UNIVERSIDAD SANTO TOMÁS

Generación de Movimientos Coordinados de Enjambre en Múltiples Drones a través de Algoritmos de Aprendizaje Profundo

Realizado por

Gómez Garzón Nicolás David

Peña Castro Néstor Harbey

Proyecto de Grado presentado en cumplimiento del requisito
para optar por el grado de Ingeniería Electrónica

Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)
Facultad de Ingeniería Electrónica
División de Ingenierías

Agosto de 2022

Generación de Movimientos Coordinados de Enjambre en Múltiples Drones a través de Algoritmos de Aprendizaje Profundo

Realizado por

Gómez Garzón Nicolás David

Peña Castro Néstor Harbey

Proyecto de Grado presentado en cumplimiento del requisito
para optar por el grado de Ingeniería Electrónica

Dirigido por

Calderón Chávez Juan Manuel PhD

Co-Dirigido por

Camacho Poveda Edgar Camilo MSc

Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)
Facultad de Ingeniería Electrónica
División de Ingenierías

Agosto de 2022

«El trabajo de grado titulado **Generación de Movimientos Coordinados de Enjambre en Múltiples Drones a través de Algoritmos de Aprendizaje Profundo**, realizado por los estudiantes **Nicolás David Gómez Garzón** y **Néstor Harbey Peña Castro**, cumple con los requisitos exigidos por la **Universidad Santo Tomas** para optar al título de **Ingeniero Electrónico.**»

Ing. Juan Manuel Calderón Chávez
Docente Tutor

Ing. Edgar Camilo Camacho Poveda
Docente Tutor

Ing. José Guillermo Guarnizo Marín
Evaluador del proyecto

Ing. Sindy Paola Amaya
Evaluadora del proyecto

UNIVERSIDAD SANTO TOMÁS

Ingeniería Electrónica

Este proyecto de grado fue aceptado para optar al título de *Ingeniero Electrónico* en Mes Año.

Director: Ing. Juan Manuel Calderón Chávez.
Magíster en Ingeniería Electrónica
Universidad de los Andes
Bogotá, Colombia.

Ing. Edgar Camilo Camacho Poveda.
Magíster en Ingeniería Electrónica
Universidad Nacional
Bogotá, Colombia.

Jurados: José Guillermo Guarnizo Marín.
Doctor en Automática, Robótica e Informática Industrial.
Universidad Politécnica de Valencia
Bogotá, Colombia.

Sindy Paola Amaya
Magíster en Ingeniería Electrónica y de Computadores con énfasis en control y automatización
Universidad de los Andes
Bogotá, Colombia.

Sustentación proyecto de grado: Dia Mes Año, Bogotá, Colombia.

Nicolás David Gómez Garzón _____

Néstor Harbey Peña Castro _____

El presente trabajo está dedicado a nuestras familias, seres queridos y todas aquellas personas que con su paciencia, dedicación y acompañamiento nos apoyaron e hicieron esto posible. Agradecimientos especiales a nuestras mamás, Mariana Garzón Acuña y Aneida Peña y a toda la planta educativa de la universidad Santo Tomás.

Agradecimientos

Agradecemos a nuestras familias por mantener su apoyo incondicional en nuestro proceso de aprendizaje y crecimiento como futuros profesionales. Así mismo, queremos agradecer al ingeniero Juan Manuel Calderón Chavez por su dedicación, paciencia y guía durante toda la trayectoría de este proyecto; además por escucharnos en cada oportunidad y mantenernos en la dirección correcta a la que queríamos llegar con la finalidad de este proyecto.

Resumen

EL presente trabajo de grado plantea un algoritmo de aprendizaje profundo basado en *Q learning* que permite a un grupo de agentes representar un movimiento de enjambre, específicamente *leader follower* implementando una repulsión entre agentes y evasión de obstáculos fijos. El modelo de aprendizaje incluye dos métodos para disminuir el riesgo de divergencia del algoritmo, el primero de ellos es la inclusión de una memoria de experiencias para el sistema y por otro lado el uso de una segunda .

La convergencia del Algoritmo lograda en menos de 6000 episodios se verificó con ayuda de la librería MATPLOT para posteriormente ser implementando en el ambiente de simulación del software CoppeliaSim.

La evaluación del sistema de implementación del modelo se realizó por medio de 6 experimentos, cada uno de ellos representando distintas situaciones de evasión de obstáculos y seguimiento de líder demostrando que el modelo entrenado cumple correctamente con lo esperado.

Índice general

Índice de figuras

Introducción

En las últimas décadas el desarrollo de la robótica de enjambre se ha abierto más al público, como se ha visto en diversas ocasiones como la inauguración de un nuevo operador en Colombia o la celebración del año nuevo 2020 en China, pero a parte de entretenimiento es casi imposible encontrar noticias sobre su uso en rescate, mapeo de zonas, u otras funciones para beneficio de la sociedad puesto que hoy en día la adaptación de estos sistemas en ambientes cambiantes es aún un reto. En este trabajo se propone la generación de movimientos coordenados para sistemas de enjambre con el uso de Aprendizaje por refuerzo con el fin de contribuir a esta necesidad.

En la primer parte de este trabajo se realiza una búsqueda de artículos y/o documentos relacionados con este proyecto en las diferentes bases de datos. También se realizan pruebas en distintos simuladores, con el objetivo de identificar el más apto para nuestros objetivos.

Sucesivamente se define el sistema de aprendizaje profundo resultante del análisis de las acciones, observaciones y estados (A.O.E) que permiten al grupo de robots comportarse como un enjambre y tomar las decisiones adecuadas según su estado de aprendizaje. Posteriormente se implementa el algoritmo para un único agente, realizando simulaciones para verificar su convergencia, para este fin se dispone de herramientas como *Matlab* y *Python*.

Siguiendo el desarrollo práctico y verificando la convergencia del algoritmo se procede a realizar la simulación de este en un ambiente que permita implementar varios agentes con el objetivo de observar su funcionamiento en modelos virtuales de robots reales, para ello se usará el simulador escogido en la primera etapa.

Luego de obtener los datos de la simulación se evalúan los resultados teniendo como parámetro de estimación el comportamiento del enjambre generado con el algoritmo.

Este trabajo de grado hace parte de un proyecto de investigación basado en el desarrollo de comportamiento de enjambres en sistemas multirobot mediante el uso de técnicas de aprendizaje. Este proyecto es desarrollado gracias al apoyo del Grupo de investigación y Desarrollo en Robótica de la Universidad Santo Tomás (G.E.D.). Este grupo se enfoca principalmente en áreas de la robótica e inteligencia artificial relacionadas con el fútbol de robots, robótica cooperativa, desarrollo de plataformas robóticas móviles, sistemas de aprendizaje automático y robótica educativa entre otros. El grupo G.E.D. ha participado activamente en RoboCup [1] como parte del área de robótica cooperativa y sistemas inteligentes. Dicha participación se ha centrado en diseño del equipo Stox's el cual fue parte de la liga Small Size [2], [3] desde 2011 a 2017. Además, en el área de robótica multiagente se han enfocado en el desarrollo de algoritmos de navegación [4] y generación de comportamientos colectivos [5]. En el campo de desarrollo de plataformas robótica se ha trabajado en el desarrollo de un laboratorio para la evaluación de algoritmos de enjambre [6], ademas del diseño e implementación de robots de asistencia casera y desinfección para covid-19 [7], [8]. En el área de los sistemas de aprendizaje se han realizado proyectos encaminados al uso de sistemas de recomendación de estilos de vestir [9] y asistentes inteligentes para el consultorio jurídico de la Universidad Santo Tomas [10].

Capítulo 1

Planteamiento del problema

1.1. Planteamiento

En los últimos 50 años el desarrollo de la robótica ha tenido un gran impacto tanto en la sociedad como en la economía. Debido a que el uso de plataformas robóticas proporciona mayor seguridad, precisión y rendimiento en procesos industriales, médicos, logísticos, entre otros, por lo que cada vez son más los campos en donde se aplica [11]. En la hoja de ruta para la robótica, del gobierno de los Estados Unidos [12], varias universidades analizan el impacto que ha tenido la robótica y exponen cómo probablemente, en las próximas décadas llegará a ser un elemento omnipresente como lo es la informática hoy en día.

A partir de esto, el uso de Drones ha estado en la mente de los desarrolladores de robótica, pensado en estos como una herramienta para facilitar o mejorar el trabajo. Un claro ejemplo de esto es su uso en actividades como lo es la mensajería [13], agricultura [14] exploración en áreas de difícil acceso para el humano [15]; además de su uso en la industria cinematográfica no solo para capturar imágenes desde gran altura sino para preservar la seguridad del lugar. Sin embargo, el manejo que se le está dando a los drones no alcanza su máximo potencial, ya que, en primera medida solo se suele emplear un único dron para estas actividades.

La implementación de sistemas multiagente no es una tarea sencilla, dado que se deben tener en cuenta diversos factores, tales como la distancia entre cada dispositivo y la dirección de vuelo [16], también es importante tener en cuenta los retos que ofrecen los sistemas multiagente, entre ellos están la no-estacionariedad. Este problema radica en que las acciones y comportamientos de un Agente van a afectar al medio y consigo a los demás, a diferencia de un sistema con un único agente donde sus acciones repercuten en el medio en que trabaja y en sí mismo. En un

sistema multiagente cada decisión tomada por cada agente individual repercute en los demás y en la finalidad del sistema. Un ejemplo de estas dificultades se puede ver en un sistema diseñado para jugar fútbol de robots como lo expone Nguyen [17].

Por ende, es necesario desarrollar un algoritmo que permita a los drones tomar decisiones pertinentes en cada situación en la que se vean involucrados, de manera que sin importar el número de dispositivos usados, asuman un comportamiento enfocado al trabajo de enjambre [18].

1.2. Pregunta problema

Basados en lo descrito anteriormente se propone la siguiente pregunta problema:

¿Cómo generar movimientos coordinados de un enjambre de robots a través de algoritmos de aprendizaje profundo, para aplicaciones en vehículos aéreos no tripulados?

Capítulo 2

Antecedentes

Con el fin de contextualizar se presentan a continuación una recopilación de trabajos realizados con medios y fines similares al propuesto. En 1992 Watkins y Dayan [19] uniendo varias herramientas de la época tales como la ecuación de Bellman, el proceso de decisión de Markov y el aprendizaje de diferencia temporal desarrollaron el *Q-Learning*, siendo este método de Aprendizaje maquina incapaz de resolver problemas con un nivel de dificultad elevado por sus limitaciones frente al número de entradas posibles para procesar, sumado al intento por solucionar día a día problemas cada vez más complejos dio lugar a que en 2015 Mnih [20] realizara un importante avance al unir el Deep learning con el *Q learning* y el aprendizaje por refuerzo implementado mucho antes en 1898 por el psicólogo Thorndike [21] resultando en el origen del *"Deep reinforcement learning"* o aprendizaje profundo por refuerzo.

Al día de hoy la complejidad de los problemas es tal que el uso de sistemas individuales para la solución de problemas está quedando atrás, un ejemplo de ello son los videojuegos de multijugador masivo, el ámbito de exploración aeroespacial y los sistemas de defensa tanto terrestres como aéreos con el uso de drones en donde los sistemas individuales pierden Escalabilidad, efectividad y son susceptibles a fallas totales con el paso del tiempo [17].

Con el fin de mitigar los inconvenientes que, trae un agente individual para la solución de problemas, se propuso el uso de sistemas multi agente [22]. Busoniu realiza una descripción y comparativa entre el uso de aprendizaje reforzado tanto en un solo agente como en múltiples agentes. Aquí expone como la técnica de aprendizaje reforzado para un solo agente está descrita por el proceso de decisión de Markov y, en múltiples agentes, por el juego estocástico; el cual es una modificación que generaliza el proceso de decisión de Markov para el caso de múltiples agentes. El autor resalta los beneficios de esta configuración puesto que pueden realizar tareas

de una forma más rápida ya que cuentan con la capacidad de procesamiento paralelo, son capaces de compartir información con otros agentes logrando así que puedan aprender de su entorno en conjunto y facilitando su Escalabilidad ya que al introducir un nuevo agente al sistema, los agentes antiguos le comparten la información y de igual manera si un agente llega a fallar o deja de ser parte del sistema, este puede continuar desempeñando su labor.

Posterior a la introducción de los sistemas multi agente y con el fin de optimizar su operación se realiza la introducción de los sistemas multi agente bio inspirados, entre ellos se encuentran los sistemas de enjambre, estos sistemas están presentes ampliamente en la naturaleza, cardúmenes de peces, bandadas de aves, manadas de animales, colonias de insectos y hasta de bacterias [23]. En 2006 WeiXing [24] propone unos algoritmos para la coordinación, formación y agregación de nuevos agentes, para esto se basa en el comportamiento de organismos grupales de la naturaleza proponiendo mejorar la adaptación de sistemas en ambientes complejos donde cambiar su formación sería retador. El sistema que expone WeiXing se basa en un término acuñado por el biólogo Paul Grasse *"stigmergy"* donde por la dificultad de comunicación entre individuos debido al entorno, estos se coordinan entre sí al modificar este, un ejemplo son los cardúmenes de peces, donde el movimiento que realiza un pez genera una pequeña corriente llamada zona de estabilidad que a su vez es interpretada por su vecindario haciendo que estos corrijan su posición para mantener una distancia segura entre sí, el algoritmo propuesto por WeiXing sigue esta modalidad de coordinación para un grupo de robots en un ambiente submarino.

Por otro lado, Stirling y Floreano exponen en su articulo [25] el problema que representa controlar la implementación de enjambres de robots voladores en lugares desconocidos, dado que si se despliegan muchos de estos hacia ubicaciones innecesarias se puede exponer al sistema a un gasto energético desaprovechado; de igual forma, si los robots no son suficientes para la tarea, esta podría resultar inalcanzable. Por esto, se propone construir una red en la cual los robots se colocan en los techos para ahorrar energía, desplegándose sólo en el momento en que son requeridos. Para dicha metodología se proponen tres estrategias. La primera y la más simple no requiere comunicación ni ningún sensor adicional, consiste en desplegar los robots con un intervalo de tiempo fijo entre lanzamientos consecutivos. La segunda estrategia no requiere de sensores adicionales, pero sí debe existir algún tipo de comunicación entre los robots, estos deben comunicarse entre ellos para calcular la densidad de robots voladores, de esta forma se evita la congestión. Por último, la tercera estrategia no requiere de comunicación, pero si de un sensor adicional que permita percibir el alcance de los robots más cercanos con el fin nuevamente de evitar que se cree una congestión.

En el año 2010 Liu-Windfield [26] propone un modelo probabilístico para la adaptación de enjambres en un barrido de terreno. Así mismo, cada robot del sistema de enjambre tiene la capacidad de ajustar sus parámetros para maximizar la energía del enjambre. Esto lo realizan usando una máquina de estados probabilísticos y una serie de ecuaciones diferenciales desarrolladas por ellos. Finalmente, el modelo propuesto facilita un entendimiento del efecto de las características individuales de cada robot en el desempeño del enjambre; lo que a su vez, permite la optimización individual de los mecanismos de control y físicos de cada robot a partir de un modelo probabilístico macroscópico.

Es importante analizar el trabajo realizado por Brambilla en el año 2012 donde ejecuta un análisis a las configuraciones de enjambre y propone dos taxonomías, en la primera clasifica la literatura analizada por diseño y análisis de métodos y en segundo lugar pero de igual importancia por el comportamiento colectivo [18]. Por último Brambilla realiza una discusión sobre los límites para la época de la robótica de enjambre y su futuro.

En el artículo de Ferrante [27], se presenta un método con el cual tres robots, los cuales están físicamente conectados a un determinado objeto, deben transportar dicho objeto de un punto A hasta un punto B, teniendo que enfrentar en el recorrido algunos obstáculos. Esto se debe a que una de las dificultades más desafiantes para este sistema es que los robots tienen una percepción diferente del entorno, dado que existe la posibilidad que el objetivo y los obstáculos sean percibidos sólo por alguno de ellos. Por lo que se deberá desarrollar simulaciones con las cuales puedan analizar eficiencia y robustez. Los robots en enjambre deben dividirse las direcciones de rumbo, las cual deben tener en cuenta las percepciones de todos estos, ya que son diferentes para cada uno de ellos.

El problema de coordinación de múltiples agentes aéreos con el objetivo de transportar una carga sujetada por cuerdas [28] ha sido ampliamente estudiado, para este caso Cardona ha enfrentado el problema con diferentes enfoques de control. En [29] se realiza el modelado del sistema de quadrotors y la carga suspendida a ellos, y a través del uso de técnicas de control geométrico se realiza el diseño del control de transporte. En [30], [31] se continúa desarrollando en mismo problema pero en este caso se hace uso de técnicas de control adaptativo óptimo y se asume que los agentes son heterogéneos introduciendo más incertidumbre y realismo al sistema. En [32], Cardona plantea el uso de control robusto adaptativo para sincronizar los quadrotors heterogéneos. Aunque este trabajo de grado no se enfoca en el transporte de carga, a futuro se planea utilizar técnicas de aprendizaje reforzado profundo para enfrentar el mismo problema y utilizar como base de comparación algunos de los resultados obtenidos en estos trabajos.

En 2018 Wilson Quesada, en la conferencia Internacional sobre Inteligencia Artificial y *Soft Computing* (ICAISC), presentó su trabajo *Leader-Follower Formation for UAV Robot Swarm Based on Fuzzy Logic Theory* [33], dicho trabajo consiste en guiar un enjambre de robots capaz de mantener una formación donde existe un líder y los demás lo siguen sin chocar con los otros agentes del enjambre. El modelo matemático de este sistema fue trabajado en Matlab, donde es posible observar el comportamiento descrito, sucesivamente se implementa una simulación en Vrep para obtener resultados 3D. Cabe resaltar que las simulaciones nos permiten observar cómo se obtienen algunos comportamientos bio-inspirados, dado que los agentes rodean al líder cuando este se encuentra en una posición estática posicionándolo en el centro y son capaces de seguirlo sin colisionar entre ellos cuando este se encuentra en movimiento.

Siguiendo la taxonomía propuesta por Brambilla es posible analizar el artículo de Gustavo A. Cardona y Juan M. Calderon, *Robot Swarm Navigation and Victim Detection Using Rendezvous Consensus in Search and Rescue Operations* [15], publicado en 2019, los autores exponen la posibilidad de usar sistemas multi-robot para la búsqueda y rescate de sobrevivientes de catástrofes, logrando así evitar riesgos adicionales. La investigación está dirigida hacia la navegación del enjambre de robots aplicando la detección de víctimas, basando la navegación en la teoría del enjambre de partículas, donde las fuerzas de repulsión y atracción son utilizadas para evitar obstáculos, mantener el grupo compacto y navegar hasta un objetivo determinado. Una vez identificada la posible víctima los agentes se separan, creando de esta forma un sub-enjambre, esto permite que los demás robots continúen con la búsqueda de posibles víctimas mientras el sub-enjambre detecta si efectivamente se trata de una situación de peligro.

Hüttenrauch y Neuman [34] exponen la aplicación de un método de *deep reinforcement learning* aplicado a enjambres, en este se recopilan características propias de un enjambre como se analizó anteriormente en la taxonomía de Brambilla. Se resaltan puntos específicos de los enjambres como lo son la multivalencia entre los agentes o que ninguno cumpla un rol diferente a los demás y la irrelevancia del número de agentes. El modelo propuesto se basa en la inclusión de distribuciones medias para tratar a los agentes como muestras del modelo y un proceso llamado incorporación media empírica como base para una política de aprendizaje descentralizado. Por otra parte evalúan el sistema propuesto en dos ambientes, el encuentro de agentes y la evasión de obstáculos en una persecución.

Para el año 2022, Xudong y Fan introducen un esquema de aprendizaje por refuerzo enfocado a enjambres de robots con el fin de responder al problema de la seguridad de los datos adquiridos del entorno que se presenta en sistemas multiagente centralizados, puesto que al comprometerse alguno de los agentes, la información completa del sistema se vería afectada. En este artículo proponen el uso de un sistema de enjambre en donde por medio de la descentralización del

sistema los agentes logran comunicar únicamente su experiencia de aprendizaje y cada uno es capaz de actuar por si mismo. En este mismo artículo se presenta el esquema de aprendizaje propuesto como un método de aceleración del proceso de aprendizaje y optimización del proceso de observación en campo, puesto que cada agente recopila información por separado que luego se comparte, esto hace que entre más agentes haya en el sistema, más rápido aprende este mismo [35].

Posteriormente en este mismo año Sitong y Yibing presentan el problema de planificación de rutas de navegación para agentes autónomos. Para este análisis toman en cuenta que el entorno es cambiante y dinámico y lo asocian a ambientes de búsqueda y rescate. El método de solución está basado en *deep reinforcement learning*, específicamente para su aplicación en vehículos aéreos no tripulados. El modelo propuesto se basa en el algortimo de *twin delayed deep deterministic policy*. Para dar una predicción al impacto del ambiente dinámico en el agente se utiliza una técnica llamada red Actor-Critico que lo implementan como una forma de definir las entradas del sistema. Este modelo se presenta como una simulación del agente en un entorno virtual y como resultado es posible observar que en efecto el mecanismo implementado permite que de forma autónoma el agente recalcule su ruta en respuesta a estímulos del ambiente de una forma eficiente y con posibilidades de generalizarse para dar solución a otros problemas[36].

Capítulo 3

Justificación

Según las hoja de ruta *Robotics 2020"* y "del internet a la robótica"[37] [38] La robótica está teniendo un impacto creciente en diferentes sectores sociales e industriales, es por esto que un sistema que ayude a mejorar el cumplimiento de las tareas de estos sectores tiene gran cabida.

Los sistemas de enjambre proporcionan ventajas sobre los sistemas convencionales, una de ellas es su confiabilidad, debido a que una tarea específica puede ser realizada aun cuando se presentan errores tales como el fallo de algún agente o la presencia de factores que no se han considerado [37]. Por ello, la implementación de un sistema de enjambre en drones, los cuales son usados para todo tipo de trabajos, resulta siendo una necesidad de innovación en el campo tecnológico, puesto que de esta forma el riesgo de actividades peligrosas para las personas disminuye y el rendimiento de estas aumenta exponencialmente.

Dentro de los beneficios que traería usar un enjambre de drones está el hecho de que son escalables, es decir, ante la falla de un agente no se comprometería la tarea principal; además, se facilitaría y permitiría desarrollar algunas tareas de manera más rápida y eficiente, como por ejemplo la exploración de lugares desconocidos.

Dichos sistemas idealmente deben ser adaptativos, es decir, capaces de ajustarse automáticamente a entornos estocásticos, dinámicos y no estacionarios [12] característica que los hace útiles cuando se necesita tomar acciones en entornos difíciles o desconocidos. También, dichas ventajas permiten al sistema ser de gran utilidad en situaciones de desastres [37], donde se desconocen muchos factores, siendo necesario intervenir de manera rápida y eficiente; los drones pueden cumplir funciones como el transporte de equipos necesarios para la realización del trabajo de los rescatistas, realizar un mapeo de puntos críticos y zonas donde se encuentran las víctimas; otorgando así un mayor rango de efectividad a la hora del rescate [39].

Finalmente, el desarrollo de este tipo de proyectos va a incentivar el uso de estas tecnologías a nivel nacional dado que es aplicable en muchos campos, tales como agronomía, seguridad, rescate, entre otros. Además, a nivel universitario, sobre todo en cuanto al grupo de investigación GED, se impulsará el desarrollo de sistemas de control multiagente junto con el uso de tecnologías como los drones para futuras investigaciones, con el fin de llegar a implementar este tipo de soluciones.

Capítulo 4

Impacto Social

Generar movimientos coordinados de enjambre en un sistema multi robot de drones por ejemplo puede suplir necesidades de vigilancia, control de plagas, control de deforestación, monitoreo de cultivos pasando de un proceso común de agricultura a uno de precisión, generando más trabajo y haciendo más eficiente el uso del suelo en [37] [38] de igual forma se muestra que para el año 2050 solamente estarán disponibles 0.15 hectáreas de tierra cultivable por persona reduciendo cada vez más el margen de fracaso para los cultivos.

En el área civil se plantea gracias al incremento de la población mundial un aumento en la concentración de personas en las zonas urbanas generando un incremento de la inseguridad y haciendo más difícil el reconocimiento del delito. Sistemas como el propuesto en este trabajo pueden contribuir con labores de reconocimiento, seguimiento y apoyo en tiempo real a las entidades de emergencia de las metrópolis. Entre las oportunidades actuales y futuras planteadas en "Robotics 2020" [37] [38] existen varias aplicaciones donde es posible aplicar los sistemas de enjambre de drones, como por ejemplo en el control y monitoreo de bosques para la detección temprana de focos de incendios, monitoreo de seguridad en lugares de alta importancia estratégica como aeropuertos, plantas nucleares o de energía, oleoductos, entre otros, monitoreo y desarrollo de tareas en lugares donde por las condiciones medioambientales, sociales o la impopularidad de la tarea a realizar sea de extrema dificultad el uso de personal humano.

Capítulo 5

Marco teórico

A continuación, se presenta una serie de información recolectada, la cual ha sido base para este proyecto, se realiza una división entre referencias teóricas y referencias conceptuales con el fin de enriquecer el conocimiento previo para la comprensión de los resultados y análisis, de manera que se presenta información desde las generalidades hasta las especialidades que han permitido el desarrollo de la propuesta.

5.1. Robótica

El acercamiento que se realiza a la robótica tiene como fin la automatización de un agente para la realización de tareas por medio de la interacción con el ambiente, cumpliendo precisamente con dos requisitos: un sistema automático y una capacidad de razonamiento para la función integral de sus sensores [40]. El origen del término robot hace referencia a estas funciones, desde la lengua checa *"robota"* hace alusión al trabajo obligatorio que hace el ser humano a la máquina.

La transformación de la robótica tiene su origen desde Herón de Alejandría con sus primeras automatizaciones, pasando a través del tiempo dónde se evidencia la transformación cultural, social y tecnológica originando diferentes necesidades e innovaciones, hasta la creación del telar de Jacquard, sin embargo, es G.C. Devol, quien da paso a la robótica industrial abriendo las puertas al uso de computación para la automatización [41].

5.2. Robótica de enjambre

La robótica de enjambre es impulsada por la inteligencia de enjambre analizada por Gerardo Beni y Jin Wang, quiénes a partir de un modelo biológico evidenciaron un comportamiento globalizado generado por la auto-organización, la comunicación y la descentralización del control por parte de los individuos. Esta observación se plantea en el ámbito de la naturaleza, exactamente en la vida animal, organismos como abejas que construyen panales a través de patrones, aves en temporada de migración y hormigas que trabajan organizadamente por la obtención de alimento, constituyen una serie de comportamientos que luego son analizados y sistematizados [42]. Estos comportamientos naturales impulsaron la creación de algoritmos en los que Beni y Wang se basaron; tomaron esos patrones de la naturaleza y los enfocaron a la resolución de problemas a través de los comportamientos de enjambres de colonias y otros grupos de animales [23].

La robótica de enjambre se inspira en los comportamientos organizados de animales sociales como hormigas, aves, peces, entre otros para coordinar y organizar sistemas con múltiples robots.

En el trabajo de Brambilla [18] se presentan algunas características de un sistema de enjambre entre las cuales resalta el hecho que los robots deben ser autónomos, por lo que no debe existir una unidad central que los controle, además, los robots deben ser simples, deben tener capacidad de comunicación y detección local, esto reduce el costo de cada robot, permitiendo que el enjambre pueda ser ampliamente escalable [15]. Se desarrollan unas características base para los robots de enjambre:

- Comunicación: la base de la organización puede ser por medio de sensores, comunicaciones o por el medio ambiente que lo rodea.
- Simplicidad: la construcción física del robot para una acción rápida y su flexibilidad para trabajar en cualquier entorno.
- Control del sistema: tiene la posibilidad de ser centrado o descentralizado.
- Inspiración biológica: el comportamiento puede estar generado de manera simple a un nivel mínimo, o avanzado con capacidades distintivas.
- Autonomía: el trabajo autónomo sin servidores o personas a cargo, con posibilidad de un líder del enjambre.

Entre los principales beneficios que presentan este tipo de sistemas se tiene:

- Escalabilidad y estabilidad: se puede ausentar y unir agentes sin afectar al grupo gracias a la comunicación entre ellos.
- Paralelismo: multifuncional por la cantidad de agentes.
- Flexibilidad: pueden realizar cambios en sus comportamientos según el ambiente.
- Económico: el costo de producción es bajo.

5.3. *Deep reinforcement learning DRL*

El aprendizaje por refuerzo profundo es un subcampo relativamente nuevo del aprendizaje automático que combina el aprendizaje por refuerzo (RL) y el aprendizaje profundo. Este enfoque ha sido capaz de resolver una amplia gama de tareas complejas de toma de decisiones que antes estaban fuera del alcance de una máquina. Como resultado, Deep RL está abriendo muchas aplicaciones nuevas en dominios como el cuidado de la salud, la robótica, las finanzas y muchos más.

Esta variante es una fusión entre las deep neural networks que ayudan gracias a los pesos de los nodos y policy gradients, haciendo uso de redes neuronales y múltiples algoritmos, trabajan para lograr la recompensa. Es decir que se unen las técnicas de recompensas de las máquinas y el complejo aprendizaje de la inteligencia artificial [43].

A continuación algunos campos que en conjunto conforman el DRL.

5.3.1. *Q learning*

Q learning es un algoritmo que hace parte del aprendizaje automático y toma como punto de partida los procesos de decisión de Markov (MPS). El objetivo que tiene es la conexión entre el agente y el ambiente por medio de la percepción y las acciones. Es una cadena continua en la cuál el agente recibe información en un tiempo acerca del ambiente, responde ante este veredicto generando un cambio, vuelve a evaluar el ambiente y a generar otro Estado nuevo [44].

La versión base de *Q learning* ocupa una tabla de valores en la cual para cada una de las entradas del sistema se asigna un par de datos, la acción y el estado del agente. El valor óptimo de la función se representa con la ecuación de Bellman que está definida por:

$$Q^*(s,a) = r + \gamma max_{a'} Q^*(s',a') \tag{5.1}$$

En donde r es la Recompensa, γ el factor de descuento, (s,a) el par de valores de estado y acción y (s',a') el par de valores próximos de estado y acción. El objetivo del agente en este algoritmo se basa en encontrar por medio de la anterior función la acción correcta para cada estado acumulando recompensas. Cuando las respuestas se dan a causa de la acción previa y el estado, sin depender del ambiente y estados anterior se aplican los siguientes casos de MPS:

- Se pueden presentar probables transiciones: P(s, a, s')
- Recompensas deseadas: R(s,a,s')
- Para hallar una política: π*: S ->A

El rasgo más importante de *Q learning* se basa en la propuesta de Watkins del desarrollo de un algoritmo de control TD y está definido por la ecuación:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \underset{a}{max} Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{5.2}$$

Este modelo aproxima la función de valor de la acción que aprende con la función de valor óptima, lo que permite las pruebas de convergencia. En este modelo las políticas no son de importancia, pero sí determinan la actualización de los estados, lo importante es la actualización continua. Este proceso se realiza indefinidamente y funciona por episodios, el *Q learning* lanza de manera estimada Q a medida que se va moviendo el agente.

5.3.2. Aprendizaje por refuerzo

El Aprendizaje por refuerzo es un tipo de aprendizaje utilizado en la inteligencia artificial, este método es aplicado a un agente que dependiendo de la conclusión de su tarea recibe a cambio una retroalimentación de lo que ha hecho correctamente. Este aprendizaje también es inspirado por los comportamientos naturales en donde los individuos responden a estímulos

de relaciones entre la causa y su respectivo efecto. Hay cuatro elementos que son la base de este sistema [45]:

- Función de recompensa: Como respuesta a sus Acciones se otorga un número que simboliza una recompensa para el agente, el cual determina si la acción realizada es oportuna.
- Política: Define las acciones apropiadas del agente en un Entorno a modo de asociación o regla.
- Modelo: Es un conjunto de espacio de estados y acciones en donde se incluye una función que describe las transiciones que el agente realiza así como la recompensa que se le da en cada momento.
- Función de valor: Esta función describe el valor de un estado cuando el agente implementa una política.

En el aprendizaje por refuerzo, el agente aprende a comportarse a través del ensayo y error y es capaz de hacerlo sin necesidad del uso de grandes cantidades de datos en su entrenamiento. Es este tipo de aprendizaje máquina se usa el termino de recompensa con el fin de reforzar el comportamiento que se quiere lograr dando lugar a un premio numérico para el agente cuando se alcanza la meta propuesta. Para maximizar esta recompensa no se le dice al agente qué acciones tomar, en su lugar el debe descubrir qué acciones producen la mayor recompensa posible probándolas [46].

En el aprendizaje por refuerzo se puede describir un modelo generalizado para su funcionamiento, este consiste en determinar un conjunto de estados dependientes del entorno, un conjunto de acciones dependientes del agente, unas reglas para la puesta en marcha de los estados y la determinación de la recompensa.

Para explicar a grandes rasgos el funcionamiento de estos elementos Barto y Sutton crean un escenario de un juego llamado “Tic-Tac-Toe” también conocido como “triqui”. La manera de razonar es la siguiente: existe un agente que sería el jugador A, debe determinar sus acciones a través de las políticas que serían planteadas al momento de empezar el juego, pues el jugador B realiza un determinado movimiento al cual A debe responder de manera correcta, ante las múltiples opciones de movimiento que puede hacer el jugador B, A debe saber responder a todas ellas, pero solo será posible a través de juegos previos o durante el juego. Cada movimiento ganado que funciona como recompensa es jugado con el fin de llegar al objetivo final que es ganar el juego con tres Xs ú Os tachadas, y correspondería finalmente a la función de valor, en

este caso como el juego actual se está basando en las conclusiones de juegos anteriores, se está usando un modelo. [46]:

En la figura 1 se observa un esquema básico de la interacción agente-entorno en un proceso de aprendizaje por refuerzo.

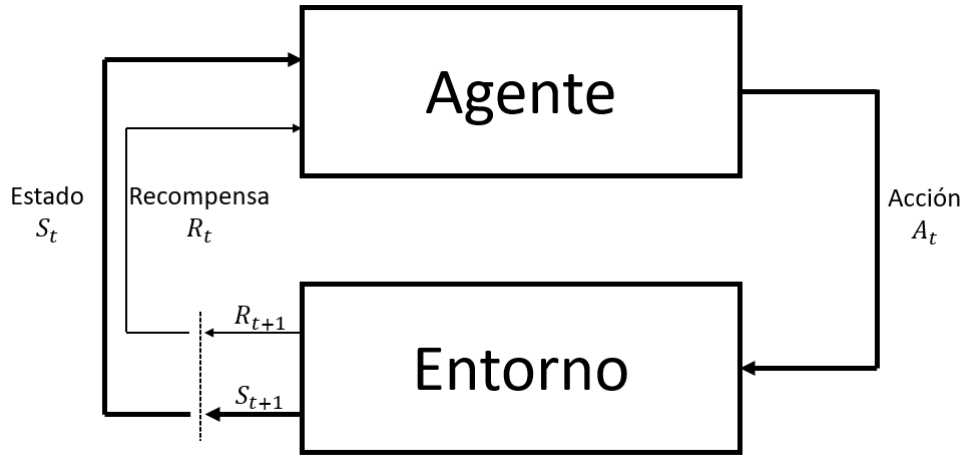

FIGURA 1: Proceso de decisión de Markov [46]

El agente observa el entorno en el que se encuentra, a partir de ello toma una decisión sobre como debería actuar, ejecuta la acción y a partir de ello recibe una recompensa, basándose en la experiencia aprende, sucesivamente repite el proceso hasta encontrar una estrategia óptima[47].

Para representar dicho proceso se define $Q(s, a)$ la recompensa ante una acción en un determinado estado. La función representa la recompensa máxima que se puede recibir al tomar la acción a cuando se encuentra en el estado s está dada por:

$$Q(s_t, a_t) = max(R_{t+1}) \tag{5.3}$$

También se define como π la política a tomar dado un determinado estado, por lo que la política óptima está dada por:

$$\pi^*(s) = argmax_a Q(s, a) \tag{5.4}$$

De esta forma se llega a la ecuación de Bellman, la cual es la base del aprendizaje por refuerzo:

$$Q(s, a) = r(s, a) + \gamma max Q(s', a') \tag{5.5}$$

5.3.3. Deep neural networks

El aprendizaje profundo es un subcampo del aprendizaje automático que se inspira en las estructuras de las redes neuronales del ser humano para realizar análisis de datos en forma de red y multitarea, con un funcionamiento a través de capa dónde recibe, procesa y exterioriza la información, evita la tradicional forma lineal, de esta forma permite a la máquina aprender como hace el ser humando a través de imitaciones o ejemplos [48].

Esta técnica utiliza información multimodal para el aprendizaje y los recopila permitiendo una mayor exactitud en sus acciones. Esquema de la red Neuronal artificial

5.3.4. Deep Q networks

Esta técnica resulta ser aún más específica, consiste en usar las redes neuronales (input) para la aproximación a la función valor Q (output), resultando ser un proceso más complejo [49].

Lo que caracteriza las redes Q profundas es que contienen mucha más memoria, actúan con el máximo de salidas. El error que se contempla aquí radica entre el valor predicho y el objetivo, llamado problema de regresión. El resultado es una adición a la ecuación usada en Q learning:

El objetivo predice su propio valor, pero siempre toma la decisión por la recompensa R así que la red se actualiza por medio de retro propagación y converge.

5.4. Simulación

La simulación permite realizar análisis sin estar en el campo real de estudio, estos análisis son basados en la experimentación para arrojar resultados, en el caso del campo computacional se hace uso de programas que tienen la información necesaria para crear el ambiente simulado, pueden funcionar como generadores de números, desarrolladores de eventos o funciones estadísticas [50].

5.5. Vehículos aéreos no tripulados

Dentro de los tipos de vehículos aéreos se usará en este trabajo el término dron, se acuñó después de la segunda guerra mundial por su uso de blanco aéreo, por ende existen drones militares y civiles, estos últimos no se utilizan para fines militares y se han usado para fines tecnológicos, así mismo existen dedicados o aficionados según su uso. Otra categoría que tienen es según sus características de vuelo, pues puede ser de alas fijas con hélices horizontales de la superficie o de alas móviles como los cuadricópteros [51].

5.6. Agentes

En el lenguaje del aprendizaje por refuerzo se puede identificar al agente como un modelo que aprende, es el que recibe el entrenamiento, dentro de esas operaciones a realizar se encuentra la toma de decisiones dentro de un entorno.

Capítulo 6

Trabajos previos

En este capítulo se mostrará el trabajo que se ha desarrollado previamente a este proyecto de grado, titulado: "Leader-follower Behavior in Multi-agent Systems for Search and Rescue Based on PSO Approach" el cual fue presentado en la conferencia IEEE SoutheastCon 2022 [52].

6.1. Modelo Propuesto

En este trabajo se aborda el problema de la navegación multiagente, proponiendo una función objetivo para dictar la ruta de navegación o al menos el control de la decisión de navegación. La función de costo se basa en la interacción de las fuerzas de repulsión y atracción que se ejercen sobre el agente.

La función de costo de navegación se describe como un plano donde las fuerzas de repulsión se representan mediante funciones gaussianas para garantizar las distancias mínimas entre los agentes y evitar colisiones. La segunda parte de la función objetivo es una función gaussiana negativa que funciona como un mínimo global y está determinada por la distancia y la posición del agente principal. Dado que los agentes y el líder están navegando, la función objetivo es dinámica y los planos de búsqueda evolucionan en el tiempo debido a la navegación y la interacción de los agentes.

En base al análisis previo de la función objetivo, tendrá un parámetro que modifica dinámicamente la función dentro del plano de tal manera que se pueda simular una ruta. Figura 2 Figura 3.

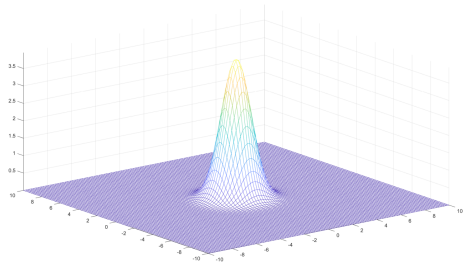

FIGURA 2: Primera propuesta de la función objetivo para el agente líder

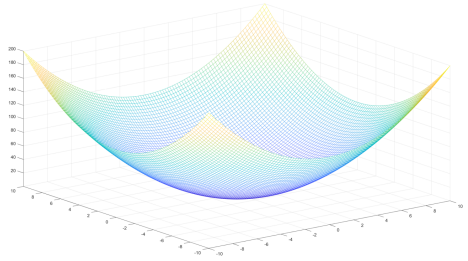

FIGURA 3: Segunda propuesta de la función objetivo para el agente líder

En cuanto a la distancia entre agentes, habrá una fuerza de repulsión, que se puede simbolizar por medio de un función gaussiana para cada uno de los agentes, de tal forma que se crea un vector a partir de la distancia del más cercano con el más lejano obteniendo así un vector de resultado, un adecuado distancia se obtendrá para cada uno de los agentes como se muestra en figura 4.

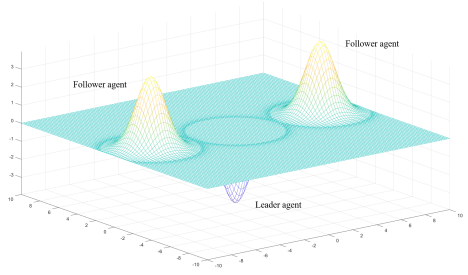

FIGURA 4: Primera propuesta, repulsión entre agentes.

Como segunda alternativa, se evalúa la distancia de un agente frente a los demás agentes de tal manera que con el cálculo de las distancias y los vectores se encuentre un vector que pueda ubicar al agente en una posición adecuada. Esta solución se ilustra en la Figura 5.

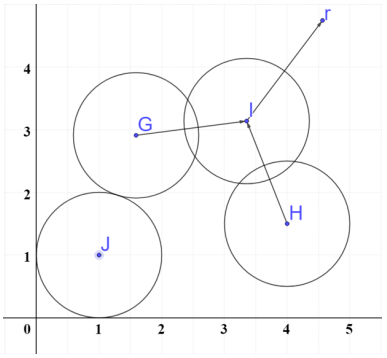

FIGURA 5: Segunda propuesta, repulsión entre agentes.

El uso de un algoritmo bioinspirado como el PSO permite minimizar una función llamada función objetivo. Para este caso, el agente líder corresponde a la posición mínima, por lo que es necesario definir una función convexa, es decir, solo tiene un mínimo, en el que al cambiar o modificar sus parámetros sigue teniendo un único mínimo global. Siguiendo este principio, la función puede tener forma de campana gaussiana o ser una función más simple como un paraboloide, que en este caso fue lo que se utiliza como función objetivo como se observa en la Ecuación 6.1,

$$F.O = (-x + Px)^2 + (-y + Py)^2 \tag{6.1}$$

donde Px y Py, además de ser valores que cambian el punto mínimo de la función objetivo, delimitan la posición del dron líder.

En cuanto al algoritmo del PSO, se cambiaron los coeficientes de aceleración e inercia de tal manera que el coeficiente social tuviera más peso para reducir la exploración y aumentar la convergencia. Esto fue propuesto como se sabe de antemano; no hay mínimos locales por los cuales solo habrá un mejor valor.

De acuerdo con las ecuaciones de posición y velocidad, se incluyó un parámetro ponderado K el cual se encontrará entre 0-1, siendo 0 para no tomar en cuenta la velocidad y 1 para tomarlo en todo su valor. De acuerdo con las ecuaciones de posición y velocidad se incluyó un parámetro k entre 0-1 que multiplicara a la velocidad, esto para que el vector resultante no cambiara bruscamente y perjudicara a la nueva posición. Este parámetro influye en el tiempo en el que converge el sistema, cuanto mayor sea k mas rápido va a converger y cuanto menor sea K más

lento va a converger, es decir, el vector de velocidad resultante al ser multiplicado con el parámetro K es pequeño, de forma tal que la nueva posición respecto a la anterior tenga distancias más cortas.

$$V_{i,j}\,(t+1) = wV_{i,j}\,(t) + C_c(P_{i,j}\,(t) - X_{i,j}\,(t)) + ... \tag{6.2}$$

$$...C_s(G_{i,j}\,(t) - X_{i,j}\,(t))$$

$$X_{i,j}\,(t+1) = X_{i,j}\,(t) + V_{i,j}\,(t+1) * K \tag{6.3}$$

Finalmente, al analizar el comportamiento del sistema de esta manera se encontró que, aunque efectivamente el enjambre seguía a un líder, la posición de los agentes no tenía en cuenta el tamaño de estos, haciendo que su implementación no fuese posible por las colisiones que se ocasionaban. Es por esto que se decidió recurrir a un algoritmo de repulsión.

6.1.1. Repulsión

El principio de funcionamiento de la repulsión entre agentes se basa en tener en una distancia mínima entre estos, es decir cada agente se puede representar a partir de una posición, además de una distancia que para fines explicativos se representará a partir de una circunferencia donde el diámetro es la distancia mínima

En la Figura 6 se evidencia la relación entre la circunferencia y los agentes, así mismo en la Figura 7 se muestra un ejemplo donde se muestra varios agentes seguidores (partículas azules) y un agente líder (partícula roja).

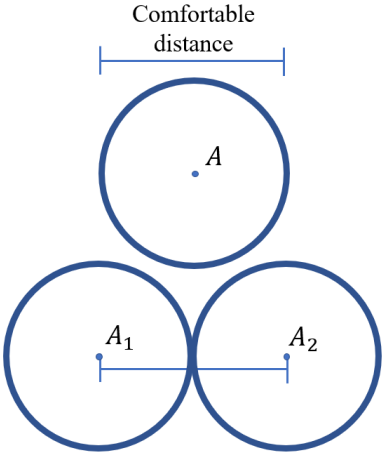

FIGURA 6: Distancia de repulsión entre agentes.

En la Figura 8 se puede evidenciar que el agente B y el agente A no cumplen con esa distancia mínima, por lo cual es necesario repeler el agente B con respecto al agente A, ya que para este caso el agente B es el que está evaluando la distancia respecto a los demás agentes. Para efectuar la repulsión entre agentes se elaboró el siguiente diagrama de flujo.

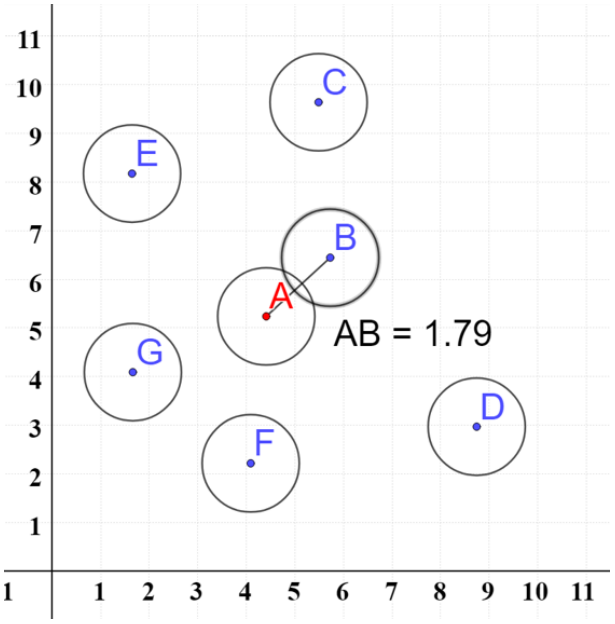

FIGURA 7: Agentes dentro de la distancia de repulsión.

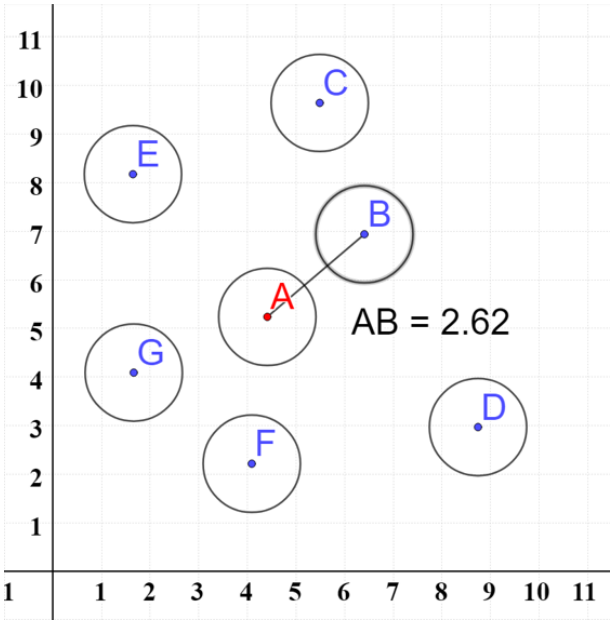

FIGURA 8: Agentes fuera de la distancia de repulsión.

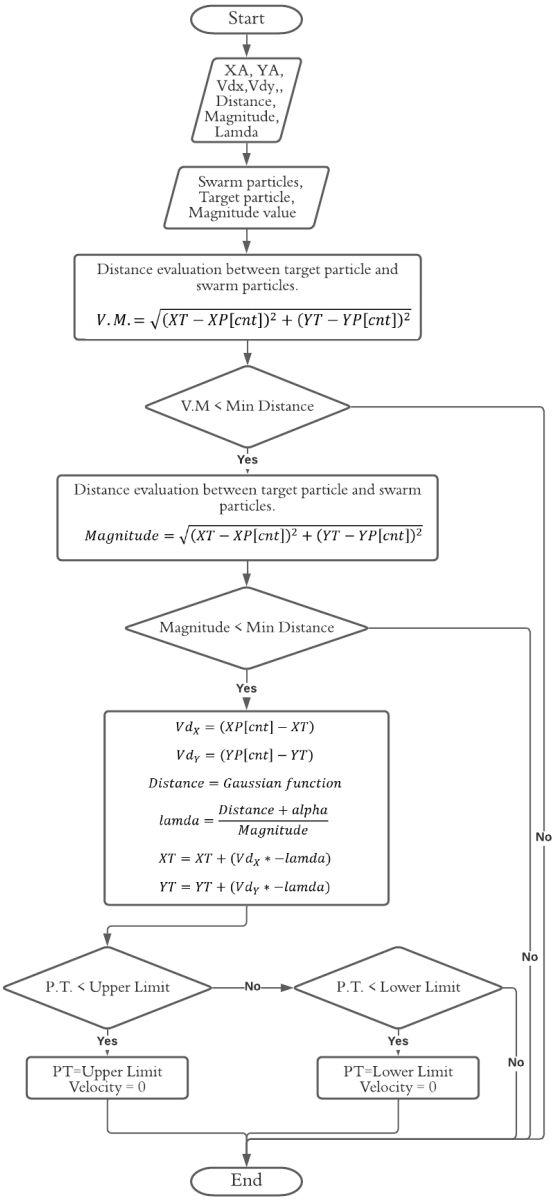

FIGURA 9: Algoritmo de repulsión.

En esencia el diagrama de la figura 9 se basa en calcular la magnitud entre el agente a evaluar y los demás agentes, es decir, en la Figura 7 el agente que evalúa la distancia entre los demás agentes incluyendo al líder es el agente B, por lo cual se mira si la distancia entre agente es inferior a la distancia mínima, si es así, se procede a efectuar la repulsión teniendo en consideración que el único agente que va a cambiar su posición es el agente a evaluar, que para este caso es el agente B. Para saber la distancia a repeler se hace uso de una función gaussiana expresada en la ecuación 6.4.

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{6.4}$$

Donde x es la magnitud, σ y μ son valores entre 0-1 suponiendo que la distancia entre agentes es inferior a 1.

El α que se observa en el diagrama de flujo es un valor entre 0-1 y su función es darle una tolerancia a esa distancia mínima como se muestra en la figura 10.

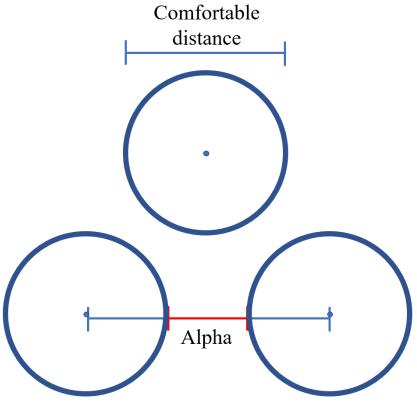

FIGURA 10: Visualización de la variable alfa.

El uso de una función gaussiana permite que al modificar la posición de los agentes estos no den saltos muy bruscos como pasaría si únicamente se restara un escalar a la magnitud, de forma tal que el uso de esta función beneficia a que los agentes se mueven de manera más uniforme, no obstante, esto conlleva a que se demore más tiempo en converger el sistema cuando los agentes están muy juntos. Es necesario recalcar que la dirección a la cual se va a mover el agente está sujeta el vector dirección que tiene el agente que va a evaluar con el agente que va a repeler, esto se puede ver en la Figura 7 y Figura 8.

Para representar lo dicho anteriormente con el diagrama de flujo con varios agentes se muestra una simulación donde el agente B es repelido por el agente A (líder) y el agente C (seguidor) por medio de las Figuras 7, 11 y 12

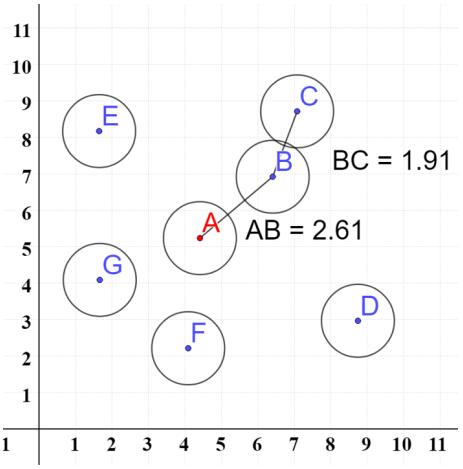

FIGURA 11: Fuerza de repulsión entre dos agentes.

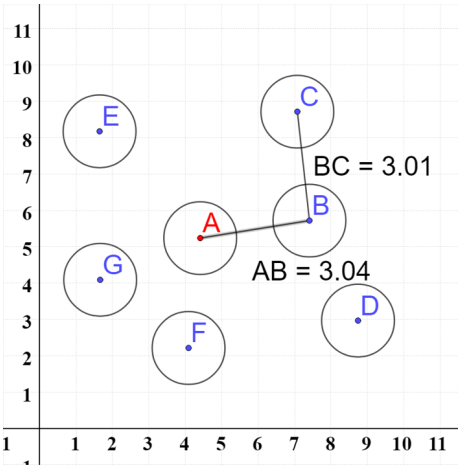

FIGURA 12: Fuerzas equilibradas de repulsión-atracción entre agentes y líder.

6.2. Resultados de la simulación

Con el fin de representar experimentalmente el modelo propuesto en la sección anterior, se usaron dos sistemas de simulación, el primero es una representación visual del modelo matemático en Python con ayuda de la librería de MATPLOTlib, lo que permite evaluar el modelo sin interferencias físicas y ver con mayor claridad su comportamiento, posteriormente se evaluó el modelo en Pybullet simulando agentes reales.

Estos dos ambientes de simulación permiten la implementación de cuatro experimentos en donde dos experimentos son realizados evaluando el modelo en Python siendo el primero de ellos la agrupación inicial de los agentes y el segundo el seguimiento de líder, repitiéndose de la misma forma para la implementación en Pybullet.

6.2.1. Convergencia de múltiples agentes reunidos

Este experimento tiene como objetivo verificar la convergencia del modelo propuesto, para esto se utilizó la librería MATPLOTlib y el modelo representado en un algoritmo en Python teniendo presente que se implementó en un ambiente de dos dimensiones. En primera instancia se verificó la convergencia del algoritmo para realizar la agrupación de los agentes, ubicando un agente líder en un punto alejado del plano XY y los demás agentes no líder de forma aleatoria alrededor del mismo plano, en la Figura 13 se representa lo anteriormente descrito.

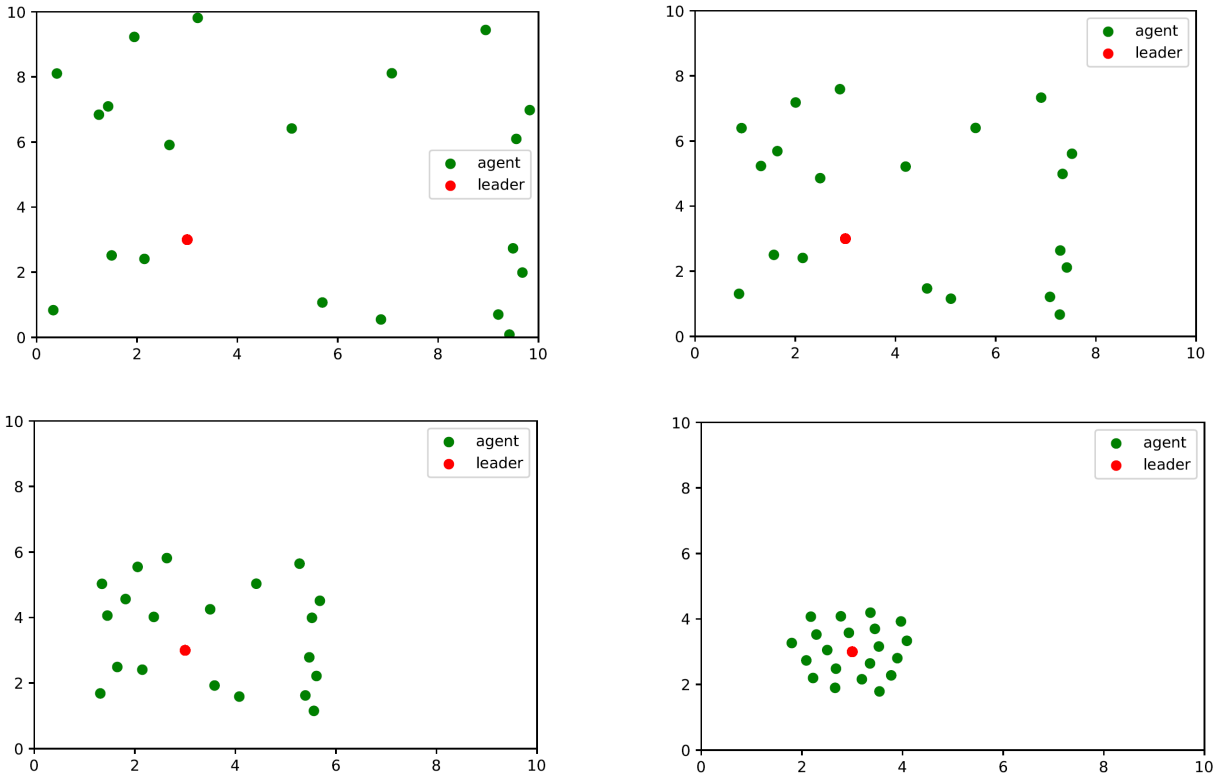

FIGURA 13: Convergencia del primer experimento

En la Figura14 se muestra la trayectoria de cada agente como resultado de la adición de cada uno de los vectores de movimiento. En esta figura se observan cruces entre las trayectorias, estos cruces se dan en diferentes espacios temporales.

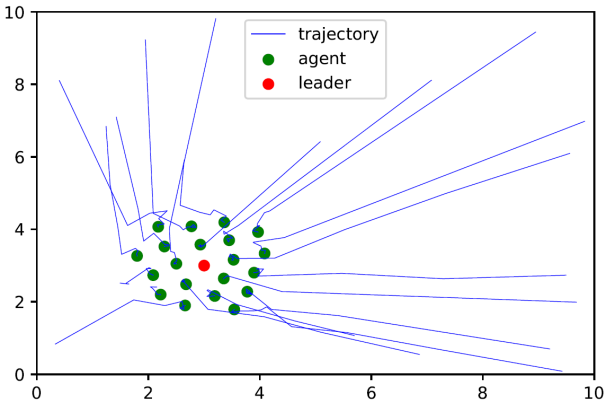

FIGURA 14: Trayectoria de partículas alrededor del líder

6.2.2. Convergencia de múltiples agentes siguiendo a un líder

En la siguiente parte del experimento se observa como al cambiar la posición del líder, el grupo de agentes inician su movimiento hacia el de forma organizada y logrando la repulsión entre ellos evitando colisionar, esto se evidencia es la Figura 15

Las trayectorias dibujadas por cada agente se aprecian en la Figura 16 donde se observa cómo cerca de su objetivo de reunión ajustan sus posiciones para lograr mantenerse equidistantes los unos de los otros.

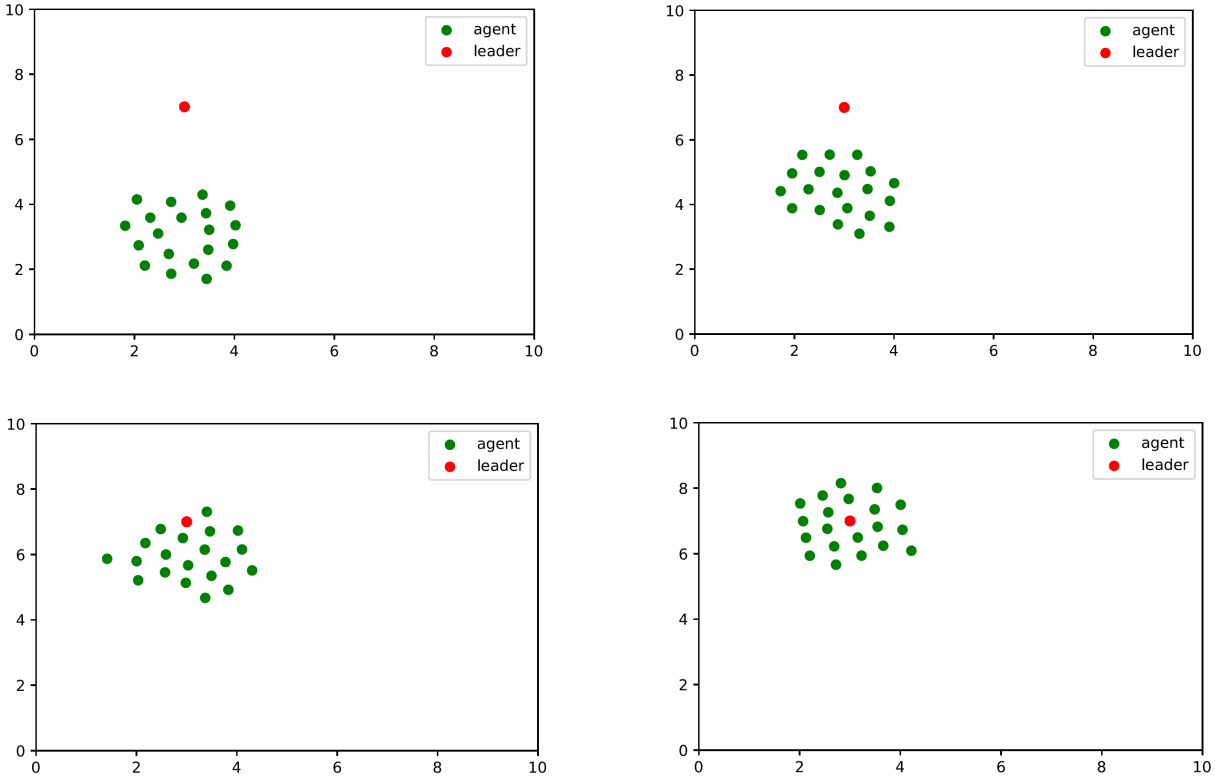

FIGURA 15: Convergencia del segundo experimento

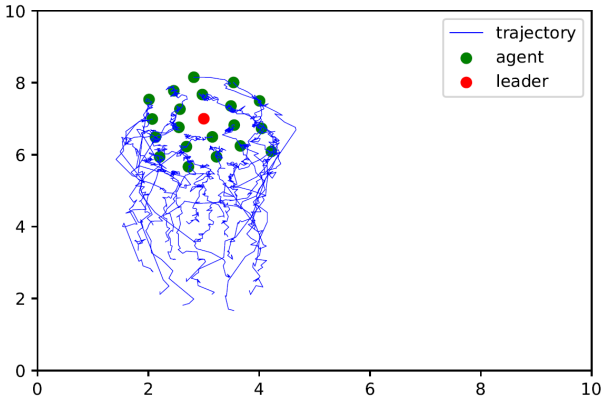

FIGURA 16: Trayectoria de las partículas siguiendo al líder

6.2.3. Simulación física de reunión de múltiples agentes.

Con la convergencia del algoritmo comprobada se procedió a comprobar el funcionamiento en un ambiente de simulación con modelos de drones reales, para este caso se utilizó Pybullet y se realizó la simulación manteniendo una altura constante y trabajando en el plano xy como se muestra en figura 17, obteniendo el mismo resultado que con las partículas.

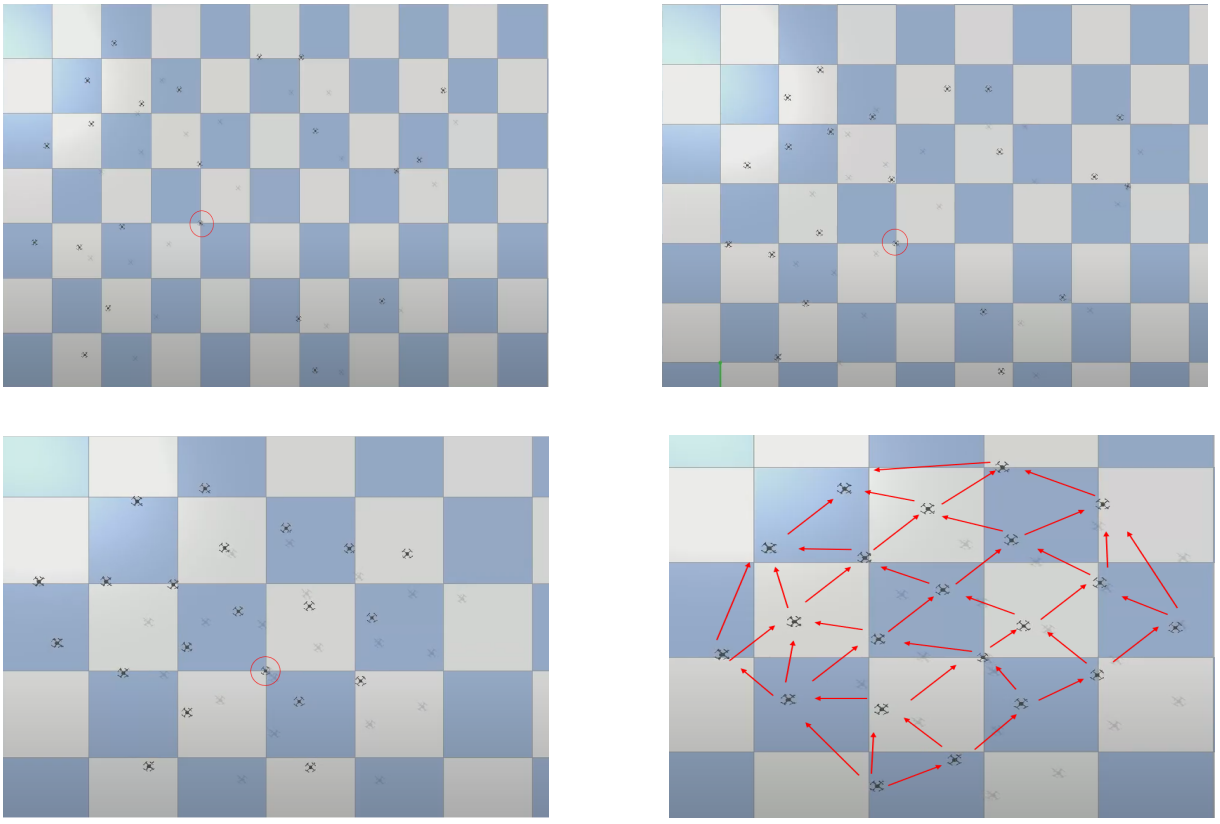

FIGURA 17: Primer experimento PyBullet.

6.2.4. Simulación física de múltiples agentes siguiendo a un líder

Al igual que en el experimento inicial se condicionó el grupo de agentes para seguir a un lider en otra posición luego de la primera reunión, estos se mueven de forma ordenada y evitando colisionar unos con otros logrando llegar al destino de forma satisfactoria comportamiento evidenciado en la Figura 18, logrando así la misma organización de la reunión inicial manteniéndose equidistantes unos de otros.

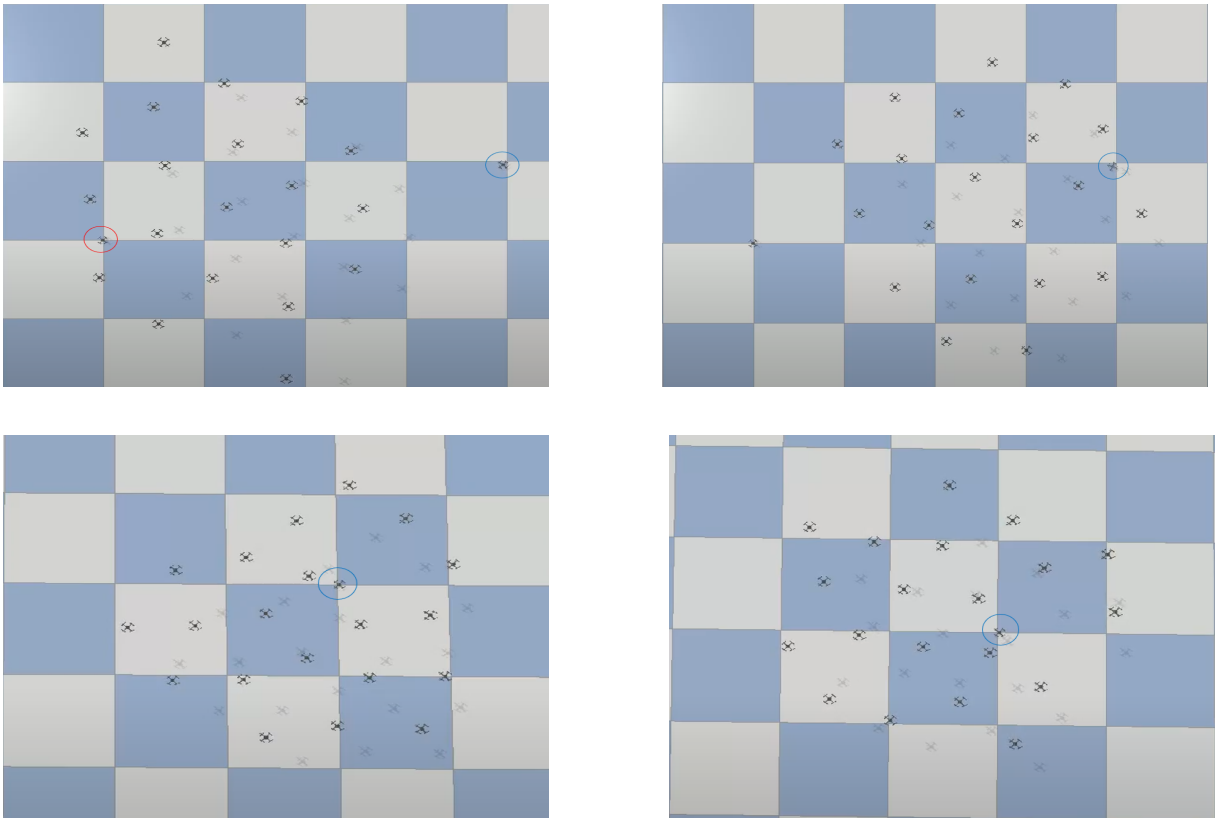

FIGURA 18: Segundo experimento Pybullet

La figura 19 muestra picos en las iteraciones 0 y 200, esto ocurre porque el agente líder cambia de posición, por lo tanto los agentes seguidores en esos puntos tendrán un alto error en cuanto a que la distancia al líder es mayor. Por otro lado, después de cada pico es evidente que los agentes buscan al líder de tal manera que se reduce el error. Además, esto evidencia el comportamiento de agregación en la iteración 0 y líder-seguidor en la iteración 200.

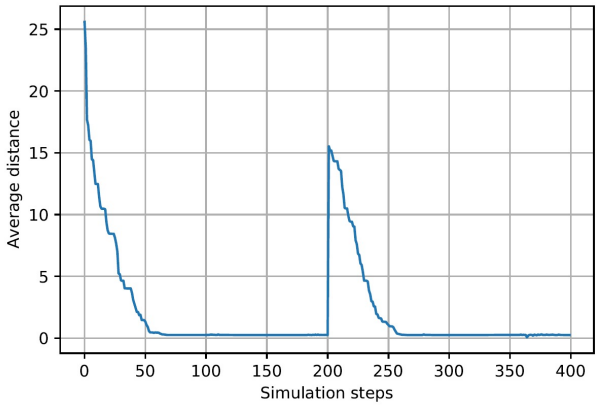

FIGURA 19: Gráfica de error

Glosario

Acciones Medio por el cual se comunica el agente con el entorno.. 21

Agente ref; Drone, Miembro del sistema de aprendizaje.. 7

Algoritmo Conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas.. 1

Aprendizaje maquina campo de estudio de los sistemas que usan la experiencia para la toma de decisiones, su objetivo es generalizar una regla para sus datos.. 9

Aprendizaje por refuerzo Tipo de método de aprendizaje automático en el que el agente recibe una recompensa según su desempeño en cada iteración del sistema. 5

Deep learning Es un campo del aprendizaje máquina, donde por medio del uso de redes neuronales se logra un avance más progresivo y significativo en la generalización de la regla de datos. 9

Drone vehículo aéreo no tripulado.. 7

Entorno Medio en el cual se desempeña el sistema de aprendizaje máquina y obtiene sus entradas.. 21

Escalabilidad capacidad de adaptación y respuesta de un sistema con respecto al rendimiento del mismo a medida que aumentan los componentes de este.. 9, 10

Estado Conjunto de valores que definen la interacción actual del agente en el entorno. 19

Neuronal Ref; Neurona. Aproximación matemática de una neurona biológica. Requiere un vector de entradas, transforma este vector en una única salida.. 23

Recompensa Valor numérico que siguiendo una función se asigna a un agente dependiendo su desempeño en el sistema.. 20

Bibliografía

[1] Saith Rodrıguez y col. «Fast path planning algorithm for the robocup small size league». En: *Robot Soccer World Cup*. Springer. 2014, págs. 407-418.

[2] Saith Rodrıguez y col. «STOx's 2016 Team description paper». En: (2013).

[3] Saith Rodrıguez y col. «STOx's 2015 Extended Team Description Paper». En: *Joao Pessoa, Brazil* (2014).

[4] Jose León y col. «Robot swarms theory applicable to seek and rescue operation». En: *International Conference on Intelligent Systems Design and Applications*. Springer. 2016, págs. 1061-1070.

[5] Juan D Pabon y col. «Event-Triggered Control for Weight-Unbalanced Directed Robot Networks». En: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, págs. 5831-5836.

[6] Nestor I Ospina y col. «Argrohbots: An affordable and replicable ground homogeneous robot swarm testbed». En: *IFAC-PapersOnLine* 54.13 (2021), págs. 256-261.

[7] Edgar C Camacho, Nestor I Ospina y Juan M Calderón. «COVID-Bot: UV-C Based Autonomous Sanitizing Robotic Platform for COVID-19». En: *Ifac-papersonline* 54.13 (2021), págs. 317-322.

[8] Edgar C Camacho, Jose Guillermo Guarnizo, Juan M Calderon y col. «Design and Construction of a Cost-Oriented Mobile Robot for Domestic Assistance». En: *IFAC-PapersOnLine* 54.13 (2021), págs. 293-298.

[9] Laura J Padilla Reyes y col. «Adaptable Recommendation System for Outfit Selection with Deep Learning Approach». En: *IFAC-PapersOnLine* 54.13 (2021), págs. 605-610.

[10] Daniel A Rincón-Riveros y col. «Automation System Based on NLP for Legal Clinic Assistance». En: *IFAC-PapersOnLine* 54.13 (2021), págs. 283-288.

[11] Bharat Rao, Ashwin Goutham Gopi y Romana Maione. «The societal impact of commercial drones». En: *Technology in Society* 45 (2016), págs. 83-90.

[12] U S Robotics. «A Roadmap for US Robotics». En: *Robotics* (2020), págs. 1-90. URL: http://www.hichristensen.com/pdf/roadmap-2020.pdf.

[13] Joshuah K Stolaroff y col. «Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery». En: *Nature communications* 9.1 (2018), págs. 1-13.

[14] Frank Veroustraete. «The rise of the drones in agriculture». En: *EC agriculture* 2.2 (2015), págs. 325-327.

[15] Gustavo A. Cardona y Juan M. Calderon. «Robot swarm navigation and victim detection using rendezvous consensus in search and rescue operations». En: *Applied Sciences (Switzerland)* 9.8 (2019). ISSN: 20763417. DOI: 10.3390/app9081702.

[16] Eliseo Ferrante y col. «"Look out!": Socially-Mediated Obstacle Avoidance in Collective Transport». En: (sep. de 2010), págs. 572-573. DOI: 10.1007/978-3-642-15461-4_66.

[17] Thanh Thi Nguyen, Ngoc Duy Nguyen y Saeid Nahavandi. «Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications». En: *IEEE transactions on cybernetics* (2020).

[18] Manuele Brambilla y col. «Swarm robotics: A review from the swarm engineering perspective». En: *Swarm Intelligence* 7.1 (2013), págs. 1-41. ISSN: 19353812. DOI: 10.1007/s11721-012-0075-2.

[19] Christopher JCH Watkins y Peter Dayan. «Q-learning». En: *Machine learning* 8.3-4 (1992), págs. 279-292.

[20] Volodymyr Mnih y col. «Human-level control through deep reinforcement learning». En: *nature* 518.7540 (2015), págs. 529-533.

[21] Edward Lee Thorndike. «Animal intelligence: An experimental study of the associate processes in animals.» En: *American Psychologist* 53.10 (1998), pág. 1125.

[22] Lucian Busoniu, Robert Babuska y Bart De Schutter. «A comprehensive survey of multiagent reinforcement learning». En: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.2 (2008), págs. 156-172.

[23] Eric Bonabeau y col. *Swarm intelligence: from natural to artificial systems*. 1. Oxford university press, 1999.

[24] Feng WeiXing y col. «Novel algorithms for coordination of underwater swarm robotics». En: *2006 International Conference on Mechatronics and Automation*. IEEE. 2006, págs. 654-659.

[25] Timothy Stirling y Dario Floreano. «Energy Efficient Swarm Deployment for Search in Unknown Environments.» En: (ene. de 2010), págs. 562-563.

[26] Wenguo Liu y Alan FT Winfield. «Modeling and optimization of adaptive foraging in swarm robotic systems». En: *The International Journal of Robotics Research* 29.14 (2010), págs. 1743-1760.

[27] Eliseo Ferrante y col. «Socially-mediated negotiation for obstacle avoidance in collective transport». En: *Distributed autonomous robotic systems*. Springer, 2013, págs. 571-583.

[28] Patricio Cruz y Rafael Fierro. «Autonomous lift of a cable-suspended load by an unmanned aerial robot». En: *2014 IEEE conference on control applications (CCA)*. IEEE. 2014, págs. 802-807.

[29] GA Cardona, D Tellez-Castro y E Mojica-Nava. «Cooperative transportation of a cable-suspended load by multiple quadrotors». En: *IFAC-PapersOnLine* 52.20 (2019), págs. 145-150.

[30] Miguel F Arevalo-Castiblanco y col. «An adaptive optimal control modification with input uncertainty for unknown heterogeneous agents synchronization». En: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, págs. 8242-8247.

[31] Gustavo A Cardona y col. «Adaptive Multi-Quadrotor Control for Cooperative Transportation of a Cable-Suspended Load». En: *2021 European Control Conference (ECC)*. IEEE. 2021, págs. 696-701.

[32] Gustavo A Cardona y col. «Robust adaptive synchronization of interconnected heterogeneous quadrotors transporting a cable-suspended load». En: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, págs. 31-37.

[33] Wilson O Quesada y col. «Leader-Follower formation for UAV robot swarm based on fuzzy logic theory». En: *International Conference on Artificial Intelligence and Soft Computing*. Springer. 2018, págs. 740-751.

[34] Maximilian Hüttenrauch, Sosic Adrian, Gerhard Neumann y col. «Deep reinforcement learning for swarm systems». En: *Journal of Machine Learning Research* 20.54 (2019), págs. 1-31.

[35] Xudong Zhu, Fan Zhang y Hui Li. «Swarm Deep Reinforcement Learning for Robotic Manipulation». En: *Procedia Computer Science* 198 (2022), págs. 472-479.

[36] Sitong Zhang, Yibing Li y Qianhui Dong. «Autonomous navigation of UAV in multi-obstacle environments based on a Deep Reinforcement Learning approach». En: *Applied Soft Computing* 115 (2022), pág. 108194.

[37] E U Robotics AISBL. «Robotics 2020 Multi-Annual Roadmap n for Robotics in Europe, Call 1 ICT23–Horizon 2020». En: *Initial Release B* 15.01 (2014), pág. 2014.

[38] Henrik I Christensen y col. «A roadmap for us robotics: from internet to robotics». En: *Computing Community Consortium* 44 (2009).

[39] David Baldazo, Juan Parras y Santiago Zazo. «Decentralized Multi-Agent deep reinforcement learning in swarms of drones for flood monitoring». En: *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE. 2019, págs. 1-5.

[40] Adrian Cervera Andes. «Coordinación y control de robots móviles basado en agentes». Tesis doct. Universitat Politècnica de València, 2011.

[41] Pedro José Sanz Valero. *Introducción a la robótica inteligente*. 2006.

[42] Rebeca Solıs-Ortega. «Enjambres de robots y sus aplicaciones en la exploración y comunicación». En: *Memorias de congresos TEC*. 2017.

[43] Lukasz Kaiser y col. «Model-based reinforcement learning for atari». En: *arXiv preprint arXiv:1903.00374* (2019).

[44] GA Cardona y col. «Autonomous navigation for exploration of unknown environments and collision avoidance in mobile robots using reinforcement learning». En: *2019 SoutheastCon*. IEEE. 2019, págs. 1-7.

[45] Pablo San José Barrios. «Comparación de técnicas de aprendizaje por refuerzo jugando a un videojuego de tenis». Tesis doct. ETSI_Informatica, 2019.

[46] Richard S Sutton y Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[47] Peter Dayan. «Reinforcement learning». En: *Stevens' Handbook of Experimental Psychology* (2002).

[48] Alba Centeno Franco. «Deep learning». En: (2019).

[49] Ankit Choudhary. «A hands-on introduction to deep Q-learning using OpenAI gym in Python». En: *Retrived from https://www. analyticsvidhya. com/blog/2019/04/introduction-deep-q-learningpython* (2019).

[50] Rafael Berlanga Llavori. «Apuntes de Simulación Informática (curso 2009-2010)». En: (2010).

[51] Israel Garcıa Garcıa y col. «Estudio sobre vehıculos aéreos no tripulados y sus aplicaciones». En: (2017).

[52] Nicolás Gómez y col. «Leader-follower Behavior in Multi-agent Systems for Search and Rescue Based on PSO Approach». En: *SoutheastCon 2022*. IEEE. 2022, págs. 413-420.