UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Implementación de enjambre de robots en operaciones de búsqueda y rescate

Trabajo de graduación presentado por Juan Pablo Cahueque Pérez para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2019

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Implementación de enjambre de robots en operaciones de búsqueda y rescate

Trabajo de graduación presentado por Juan Pablo Cahueque Pérez para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2019

Vo.Bo.:

(f) _____
MSc. Miguel Zea

Tribunal Examinador:

(f) _____
MSc. Miguel Zea

(f) _____
PhD. Luis Rivera

(f) _____
MSc. Pablo Oliva

Fecha de aprobación: Guatemala, 3 de diciembre de 2019.

Prefacio

El trabajo presentado a continuación lleva como título *Implementación de enjambre de robots en operaciones de búsqueda y rescate* y se ha elaborado como requisito de graduación para la licenciatura de Ingeniería Mecatrónica en la Universidad del Valle de Guatemala.

El objetivo de desarrollar como tema de investigación la implementación de teoría de enjambre ha sido el de ser la segunda fase del proyecto Robotat elaborado en la Universidad del Valle de Guatemala, el cual es un proyecto enfocado en impulsar el conocimiento de inteligencia de enjambre, siendo este un proyecto pionero en América Latina. Espero que, como a mí, para el lector este trabajo sea el inicio del interés sobre el tema de Swarm Intelligence y que despierte un interés en aplicar el conocimiento recibido en soluciones innovadoras.

Quiero agradecer a mis padres por su apoyo incondicional durante mi carrera universitaria, a la Fundación Juan Bautista Gutiérrez por darme la oportunidad de obtener una educación superior en la universidad que deseaba, y a mis amigos Aldo Aguilar, Fredy España y Antonio Ixtecoc que durante estos cinco años de carrera universitaria me acompañaron en muchas horas de trabajo. Finalmente, quiero agradecer a mi asesor de tesis, M.Sc. Miguel Zea, por su apoyo para poder realizar un trabajo interesante y relevante como culminación de mi carrera universitaria.

Índice

Lista de figuras

Lista de cuadros

Resumen

El trabajo de búsqueda y rescate es un área social en la que se puede aportar mucho tecnológicamente para la optimización del trabajo en materia de optimización del tiempo usado hoy en día y los riesgos que este trabajo requiere. El objetivo principal de este proyecto consiste en implementar la teoría de enjambres de robots en el área de búsqueda y rescate de personas en zonas de desastres.

Primero, se definieron las funciones artificiales de potencial de Choset y de Kim, Wang y Shin con comportamiento multiplicativo y aditivo como los modelos matemáticos apropiados que convertían la situación del desastre en la función a optimizar por el algoritmo de Particle Swarm Optimization (PSO). Después, se implementó un algoritmo de optimización de partículas en un sistema multiagentes que simuló bajo condiciones ideales el comportamiento del enjambre. Se realizaron iteradas simulaciones para obtener los parámetros del algoritmo PSO que le daban un mejor comportamiento al sistema multiagentes. Luego, se realizaron iteraciones para poder determinar que el modelo de funciones artificiales de potencial de Choset con comportamiento multiplicativo era el modelo más valioso para el trabajo de búsqueda y rescate.

Finalmente se hicieron pruebas en un entorno de simulación de robots en donde se verificó que tanto el modelo propuesto como el algoritmo PSO eran de utilidad en una situación con condiciones más realistas y con un modelo de robot definido en donde el enjambre debía llegar hacia una meta indicada evadiendo obstáculos. Se realizaron diversas simulaciones probando distintos controladores de trayectoria y de pose en los robots lo cual permitió determinar el controlador que le daba un mejor comportamiento al sistema multiagentes enfocado a la evasión de obstáculos en los casos planteados. A través de estas simulaciones se logró determinar además que los parámetros del algoritmo PSO y del modelo de la situación no daban el mismo comportamiento al sistema que condiciones ideales, por lo que se determinaron los parámetros que le daban un mejor comportamiento al sistema en condiciones realistas.

Abstract

The search and rescue field is a field that presents many opportunities to implement technological solutions in order to optimize the time used in present days to perform it and to lower the risks that this field involves. This project is focused on implementing Swarm Theory and evaluating if it is useful to search and rescue people in disaster zones.

First, artificial potencial fields based on models of Choset and Kim, Wang and Shin were proposed as mathematical models in order to transform the situation of a disaster zone in an optimizable function. The idea behind this was to use the Particle Swarm Optimization algorithm as a tool to optimize the function and finding a path for the agents of the swarm. Then, the PSO algorithm was implemented in ideal conditions to a swarm of agents. From this tests, the best ideal parameters of the PSO algorithm and the situational model were defined. Then, iterations were made in order to determine that the Choset model with multiplicative behaviour was the most valuable model for the search and rescue field.

Finally, simulations were run in the robot simulation software, Webots by Cyberbotics. Physical conditions of the developed models of the situations were implemented in this software so that simulations could give an idea of the modifications that have to be done to the PSO algorithm and to the situational model in order to implement them in the real world. Aditionally, several controllers were tested to control the velocity and trajectories of the robots and to determine if the behaviour of the swarm was valuable for the search and rescue field.

CAPÍTULO 1

Introducción

El algoritmo de Particle Swarm Optimization fue por primera vez introducido por Kennedy y Eberhart en su trabajo *Particle swarm optimization (PSO)* [1] en 1996 y desde entonces ha sido un algoritmo de optimización en auge en el área de informática y objeto de muchas investigaciones. Asimismo, el área de búsqueda y rescate ha sido de gran objeto de estudio y se han desarrollado diversos trabajo de robótica para poder asistir en la búsqueda de las personas de manera efectiva y en vistas a disminuir el riesgo para los rescatistas. Sin embargo, en trabajos como [2], se menciona en su mayoría el diseño mecánico y eléctrico de los robots, pero hace falta la característica que vuelva a su enjambre autónomo y que haga su trabajo de una manera colaborativa y eficiente.

En el trabajo a continuación se realizaran las pruebas en condiciones ideales, sin tomar en cuenta restricciones de un robot en específico, para poder determinar el modelo matemático que en combinación con el algoritmo de Particle Swarm Optimization puedan llegar a optimizar el rendimiento del sistema de agentes robóticos para llegar a una meta.

Finalmente, se realizaran simulaciones en el programa de simulación de robots WeBots de Cyberbotics, en donde se modelaran las características físicas de un robot diferencial utilizado en el área de aprendizaje de enjambres de robots de la École Polytechnique Fédérale de Lausanne (EPFL). En este entorno de programación se modelaran físicamente las situaciones de prueba para los modelos planteados. Por medio de las simulaciones con un enjambre de robots en condiciones realistas se verificará que tanto el modelo propuesto como el algoritmo PSO son implementables y valiosos para el trabajo de búsqueda y rescate.

CAPÍTULO 2

Antecedentes

2.1. Trabajos académicos previos

Principalmente se debe mencionar la fase previa a este trabajo: la primera fase del Robotat de la Universidad del Valle de Guatemala, la cual consistió en el diseño y manufactura de la mesa de pruebas, el prototipo de los primeros agentes y la implementación de una red Wi-Fi en la plataforma de pruebas. El trabajo logró movilizar a dos agentes con una trayectoria y presentó una oportunidad para poder hacer pruebas de optimización de los modelos matemáticos que pudieran coordinar la movilidad de un sistema multiagentes a través de la mesa en trayectorias mas complejas.

Como otro trabajo previo se puede mencionar el trabajo de Stormont y Berkemeier [2], quienes realizaron una descripción sobre el proyecto conocido como Blue Swarm, el cual es un esfuerzo por desarrollar un enjambre de robots de rescate completamente funcional. En este trabajo se mencionan las características que debería tener el enjambre con el fin de ser de utilidad para los encargados de búsqueda y rescate. Se menciona que los agentes del enjambre deberían ser completamente autónomos para que no se necesite a una persona especializada que maneje los robots. Además, estos agentes no deben representar ningún peligro tanto para las víctimas como para la seguridad de los rescatistas, por lo que establece que los robots deben ser pequeños para no ser obstáculos en el caminos de las personas. Finalmente, el costo de los robots debe ser mínimo ya que se debe tomar en cuenta la posibilidad de que se pierdan algunos durante el trabajo de búsqueda, y que la pérdida de uno o varios sea fácilmente superado.

En dicho trabajo se describe el desarrollo que ha tenido el proyecto a lo largo de las competencias en las que ha participado. Se describe principalmente el diseño de los prototipos a un nivel de abstracción bajo, pero no se menciona el control o el algoritmo usado para manipular al enjambre, únicamente se menciona la futura implementación de un algoritmo de comportamiento de enjambres que permita que los agentes sean repelidos por otros robots o por objetos que muestren señales de ser obstáculos y que no muestren señales de ser una víctima.

Otro trabajo similar se puede encontrar en el trabajo *Robot swarms theory applicable to seek and rescue operation* [3], en donde se muestra la implementación de un algoritmo PSO en un enjambre de robots simulado en MATLAB. En dicho trabajo se menciona el uso de un algoritmo PSO denominado "darwiniano", cuyo nombre se inspira en la característica de evolución, o añadimiento de más agentes, de un sub-enjambre a medida que la búsqueda se vuelve exitosa. Si el sub-enjambre no es exitoso en el trabajo de búsqueda, éste es propenso a desaparecer, lo que se traduce en la dispersión de sus agentes.

En este trabajo se define el modelo matemático del comportamiento que cada agente debe tener tanto con el entorno que lo rodea, como con otros agentes circuncidantes. En éste modelo se definen fuerzas de repulsión de los obstáculos y otros agentes, y fuerzas de atracción hacia la meta. Finalmente se muestran resultados gráficos de las simulaciones realizadas en entornos significantes.

De todos los trabajos realizados con el algoritmo de Particle Swarm Optimization, el 3.4 por ciento se ha realizado en el área de robótica, lo cual incluye aplicaciones de control de manipuladores y brazos, planeación de trayectorias, búsqueda colectiva, evasión de obstáculos y róbotica de enjambres [4].

2.2. Oportunidad de solucionar una problemática nacional

Respecto a la situación nacional, se puede mencionar que la mayor parte de la población vive en situación de altísimo riesgo o en condiciones de amenaza de derrumbe e inundación, como lo indica el trabajo *¿ Se repetirá la tragedia de El Cambray II* [5]. Además, el potencial sísmico de la región es variable, pudiendo generarse terremotos de magnitudes máximas con distintas leyes de recurrencia gracias a sismos asociados a la Depresión de Honduras, la actividad de la fuente de volcanes en Centroamérica, el sistema de fallas Chixoy-Polochic-Motagua y la fosa de subducción asociada al límite de placas Cocos y Caribe [6].

Según esto, se realizó una entrevista al encargado de formación de la Brigada de Búsqueda y Rescate del Cuerpo de Bomberos Voluntarios de Guatemala, Hector Sicajá, para recabar información desde la experiencia y el punto de vista local. En dicha entrevista se estableció que los casos mas recurrentes de estructuras localmente (esto incluye México, Guatemala, y El Salvador) son los denominados en campo como nido de golondrinas, 'en V' y 'de panqueque', los cuales corresponden a nido de golondrinas, oblicuo y superposición de planos respectivamente en [7]. También se estableció que, en el caso de superposición de planos, el acceso a espacios de vida es muy difícil y es el caso donde el tiempo crítico se reduce debido a que los espacios de vida suelen ser más pequeños.

Los trabajos y hechos presentados anteriormente respaldan la viabilidad y necesidad de realizar el siguiente trabajo, ya que en trabajos similares se ha logrado elaborar un prototipo, pero se necesita un algoritmo que permita realizar eficientemente la navegación del enjambre, mientras que la situación nacional es propensa a desastres naturales como derrumbes y colapso de estructuras y por consiguiente generan la necesidad y oportunidad de soluciones óptimas para el trabajo de búsqueda de personas en zonas damnificadas.

CAPÍTULO 3

Justificación

Guatemala es un país centroamericano único en el continente por su número sin igual de volcanes activos, una topografía diversa y una variedad interesante de microclimas subtropicales. Aún así, como cualquier otro país, Guatemala se ve afectada constantemente por los acontecimientos que por su naturaleza son impredecibles o imparables por el ser humano. Por ejemplo, la actividad sísmica fuerte en el caso del terremoto de 1972, recientemente el terremoto que afectó el suroccidente en 2012 o la erupción del Volcán de Fuego en 2018 que afectó las comunidades aledañas por la expulsión de material piroclástico [6]. Algunos otros acontecimientos cuentan con más tiempo para realizar un estudio y así poder realizar la preparación adecuada. Lastimosamente, siendo Guatemala un país en vías de desarrollo no se le da una alta prioridad al plan de acción y ocurren tragedias como el derrumbe ocurrido en el Cambray II [5].

Debido a que el territorio es muy propenso ante estos desastres y existe un problema en la preparación ante ellos, se presenta la oportunidad de ayudar ante la necesidad de la búsqueda y rescate de personas que se ven afectadas en el lugar de la catástrofe. Actualmente, este trabajo se realiza por parte de personal de la CONRED, la Cruz Roja Guatemalteca, bomberos locales y grupos de voluntarios. Pero en muchos de estos casos, los lugares pueden ser inestables o representar un peligro para el ser humano. Por ejemplo, se pueden dar condiciones extremas, como en el caso de altas temperaturas, toxicidad del ambiente, combinación de material piroclástico y lluvias constantes, inestabilidad de estructuras colapsadas, o simplemente espacios demasiado angostos de alcanzar por los voluntarios.

Una vez planteado el problema anterior, se presenta ante mí una oportunidad para poder integrar los conocimientos obtenidos a lo largo de la carrera de ingeniería mecatrónica en mi proyecto de graduación. Haciendo uso de estos conocimientos se plantea como solución la implementación de robots móviles, ya que estos tienen ventajas sobre los humanos ante los ambientes tan difíciles que dejan las catástrofes. Estos podrían atravesar espacios mas reducidos, soportar condiciones extremas y representar un menor costo en el caso de perderlo, en comparación a la vida de un rescatista [2]. Además, se busca manejar estos robots con algoritmos inspirados en el comportamiento de los enjambres para poder realizar un trabajo

óptimo en estos casos de vida o muerte.

Finalmente, el trabajo a realizar representa una satisfacción personal en el área de servicio social ya que tengo la firme convicción de que como profesionales guatemaltecos tenemos el deber de ser agentes de cambio en nuestras comunidades y aprovechar nuestra posición para el bien del prójimo.

CAPÍTULO 4

Objetivos

4.1. Objetivo general

Implementar un algoritmo de optimización de partículas organizadas en enjambre enfocada al trabajo de búsqueda y rescate.

4.2. Objetivos específicos

- Plantear un modelo de optimización de enjambre de partículas de relevancia para el trabajo de búsqueda y rescate de personas en zonas de desastres naturales.
- Producir simulaciones computarizadas sobre el comportamiento de las partículas organizadas en enjambre para validar el comportamiento en condiciones ideales.
- Implementar el modelo propuesto en el entorno de simulación Webots de Cyberbotics para validar el comportamiento en condiciones realistas.

CAPÍTULO 5

Alcance

Este proyecto tiene como alcance implementar un algoritmo de enjambres de partícula que permita la navegación de un sistema multiagentes en simulaciones de entornos desarrollados en base al trabajo que se hace para la búsqueda de personas atrapadas en estructuras colapsadas.

Para el desarrollo de esta propuesta, se elaboraran modelos de entornos que ejemplifiquen los casos mas comunes en el trabajo de búsqueda de personas realizado localmente para que sea de mayor relevancia en el ámbito nacional. Asimismo, se implementaran varios modelos matemáticos para poder simular tanto los entornos como el algoritmo de navegación de los agentes hasta la meta, la cual simulará ser una persona en la zona del desastre. El alcance del proyecto consiste llegar a realizar simulaciones tanto con condiciones ideales como con condiciones realistas que determinen si el modelo matemático es ideal para la navegación de los agentes. Las simulaciones generadas en el entorno Webots en condiciones realistas contienen ciertas ventajas en comparación con realizarlas en una plataforma real, tales como acceder a las coordenadas globales de los robots en cualquier momento y de cualquier agente y lograr detectar obstáculos en todas las direcciones con el modelo de robot utilizado.

Las simulaciones generadas, especialmente las realizadas con condiciones realistas, pretenden ser de utilidad para determinar si es posible implementarlos en un futuro proyecto de sistema multiagentes, que pueda ser probado en la plataforma de pruebas Robotat de la Universidad del Valle de Guatemala.

CAPÍTULO 6

Marco teórico

6.1. Estructuras colapsadas

6.1.1. Derrumbes

Se considera un derrumbe como el colapso y por consiguiente el total o parcial desprendimiento de una construcción [7].

Los derrumbes pueden ser clasificados en dos grandes grupos debido a su origen o causa: naturales y antrópicos.

Derrumbes por causas naturales

Se caracterizan por originarse a partir de fenómenos naturales como lluvia, terremotos, huracanes, inundaciones, entre otros que sobrepasan una escala normal de intensidad. A continuación, se detallan los mas comunes.

- Terremoto: consiste en sacudidas bruscas y pasajeras de corteza terrestre, producida por liberación de energía acumulada en forma de ondas sísmicas. La liberación de energía puede ser causada tanto por fenómenos naturales (i.e. movimiento de placas tectónicas, procesos volcánicos o ruptura de fallas geológicas) como por actividades humanas (i.e. 'fracking' y 'explosiones nucleares').
- Alud: consiste en en el deslizamiento de grandes masas de tierra, rocas, árboles y casas provocado por terremotos, erupciones volcánicas, o inestabilidad en el área. Entre las modalidades que se da este fenómeno se puede mencionar el deslizamiento, el flujo de arcilla, licuefacción y reptación.
- Riesgos Volcánicos: consiste en el riesgo debido al ascenso de material volcánico debido a la actividad violenta de los volcanes. Las avalanchas de material volcánico

constituyen uno de los mayores riesgo de este tipo ya que poseen una gran capacidad destructiva. Por lo general estas incluyen lava, ceniza volcánica, barro acumulado y gases provenientes de la tierra.

Derrumbes por causas antrópicas

Este tipo de derrumbes tienen en común la acción humana, y sus causas mas comunes son falta de mantenimiento, incendios y ataques terroristas.

A continuación se detallan las causas de derrumbes de origen antrópico:

- Falta de mantenimiento en la edificación
- Reformas en la edificación
- Reformas en la edificación
- Construcción de edificaciones paralelas
- Explosiones de gas
- Atentados terroristas
- Incendios
- Mala ejecución de derribos
- Impacto de vehículos

6.1.2. Tipos de derrumbes

Desde el punto de vista del cuerpo de bomberos el estudio de derrumbes se centra en espacios se puedan encontrar sobrevivientes con mayor probabilidad. Dichos espacios son conocidos como 'huecos de vida' y según estas se clasifican las estructuras colapsadas en distintas clasificaciones.

A continuación, se describen las tipologías básicas de especial importancia:

Figura 1: Tipos de estructuras colapsadas, de izquierda a derecha:
oblicuo, derrumbe total, y marquesina

Oblicuo, lateral o de plano inclinado

Se define como el colapso de pilares o muros de carga de un edificio de manera que colapsa el forjado superior quedando parcialmente apoyado en pilar gemelo y en el forjado inferior formando huecos de vida.

En este caso las víctimas que se encontraban el el forjado superior yacen al pie del plano inclinado, mientras que se encontraban el plano inferior pueden encontrarse bajo los escombros.

Superposición de planos o derrumbe total

En este caso, se define como el colapso de varias plantas, unas sobre otras, quedando en posiciones mas o menos horizontales o inclinadas. Las víctimas se encontrarán entre los estratos y los posibles sobrevivientes entre las oquedades.

La localización de las víctimas es sumamente difícil y se realiza regularmente con agentes caninos y equipos de escucha.

Marquesina

Se define como el colapso estructural que genera un hueco libre entre escombros, comprendido entre un plano inclinada, un muro y el suelo.

Espacio relleno

Se define como el colapso de la totalidad del interior de una estancia, que queda completamente llena de escombros.

En estos casos la formación de huecos es casi inexistente.

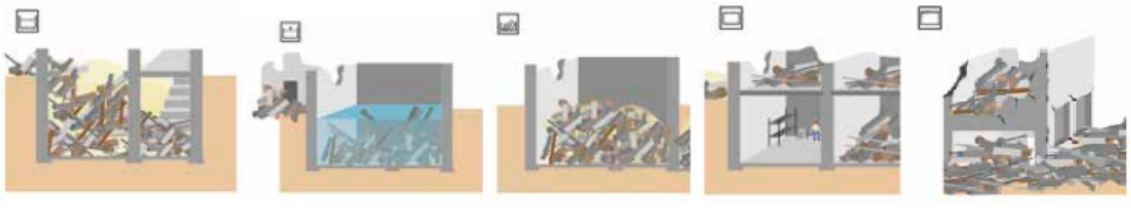


Figura 2: Tipos de estructuras colapsadas, de izquierda a derecha: espacio relleno, espacio inundado-embarrado, espacio estratificado, tapón de escombros y local impactado.

Espacio inundado-embarrado

Este caso es similar al espacio relleno, pero ahora cubierto de agua y lodo.

Los escombros forman una masa consistente con muy elevado riesgo de asfixia o ahogamiento, por lo que las posibilidades de supervivencia son muy bajos.

Espacio estratificado

Se define como un recinto cuyos muros permanecen de pie y lleno de escombros estratificados o comprimidos.

Tapón de escombros

En este caso se genera por el colapso parcial de edificaciones, las cuales bloquean zonas de acceso. Las víctimas corren el riesgo de asfixia por falta de aire, intoxicación con gas o ahogamiento por una fuga de agua.

Antes de desalojar el tapón, se debe comprobar que no existen escombros inestables pendientes de caída.

Local impactado

Se define como un recinto que ha sufrido los efectos de una explosión y conserva su estructura principal, pero con una solidez dudosa.

Nido de golondrinas

Se define como un recinto que ha sufrido un colapso en los cerramientos circulares de una explosión, pero que conserva en mayor medida el suelo o pisos. En este caso la estructura presenta una gran inestabilidad y el peligro de hundimiento es muy grande.

Figura 3: Tipos de estructuras colapsadas, de izquierda a derecha: nido de golondrinas, escombros adosados al exterior, escombros dispersos en el exterior, derrumbamiento de cono de escombros.

Escombros adosados al exterior

Consiste en un cúmulo de escombros en el exterior de un recinto dañado. En cualquier parte de los escombros se pueden hallar víctimas, tales como personas que hayan sido expulsadas del edificio o transeúntes que fueron sorprendidos. Generalmente, se forma un cúmulo compacto y continuo al pie de la fachada.

Escombros dispersos en el exterior

Consiste en cúmulos de escombros proyectados fuera de un edificio o escombros de objetos destruidos en el exterior sobre la vía pública. En este caso también se pueden encontrar víctimas debajo de los escombros.

Derrumbamiento de cono de escombros

Consiste en un cúmulo de escombros correspondientes al derrumbamiento total de un edificio, con la estructura totalmente destruida y desmembrada. Este tipo de colapso puede contener otros tipos de derrumbes y puede tener víctimas en cualquier lugar del cono.

6.2. Funciones artificiales de potencial

Según Choset[8], una función de potencial es una función derivable definida en los números reales $U : \Re \longrightarrow \Re^m$ cuyos valores pueden ser vistos como energía, y por lo tanto su gradiente se puede ver como una fuerza. El gradiente es un vector que apunta en la dirección que maximiza a la función U localmente. El enfoque de usar Funciones Artificiales de Potencial dirige al robot al movimiento de una trayectoria como si fuera una partícula cargada positivamente que es atraída por una fuerza negativa, en este caso la meta, y repelida por fuerzas positivas representadas en este caso por obstáculos. La combinación de las fuerzas negativas y las fuerzas positivas conducen al robot esperadamente a la meta evadiendo los obstáculos.

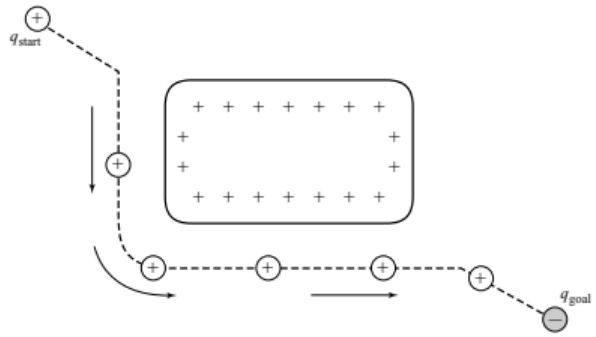


Figura 4: Combinación de fuerzas de potencial que conducen al robot a la meta [8]

6.2.1. Función de Choset

La suma de los efectos de repulsión y atracción en el robot constituyen a la forma más simple de función de potencial.

$$U(q) = U_{att}(q) + U_{rep}(q). \tag{1}$$

Potencial de atracción

La función de potencial de atracción según Choset está definida como una función por partes que satisface ciertos criterios.

$$U_{att}(q) = \begin{cases} \frac{1}{2}\zeta d^2(q, q_{goal}) & d(q, q_{goal}) \leq d^*_{goal} \\ d^*_{goal}\zeta d(q, q_{goal}) - \frac{1}{2}\zeta (d^*_{goal})^2 & d(q, q_{goal}) > d^*_{goal} \end{cases} \tag{2}$$

Donde ζ es el parámetro que escala el efecto de atracción del potencial, $d^2(q, q_{goal})$ es la distancia de q a q_{goal} y d^*_{goal} es un valor de umbral de distancia donde la función realiza una frontera entre las dos partes de la función.

Como se observa, la función está definida por dos partes; la primera trata de una función cuadrática que atrae al robot cuando se encuentra cerca de q_{goal} y la segunda atrae al robot cuando el robot se encuentra más distante de q_{goal} que el umbral definido. Esta función es monótonamente creciente en relación a la distancia hacia q_{goal}, es continuamente diferenciable incluso en la frontera entre las partes cuadrática y cónica y es cero cuando el agente llega a q_{goal}.

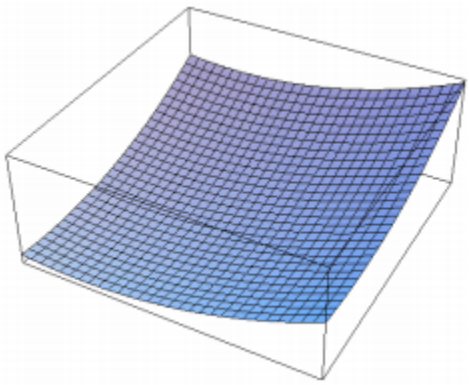


Figura 5: Ejemplo de modelo de potencial de atracción según Choset[8]

Potencial de repulsión

La función de repulsión, como su nombre lo indica, mantiene al agente lejos de un obstáculo dependiendo de la distancia entre estos dos. Mientras mas cerca se encuentre el robot al obstáculo, más fuerte sera la energía de repulsión entre estos dos.

La función de potencial de repulsión está definida en términos de la distancia mínima del agente al punto más cercano de cada obstáculo individualmente gracias a la función $d_i(q) = \min_{c \in QO_i} d(q, c)$ que devuelve el menor $d(q, c)$ para los puntos c en QO_i.

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta(\frac{1}{d_i(q)} - \frac{1}{Q^*})^2 & d_i(q) \leq Q^* \\ 0 & d_i(q) > Q^* \end{cases} \tag{3}$$

Donde $Q^* \in \Re$ es un factor que define el tamaño del dominio de influencia del obstáculo QO_i sobre el robot y η es un parámetro de ganancia para la repulsión. Tanto estos parámetros como el factor Q^*son determinados heurísticamente a lo largo del proceso de prueba del algoritmo.

Luego de calcular el potencia de repulsión para cada objeto QO_i se obtiene el potencial de repulsión total $U_{rep}(q) = \sum_{i=1}^{n} U_{rep_i}(q)$ Asumiendo que solamente existen objetos convexos o no convexos que se pueden descomponer en piezas convexas, no ocurren oscilaciones ya que no existen cambios radicales hacia el punto mas cercano.

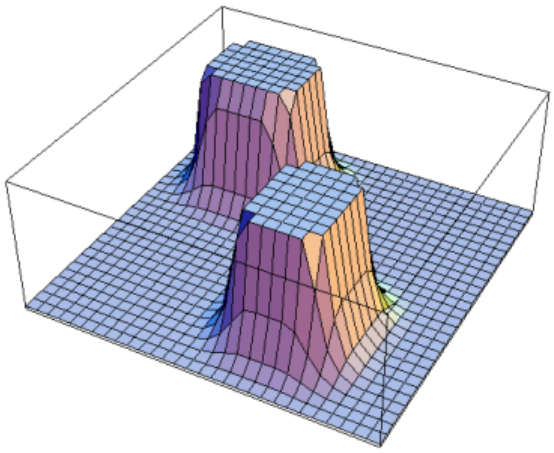


Figura 6: Ejemplo de modelo de potencial de repulsión según Choset [8]

Superposición de potenciales

Luego de calcular los potenciales de atracción y de repulsión, estos se suman para crear el potencial total del robot en el campo definido:

$$U(q) = U_{att}(q) + U_{rep}(q)$$

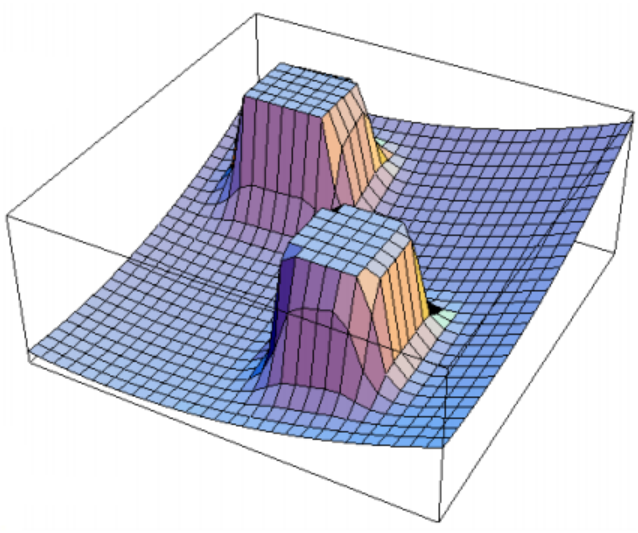


Figura 7: Ejemplo de modelo de superposición de potenciales según Choset [8]

6.2.2. Función de Kim, Wang y Shin

En este caso, se tiene la misma idea que el modelo anterior, la cual consiste en la migración del grupo hacia una meta evitando la colisión con los obstáculos. Se plantea una nueva arquitectura de comportamiento para cada agente, aunque también se puede formar la función de potencial artificial superponiendo las funciones de atracción y de repulsión. [9]

Potencial de atracción

Se define un vector de posición relativa del agente respecto de la meta como

$$\psi_i^g = P_i - P_{goal} \tag{4}$$

donde P_i es la posición del i-ésimo agente y P_{goal} es la posición de la meta. Dicho vector de posición indica que la posición de la formación es independiente de la posición global del grupo.

Habiendo definido el vector de posición relativa, el vector de atracción de los agentes a la meta para la migración se define así:

$$U_i^g = c_g(1 - \exp\{-\frac{\|\psi_i^g\|^2}{l_g^2}\}) \tag{5}$$

donde c_g es la distancia de intensidad y l_g es la distancia de correlación para migración grupal. El término c_g del lado derecho de la ecuación se emplea para hacer cero a U_i^g cuando $\psi_i^g = 0$.

Potencial de repulsión

Se define un vector de posición relativa del agente respecto de un obstáculo como

$$\psi_j^o = P_i - O_j \tag{6}$$

donde P_i es la posición del i-ésimo agente y O_j es la posición del obstáculo j el cual es próximo al agente i.

La función de potencial de repulsión para evitar obstáculos se define de la siguiente manera.

$$U_i^o = \sum_{j \in \aleph} \{c_o \exp\{-\frac{\|\psi_i^o\|^2}{l_o^2}\}\} \tag{7}$$

donde c_o es la distancia de intensidad y l_g es la distancia de correlación para repulsión de obstáculos.

Potencial con estructura aditiva

El potencial total para una configuración convencional se construye combinando los potenciales de atracción y de repulsión en una estructura aditiva.

$$U_i^{og} = U_i^o + U_i^g \tag{8}$$

$$U_i^{og} = \sum_{j \in \aleph} \{c_o \exp\{-\frac{\|\psi_i^o\|^2}{l_o^2}\}\} - c_g \exp\{-\frac{\|\psi_i^g\|^2}{l_g^2}\} + c_g \tag{9}$$

Potencial con estructura multiplicativa

Por motivos de evitar problemas en los mínimos locales existe una estructura multiplicativa y aditiva entre los potenciales de migración del grupo y de la repulsión hacia obstáculos.

$$U_i^{og} = \frac{1}{c_g} U_i^o \cdot U_i^g + U_i^g \tag{10}$$

$$U_i^{og} = \sum_{j \in \aleph} \{c_o \exp\{-\frac{\|\psi_i^o\|^2}{l_o^2}\}\}(1 - \exp\{-\frac{\|\psi_i^g\|^2}{l_g^2}\}) - c_g \exp\{-\frac{\|\psi_i^g\|^2}{l_g^2}\} + c_g \tag{11}$$

6.3. Inteligencia de enjambre

El concepto de inteligencia de enjambre se refiere a la habilidad de resolver problemas mediante la interacción entre unidades simples de procesamiento de información. El hecho de que se incluya el término enjambre sugiere que se agrega multiplicidad y aleatoriedad al concepto. Las unidades de procesamiento de información pueden ser, en términos generales,

de cualquier tipo: animales, modelos matemáticos, simulaciones o humanos; pero todas estas deben cumplir con el requisito de intercambio de información. En especial, el enfoque de *Particle Swarm Optimization*, también conocido como PSO por sus siglas en inglés, es un paradigma que ha sobresalido como uno de los enfoques heurísticos más populares de los últimos años. Este algoritmo se basa en replicar las dinámicas del comportamiento social observadas en la naturaleza, por ejemplo, enjambres de abejas, bancos de peces, o bandadas.

6.3.1. Particle swarm optimization

En general, cualquier algoritmo de optimización asociado al PSO cuenta con tres características esenciales. La primera consiste en emplear una población de partículas, cuyo número no está establecido pero que por lo usual se encuentra entre los veinte o cincuenta agentes.

La segunda característica describe la topología de las interacciones entre los miembros de la población. Las topologías tradicionales son *gbest* y *lbest*. La topología *gbest* está asociada al efecto que una población que se encuentra completamente conectada. Esta topología sirve para mantener el dato de la partícula con una mejor solución entre toda la población. La topología *lbest* se encarga de mantener conectada a cada partícula con las partículas más próximas a ella. La ventaja de esta topología sobre *gbest* es que las subpoblaciones pueden converger a diferentes valores óptimos en el espacio del problema.

La tercera característica consiste en la asignación de una regla de cambio para cada partícula. A pesar de las versiones que se han creado para estas reglas de cambio, por lo general se usa la topología en la que el siguiente paso depende tanto del paso actual como del éxito logrado anteriormente y el éxito logrado por los demás.

6.3.2. Algoritmo PSO con preservación de diversidad

El trabajo de Chowdhury[10] ofrece una variante del PSO convencional, en donde se agrega un vector de diversidad para evitar una congregación prematura de las partículas. Este factor es de especial interés para el trabajo de exploración, especialmente en el caso del trabajo de búsqueda, ya que se necesita maximizar el área cubierta por los agentes.

En este caso, la dinámica de movimiento de cada partícula se modela de la siguiente manera:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{12}$$

$$V_i^{t+1} = \alpha V_i^t \beta_l r_1 (P_i - X_i^t) + \beta_g r_2 (P_g - X_i^t) + \gamma r_3 \hat{V}_i^t \tag{13}$$

donde,

- X_i^t y X_i^{t+1} son las posiciones de la i-ésima partícula en la iteración t y $t+1$ respectivamente;
- V_i^t y V_i^{t+1} son los vectores de velocidad de la i-ésima partícula en la iteración t y $t+1$ respectivamente;

- r_1, r_2 y r_3 son números reales aleatorios entre 0 y 1;
- P_i es la mejor solución candidata encontrada para la i-ésima partícula;
- P_g es la mejor solución candidata encontrada para toda la población (también conocida como *gbest*);
- α, β_l y β_g son los coeficientes definidos por el usuario para controlar los atributos inerciales, explorativos y de *exploit* respectivamente del movimiento de cada partícula;
- γ es el coeficiente de preservación de diversidad; y
- el término $\gamma r_3 \hat{V}_i^t$ es el término de preservación de diversidad.

Figura 8: Dinámica modificada de las partículas de acuerdo a Chowdhury [10]

6.4. Robot diferencial e-puck

El e-puck es un robot móvil didáctico desarrollado por la École Polytechnique Fédérale de Lausanne (EPFL). Tanto el hardware como el software del e-puck son *open source* y se incluye un modelo en la librería de Webots. Este robot cuenta con una gran variedad de sensores, es de una escala pequeña y su programación es amigable por lo que ha sido usado por varios años como herramienta de aprendizaje en cursos de Swarm Intelligence [11].

El e-puck fue desarrollado para cumplir con las siguientes características [12]:

- Una arquitectura electro-mecánica simple pero robusta.
- Flexibilidad y variedad en los sensores, lo que le da una rango amplio de actividades educacionales e ingenieriles.
- Amigable con el usuario y de programación y control sencillo.
- Costo reducido para el alcance del presupuesto universitario.
- Robusteza suficiente para poder despreocuparse del mantenimiento en un largo período de tiempo.

6.4.1. Características del modelo físico

A contiuación se describen las características físicas principales del modelo e-puck implementado en el programa de simulación Webots de Cyberbotics.

Característica	Valor
Diámetro	71 mm
Altura	50 mm
Radio de la rueda	20.5 mm
Largo del eje	52 mm
Peso	0.16 kg
Máxima velocidad de avance/retroceso	0.25 m/s
Máxima velocidad de rotación	6.28 rad/s

Cuadro 1: Características físicas del robot e-puck

6.4.2. Equipamiento del e-puck

A continuación se describen las características principales de los sensores y los componentes principales equipados en el robot e-puck.

Característica	Descripción
Batería	3 horas provistas por la batería recargable de LiIon de 5Wh
Procesador	Microchip dsPIC 30F6014A @ 60MHz
Motores	2 motores stepper con 20 pasos por revolución y un engrane de reducción 50:1
Sensores IR	8 sensores infrarrojos que miden la luz del ambiente y la proximidad de obstáculos con un rango de 4 cm
Cámara	Cámara a color con una resolución máxima de 640x480
Micrófonos	3 micrófonos omni-direccionales para localización de sonido
Acelerómetro	Acelerómetro 3D a lo largo del eje X, Y y Z
LEDs	8 LEDs rojos y uno verde
Speaker	Speaker capaz de reproducir archivos WAV
Switch	Switch de rotación de 16 posiciones
Bluetooth	Bluetooth para comunicación inalámbrica robot-computadora y robot robot
Programación	Programación en lenguaje C a través del compilador GNU GCC

Cuadro 2: Descripción de equipamiento del robot e-puck

CAPÍTULO 7

Generación de funciones artificiales de potencial aplicados al trabajo de búsqueda y rescate

7.1. Generación de funciones artificiales de potencial

Las funciones artificiales de potencial se elaboraron en la versión R2017a de MATLAB, el cual es un entorno de desarrollo creado por MathWorks integrado con un lenguaje de programación propio para ingenieros y científicos. Los modelos generados se basaron en un cuadro de 20 por 20 unidades para simular una mesa de pruebas, como el Robotat de la Universidad del Valle de Guatemala por ejemplo.

Al utilizar MATLAB para elaborar las funciones de potencial se logró programar de forma matricial la generación de estas funciones por lo que se pudo optimizar el tiempo en cada simulación. El uso de las funciones *inpolygon*, *repmat*, *min* y condicionales (<,> y =), cuyo valor de retorno eran matrices binarias que cumplían estas condiciones, permitieron una programación óptima de las funciones de distancia hacia la meta y de distancia mínima de un punto hacia un obstáculo necesarias para poder calcular los potenciales de atracción y de repulsión tanto en el modelo de Choset [8] como en el modelo de Kim, Wang y Shin [9].

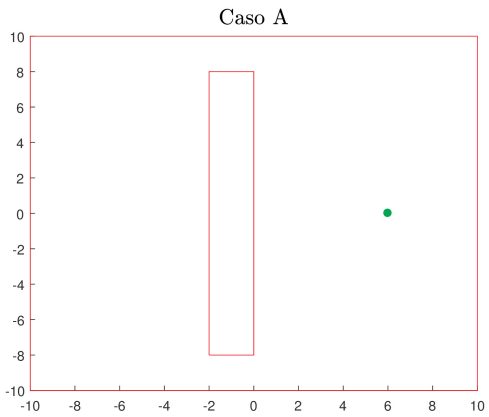
7.2. Casos de interés para el trabajo de búsqueda y rescate

Se generaron tres tipos de casos de acuerdo a lo recabado en la entrevista con el especialista de rescate, el cual estableció que los casos más comunes eran los de colapso oblicuo, colapso total y nido de golondrinas. Además de la entrevista, se tomó como base lo observado en trabajos similares, tal como [3], y se tomó la suposición de que la generación de estos

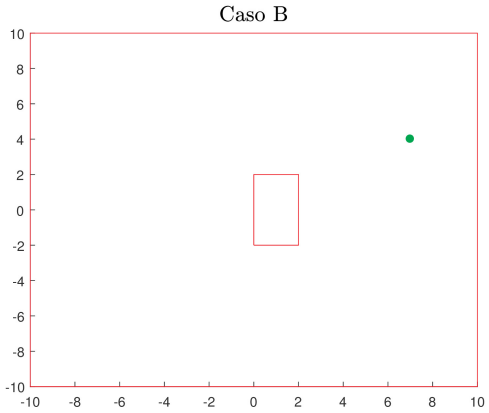
modelos es posible gracias a un observador omnisciente, que en el trabajo de búsqueda y rescate podría tratarse de un cuadricóptero.

Se generó un caso A, el cual se puede observar en la Figura 9a,en el que se interpone un obstáculo significativamente más grande que cada agente individualmente, simulando la obstaculización de objetos grandes que usualmente se encuentran en oficinas y casas, tales como escritorios, mesas, camas, paredes y vigas colapsadas. También se generó un caso B, el cual se puede observar en la Figura 9b, en el que se simula la obstaculización del paso por objetos de mediano tamaño en comparación al tamaño de los agentes. Finalmente se generó un caso C, el cual se puede observar en la Figura 9c, en el cual se obstaculiza el paso de los agentes por pequeños objetos, pero estos en mayor número y dispersos en el espacio de búsqueda.

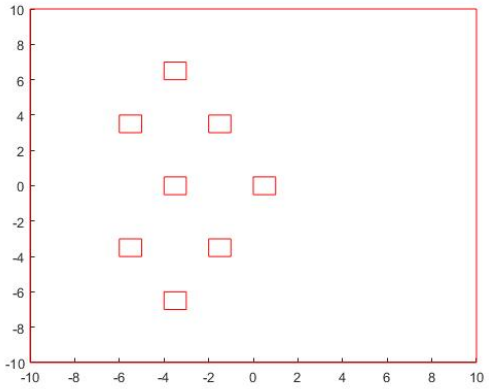
La idea detrás de haber generado estos casos es poder representar de forma general los casos más comunes de obstaculización que se puedan encontrar los agentes en un área de estructura colapsada. Además cada caso representa un nivel de dificultad distinto, ya que se supone que la navegación en un espacio con obstáculos pequeños debería ser más sencilla y su dificultad va incrementando de acuerdo a la cantidad de obstáculos dispersos y el tamaño de los obstáculos.



(a) Caso generado A



(b) Caso generado B

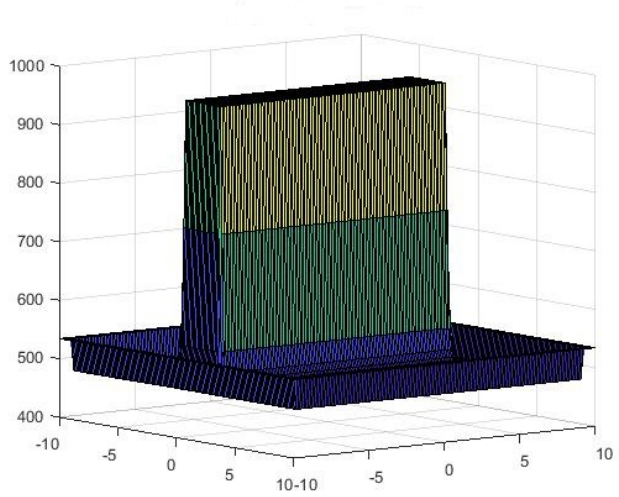


(c) Caso generado C

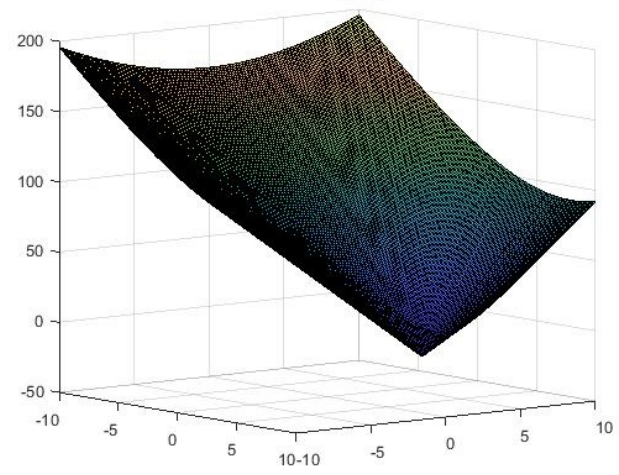
Figura 9: Casos de prueba para el algoritmo PSO

7.3. Función artificial de potencial basada en la función de Choset

Se generaron las funciones artificiales de potencial tomando el modelo de Choset [8]. Para la función de repulsión (ecuación 3) se usaron los valores de 10 para η y 5 para Q^*. Para la función de atracción (ecuación 2) se usaron los valores de 5 para ζ y 2 para d^*_{goal} hallados heurísticamente. Se puede observar, en las figuras 10, 11 y 12 que se generaron las gráficas de la función de repulsión y de la función de atracción del caso A, lo cual confirma lo aprendido en el trabajo de Choset [8]. Se puede observar en las figuras 10a, 11a y 12a que el potencial de un obstáculo es de un valor numérico mas grande a medida que la distancia hacia el obstáculo es menor, mientras que en el caso del campo de potencial de atracción el valor tiende a decrecer mientras se aproxima a una meta, como se puede ver en las figuras 10b,11b y 12b.

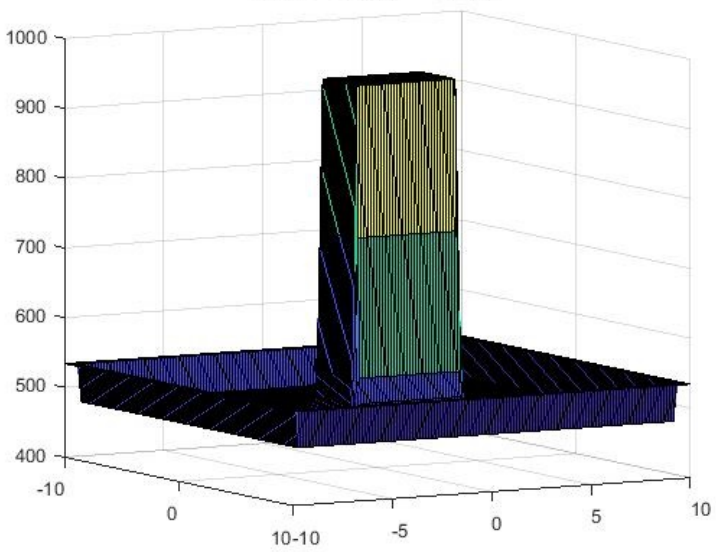


(a) Función de repulsión

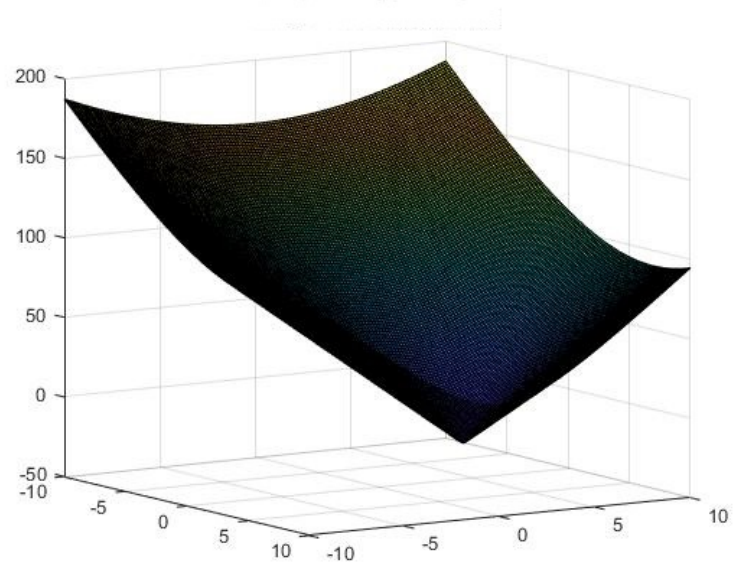


(b) Función de atracción

Figura 10: Funciones artificiales de potencial de repulsión y atracción generadas para el caso A

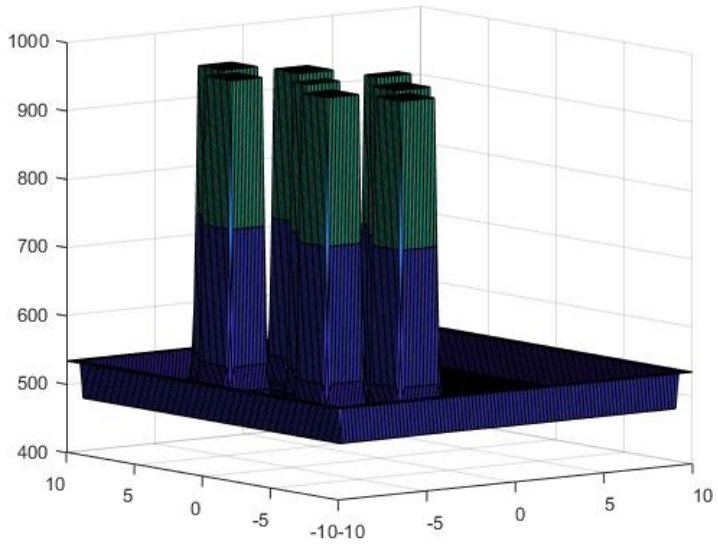


(a) Función de repulsión

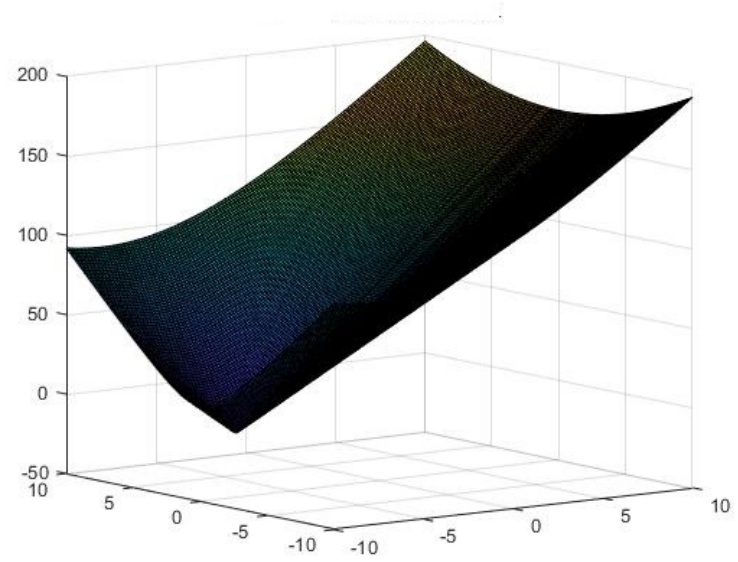


(b) Función de atracción

Figura 11: Funciones artificiales de potencial de repulsión y atracción generadas para el caso B



(a) Función de repulsión



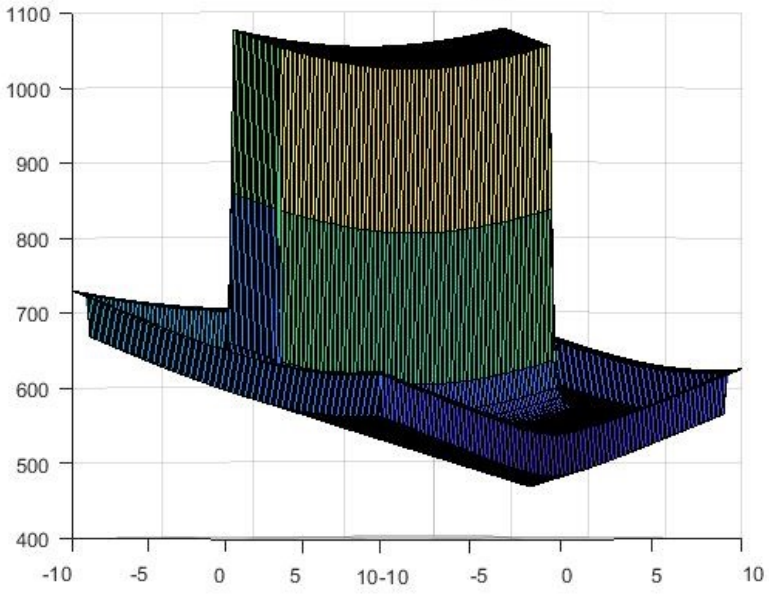
(b) Función de atracción

Figura 12: Funciones artificiales de potencial de repulsión y atracción generadas para el caso C

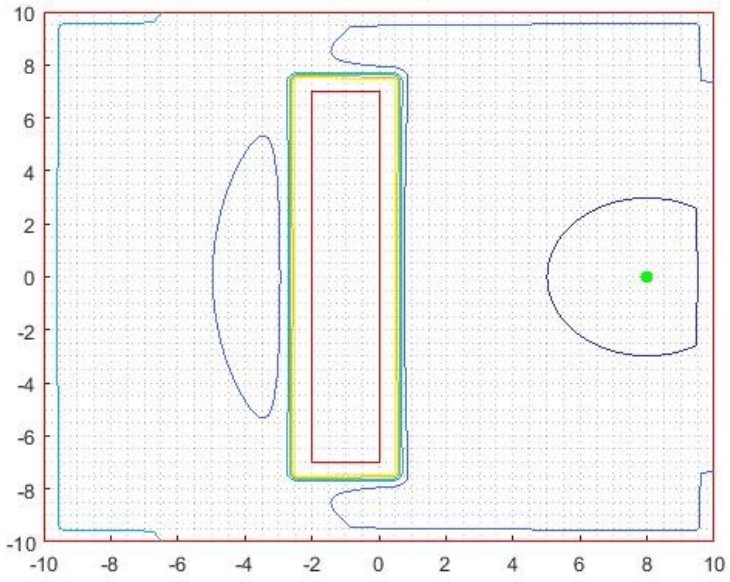
7.3.1. Comportamiento aditivo

Según el modelo descrito por Choset [8], los potenciales se combinan linealmente para generar la función de potencial total del entorno de interés. Se realizó dicha función final para cada caso (ecuación 1), sumando los respectivos potenciales de repulsión y de atracción. Se puede observar, en las figuras 13a, 14a, 15a que el costo decrece mientras mas cerca se esté de la meta y el costo de llegar a un obstáculo es muy grande. Además, se observa que el cambio de potencial hacia un obstáculo cumple con ser continuo, pero es pronunciado.

De acuerdo a las curvas de nivel del modelo, observables en las figuras 13b, 14b y 15b, el gradiente en el potencial de atracción no es muy grande.

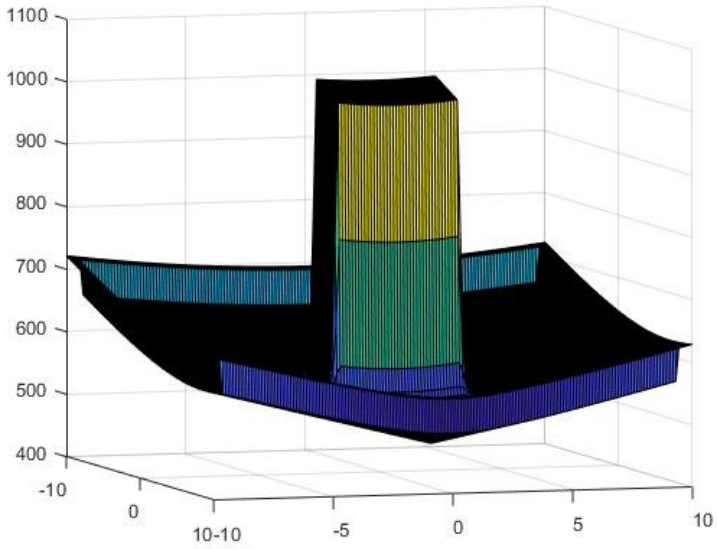


(a) Función artificial de potencial total para el caso A

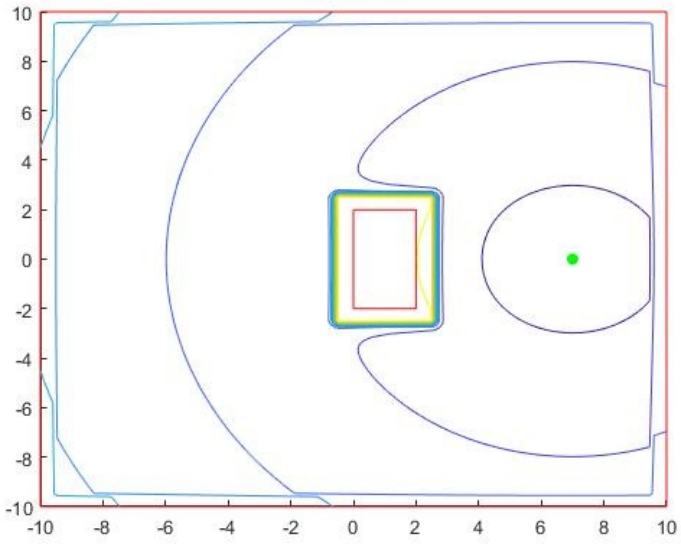


(b) Curvas de nivel para caso A

Figura 13: Función artificial generada con comportamiento aditivo para caso A

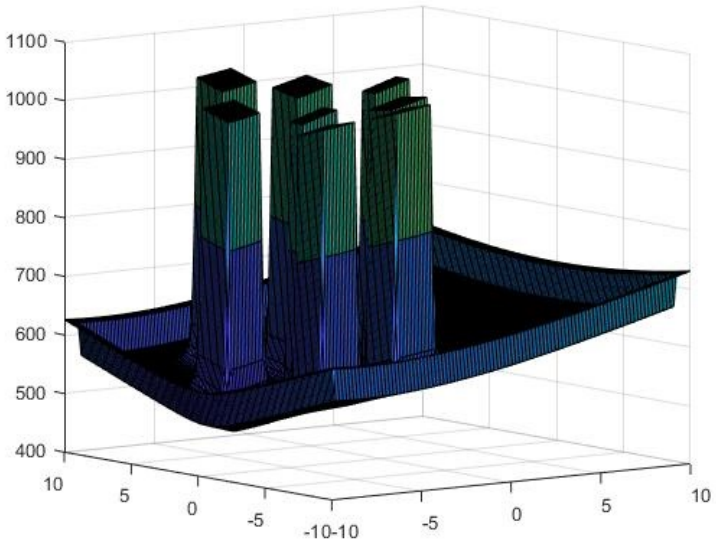


(a) Función artificial de potencial total para el caso B

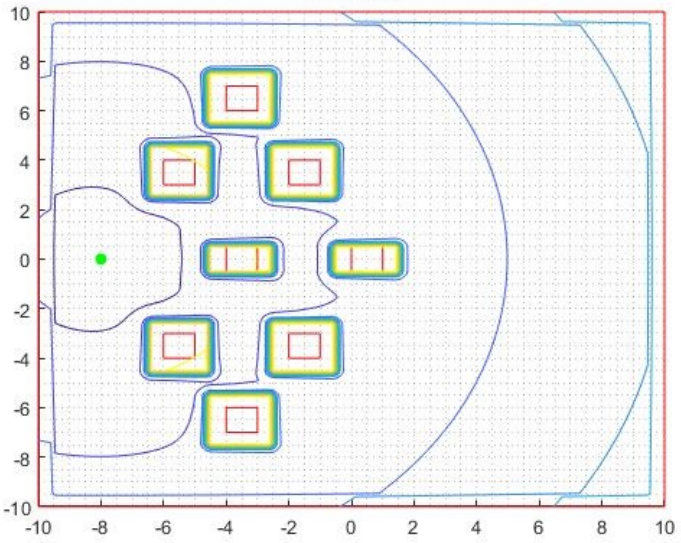


(b) Curvas de nivel para caso B

Figura 14: Función artificial generada con comportamiento aditivo para caso B



(a) Función artificial de potencial total para el caso C

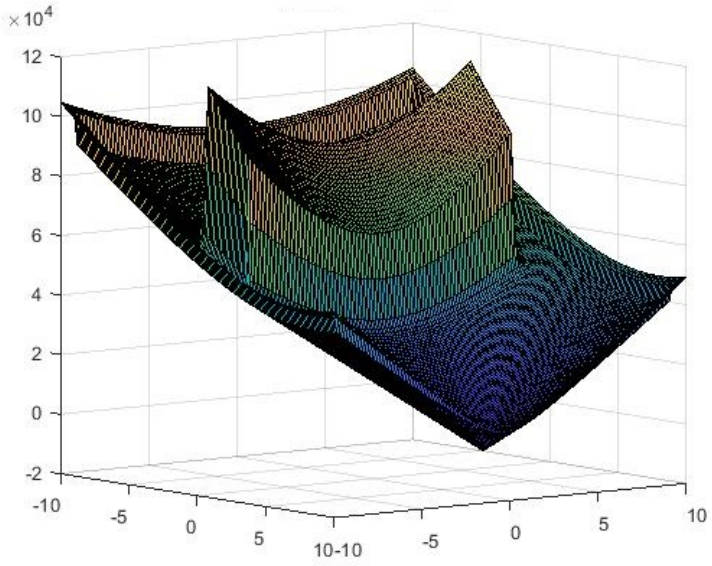


(b) Curvas de nivel para caso C

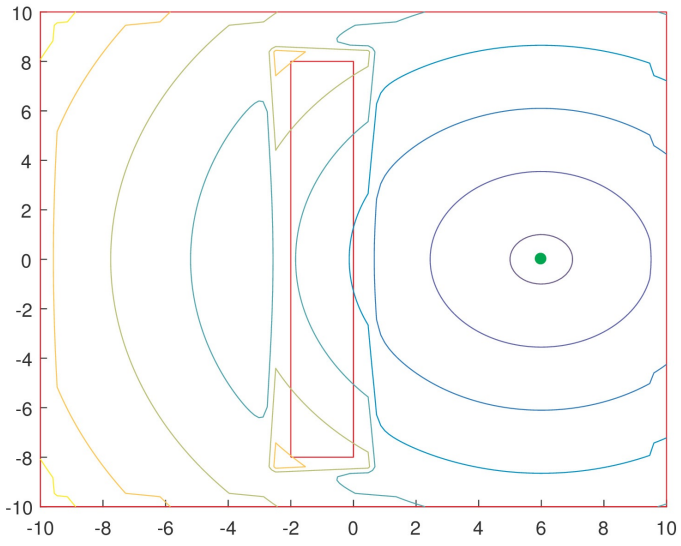
Figura 15: Función artificial generada con comportamiento aditivo para caso C

7.3.2. Comportamiento multiplicativo

Tomando como referencia el trabajo de Kim, Wang y Shin [9], se implementaron los mismos modelos matemáticos de Choset para las funciones de potenciales (ecuaciones 2 y 3) pero con el comportamiento multiplicativo planteado en el trabajo de Kim, Wang y Shin [9], sin tomar en cuenta el parámetro c_g de la ecuación 10. Se puede observar en la figura 16b que existe un gradiente mayor que el presentado en el modelo aditivo, figura 13b. De igual manera se pueden comparar las figuras 14b y 17b y las figuras 15b y 18b.

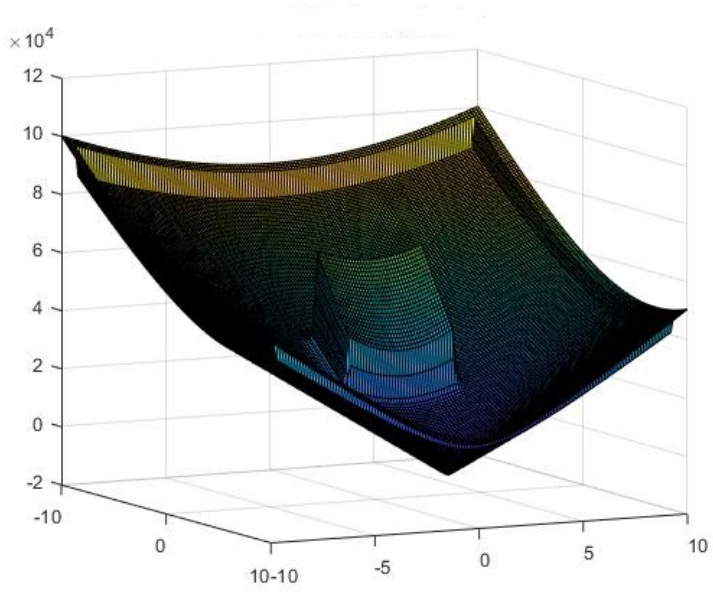


(a) Función artificial de potencial total para el caso A

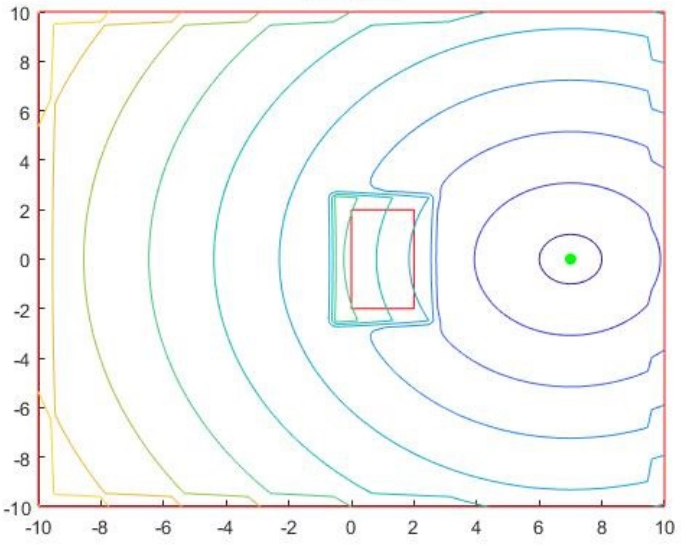


(b) Curvas de nivel para caso A

Figura 16: Función artificial generada con comportamiento multiplicativo para caso A

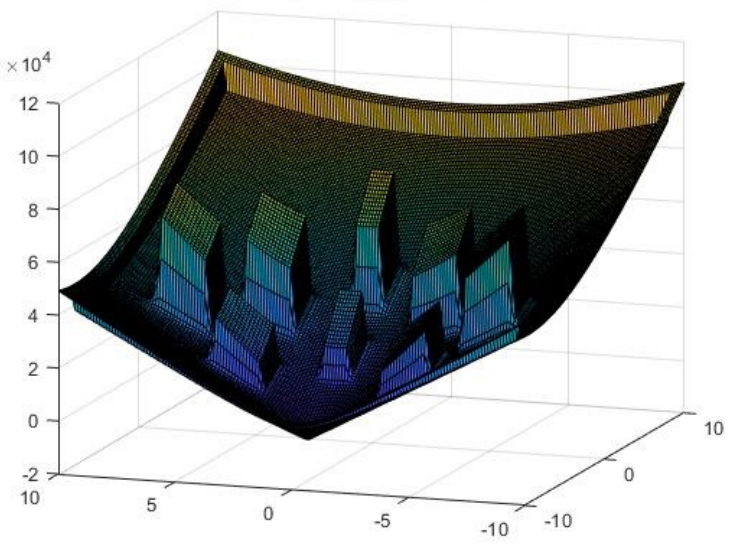


(a) Función artificial de potencial total para el caso B

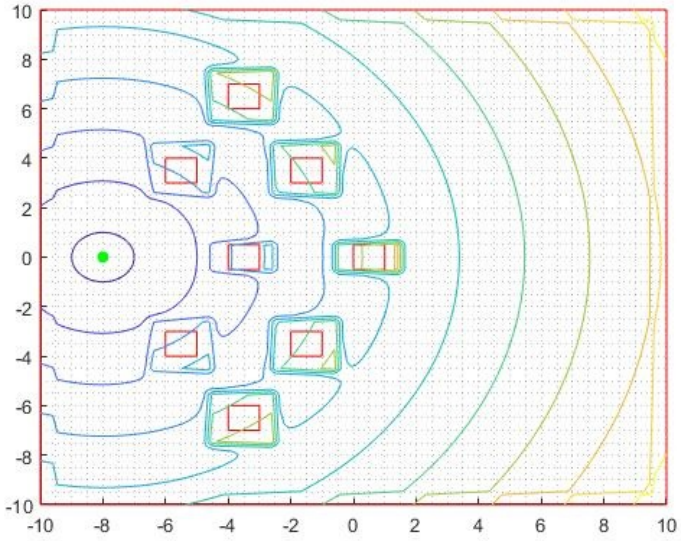


(b) Curvas de nivel para caso B

Figura 17: Función artificial generada con comportamiento multiplicativo para caso B



(a) Función artificial de potencial total para el caso C

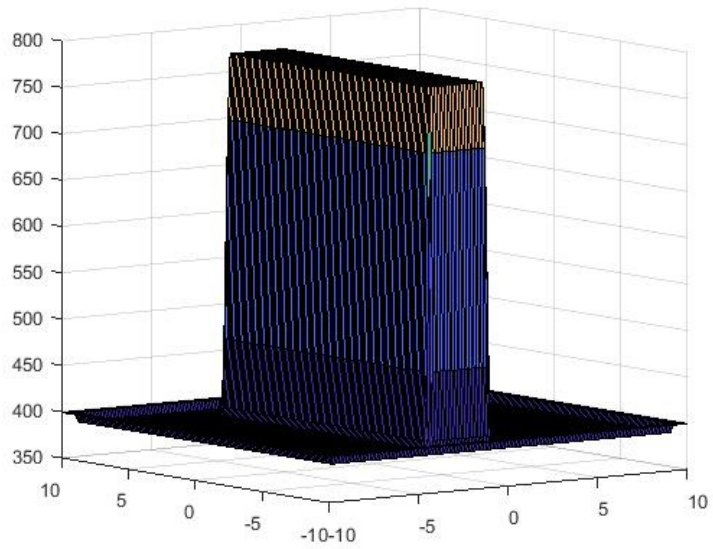


(b) Curvas de nivel para caso C

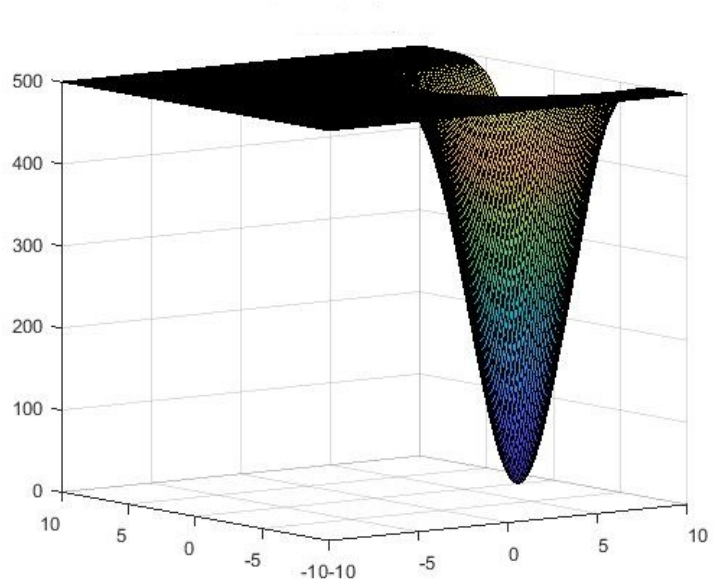
Figura 18: Función artificial generada con comportamiento multiplicativo para caso C

7.4. Función artificial de potencial basada en la función de Kim, Wang y Shin

Se implementó el modelo de Kim, Wang y Shin [9] en los casos planteados anteriormente. Para la función de atracción a la meta (ecuación 5) se usaron los valores de 500 para c_g y de 3 para l_g hallados heurísticamente. Para la función de repulsión hacia los obstáculos (ecuación 7) se usaron los valores de 500 para c_o y 0.2 para l_o hallados heurísticamente.

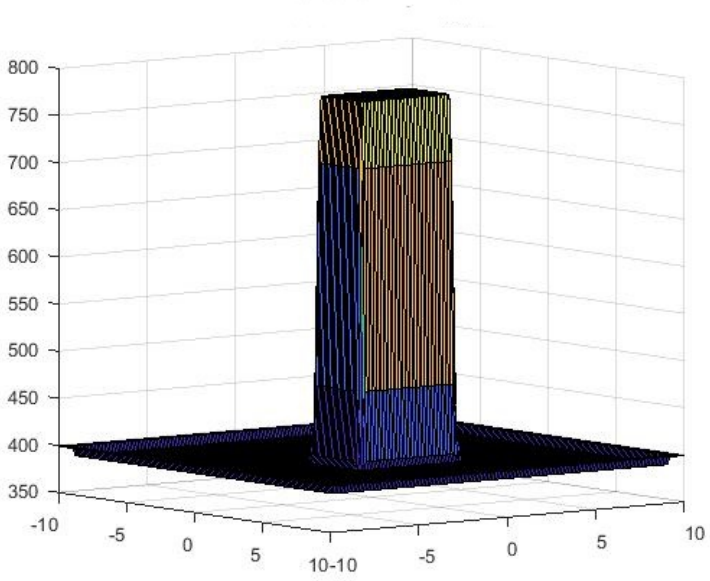


(a) Función de repulsión

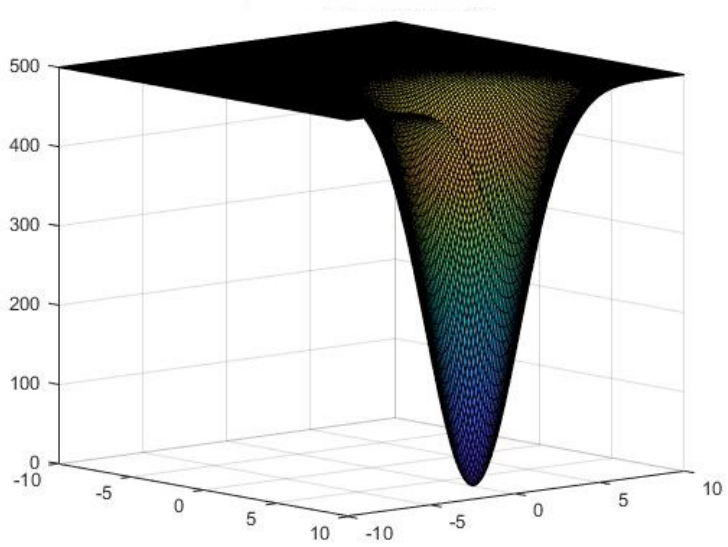


(b) Función de atracción

Figura 19: Funciones artificiales de potencial de repulsión y atracción generadas para el caso A

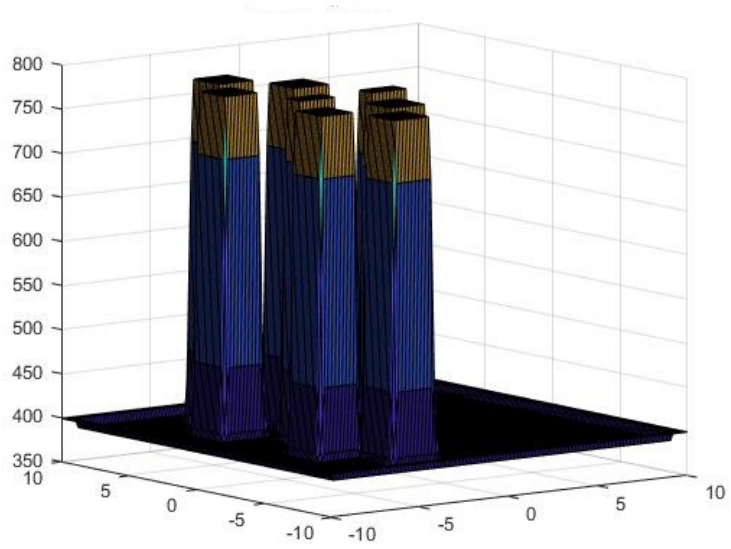


(a) Función de repulsión

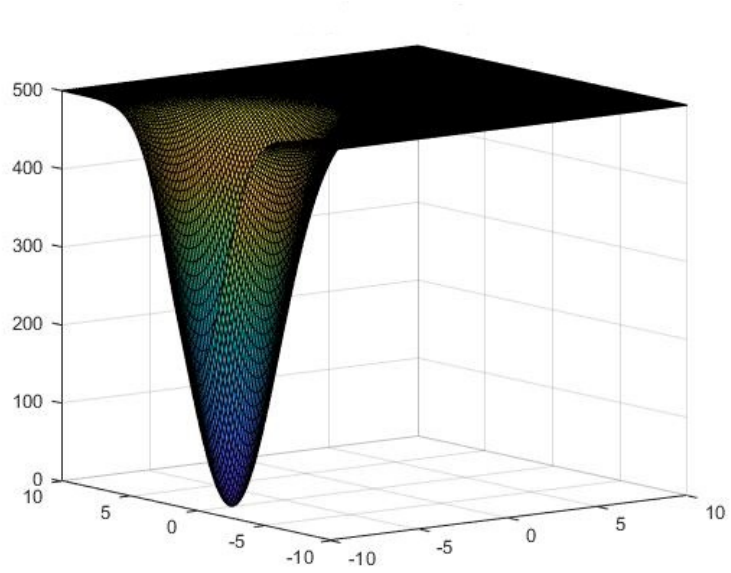


(b) Función de atracción

Figura 20: Funciones artificiales de potencial de repulsión y atracción generadas para el caso B



(a) Función de repulsión

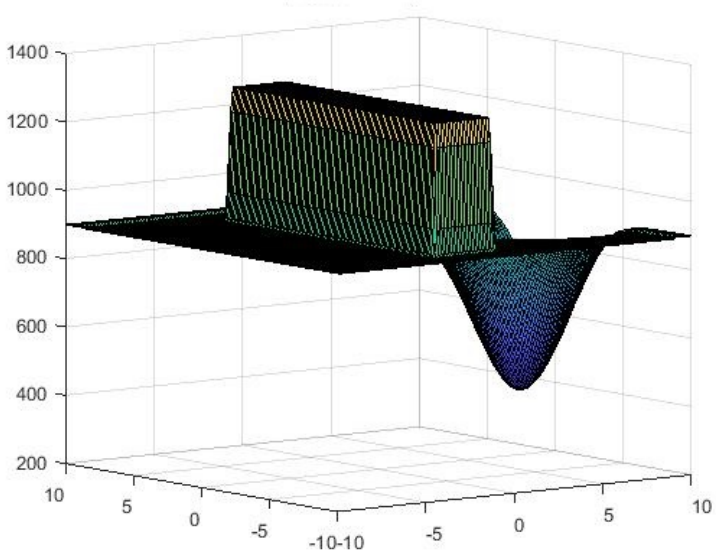


(b) Función de atracción

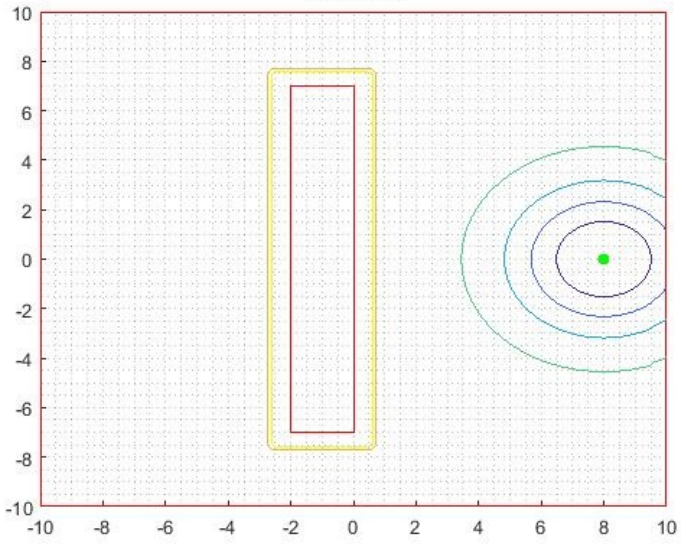
Figura 21: Funciones artificiales de potencial de repulsión y atracción generadas para el caso C

7.4.1. Comportamiento aditivo

Los potenciales de atracción y de repulsión se combinaron aditivamente (ecuación 9), y los resultados generados se muestran en las figuras 22, 23 y 21. Se observó en este caso, que no existe un cambio notable en el gradiente hasta que la distancia es corta hacia la meta o hacia un obstáculo.

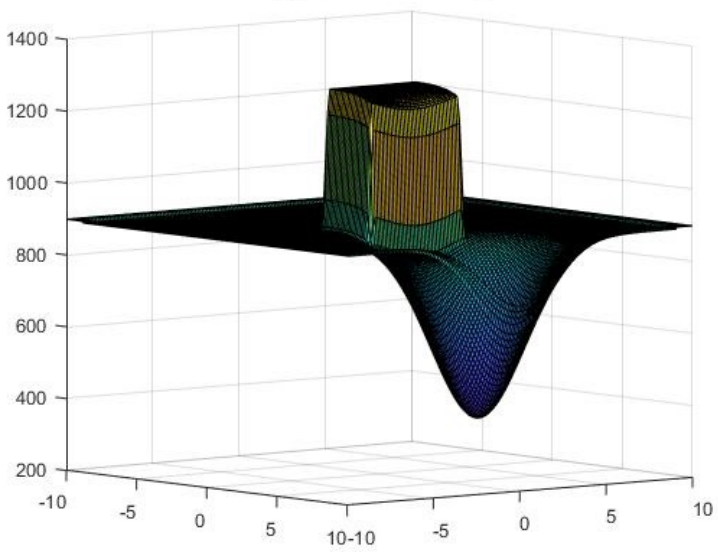


(a) Función artificial de potencial total para el caso A

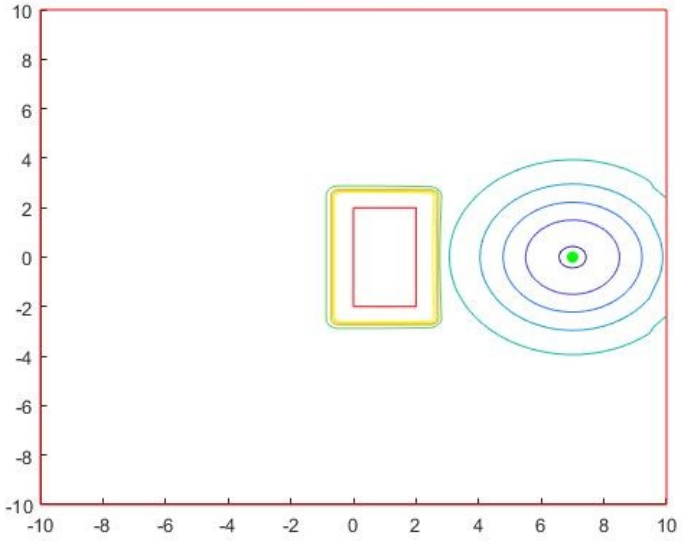


(b) Curvas de nivel para caso A

Figura 22: Función artificial generada con comportamiento aditivo para caso A

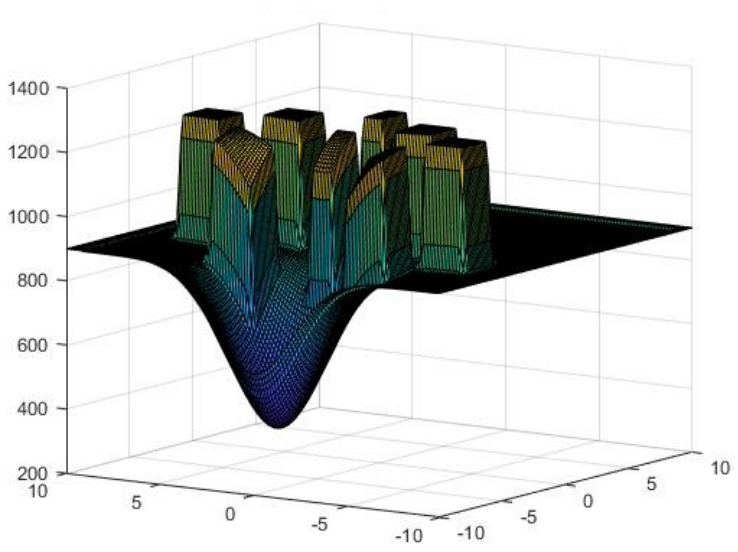


(a) Función artificial de potencial total para el caso B

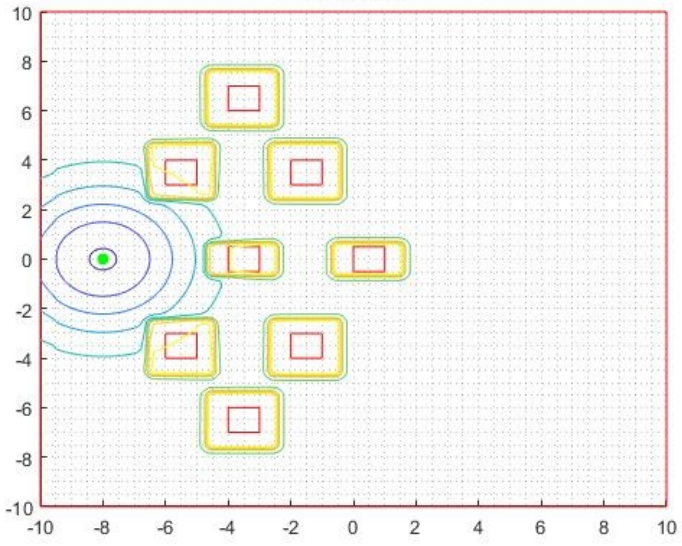


(b) Curvas de nivel para caso B

Figura 23: Función artificial generada con comportamiento aditivo para caso B



(a) Función artificial de potencial total para el caso C

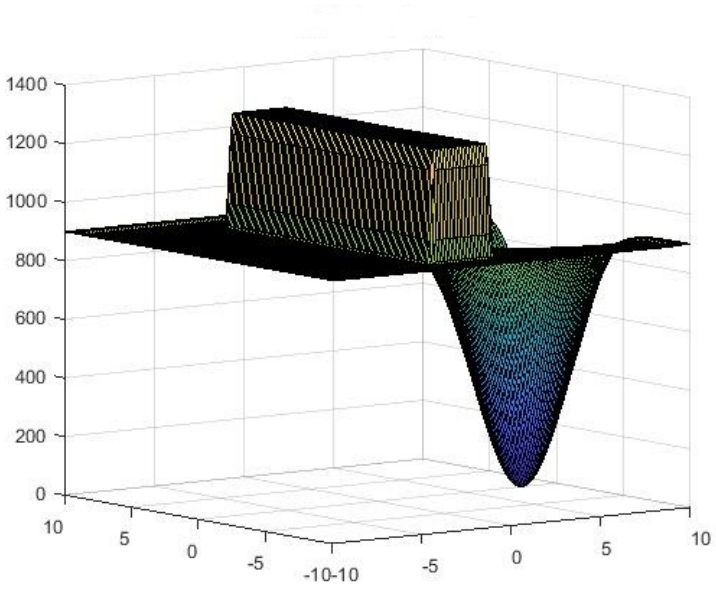


(b) Curvas de nivel para caso C

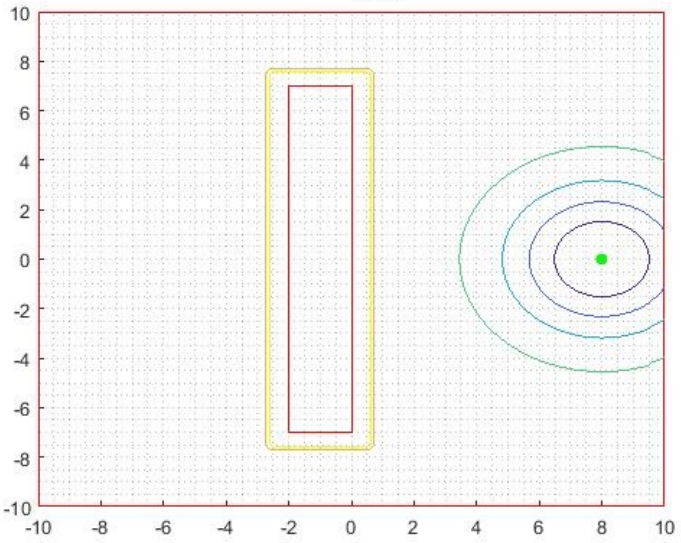
Figura 24: Función artificial generada con comportamiento aditivo para caso C

7.4.2. Comportamiento multiplicativo

Se combinaron los potenciales de repulsión y de atracción de manera multiplicativa (ecuación 10). En este caso, se nota que el cambio de gradiente se da desde una distancia un poco mayor en comparación con en modelo de comportamiento aditivo.

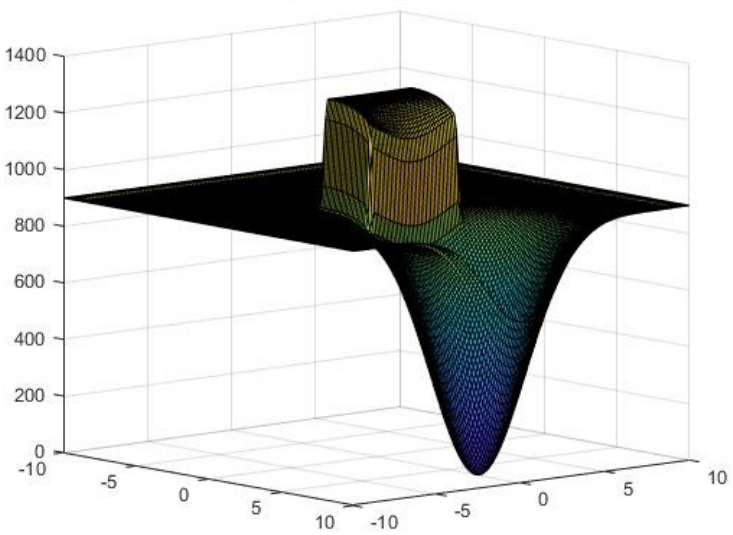


(a) Función artificial de potencial total para el caso A

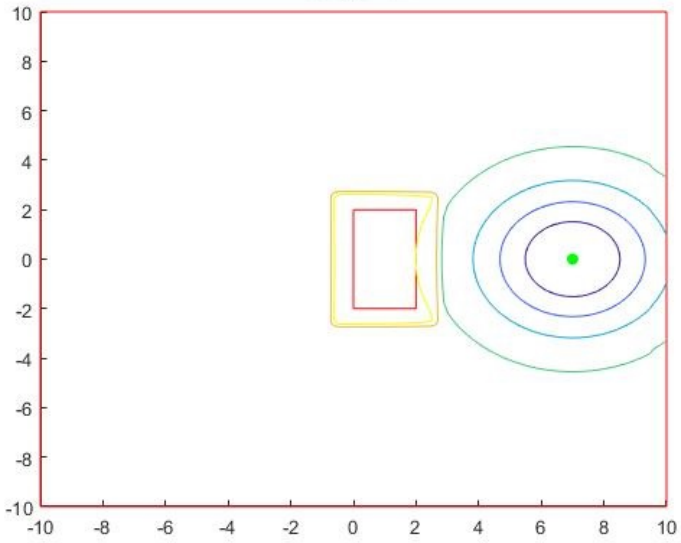


(b) Curvas de nivel para caso A

Figura 25: Función artificial generada con comportamiento multiplicativo para caso A

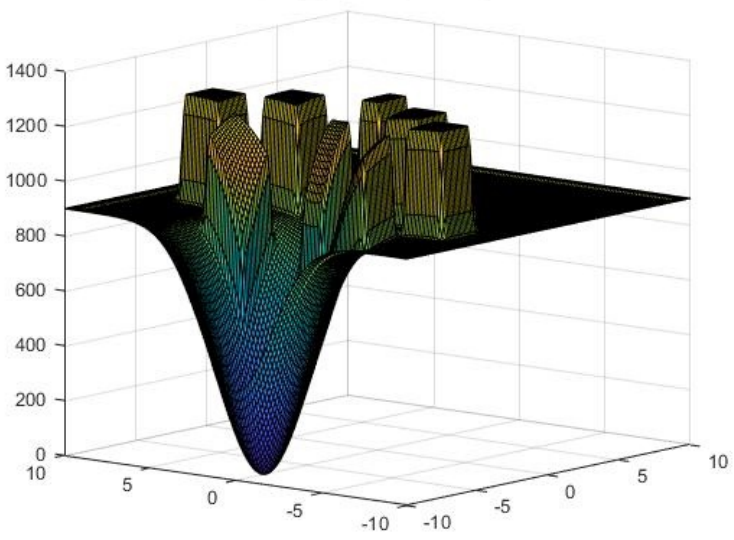


(a) Función artificial de potencial total para el caso B

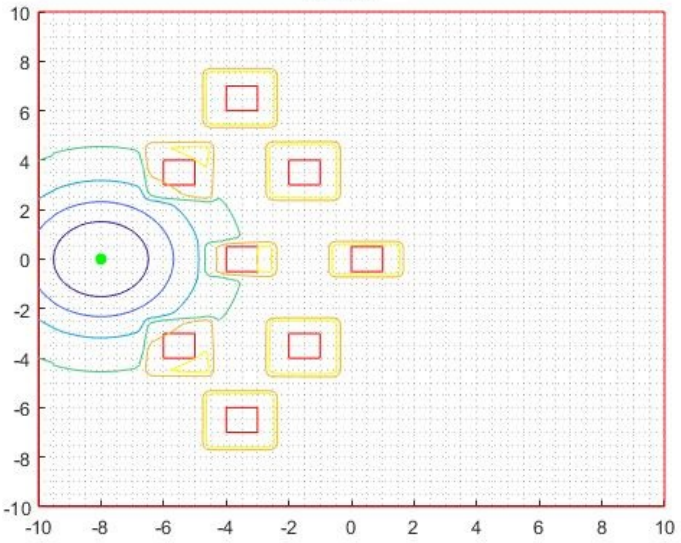


(b) Curvas de nivel para caso B

Figura 26: Función artificial generada con comportamiento multiplicativo para caso B



(a) Función artificial de potencial total para el caso C



(b) Curvas de nivel para caso C

Figura 27: Función artificial generada con comportamiento multiplicativo para caso C

CAPÍTULO 8

Implementación de algoritmo PSO en condiciones ideales

Las simulaciones del algoritmo PSO en los campos artificiales de potencial se elaboraron en la versión R2017a de MATLAB. En este entorno, se modeló puntualmente a cada agente y no se tomaron en cuenta restricciones físicas tales como la velocidad máxima de éstos o la energía perdida por la fricción de éste con el suelo.

8.1. Análisis de sensibilidad de parámetros de PSO

Se realizaron múltiples iteraciones implementando el algoritmo PSO de Chowdhury [10] para poder determinar cómo afectaban los parámetros w, α, β, y γ en el comportamiento de la navegación de los agentes controlado mediante el algoritmo PSO a través de un modelo planteado con campos artificiales de potencial. Se agregó un factor de constricción K a la formula general de PSO (ecuación 13) como una forma de regular la velocidad de los agentes, al cual se le dio el valor de 0.5 para las iteraciones realizadas.

$$V_i^{t+1} = K[\alpha V_i^t \beta_l r_1 (P_i - X_i^t) + \beta_g r_2 (P_g - X_i^t) + \gamma r_3 \hat{V}_i^t] \tag{14}$$

Se implementaron cien agentes por iteración posicionados aleatoriamente en el caso B presentado anteriormente. Conforme se realizaron las iteraciones se observó cualitativamente el comportamiento de los agentes al navegar para así poder determinar un rango de valores entre los cuales se pudiera elegir cada parámetro. Para realizar estos experimentos se varió únicamente un parámetro y se dejaron los restantes en un valor de uno siempre. Para poder realizar un análisis cuantitativo se tomó cada parámetro como un valor independiente y se tomó la cantidad de agentes que convergieron a la meta propuesta y la cantidad de iteraciones por simulación como variables dependientes. El objetivo de la recolección de estos datos era

poder graficarlos y determinar heurísticamente, según la recomendación de Choset [8], el valor adecuado de cada parámetro.

8.1.1. Analizando el parámetro w

Se realizó el análisis del parámetro de inercia w variando su valor desde 0.1 hasta 1.9, manteniendo el parámetro de constricción K en 0.5 y los parámetros α, β y γ en 1. Como se puede observar en la Figura 28, el incremento del valor de inercia w de los agentes no afecta la cantidad de agentes que convergen a la meta. Sin embargo, se puede observar en la Figura 29 que a medida que se incrementa el factor de inercia de los agentes las iteraciones necesarias para converger a la meta se van reduciendo. El valor de w mostró un límite cuando este era 2, y se debe a que se observó que el movimiento individual por cada agente era muy agresivo y lejos de lo necesario para el trabajo.

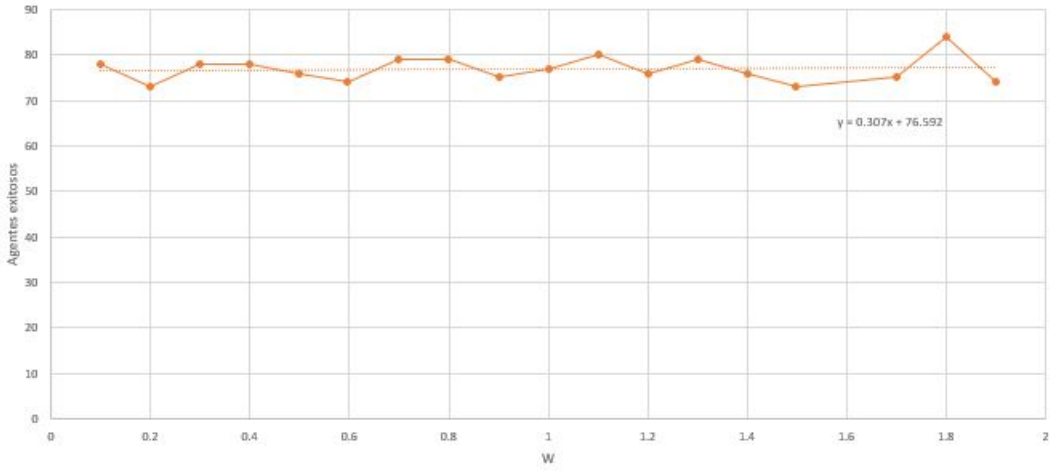


Figura 28: Análisis de sensibilidad de w en relación a agentes exitosos

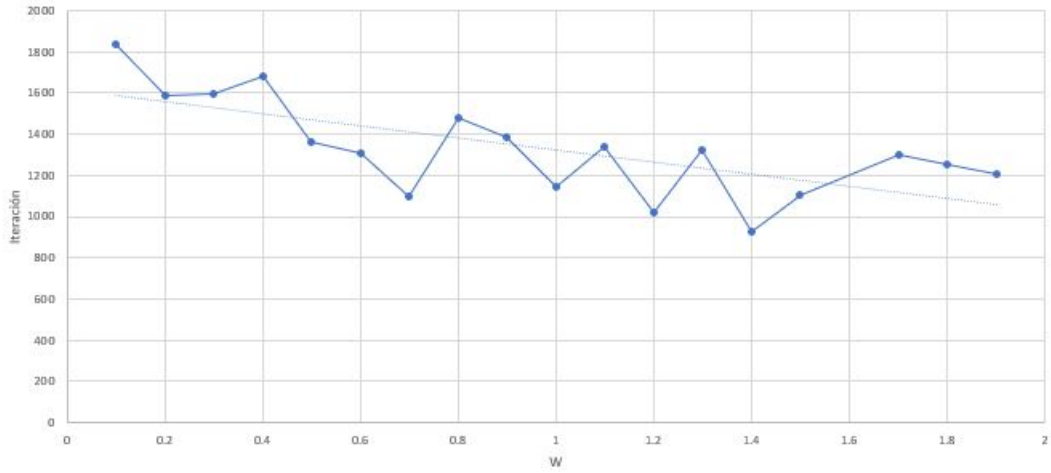


Figura 29: Análisis de sensibilidad de w en relación a iteraciones para convergencia

8.1.2. Analizando el parámetro α

Se realizó el análisis del parámetro α incrementando su valor hasta llegar a 1000. El parámetro de constricción se mantuvo en 0.5, mientras que los parámetros β, γ y w en 1. El resultado observado en la Figura 30 muestra que existe una tendencia decreciente en la cantidad de agentes que logran llegar a la meta a medida que se incrementa el parámetro α, conocido como también por ser el parámetro cognitivo de cada agente.

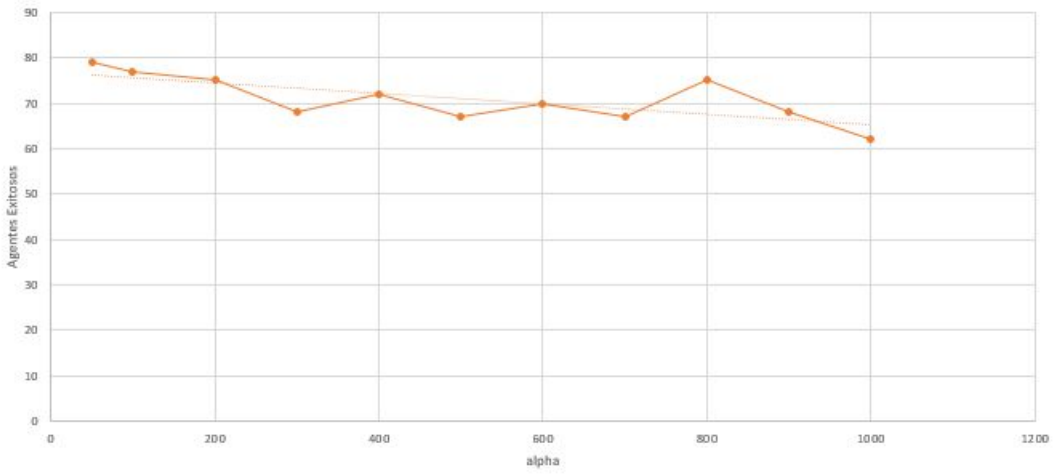


Figura 30: Análisis de sensibilidad de α en relación a agentes exitosos

En cuanto a la relación que existe entre el parámetro α y la cantidad de iteraciones necesarias para que los agentes logren llegar a la meta, se ve en la Figura 31 una tendencia decreciente exponencial a medida que se aumenta el valor de α. A medida que se siguió aumentando el valor del parámetro se pudo observar una variación mínima en la variable dependiente, por lo que se decidió no seguir mas allá del valor 1000 ya que elegir un valor tan alto reduciría la cantidad de agentes exitosos considerablemente. En este caso no se observó un comportamiento errático en los agentes a medida que se aumentaba el parámetro α.

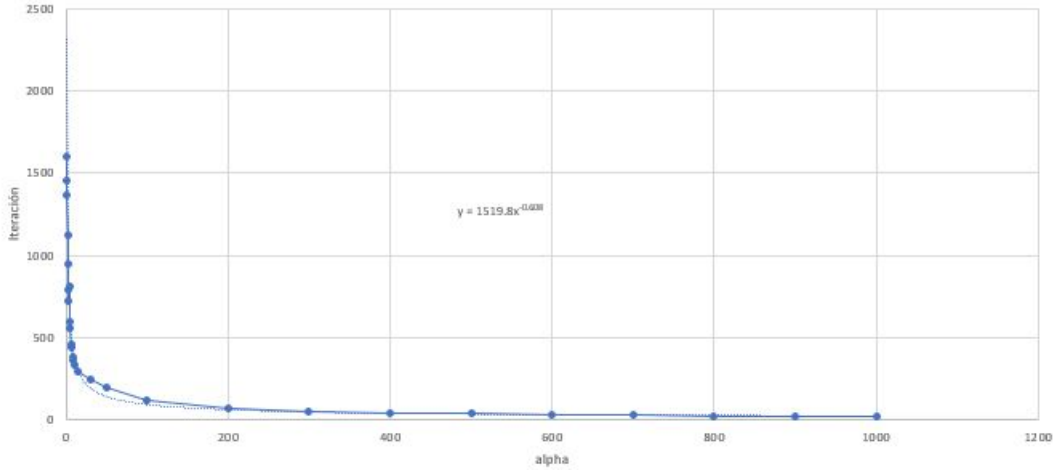


Figura 31: Análisis de sensibilidad de α en relación a iteraciones para convergencia

8.1.3. Analizando el parámetro β

Se realizó el análisis del parámetro β variando este desde 0 hasta 70 en pasos de 10. El parámetro de constricción se mantuvo en 0.5, mientras que los parámetros α, γ y w en 1. El resultado observado en la Figura 32 muestra que la cantidad de agentes que logran llegar a la meta no depende directamente del parámetro β, conocido también por ser el parámetro social de cada agente.

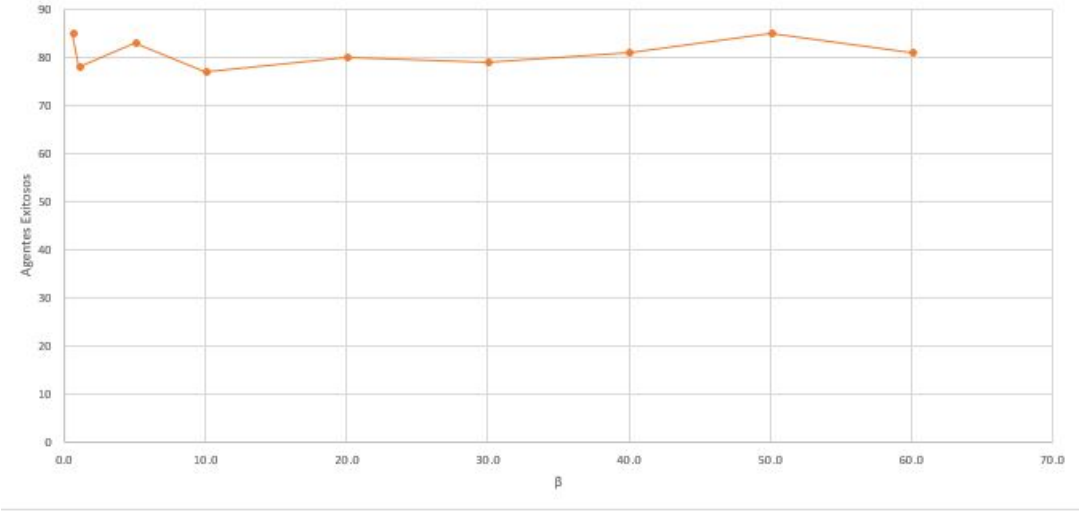


Figura 32: Análisis de sensibilidad de β en relación a agentes exitosos

En cuanto a la relación que existe entre el parámetro β y la cantidad de iteraciones necesarias, se observa en la Figura 33 una tendencia decreciente lineal hasta que el valor llega a 40. A medida que se aumentó el parámetro mas allá de 40 se observó una variación mínima en la cantidad de iteraciones necesarias para que los agentes convergieran a la meta. En este caso se observó un comportamiento muy agresivo en el movimiento de los agentes después del valor 40, por lo que se decidió no seguir aumentando el parámetro.

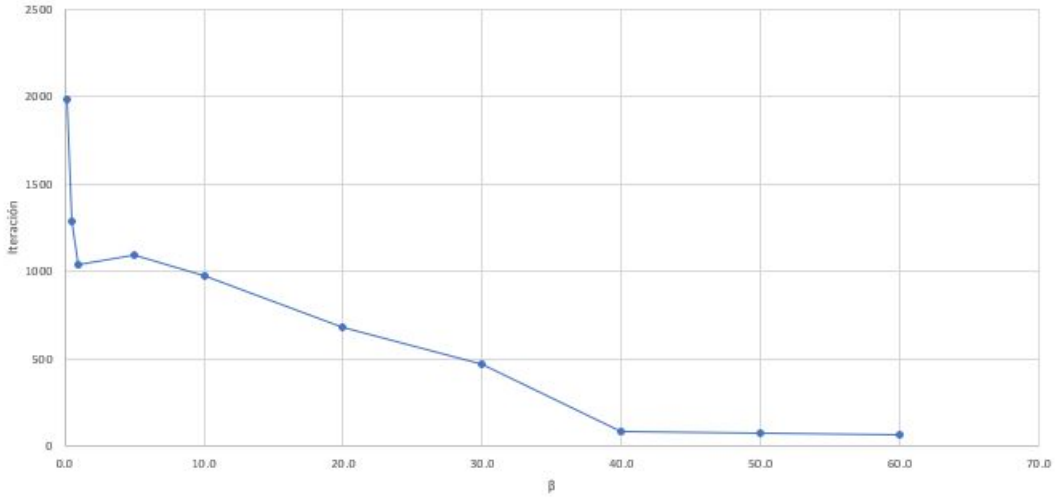


Figura 33: Análisis de sensibilidad de β en relación a iteraciones para convergencia

8.1.4. Analizando el parámetro γ

Se realizó el análisis del parámetro γ variando su valor desde 0 hasta 40. El parámetro de constricción se mantuvo en 0.5, mientras que los parámetros α, β y w en 1. El resultado observado en la Figura 34 muestra que no existe relación directa entre la cantidad de agentes que logran llegar a la meta y el valor del parámetro γ, conocido como también por ser el parámetro de diversidad en el movimiento de cada agente.

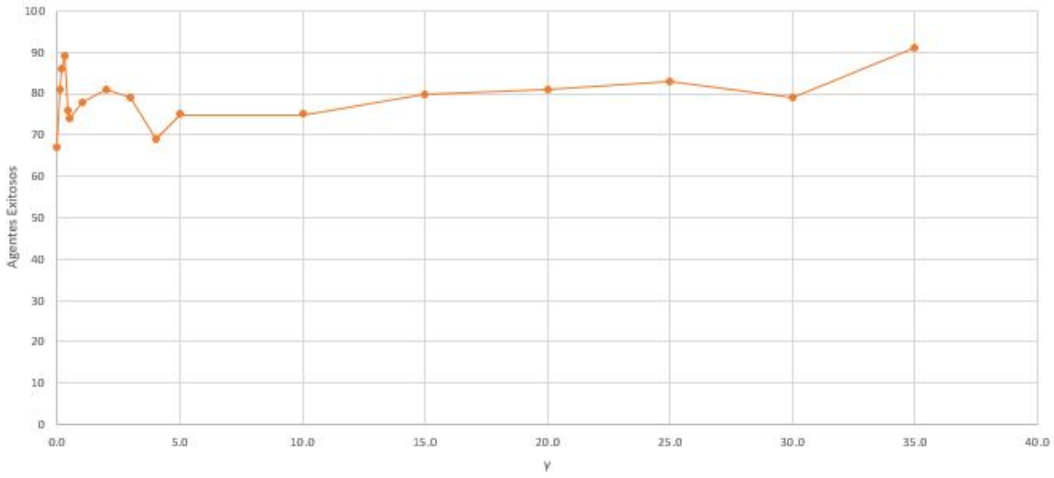


Figura 34: Análisis de sensibilidad de γ en relación a agentes exitosos

En cuanto a la relación que existe entre el parámetro γ y la cantidad de iteraciones necesarias, se observa en la Figura 33 una tendencia decreciente antes de que el valor llegue a 35. A medida que se aumentó el parámetro más allá de 5 se observó que la cantidad de iteraciones necesarias para que los agentes convergieran a la meta no disminuía tanto. En este caso se observó un comportamiento muy agresivo en el movimiento de los agentes después del valor 40.

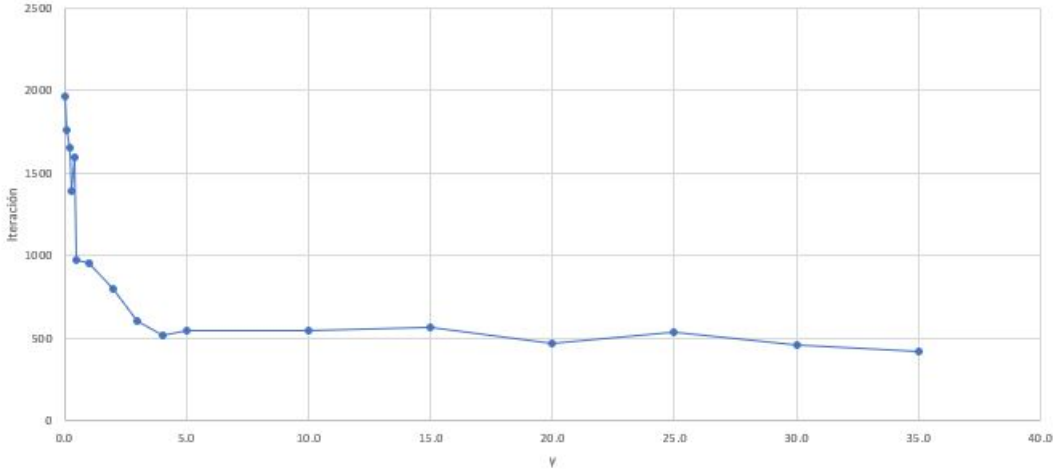


Figura 35: Análisis de sensibilidad de γ en relación a iteraciones para convergencia

8.1.5. Seleccionando los valores de los parámetros

En base a las iteraciones realizadas se eligieron los valores que se cree le darán un mejor comportamiento al movimiento de los agentes, reduciendo la cantidad de iteraciones necesarias para la convergencia y aumentando la cantidad de agentes exitosos en llegar a la meta. Se decidió que el valor de constricción se mantiene en 0.5, el valor de w debería ser entre 1.2 y 1.8, α se dejará en 400, β en 50 y γ en 35.

8.2. Selección de modelo óptimo

Luego de determinar los valores óptimos para la implementación del algoritmo PSO modificado (14), se procedió a realizar iteraciones para los campos de potencial artificiales modelados mediante el modelo de Choset (ecuaciones 2 y 3) y el modelo de Kim, Wang y Shin (ecuaciones 5 y 7), tanto en comportamiento multiplicativo (10) como en comportamiento aditivo (1).

Para los parámetros del la ecuación de PSO 13 se usó un valor de constricción de 0.5, un valor de 1 para w, un valor de 400 para α, β en 50 y un valor de 35 para γ.

Para el modelado de las funciones artificiales de Choset se usaron los siguientes parámetros: 5 para ζ, 10 para η, 2 para d^*_{goal}, 5 para Q_i en una cuadrícula de 20 unidades por 20 unidades.

Para el modelado de las funciones artificiales de Kim, Wang y Shin se usaron los siguientes parámetros: 5 para c_g, 3 para l_g, 500 para c_o y 0.2 para l_o.

8.2.1. Simulaciones para el caso A

Se realizaron veinticinco iteraciones con cincuenta agentes dispersos aleatoriamente sobre el campo del caso A. Mediante las iteraciones realizadas se obtuvo la media del porcentaje de agentes exitosos y la media de la cantidad de iteraciones que se tomó el algoritmo PSO para que pudieran alcanzar la meta planteada a través de los obstáculos, Estos resultados se tomaron para poder comparar la eficiencia de los modelos planteados.

Los resultados para las pruebas realizadas con el modelo de Choset con comportamiento aditivo se muestran en el Cuadro 3, mientras que los resultados para el modelo de Choset con comportamiento multiplicativo de muestran en el Cuadro 4. Entre estas dos pruebas el modelo de Choset con comportamiento aditivo demostró un promedio de agentes exitosos ligeramente mayor a las pruebas realizadas con el modelo de Choset con comportamiento multiplicativo, pero con una desviación estándar también mayor.

En el caso de los modelos de Kim, Wang y Shin los resultados se muestran en el Cuadro 5 para el modelo con comportamiento aditivo y en el Cuadro 6 para el modelo con comportamiento multiplicativo. En este caso el modelo de Kim, Wang y Shin con comportamiento aditivo también demostró un promedio de agentes exitosos ligeramente mayor y una desviación estándar también mayor.

Se puede observar en el diagrama boxplot que se presenta en la Figura 36 que el modelo con el mayor promedio de agentes fue el caso con el modelo de Kim, Wang y Shin con comportamiento aditivo. El caso con mayor dispersión de datos fue el modelo de Choset con comportamiento aditivo, el cual también muestra el límite inferior mas bajo de todas las pruebas. En el caso del rendimiento por iteraciones se puede observar en la Figura 37 que el modelo que en promedio requirió menos iteraciones fue el modelo de Choset con comportamiento multiplicativo. Este también demostró el valor más alto de todas las pruebas realizadas y el modelo con mayor dispersión de datos.

Modelo de Choset con comportamiento aditivo				
Simulación	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	34	16	68	25
2	38	12	76	12
3	37	13	74	13
4	35	15	70	6
5	31	19	62	33
6	37	13	74	9
7	31	19	62	10
8	24	26	48	8
9	40	10	80	18
10	38	12	76	6
11	28	22	56	10
12	32	18	64	7
13	36	14	72	7
14	23	27	46	13
15	34	16	68	41
16	34	16	68	11
17	36	14	72	7
18	36	14	72	7
19	39	11	78	11
20	35	15	70	10
21	34	16	68	7
22	33	17	66	7
23	27	23	54	29
24	33	17	66	12
25	38	12	76	8
Media	33.72	16.28	67.44	13.08
Desviacion Estandar	4.33	4.33	8.66	9.00

Cuadro 3: Resultado de simulaciones en caso A con el modelo de Choset y comportamiento aditivo

Modelo de Choset con comportamiento multiplicativo				
Iteración	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	37	13	74	19
2	27	23	54	10
3	35	15	70	12
4	33	17	66	18
5	31	19	62	10
6	31	19	62	9
7	41	9	82	10
8	34	16	68	11
9	35	15	70	11
10	32	18	64	11
11	38	12	76	11
12	33	17	66	9
13	31	19	62	11
14	31	19	62	12
15	34	16	68	10
16	33	17	66	12
17	29	21	58	16
18	35	15	70	8
19	34	16	68	14
20	36	14	72	10
21	30	20	60	11
22	35	15	70	10
23	31	19	62	9
24	33	17	66	10
25	30	20	60	26
Media	33.16	16.84	66.32	12.00
Desviación estándar	2.99	2.99	5.98	3.89

Cuadro 4: Resultado de simulaciones en el caso A con el modelo de Choset con comportamiento multiplicativo

Modelo de Kim, Wang y Shin. con comportamiento aditivo				
Iteración	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	34	16	68	9
2	35	15	70	10
3	31	19	62	18
4	39	11	78	9
5	37	13	74	11
6	35	15	70	9
7	36	14	72	20
8	37	13	74	9
9	32	18	64	10
10	32	18	64	10
11	31	19	62	24
12	29	21	58	15
13	29	21	58	11
14	36	14	72	13
15	38	12	76	14
16	36	14	72	11
17	37	13	74	11
18	39	11	78	11
19	35	15	70	31
20	28	22	56	22
21	40	10	80	12
22	31	19	62	13
23	31	19	62	9
24	33	17	66	11
25	31	19	62	25
Media	34.08	15.92	68.16	13.92
Desviación Estándar	3.37	3.37	6.74	5.86

Cuadro 5: Resultado de simulaciones en el caso A con el modelo de Kim, Wang y Shin con comportamiento aditivo

Modelo de Kim, Wang y Shin con comportamiento multiplicativo				
Simulación	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	27	23	54	13
2	29	21	58	8
3	35	15	70	9
4	36	14	72	11
5	31	19	62	10
6	39	11	78	9
7	31	19	62	8
8	34	16	68	8
9	34	16	68	7
10	36	14	72	9
11	32	18	64	9
12	36	14	72	23
13	37	13	74	9
14	32	18	64	8
15	32	18	64	8
16	35	15	70	11
17	31	19	62	8
18	28	22	56	8
19	33	17	66	9
20	37	13	74	10
21	35	15	70	12
22	36	14	72	8
23	37	13	74	13
24	36	14	72	18
25	34	16	68	9
Media	33.72	16.28	67.44	10.20
Desviación Estándar	2.99	2.99	5.97	3.49

Cuadro 6: Resultado de simulaciones en el caso A con el modelo de Kim, Wang y Shin con comportamiento multiplicativo

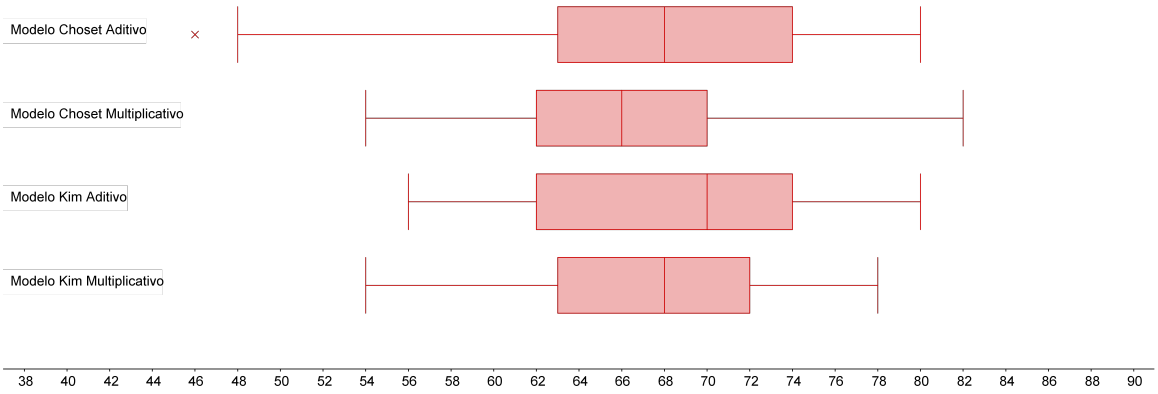


Figura 36: Diagrama boxplot de los resultados de rendimiento para el porcentaje de agentes exitosos con los modelos planteados para el caso A

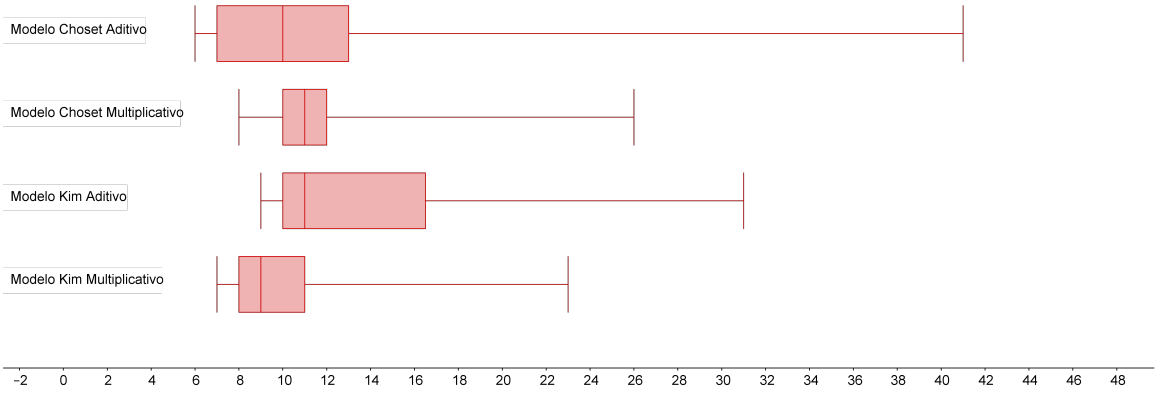


Figura 37: Diagrama boxplot de los resultados de rendimiento para la cantidad de iteraciones necesarias para la convergencia de los agentes con los modelos planteados para el caso A

8.2.2. Simulaciones para el caso B

En este caso también se realizaron veinticinco iteraciones con cincuenta agentes dispersos aleatoriamente sobre el campo del caso B. Mediante las iteraciones realizadas se obtuvieron los datos de la media del porcentaje de agentes exitosos y su desvaición estándar y la media de iteraciones que se tomó con el algoritmo PSO para que los agentes exitosos alcanzaran la meta planteada a través de los obstáculos. Con estos datos se logró comparar el rendimiento de los modelos planteados.

Las pruebas realizadas con el modelo de Choset se presentan en el Cuadro 7 para el comportamiento aditivo y en el Cuadro 8 para el comportamiento multiplicativo. Para el modelo de Kim, Wang y Shin los resultados se muestran en el Cuadro 5, mientras que para el modelo multiplicativo los resultados se muestran en el Cuadro 6. En cuanto al rendimiento se pudo comprobar que el modelo de Choset con comportamiento multiplicativo demostró

un promedio de agentes exitosos más alto que el modelo de Choset con comportamiento aditivo. Aunque el promedio de iteraciones del modelo de Choset multiplicativo es un par de iteraciones más alto que el promedio de Choset con comportamiento aditivo, el primero tiene una desviación estándar menor.

El modelo de Kim, Wang y Shin con comportamiento multiplicativo también demostró una mayor convergencia de agentes exitosos a la meta frente al modelo con comportamiento aditivo, y de igual manera que en el caso del modelo de Choset, el modelo con comportamiento multiplicativo de Kim, Wnag y Shin demostró un mayor promedio de iteraciones por prueba pero con una desviación estándar menor.

Como lo muestra el diagrama boxplot en la Figura 38, las media de convergencia de los agentes a la meta resultó ser mayor e incluso similar en el caso del modelo de Choset con comportamiento multiplicativo y de Kim Wang y Shin con comportamiento multiplicativo. Sin embargo, se muestra en el diagrama que la dispersión de los datos es menor en el caso del modelo de Choset con comportamiento multiplicativo comparado con el modelo de Kim, WAng y Shin con comportamiento multiplicativo. El modelo de Choset con comportamiento multiplicativo tuvo un mínimo de 70 agentes frente a un mínimo de 58 agentes en el caso del modelo de Kim, Wang y Shin con comportamiento multiplicativo, lo que le hace un modelo más confiable para el trabajo de búsqueda y rescate. En la Figura 41 se demuestra también que es el modelo con menor dispersión de datos en cuanto al número de iteraciones necesarias para la convergencia de los agentes en las pruebas realizadas.

Modelo de Choset con comportamiento aditivo				
Simulación	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	34	16	68	14
2	37	13	74	12
3	39	11	78	14
4	40	10	80	11
5	39	11	78	13
6	37	13	74	15
7	38	12	76	9
8	40	10	80	11
9	36	14	72	16
10	31	19	62	11
11	36	14	72	17
12	37	13	74	12
13	41	9	82	19
14	43	7	86	31
15	37	13	74	14
16	43	7	86	19
17	38	12	76	18
18	37	13	74	30
19	35	15	70	13
20	37	13	74	15
21	37	13	74	12
22	34	16	68	14
23	40	10	80	18
24	43	7	86	12
25	39	11	78	12
Media	37.92	12.08	75.84	15.28
Desviación Estándar	2.86	2.86	5.71	5.20

Cuadro 7: Resultado de simulaciones en el caso B con el modelo de Choset con comportamiento aditivo

Modelo de Choset con comportamiento multiplicativo				
Simulación	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	40	10	80	12
2	41	9	82	15
3	37	13	74	23
4	41	9	82	13
5	37	13	74	15
6	38	12	76	19
7	39	11	78	20
8	40	10	80	20
9	40	10	80	13
10	41	9	82	21
11	39	11	78	19
12	37	13	74	12
13	35	15	70	18
14	37	13	74	18
15	40	10	80	10
16	37	13	74	25
17	41	9	82	10
18	38	12	76	16
19	37	13	74	25
20	38	12	76	16
21	43	7	86	18
22	39	11	78	11
23	36	14	72	19
24	40	10	80	21
25	39	11	78	14
Media	38.80	11.20	77.6	16.92
Desviación Estándar	1.88	1.88	3.75	4.31

Cuadro 8: Resultado de simulaciones en el caso B con el modelo de Choset con comportamiento multiplicativo

Modelo de Kim, Wang y Shin con comportamiento Aditivo				
Simulación	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	38	12	76	14
2	37	13	74	15
3	38	12	76	10
4	32	18	64	25
5	38	12	76	18
6	39	11	78	32
7	40	10	80	15
8	36	14	72	19
9	41	9	82	14
10	38	12	76	14
11	35	15	70	14
12	40	10	80	11
13	41	9	82	11
14	39	11	78	18
15	36	14	72	12
16	33	17	66	13
17	36	14	72	25
18	37	13	74	17
19	38	12	76	15
20	36	14	72	23
21	32	18	64	14
22	35	15	70	15
23	41	9	82	17
24	36	14	72	13
25	29	21	58	21
Media	36.84	13.16	73.68	16.60
Desviación Estándar	2.98	2.98	5.95	5.08

Cuadro 9: Resultado de simulaciones en el caso B con el modelo de Kim, Wang y Shin con comportamiento aditivo

Modelo de Kim, Wang y Shin con comportamiento multiplicativo				
Simulación	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	36	14	72	14
2	40	10	80	17
3	42	8	84	15
4	40	10	80	18
5	39	11	78	10
6	41	9	82	18
7	36	14	72	18
8	37	13	74	22
9	42	8	84	15
10	37	13	74	18
11	39	11	78	21
12	40	10	80	24
13	33	17	66	15
14	38	12	76	9
15	40	10	80	16
16	38	12	76	19
17	31	19	62	12
18	29	21	58	16
19	37	13	74	13
20	45	5	90	23
21	44	6	88	18
22	37	13	74	21
23	36	14	72	22
24	42	8	84	26
25	40	10	80	13
Media	38.36	11.64	76.72	17.32
Desviación Estándar	3.64	3.64	7.29	4.23

Cuadro 10: Resultado de simulaciones en el caso B con el modelo de Kim, Wang y Shin con comportamiento multiplicativo

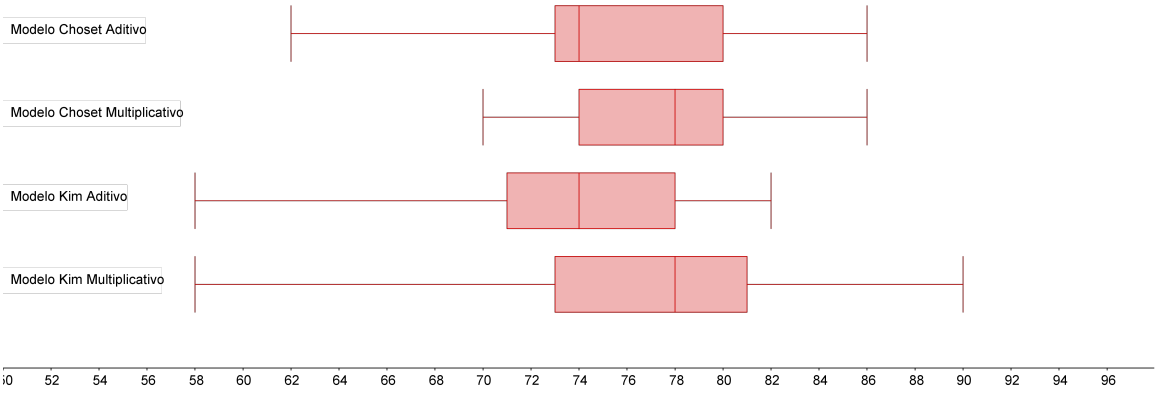


Figura 38: Diagrama boxplot de los resultados de rendimiento para el porcentaje de agentes exitosos con los modelos planteados para el caso B

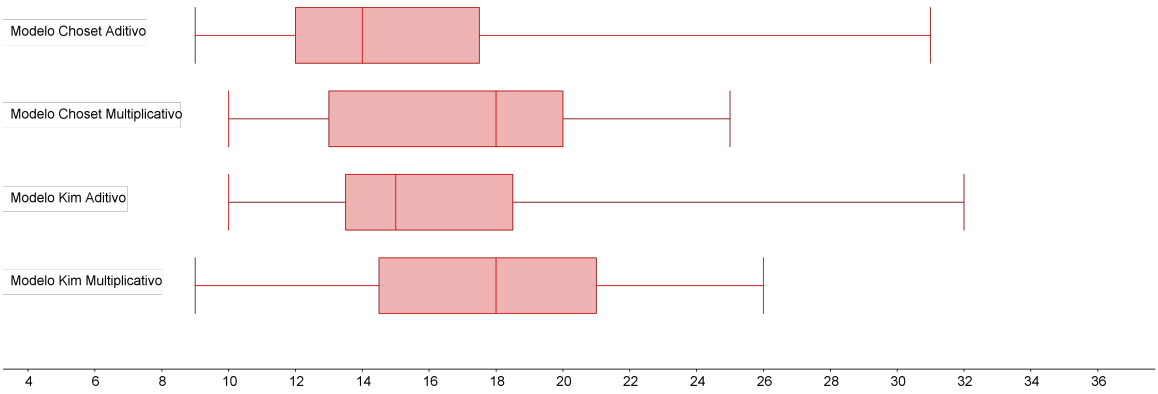


Figura 39: Diagrama boxplot de los resultados de rendimiento para la cantidad de iteraciones necesarias para la convergencia de los agentes con los modelos planteados para el caso B

8.2.3. Simulaciones para el caso C

En este caso también se realizaron veinticinco iteraciones con cincuenta agentes dispersos aleatoriamente sobre el campo del caso C. Mediante las iteraciones realizadas se obtuvieron las medias y desviaciones estándar del porcentaje de agentes exitosos y las medias y desviaciones estándar de la cantidad de iteraciones que se tomó el algoritmo PSO para que los agentes exitosos pudieran alcanzar la meta planteada a través de los obstáculos. Con estos datos se realizó una comparación del rendimiento de todos los modelos planteados.

Las pruebas realizadas con el modelo de Choset se presentan en el cuadro 11 para el comportamiento aditivo y en el cuadro 12 para el comportamiento multiplicativo. Para el modelo de Kim, WAng y Shin los resultados se muestran en el cuadro 13, mientras que para el modelo multiplicativo los resultados se muestran en el cuadro 14. Mediante las pruebas realizadas se determinó que el modelo de Choset con comportamiento aditivo y con comportamiento multiplicativo tuvieron promedios similares. El modelo de Choset con comportamiento multiplicativo tuvo una mayor dispersión de datos y desviación estándar. En el caso de las iteraciones necesarias para la convergencia de los agentes, el modelo de Choset con comportamiento multiplicativo demostró un promedio un mayor al modelo de Choset con comportamiento aditivo.

El modelo de Kim, Wang y Shin con comportamiento multiplicativo demostró una mayor convergencia de agentes exitosos a la meta frente al modelo con comportamiento aditivo. En el caso de las iteraciones necesarias para la convergencia de los agentes a la meta las pruebas realizadas demostraron que el modelo de Kim, Wang y Shin. con comportamiento multiplicativo necesitó en promedio una cantidad menor de iteraciones a comparación del modelo de Kim, Wang y Shin con comportamiento aditivo.

Como lo muestra el diagrama boxplot en la figura 40, las medias de convergencia de los agentes a la meta resultó ser mayor en el caso de los modelos elaborados con el modelo de Choset y se muestra que la dispersión de los datos es muy similar. En el caso de las iteraciones, el diagrama boxplot en la figura 41 demuestra que el modelo de Choset con

comportamiento multiplicativo tiene una menor dispersión de datos para este valor, y por lo tanto también se vuelve un modelo confiable al momento de implementarlo en el trabajo de búsqueda y rescate de personas.

Modelo de Choset con comportamiento Aditivo				
Simulación	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	48	2	96	32
2	44	6	88	19
3	44	6	88	25
4	43	7	86	33
5	48	2	96	28
6	47	3	94	24
7	48	2	96	33
8	42	8	84	46
9	48	2	96	22
10	44	6	88	25
11	44	6	88	21
12	46	4	92	23
13	48	2	96	19
14	48	2	96	21
15	48	2	96	37
16	46	4	92	33
17	47	3	94	16
18	46	4	92	25
19	47	3	94	42
20	47	3	94	22
21	47	3	94	23
22	46	4	92	19
23	47	3	94	29
24	46	4	92	44
25	43	7	86	22
Media	46.08	3.92	92.16	27.32
Desviación Estándar	1.83	1.83	3.66	8.06

Cuadro 11: Resultado de simulaciones en el caso C con el modelo de Choset con comportamiento aditivo

Modelo de Choset con comportamiento Multiplicativo				
Simulación	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	47	3	94	44
2	45	5	90	33
3	43	7	86	52
4	48	2	96	31
5	48	2	96	38
6	46	4	92	55
7	47	3	94	28
8	48	2	96	37
9	46	4	92	28
10	49	1	98	15
11	44	6	88	38
12	47	3	94	43
13	45	5	90	30
14	43	7	86	39
15	47	3	94	34
16	46	4	92	27
17	49	1	98	43
18	47	3	94	38
19	44	6	88	49
20	48	2	96	17
21	48	2	96	39
22	47	3	94	35
23	47	3	94	21
24	46	4	92	20
25	42	8	84	38
Media	46.28	3.72	92.56	34.88
Desviación Estándar	1.87	1.87	3.73	10.04

Cuadro 12: Resultado de simulaciones en el caso C con el modelo de Choset con comportamiento multiplicativo

Modelo de Kim, Wang y Shin con comportamiento aditivo				
Iteración	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	46	4	92	23
2	48	2	96	26
3	45	5	90	41
4	40	10	80	36
5	44	6	88	25
6	41	9	82	35
7	47	3	94	30
8	44	6	88	29
9	42	8	84	29
10	45	5	90	43
11	48	2	96	32
12	45	5	90	33
13	43	7	86	34
14	44	6	88	25
15	46	4	92	40
16	46	4	92	31
17	44	6	88	27
18	48	2	96	24
19	44	6	88	38
20	45	5	90	24
21	44	6	88	34
22	37	13	74	24
23	41	9	82	35
24	47	3	94	28
25	47	3	94	19
Media	44.44	5.56	88.88	30.60
Desviación Estándar	2.64	2.64	5.28	6.12

Cuadro 13: Resultado de simulaciones en el caso C con el modelo de Kim, Wang y Shin con comportamiento aditivo

Modelo de Kim, Wang y Shin con comportamiento multiplicativo				
Simulación	Agentes exitosos	Agentes no exitosos	% de éxito	Iteración de convergencia
1	47	3	94	31
2	43	7	86	24
3	44	6	88	36
4	42	8	84	22
5	48	2	96	35
6	44	6	88	16
7	43	7	86	29
8	45	5	90	26
9	47	3	94	33
10	47	3	94	21
11	48	2	96	43
12	44	6	88	30
13	42	8	84	30
14	47	3	94	36
15	45	5	90	44
16	44	6	88	44
17	38	12	76	21
18	44	6	88	27
19	46	4	92	52
20	44	6	88	23
21	46	4	92	23
22	47	3	94	32
23	40	10	80	29
24	45	5	90	35
25	39	11	78	40
Media	44.36	5.64	88.72	31.28
Desviación Estándar	2.62	2.62	5.24	8.54

Cuadro 14: Resultado de simulaciones en el caso C con el modelo de Kim, Wang y Shin con comportamiento multiplicativo

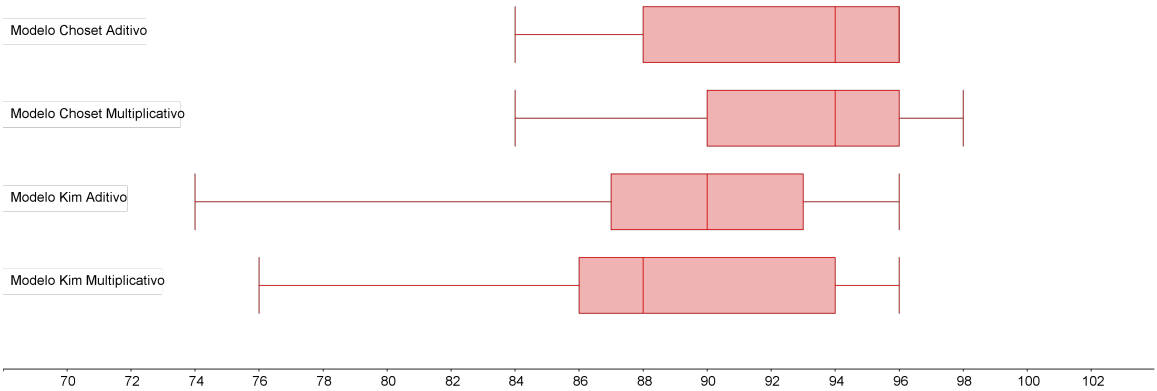


Figura 40: Diagrama boxplot de los resultados de rendimiento para el porcentaje de agentes exitosos con los modelos planteados para el caso C

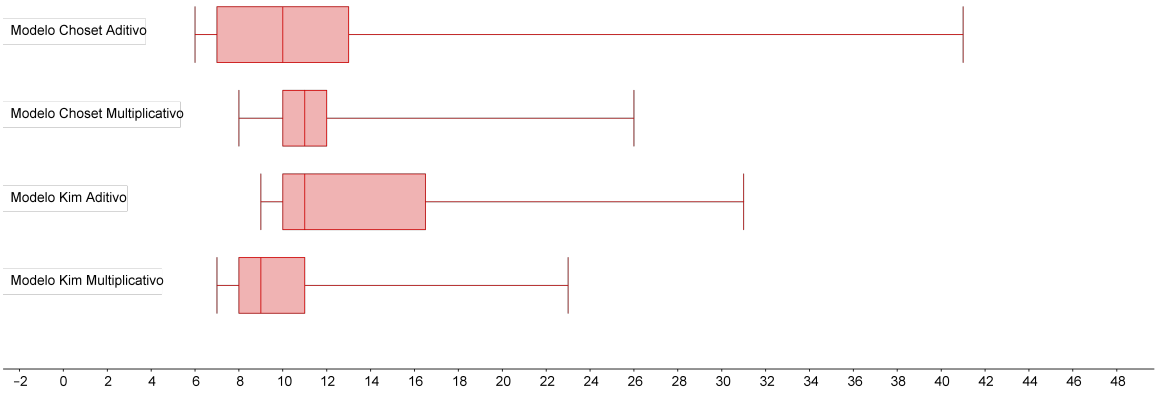


Figura 41: Diagrama boxplot de los resultados de rendimiento para la cantidad de agentes necesaria para la convergencia de los agentes con los modelos planteados para el caso C

8.2.4. Modelo a usar

En base a los resultados de las pruebas mostradas anteriormente, se determinó que el modelo de Choset con comportamiento multiplicativo demuestra una mejor combinación de resultados óptimos y confiabilidad en cuanto a variabilidad de los datos recabados en base al número promedio de agentes que convergieron a la meta planteada en cada casos A, B y C y el número de iteraciones que le tomó al algoritmo PSO para lograr que los agentes exitosos convergieran a las metas planteadas en cada simulación.

8.3. Simulación en casos planteados con modelo propuesto

Una vez realizadas las pruebas que determinaron el rango de valor óptimo de los parámetros del algoritmo PSO y el modelo matemático de funciones artificiales de potencial de mayor relevancia para trabajo de búsqueda y rescate, se procedió a realizar simulaciones que demostraran la convergencia del enjambre de agentes a la meta planteada en los casos A, B y C. Para demostrar los resultados de las pruebas realizadas se graficaron las trayectorias de los agentes en el mapa bidimensional de cada caso. Además se muestran las gráficas de las posiciones en x y en y de cada agente contra cada iteración realizada por el algoritmo PSO. Finalmente se generaron las gráficas del potencial de cada agente contra la iteración realizada por el algoritmo PSO.

Para cada simulación se usaron cien agentes dispersos aleatoriamente en el mapa generado y se le dio un máximo de 500 iteraciones al algoritmo PSO. Se dispersaron los agentes al inicio de cada simulación ya que el algoritmo de Particle Swarm Optimization tiende a hacer que los agentes naveguen juntos si se posicionan cercanos entre ellos al inicio, por lo que se realizó la suposición de que antes de aplicar el algoritmo de optimización de la función de costo se debía dar un tiempo para que los agentes se dispersaran aleatoriamente en el campo sin importar la ubicación de los obstáculos o la meta. Para los parámetros del algoritmo PSO se usó un valor de constricción de 0.5, un valor de 1 para w, un valor de 400 para α, β en 50 y un valor de 35 para γ.

Para el modelado de las funciones artificiales de Choset se usaron los siguientes parámetros: 5 para ζ, 10 para η, 2 para d^*_{goal} y 5 para Q_i en una cuadrícula de 20 unidades por 20 unidades.

8.3.1. Simulación en caso A

Se puede observar en las trayectorias de los agentes, figura 42, que una cantidad pequeña de agentes que se encontraban detrás del obstáculo lograron llegar a la meta; el resto de los agentes se quedaron atrapados en el mínimo local generado por el potencial de repulsión del obstáculo y el potencial de atracción de la meta. Este problema es un problema clásico del algoritmo de Particle Swarm Optimization, explicado en el trabajo [1]. Las gráficas de posición de los agentes a través de las iteraciones , figuras 43 y 44, permiten observar que se alcanzan las posiciones de los mínimos local y global, mientras que la gráfica de potencial permite observar como el algoritmo de Particle Swarm Optimization permite que

los agentes reduzcan su potencial a través de la función artificial hasta encontrar un mínimo local o global.

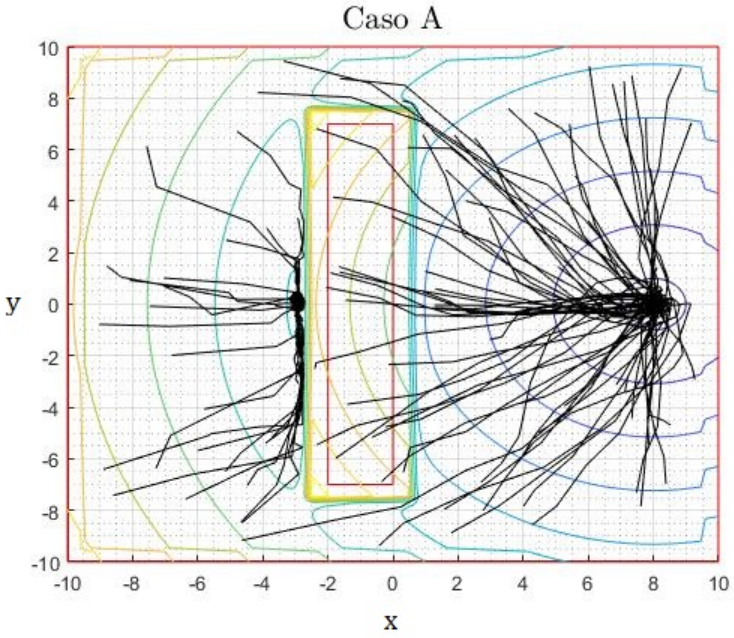


Figura 42: Trayectorias de los agentes en simulación del caso A

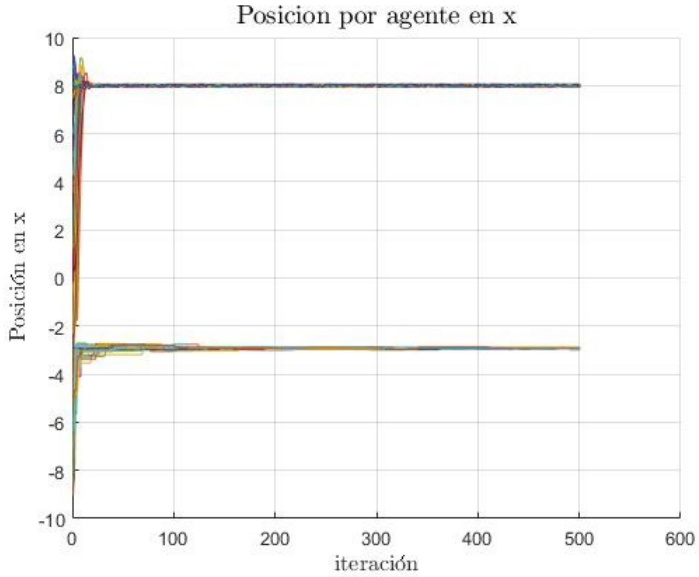


Figura 43: Posición en coordenadas X para cada agente en simulación de aaso A

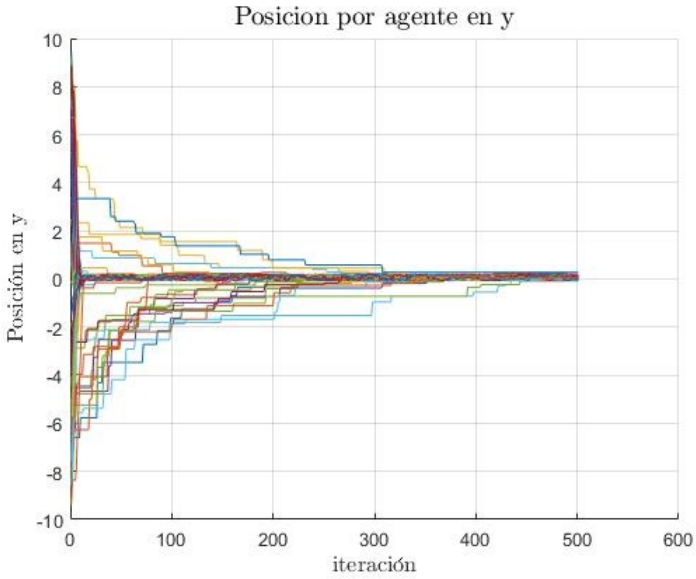


Figura 44: Posición en coordenadas Y para cada agente en simulación de caso A

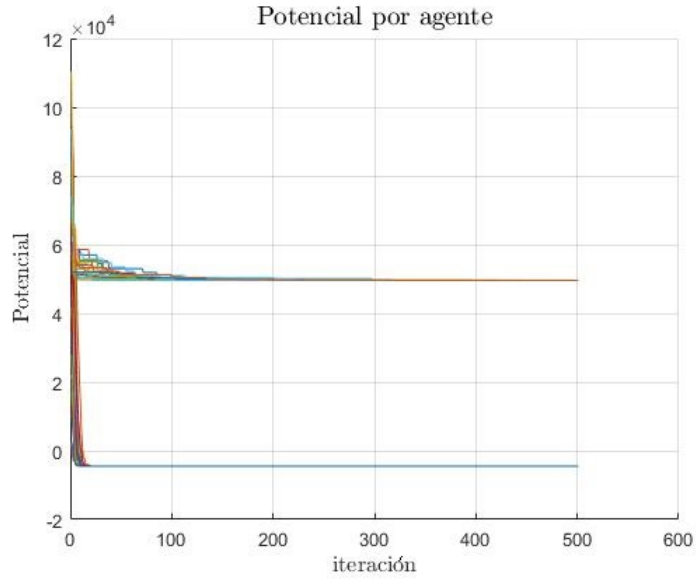


Figura 45: Potencial en modelo generado para cada agente en simulación de caso A

8.3.2. Simulación en caso B

En este caso es notable como el nivel de dificultad de evasión del obstáculo ha disminuido si se hace una comparación con el caso A. En este caso una mayor cantidad de agentes convergen a la meta, aunque sigue existiendo una cantidad de agentes que convergen al mínimo local.

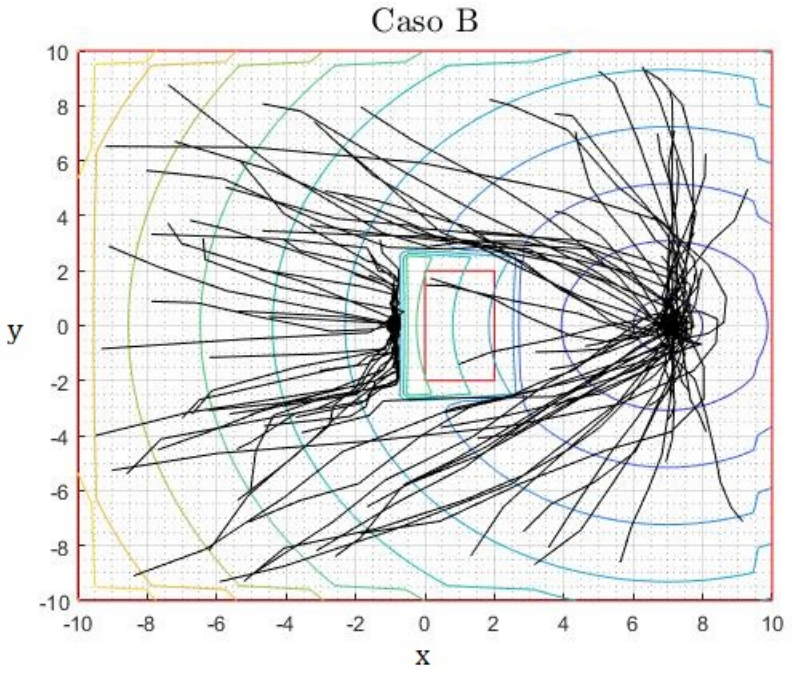


Figura 46: Trayectorias de los agentes en simulación del caso B

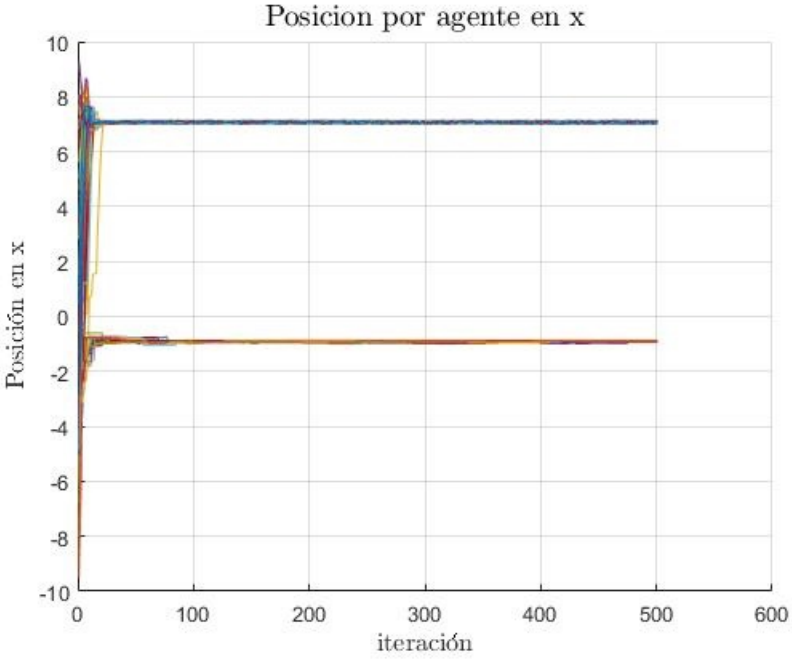


Figura 47: Posición en coordenadas X para cada agente en simulación de caso B

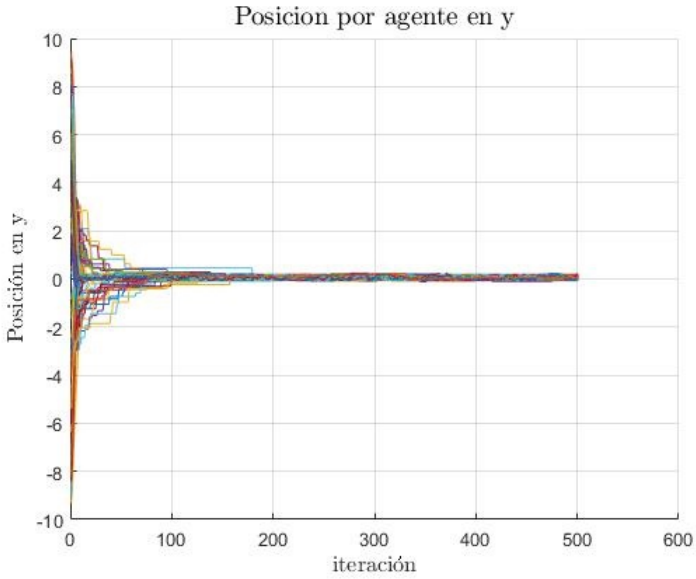


Figura 48: Posición en coordenadas Y para cada agente en simulación de caso B

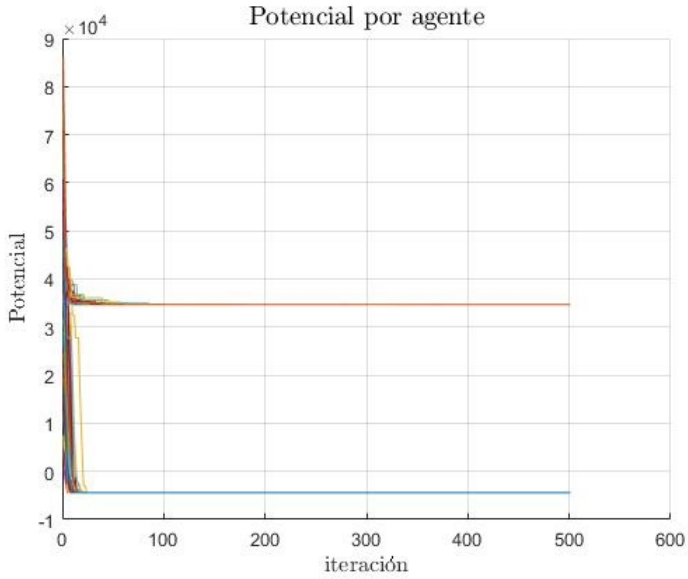


Figura 49: Potencial en modelo generado para cada agente en simulación de caso B

8.3.3. Simulación en caso C

En este caso vemos cómo una cantidad mínima de agentes han convergido a mínimos locales y la mayoría del sistema ha logrado converger al mínimo global.

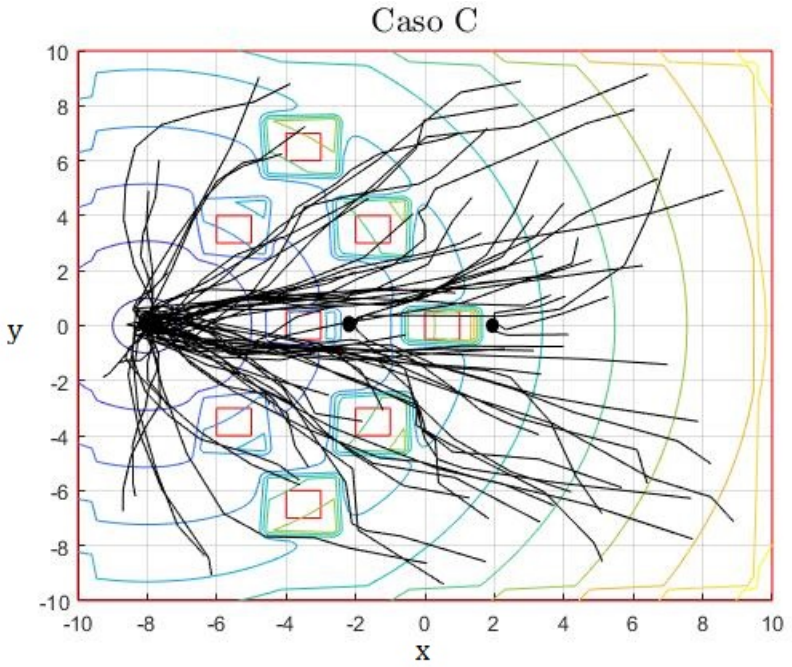


Figura 50: Trayectorias de los agentes en simulación del caso C

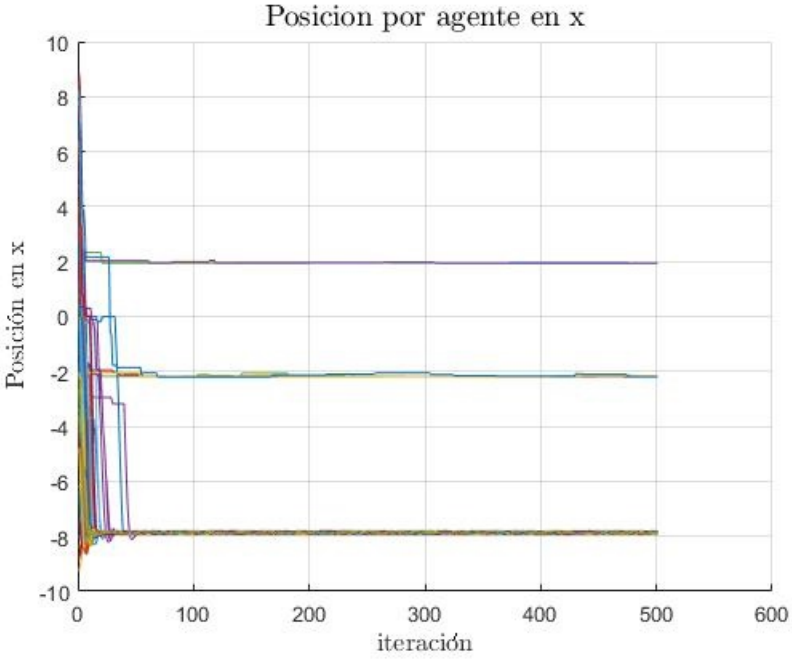


Figura 51: Posición en coordenadas X para cada agente en simulación de caso C

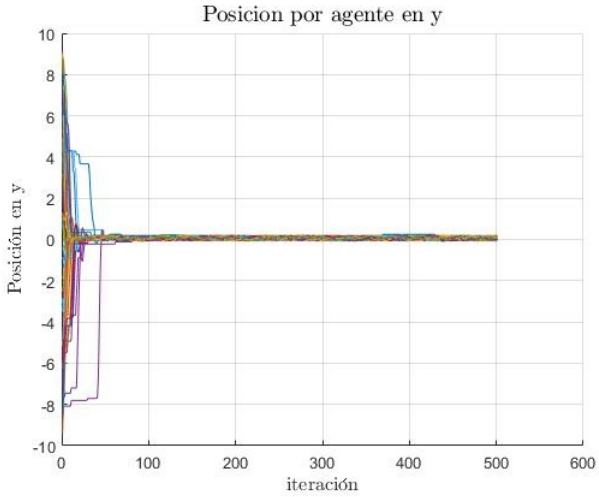


Figura 52: Posición en coordenadas Y para cada agente en simulación de caso C

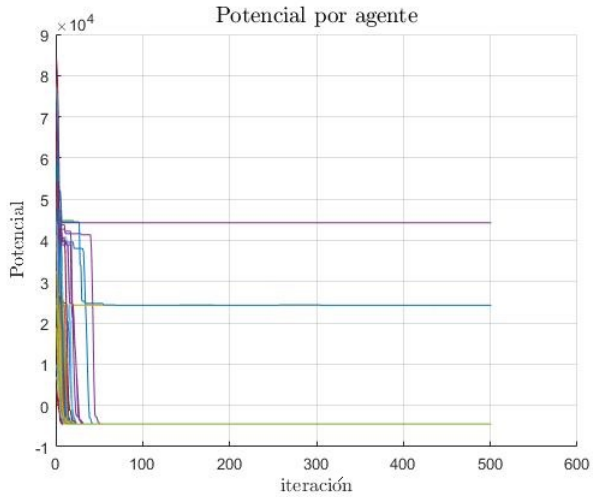


Figura 53: Potencial en modelo generado para cada agente en simulación de caso C

8.4. Solucionando el problema de convergencia

Se implementó el uso de mínimos locales artificiales para poder lograr que todos los agentes convergieran al finalizar la simulación. La idea consistió en hacer que los robots convergieran a un punto de manera artificial determinado en este caso heurísticamente luego de cierto tiempo de simulación en donde los agentes no podían avanzar mas allá de un obstáculo. Este tipo de comportamientos se puede observar en enjambres de la vida real en donde el potencial de atracción se ve influenciado por la trayectoria de los agentes que han logrado ser exitosos, como en el caso de las hormigas que se guían por las hormonas de sus compañeras. Este es otro tipo de inteligencia de enjambre conocido como *Optimización de Colonia de Hormigas* o ACO por sus siglas en inglés. Se puede conocer más sobre este tipo de inteligencia de enjambre en el trabajo *Ant Colony Optimization and Swarm Intelligence:*

En la Figura 54 se observa cómo las trayectorias de los agentes convergen hacia la meta luego de verse atrapadas en un mínimo local, mientras que en la Figura 55 se puede apreciar como el potencial de un sub-enjambre aumenta durante cierto tiempo y el potencial del otro se reduce, para que el potencial de los dos se disminuya al mínimo finamente.

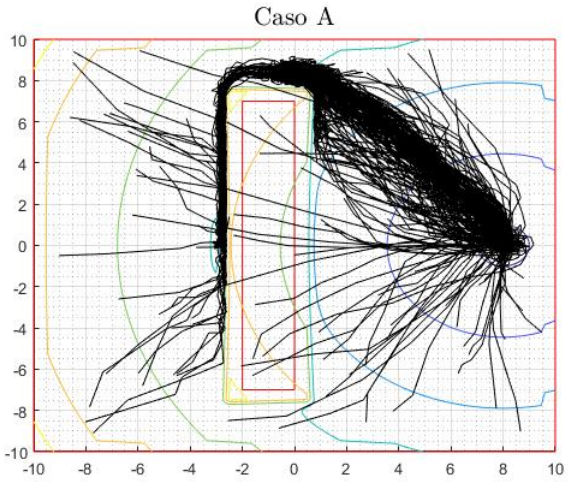


Figura 54: Mapa de simulación de caso A con mínimo artificial implementado

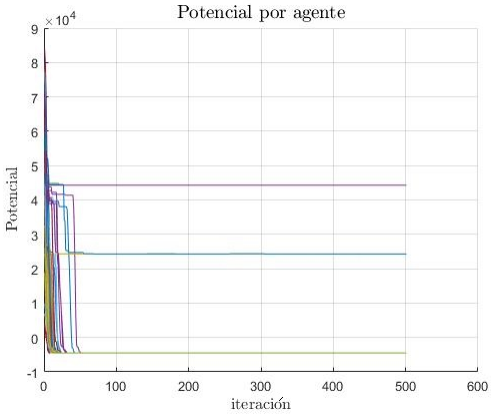


Figura 55: Potencial de agentes en simulación de caso A con mínimo artificial implementado

CAPÍTULO 9

Simulaciones generadas en entorno Webots

Se procedió a implementar el modelo planteado de funciones artificiales de potencial combinado con el algoritmo de Particle Swarm Optimization en el programa de simulación robótica Webots de Cyberbotics. Dicha implementación se realizó con el fin de poder determinar que el modelo propuesto anteriormente es implementable en condiciones reales. Para esto se realizó un modelo de cada caso planteado anteriormente en Webots con el modelo de robot e-puck y modelos de cajas de madera en modelo de arena cuadrada de 2 metros de cada lado, todos disponibles en la librería de modelos del programa.

Como se menciona en el trabajo [11], el robot e-puck ha sido desarrollado con las caracteristicas necesarias para poderlo tomar en cuenta en experimentaciones de sistemas multiagentes. Su bajo costo, su diseño a baja escala y su flexibilidad y variedad de sensores lo convierten en un modelo ejemplar para un sistema multiagentes y para el trabajo de búsqueda y rescate al cual se hace mención en el trabajo [2], por lo que es un robot ideal para evaluar el modelo de sistema multiagentes planteado en este trabajo.

Para poder implementar el modelo con función de potencial artificial y Particle Swarm Optimization se replicaron los casos descritos anteriormente. Se colocaron cajas de las proporciones similares en una arena rectangular de dos por dos metros y se dispersaron 10 agentes de modelo e-puck aleatoriamente en el espacio en cuestión.

9.1. Control de los robots en Webots

Para poder realizar el control de los robots se tomo como base el trabajo *Algoritmo Modificado de Optimizacion de Enjambre de Partículas (MPSO)* de Aguilar [13] y se añadió al controlador programado en lenguaje C la función artificial de potencial de Choset con comportamiento multiplicativo como función a optimizar.

Uno de los objetivos en el trabajo de Aguilar [13] consiste en encontrar el control adecuado para una simulación de Webots con robots e-puck que se mueven en función del algoritmo de Particle Swarm Optimization. En el trabajo mencionado se hace uso de control PID de velocidad angular con filtro de hard stops, control simple de pose de robot, control de Lyapunov de pose de robot diferencial y control de direccionamiento en lazo cerrado teniendo como resultado que el control PID es el que permite obtener las velocidades de motores más reguladas y el controlador de pose de Lyapunov el que permite obtener las trayectorias más suaves.

Los parámetros usados en [13] para la implementación física fueron de 0.8 para el parámetro de constricción K, 2 para el parámetro cognitivo α y de 10 para el parámetro social β. En este trabajo no se tomó en cuenta el parámetro de diversidad y se eligió como parámetro de inercia w un modelo natural exponencial en función de la distancia de los robots a la meta.

Por lo que para hallar el comportamiento óptimo del sistema multi-agentes en el trabajo de búsqueda y rescate tomando en cuenta un modelo realista de robot se realizaron iteraciones para cada controlador en cada caso planteado, y tanto con los parámetros de Particle Swarm Optimization encontrados en [13] como con los encontrados en este trabajo.

9.2. Simulación con controlador PID con filtro hard stops

En este caso se implementó a la velocidad angular del robot un controlador PID clásico con las constantes definidas en [13], K_p igual a 2.00, K_i igual a 0.10 y K_d igual a 0.01 junto con el filtro recomendado de cambios bruscos o *hard stops* de velocidades transmitidas a los actuadores de los robots que superase en la simulación la velocidad máxima de 6.28 m/s.

9.2.1. Simulaciones con parámetros hallados en condiciones ideales

Se realizaron las simulaciones con los parámetros K igual a 0.5, w igual a 1, α igual a 400, β igual a 50 y aplicando esta vez el parámetro de diversidad γ igual a 35. Se puede observar en las figuras 56 a 58 las trayectorias generadas por los e-pucks durante la simulación en Webots, donde las líneas azules representan las trayectorias, los puntos negros las metas a alcanzar y los rectángulos rojos los obstáculos en la arena. Se muestran en las figuras 59 a 61 los potenciales por agente en cada iteración del algoritmo simulado en Webots.

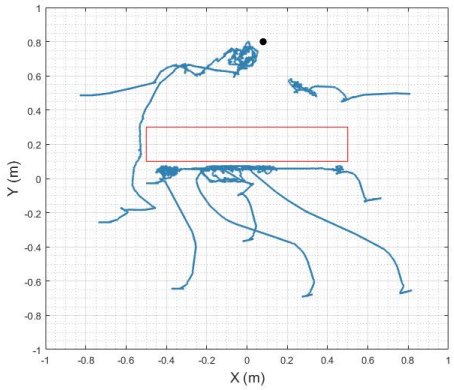


Figura 56: Trayectorias de agentes en simulación de caso A con controlador PID, filtro hard stops y parámetros hallados en condiciones ideales

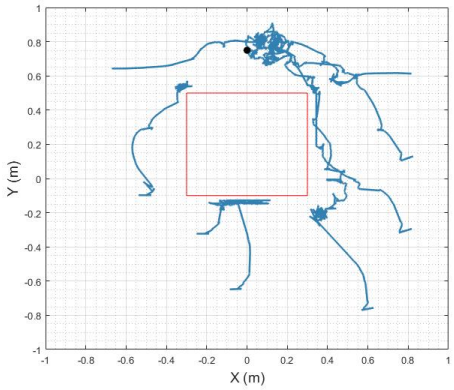


Figura 57: Trayectorias de agentes en simulación de caso B con controlador PID, filtro hard stops y parámetros hallados en condiciones ideales

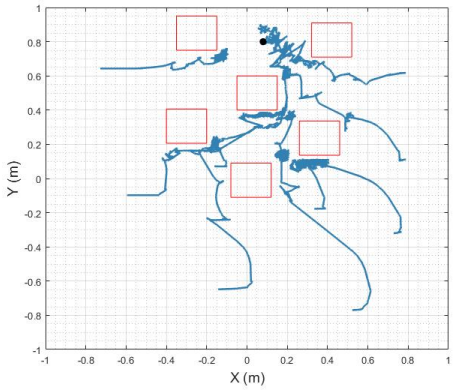


Figura 58: Trayectorias de agentes en simulación de caso C con controlador PID, filtro hard stops y parámetros hallados en condiciones ideales

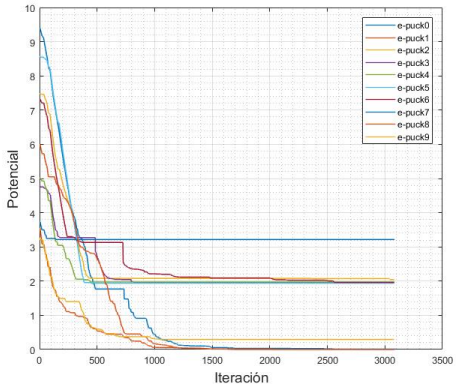


Figura 59: Potencial de agentes por iteración en simulación de caso A con controlador PID, filtro hard stops y parámetros hallados en condiciones ideales

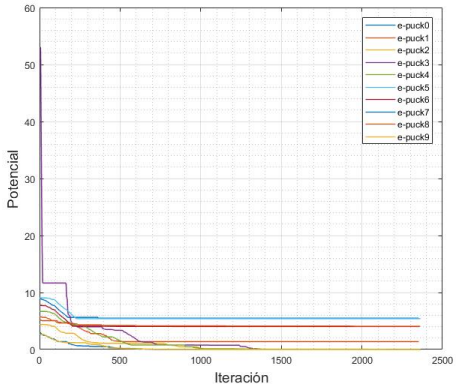


Figura 60: Potencial de agentes por iteración en simulación de caso B con controlador PID, filtro hard stops y parámetros hallados en condiciones ideales

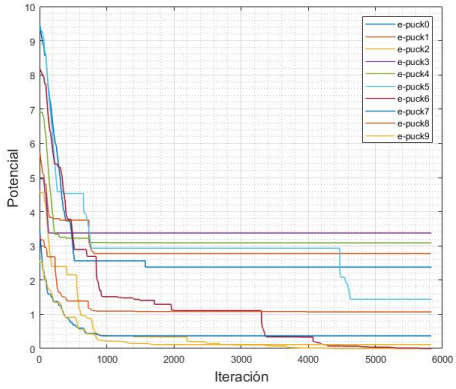


Figura 61: Potencial de agentes por iteración en simulación de caso C con controlador PID, filtro hard stops y parámetros hallados en condiciones ideales

Se puede observar que nuevamente los mínimos locales presentan un problema para la

convergencia de los agentes, y que conforme se reduce el tamaño de los obstáculos la cantidad de agentes que convergen a la meta aumenta, de acuerdo a lo esperado.

9.2.2. Simulación con parámetros nuevos

Despúes, se realizaron simulaciones con los parámetros del Particle Swarm Optimization definidos en [13]. Se puede observar en las figuras 62 a 64 las trayectorias generadas por los e-pucks durante la simulación en Webots, donde las líneas azules representan las trayectorias, los puntos negros las metas a alcanzar y los rectángulos rojos los obstáculos en la arena. Se muestran en las figuras 65 a 67 los potenciales por agente en cada iteración del algoritmo simulado en Webots.

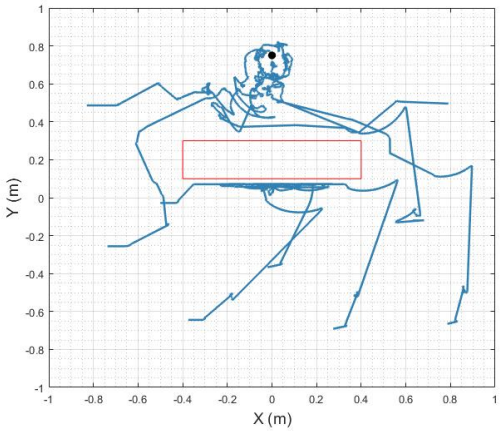


Figura 62: Trayectorias de agentes en simulación de caso A con controlador PID y filtro hard stops

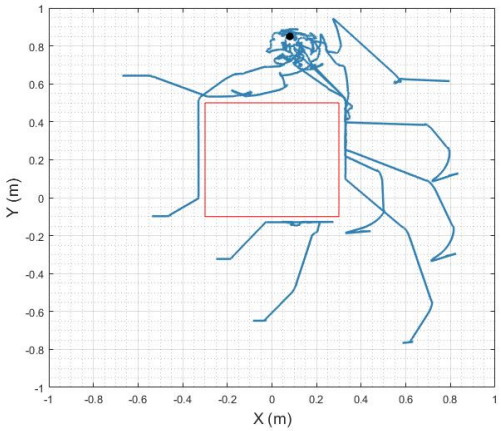


Figura 63: Trayectorias de agentes en simulación de caso B con controlador PID y filtro hard stops

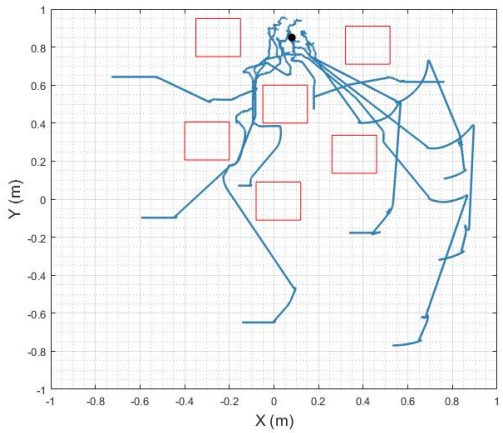


Figura 64: Trayectorias de agentes en simulación de caso C con controlador PID y filtro hard stops

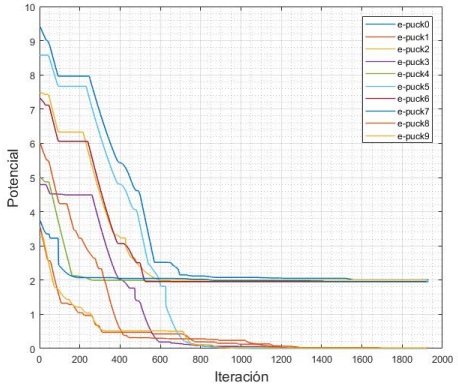


Figura 65: Potencial de agentes por iteración en simulación de caso A con controlador PID y filtro hard stops

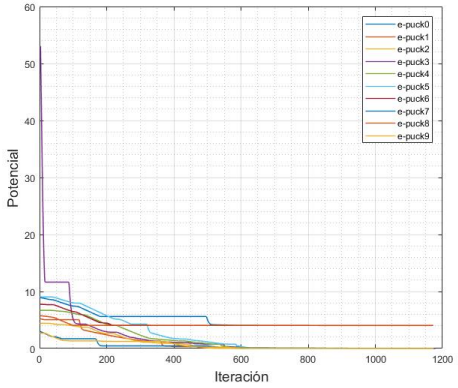


Figura 66: Potencial de agentes por iteración en simulación de caso B con controlador PID y filtro hard stops

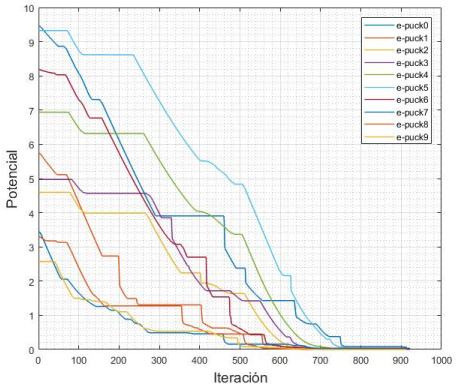


Figura 67: Potencial de agentes por iteración en simulación de caso C con controlador PID y filtro hard stops

Se puede observar como las trayectorias de los agentes no son suaves en ningún caso. Esto se debe a que se está controlando únicamente la velocidad angular de cada robot y no la pose. En el caso A se observa gracias a las trayectorias y a la gráfica de potenciales que la mitad de los agentes convergen a la meta y que se presenta nuevamente el problema de que ciertos agentes quedan atrapados en el mínimo local generado antes del obstáculo y la meta. Sin embargo, se puede observar como una mayor cantidad de agentes convergen a la meta en el caso B como se esperaba, y que todos los agentes convergen a la meta en el caso C.

9.2.3. Comparación entre simulaciones

Realizando una comparación entre simulaciones se observa que las trayectorias de los agentes son erráticas comparado con las trayectorias generadas sin el parámetro de diversidad y con los parámetros nuevos para el algoritmo de Particle Swarm Optimization. Se puede inferir que dicho comportamiento se debe a la adición del parámetro de diversidad, que, si bien en condiciones ideales ayuda a la exploración del espacio, al emplearlo en un modelo físico de un robot diferencial como el e-puck añade ruido al vector de dirección y por consiguiente a la trayectoria del agente. En estas simulaciones, el parámetro de diversidad es un factor que afectó en la convergencia de los agentes.

Además se puede observar que los potenciales de todos los agentes si tienden a disminuir conforme el algoritmo de Particle Swarm Optimization avanza, siendo la simulación con los parámetros nuevos la que mas rápido converge a un potencial menor en términos de iteraciones tomadas por simulación en todos los casos.

9.3. Simulaciones con controlador de pose de robot

Se hizo uso del controlador de pose de robot que se implementa en [13], el cual tiene como finalidad lograr trayectorias más suaves de convergencia a la meta establecida.

9.3.1. Simulaciones con parámetros hallados en condiciones ideales

Se realizaron las simulaciones con los parámetros K igual a 0.5, w igual a 1, α igual a 400, β igual a 30 y aplicando esta vez el parámetro de diversidad γ igual a 35.

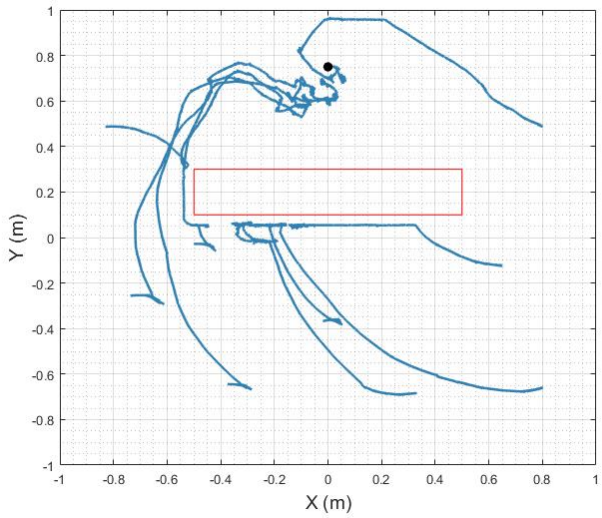


Figura 68: Trayectorias de agentes en simulación de caso A con controlador de pose simple y parámetros hallados en condiciones ideales

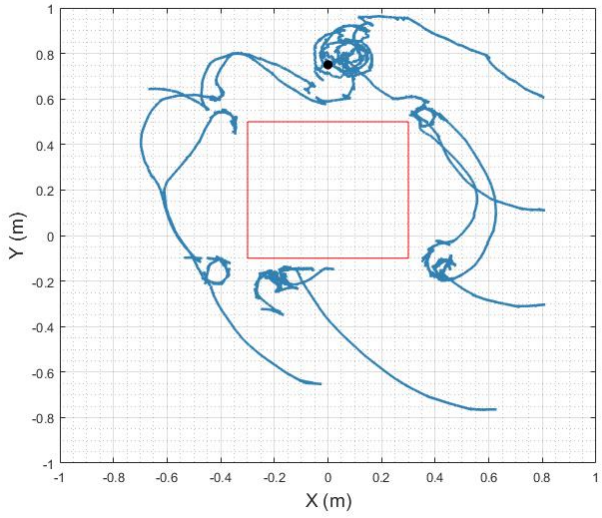


Figura 69: Trayectorias de agentes en simulación de caso B con controlador de pose simple y parámetros hallados en condiciones ideales

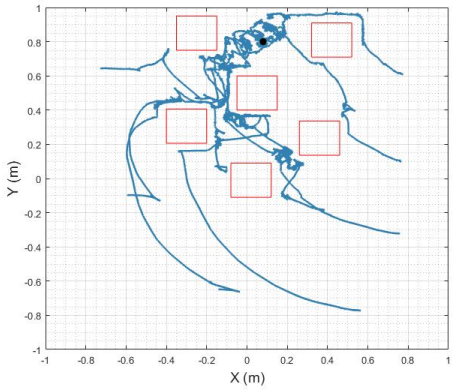


Figura 70: Trayectorias de agentes en simulación de caso C con controlador de pose simple y parámetros hallados en condiciones ideales

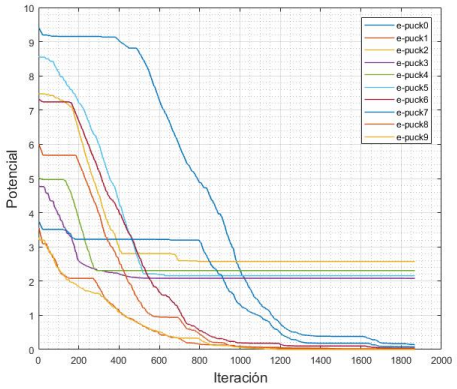


Figura 71: Potencial de agentes por iteración en simulación de caso A con controlador de pose simple y parámetros hallados en condiciones ideales

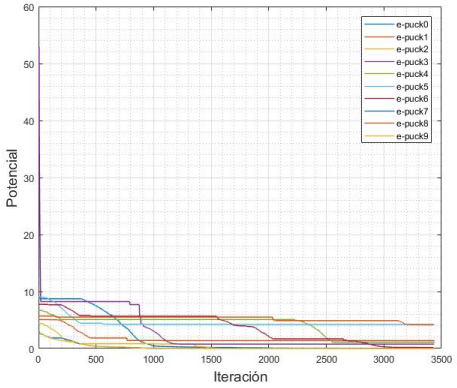


Figura 72: Potencial de agentes por iteración en simulación de caso B con controlador de pose simple y parámetros hallados en condiciones ideales

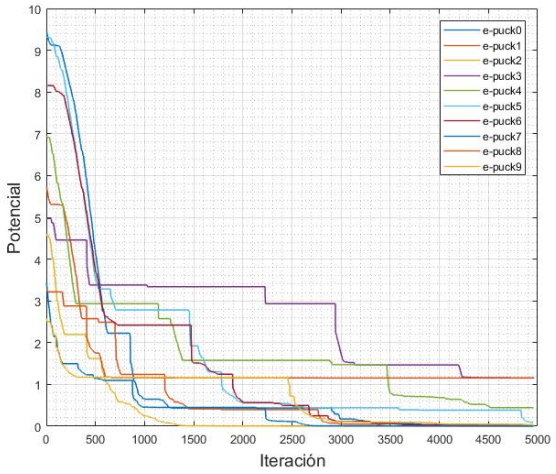


Figura 73: Potencial de agentes por iteración en simulación de caso C con controlador de pose simple y parámetros hallados en condiciones ideales

Esta vez se puede observar como el controlador permite que las trayectorias de convergencia a la meta de los agentes a la meta sean más suaves. En el la Figura 68 se observa como para el caso A, las trayectorias suaves ayudan a que los agentes puedan esquivar el obstáculo, lo que ayuda a una mayor convergencia hacia la meta. En este caso se el controlador mitigó el efecto del factor de diversidad del algoritmo de Particle Swarm Optimization.

9.3.2. Simulaciones con parámetros nuevos

Se realizaron simulaciones con los parámetros del Particle Swarm Optimization definidos en [13], en donde no se toma en cuenta el parámetro de diversidad.

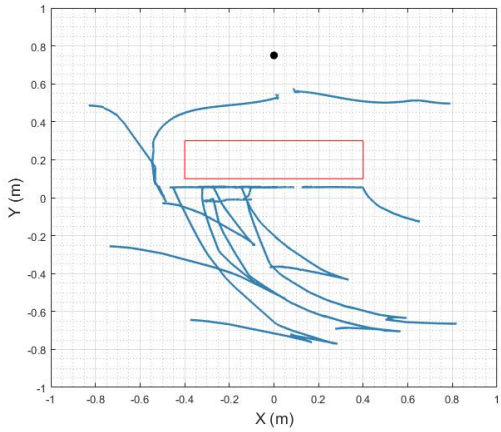


Figura 74: Trayectorias de agentes en simulación de caso A con controlador de pose simple y nuevos parámetros

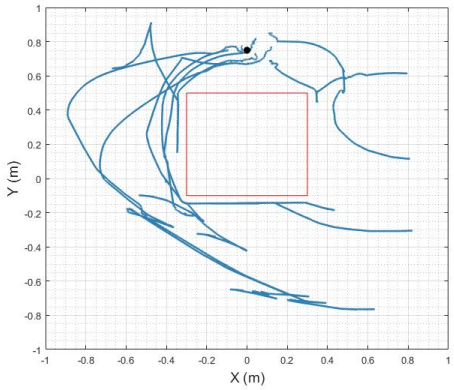


Figura 75: Trayectorias de agentes en simulación de caso B con controlador de pose simple y nuevos parámetros

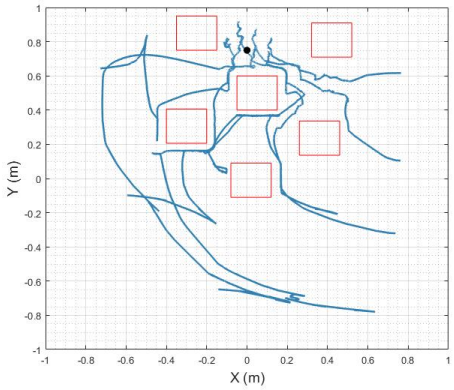


Figura 76: Trayectorias de agentes en simulación de caso C con controlador de pose simple y nuevos parámetros

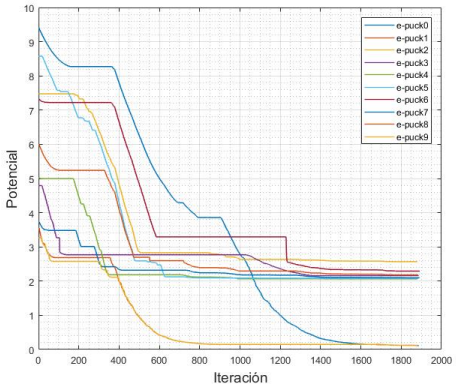


Figura 77: Potencial de agentes por iteración en simulación de caso A con controlador de pose simple y nuevos parámetros

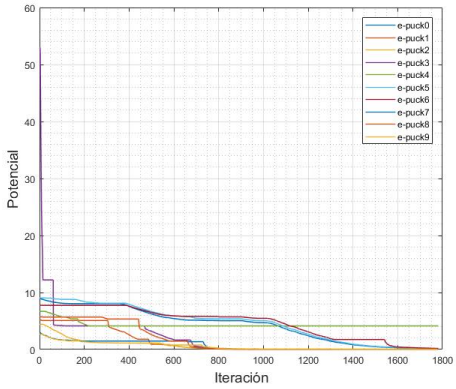


Figura 78: Potencial de agentes por iteración en simulación de caso B con controlador de pose simple y nuevos parámetros

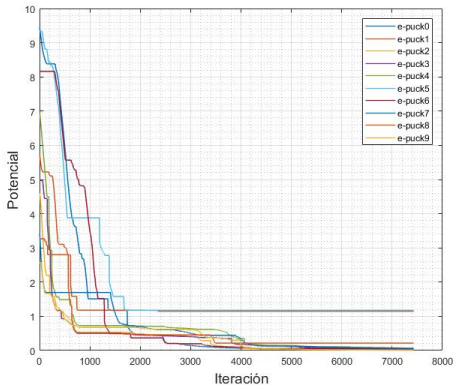


Figura 79: Potencial de agentes por iteración en simulación de caso C con controlador de pose simple y nuevos parámetros

9.3.3. Comparación entre simulaciones

Realizando una comparación entre las trayectorias del caso A, figuras 68 y 74, se puede observar que existe una mayor convergencia de agentes con los parámetros hallados en condiciones ideales. En los casos B y C la cantidad de robots que convergen a la meta es similar. Sin embargo, se puede observar en las figuras 73 y 76 que la simulación con parámetros nuevos de PSO tarda menos iteraciones en lograr que los agentes lleguen a la meta.

9.4. Simulación con controlador Lyapunov de pose de robot

Se implementó el controlador de pose de robot que se usa en [13], en donde se demuestra la convergencia asintótica estable del robot a la meta por medio de una trayectoria suave y estable por medio del Criterio de Estabilidad de Lyapunov.

9.4.1. Simulaciones con parámetros hallados en condiciones ideales

Se realizaron las simulaciones con los parámetros K igual a 0.5, w igual a 1, α igual a 400, β igual a 30 y aplicando esta vez el parámetro de diversidad γ igual a 35.

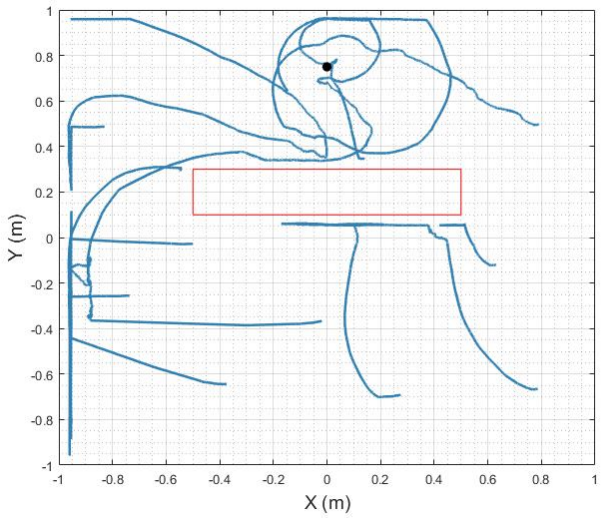


Figura 80: Trayectorias de agentes en simulación de caso A con controlador de Lyapunov y parámetros hallados en condiciones ideales

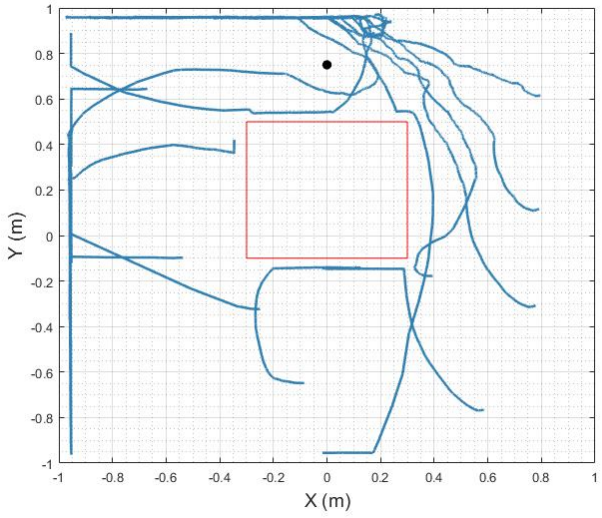


Figura 81: Trayectorias de agentes en simulación de caso B con controlador de Lyapunov y parámetros hallados en condiciones ideales

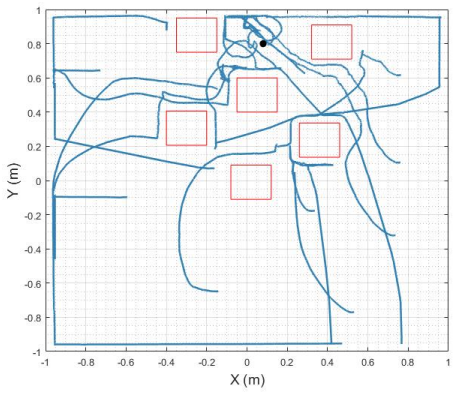


Figura 82: Trayectorias de agentes en simulación de caso C con controlador de Lyapunov y parámetros hallados en condiciones ideales

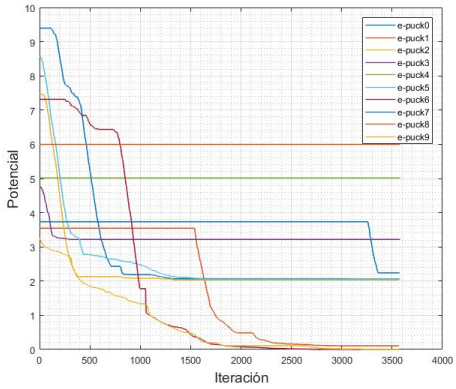


Figura 83: Potencial de agentes por iteración en simulación de caso A con controlador de Lyapunov y parámetros hallados en condiciones ideales

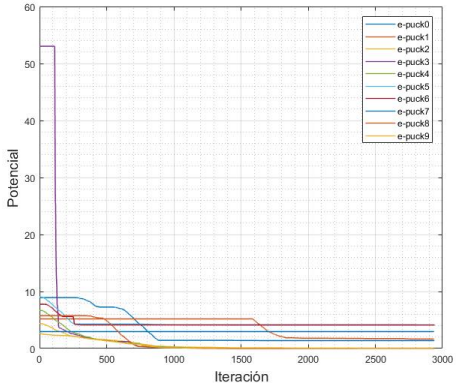


Figura 84: Potencial de agentes por iteración en simulación de caso B con controlador de Lyapunov y parámetros hallados en condiciones ideales

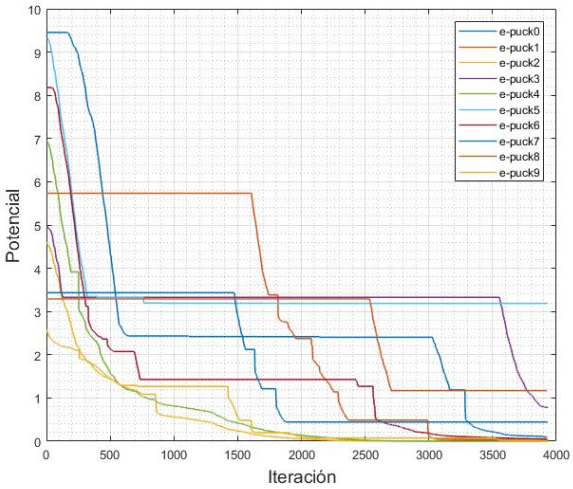


Figura 85: Potencial de agentes por iteración en simulación de caso C con controlador de Lyapunov y parámetros hallados en condiciones ideales

En el caso de esta implementación se observa como las trayectorias en estas simulaciones son suaves pero se ve afectada la convergencia de los robots a la meta.

9.4.2. Simulación con nuevos parámetros

Se realizaron simulaciones implementando el control de pose de Lyapunov con los parámetros del Particle Swarm Optimization definidos en [13], en donde no se toma en cuenta el parámetro de diversidad.

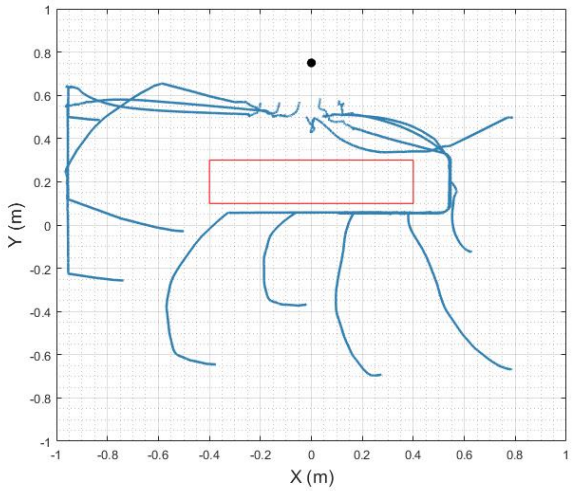


Figura 86: Trayectorias de agentes en simulación de caso A con controlador de pose de Lyapunov y nuevos parámetros

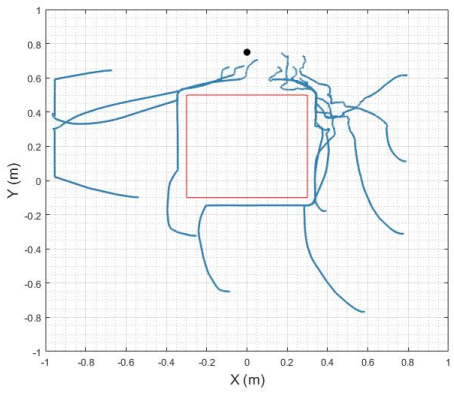


Figura 87: Trayectorias de agentes en simulación de caso B con controlador de pose de Lyapunov y nuevos parámetros

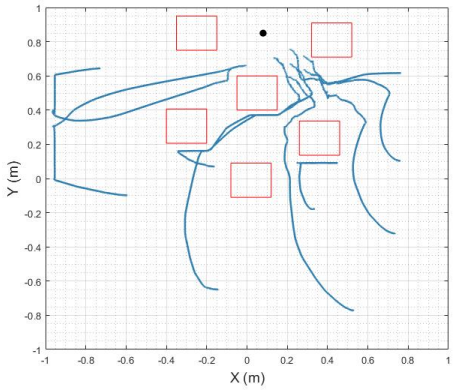


Figura 88: Trayectorias de agentes en simulación de caso C con controlador de pose de Lyapunov y nuevos parámetros

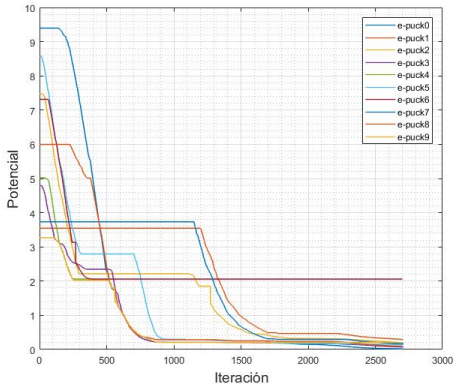


Figura 89: Potencial de agentes por iteración en simulación de caso A con controlador de pose de Lyapunov y nuevos parámetros

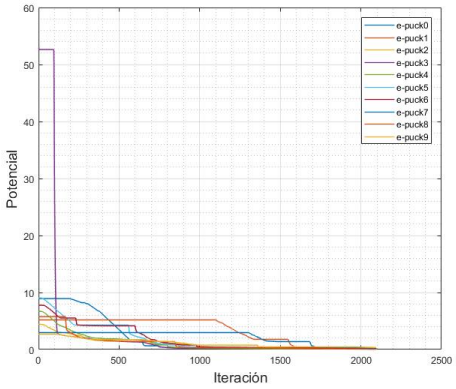


Figura 90: Potencial de agentes por iteración en simulación de caso B con controlador de pose de Lyapunov y nuevos parámetros

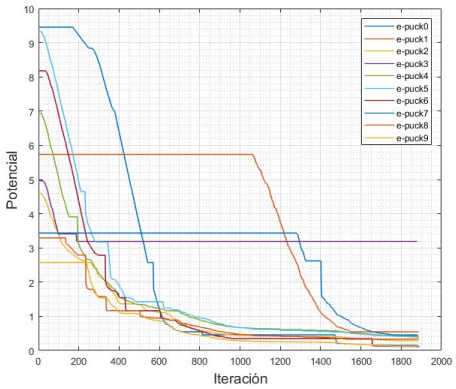


Figura 91: Potencial de agentes por iteración en simulación de caso C con controlador de pose de Lyapunov y nuevos parámetros

En la implementación de este controlador se observa que las trayectorias son mas suaves y estables en comparación a la simulación usando los parámetros en condiciones ideales. Es importante resaltar que en estas trayectorias se observa como el obstáculo es evadido en el caso A, y la convergencia es casi total en todos los casos presentados.

9.4.3. Comparación entre simulaciones

Existe una clara diferencia entre usar los nuevos parámetros y usar los parámetros hallados en condiciones ideales, y esta es de mayor importancia al destacar que se presenta una máxima convergencia en todos los casos ya que se evaden los obstáculos presentados con gran eficacia.

9.5. Simulación con controlador de lazo cerrado de direccionamiento de robot

Se implementó el controlador de lazo cerrado de direccionamiento de robot que se usa en [13].

9.5.1. Simulaciones con parámetros hallados en condiciones ideales

Se realizaron las simulaciones con los parámetros K igual a 0.5, w igual a 1, α igual a 400, β igual a 30 y el parámetro de diversidad γ igual a 35.

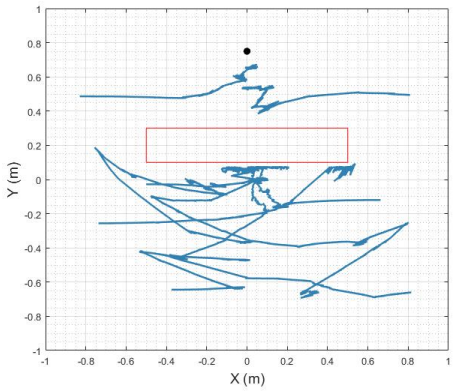


Figura 92: Trayectorias de agentes en simulación de caso A con controlador de lazo cerrado y parámetros hallados en condiciones ideales

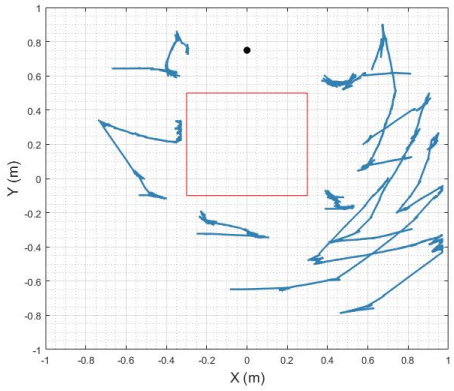


Figura 93: Trayectorias de agentes en simulación de caso B con controlador de lazo cerrado y parámetros hallados en condiciones ideales

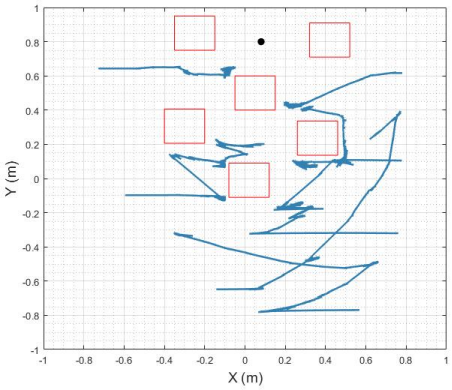


Figura 94: Trayectorias de agentes en simulación de caso C con controlador de lazo cerrado y parámetros hallados en condiciones ideales

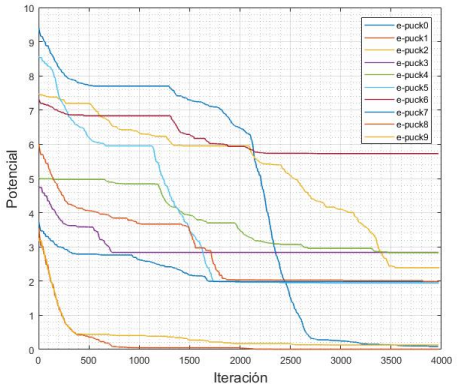


Figura 95: Potencial de agentes por iteración en simulación de caso A con controlador de lazo cerrado y parámetros hallados en condiciones ideales

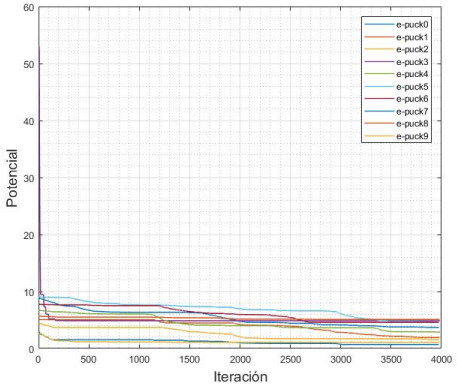


Figura 96: Potencial de agentes por iteración en simulación de caso B con controlador de lazo cerrado y parámetros hallados en condiciones ideales

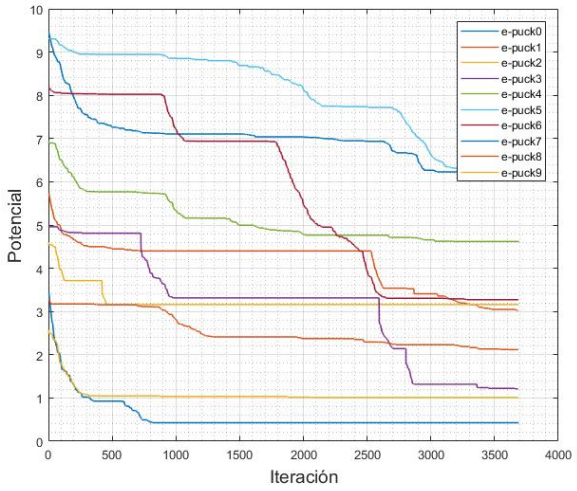


Figura 97: Potencial de agentes por iteración en simulación de caso C con controlador de lazo cerrado y parámetros hallados en condiciones ideales

En este caso se observa un comportamiento muy errático y no convergente en las trayectorias de los agentes. En ningún caso se logró la convergencia a las metas indicadas. Este comportamiento hace que automáticamente sea descartable esta combinación modelo de controlador con parámetros de Particle Swarm Optimization.

9.5.2. Simulación con nuevos parámetros

Se realizaron simulaciones implementando el control de lazo cerrado de direccionamiento de pose con los parámetros del Particle Swarm Optimization definidos en [13], en donde no se toma en cuenta el parámetro de diversidad.

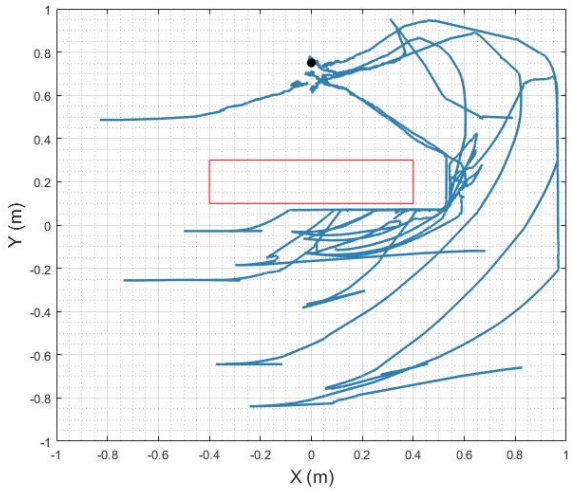


Figura 98: Trayectorias de agentes en simulación de caso A con controlador de lazo cerrado y nuevos parámetros

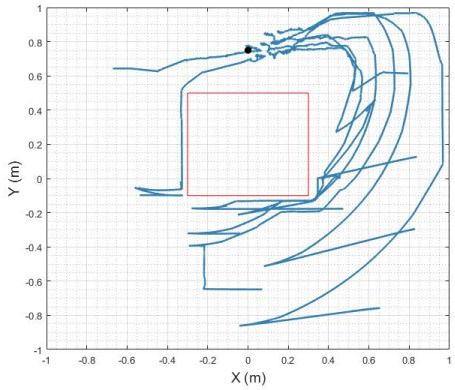


Figura 99: Trayectorias de agentes en simulación de caso B con controlador de lazo cerrado y nuevos parámetros

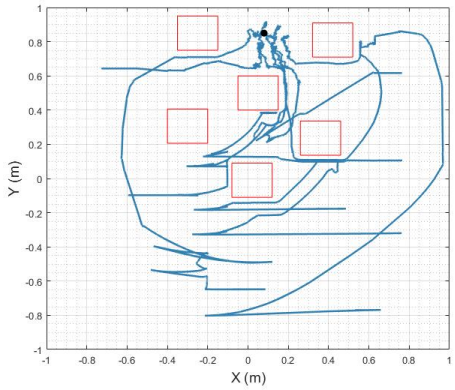


Figura 100: Trayectorias de agentes en simulación de caso C con controlador de lazo cerrado y nuevos parámetros

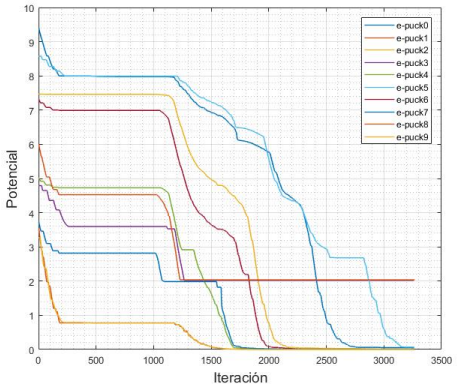


Figura 101: Potencial de agentes por iteración en simulación de caso A con controlador de lazo cerrado y nuevos parámetros

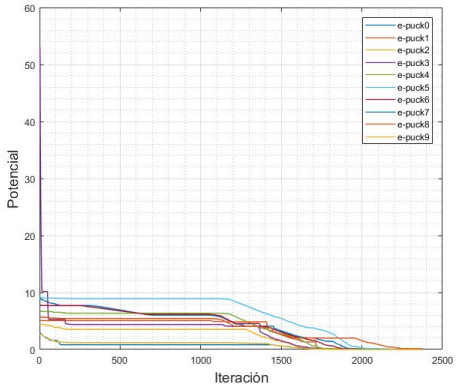


Figura 102: Potencial de agentes por iteración en simulación de caso B con controlador de lazo cerrado y nuevos parámetros

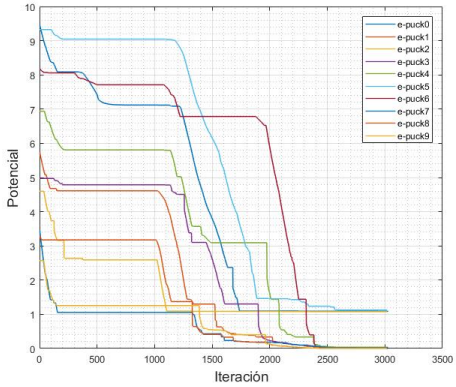


Figura 103: Potencial de agentes por iteración en simulación de caso C con controlador de lazo cerrado y nuevos parámetros

En la implementación del controlador de lazo cerrado de direccionamiento se nota una mejora con los parámetros nuevos, ya que las trayectorias logran evadir en su mayoría a los obstáculos en todos los casos presentados.

9.5.3. Comparación entre simulaciones

Se nota una gran diferencia entre las simulaciones implementando los parámetros del Particle Swarm Optimization definidos en [13] y los parámetros encontrados en condiciones ideales. Se ve además una convergencia de los agentes exitosos al mínimo potencial posible con los nuevos parámetros (figuras 101 a 103) en comparación al fallo de los agentes de llegar al potencial mínimo (figuras 95 y 97).

9.6. Simulaciones con controlador Linear Quadratic Regulator (LQR)

Se implementó el controlador Linear Quadratic Regulator (LQR) implementado en [13].

9.6.1. Simulaciones con parámetros hallados en condiciones ideales

Se realizaron las simulaciones con los parámetros K igual a 0.5, w igual a 1, α igual a 400, β igual a 30 y el parámetro de diversidad γ igual a 35.

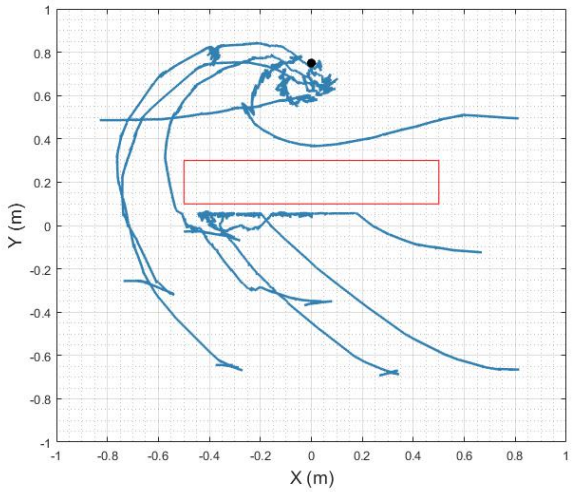


Figura 104: Trayectorias de agentes en simulación de caso A con controlador LQR y parámetros hallados en condiciones ideales

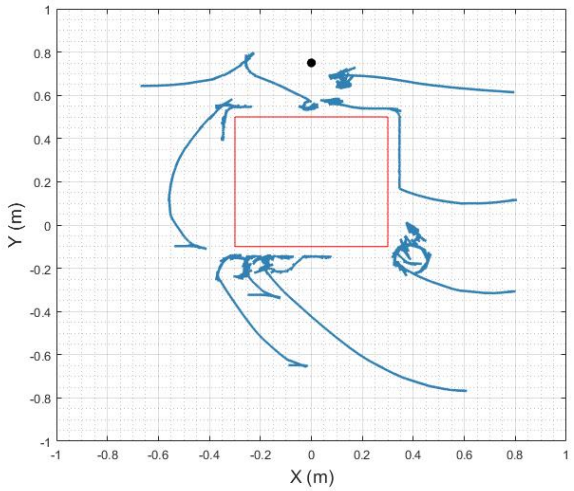


Figura 105: Trayectorias de agentes en simulación de caso B con controlador LQR y parámetros hallados en condiciones ideales

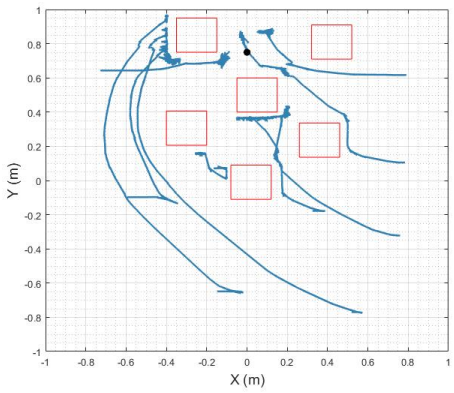


Figura 106: Trayectorias de agentes en simulación de caso C con controlador LQR y parámetros hallados en condiciones ideales

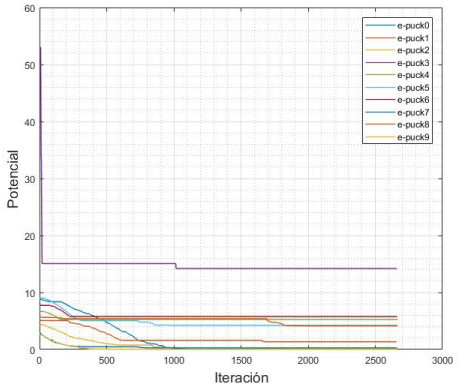


Figura 107: Potencial de agentes por iteración en simulación de caso A con controlador LQR y parámetros hallados en condiciones ideales

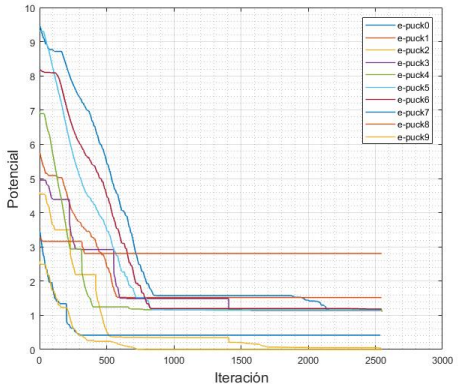


Figura 108: Potencial de agentes por iteración en simulación de caso B con controlador LQR y parámetros hallados en condiciones ideales

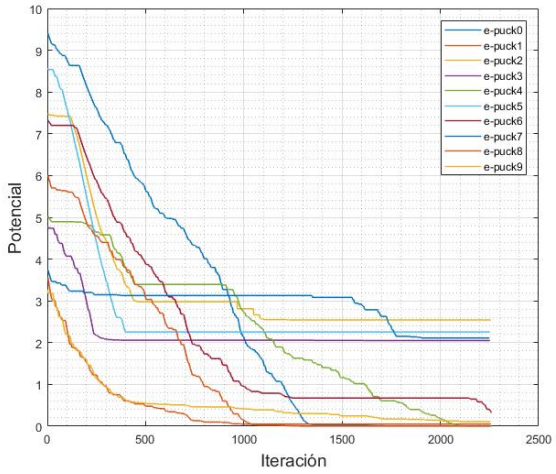


Figura 109: Potencial de agentes por iteración en simulación de caso C con controlador LQR y parámetros hallados en condiciones ideales

En este caso el comportamiento de las trayectorias generadas es suave, aunque con poca convergencia de agentes a la meta. En el caso A se nota una buena evasión del obstáculo. Sin embargo en el caso B y C el comportamiento es muy rígido al encontrarse con un obstáculo.

9.6.2. Simulación con nuevos parámetros

Se realizaron simulaciones implementando el control de lazo cerrado de direccionamiento de pose con los parámetros del Particle Swarm Optimization definidos en [13], en donde no se toma en cuenta el parámetro de diversidad.

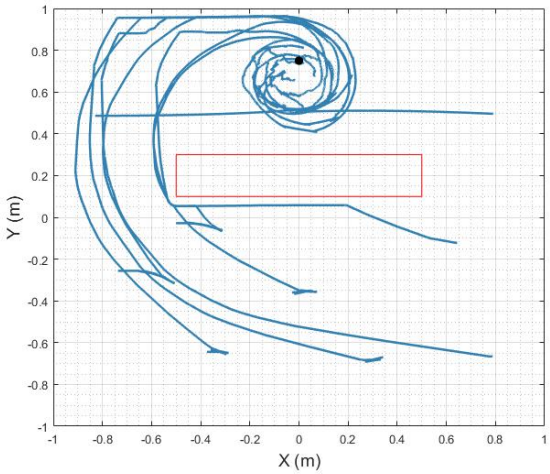


Figura 110: Trayectorias de agentes en simulación de caso A con controlador LQR y nuevos parámetros

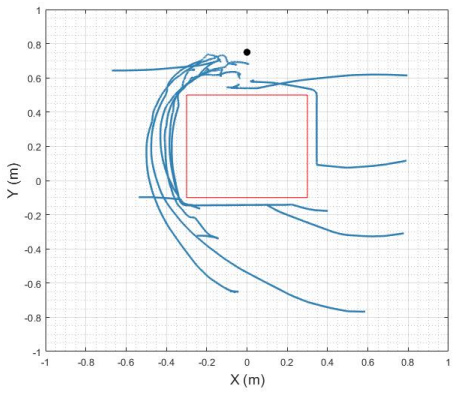


Figura 111: Trayectorias de agentes en simulación de caso B con controlador LQR y nuevos parámetros

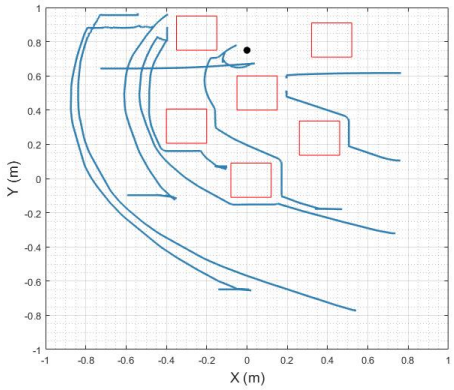


Figura 112: Trayectorias de agentes en simulación de caso C con controlador LQR y nuevos parámetros

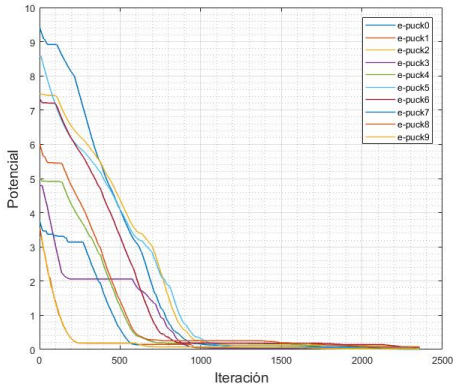


Figura 113: Potencial de agentes por iteración en simulación de caso A con controlador LQR y nuevos parámetros

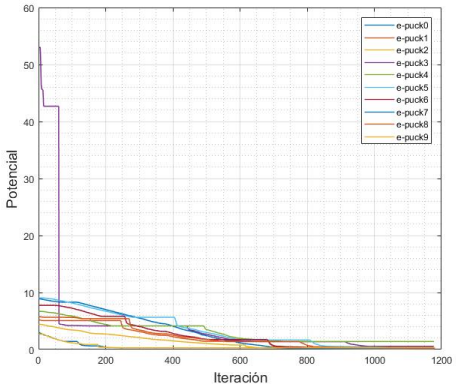


Figura 114: Potencial de agentes por iteración en simulación de caso B con controlador LQR y nuevos parámetros

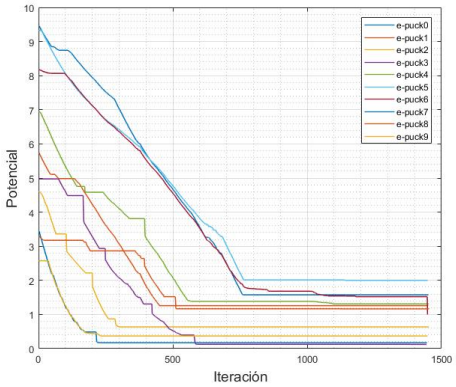


Figura 115: Potencial de agentes por iteración en simulación de caso C con controlador LQR y nuevos parámetros

En la implementación del controlador LQR se notó una gran mejora con los nuevos parámetros ya que las trayectorias son suaves y el número de iteraciones para poder llegar a la meta es bajo. La evasión de obstáculos en los casos A y B fue totalmente efectiva, logrando llevar a todos los agentes a la meta. Sin embargo, se puede notar que existe una falta de dinamismo cuando los agentes se topan con un obstáculo cuando están cerca de la meta.

9.6.3. Comparación entre simulaciones

Se nota una gran diferencia entre las simulaciones implementando los parámetros del Particle Swarm Optimization definidos en [13] y los parámetros encontrados en condiciones ideales. Se observa en las figuras 113 a 115 una mejor convergencia de los agentes al potencial mínimo en comparación al fallo de los agentes de llegar al potencial mínimo (figuras 107 y 109).

9.7. Comparando los resultados de las implementaciones de los controladores

Las simulaciones implementando el controlador PID con filtro hard stops mostraron un mejor desempeño con los parámetros encontrados en [13], al igual que el controlador de Lyapunov, el controlador de lazo cerrado de direccionamiento y el controlador LQR. Las simulaciones con el controlador de pose simple se desempeñaron mejor evadiendo obstáculos grandes con los parámetros hallados anteriormente en condiciones ideales y eso marcó la diferencia frente a la simulación con parámetros nuevos. La mejorada eficiencia en la implementación física con los nuevos parámetros del algoritmo PSO se debe a que estos son menores en por lo menos un orden de magnitud, por lo que no existe una saturación en las velocidades de los agentes.

Las simulaciones con el controlador PID demostraron tener un mejor desempeño en cuanto a iteraciones necesarias para la convergencia, a pesar de no tener el mejor desempeño evadiendo obstáculos que fueran de mayor tamaño a los dispersos. A pesar de que la cantidad de iteraciones fue la menor de todas en los casos A y B, la mitad de los agentes no lograron llegar a la meta en el caso A y tres no lograron llegar a la meta en el caso B. Sin embargo, en el caso C, la implementación del controlador PID demostró ser dinámico al llevar a todos los agentes a la meta en el menor tiempo de las simulaciones. Las simulaciones con el controlador de Lyapunov demostraron un desempeño muy bueno para evadir obstáculos grandes. En el caso A, solo dos agentes no llegaron a la meta, mientras que en el caso B todos los agentes lograron llegar. A pesar de que las simulaciones no eran las que requerían la menor cantidad de iteraciones para la convergencia de los agentes, estas son aceptables considerando que en éstas los agentes lograban evadir muy bien los obstáculos gracias a la suavidad y estabilidad de las trayectorias. Las simulaciones con el controlador LQR demostraron tener trayectorias suaves y capaces de evadir obstáculos de mediano tamaño. Sin embargo, se observó que su dinámica es reducida, ya que las simulaciones en el caso C, en donde los obstáculos son más pequeños y dispersos, demostraron que los agentes no convergían a la meta por no lograr evadir un obstáculo cuando se encontraban cerca del potencial mínimo. El número de iteraciones en el caso de las simulaciones con el controlador LQR demostraron ser iguales a las del PID, por lo que se consideran aceptables.

CAPÍTULO 10

Conclusiones

- Se elaboraron los casos A, B y C como casos que representan situaciones recurrentes en trabajos locales de búsqueda y rescate y que pueden servir para probar preliminarmente la eficiencia en la movilidad de un sistema multi-agente.

- Se elaboraron de manera eficaz lo campos artificiales de potencial de los casos significativos para el trabajo de búsqueda y rescate con el modelo matemático de Choset y el modelo matemático de Kim, Wang y Shin, combinando sus campos de atracción y de repulsión tanto con comportamiento multiplicativo como con comportamiento aditivo.

- En condiciones ideales, los parámetros del algoritmo PSO que dan un mejor comportamiento al sistema multi-agente son K igual a 0.5, w igual a 1, α igual a 400, β igual a 50 y γ igual a 35.

- El modelo de campos artificiales de potencial de Choset con comportamiento multiplicativo aplicado con los parámetros más eficaces del algoritmo PSO demostró un buen rendimiento y la confiabilidad adecuada según la cantidad de agentes que convergieron al mínimo local y la cantidad de iteraciones que le tomó al algoritmo PSO, por lo que sería el más indicado para el trabajo de búsqueda y rescate, frente a los otros modelos elaborados.

- Se logró resolver el problema de convergencia de sub-enjambres en mínimos locales mediante la implementación de mínimos artificiales que condujeron a todo el sistema al mínimo global.

- Las simulaciones con implementación realista de un sistema multi-agente que mostraron un mejor comportamiento, en general, en las trayectorias y potencial e los agentes fueron las que se generaron con los párametros de PSO K igual a 0.2, α igual a 2, β igual a 10, γ igual a 0, y el parámetro de inercia como un modelo exponencial decreciente. El parámetro de diversidad del algoritmo PSO provocó cambios erráticos en el movimiento de los robots.

- La implementación del controlador PID con filtro hard stops demostró darle un comportamiento más dinámico al sistema multi-agente, logrando evadir fácilmente los obstáculos de menor tamaño y dispersos en el campo.
- La implementación del controlador LQR demostró ser de mayor utilidad que el sistema multi-agente evadiera los obstáculos grandes en un tiempo menor al de la implementación de otros controladores.

CAPÍTULO 11

Recomendaciones

- Se recomienda usar visión de computadora en el Robotat de la Universidad del Valle de Guatemala para poder elaborar los campos artificiales de potencial y usarlos en una futura implementación.
- Para una futura implementación en un modelo físico, se recomienda empezar los modelos matemáticos con las constantes que se usaron en este trabajo para las funciones de repulsión y atracción, y ajustarlos heurísticamente según el comportamiento observado.
- Para una futura implementación física, se recomienda usar los parámetros del algoritmo de Particle Swarm Optimization que le dieron un mejor comportamiento el sistema multi-agente en las simulaciones de Webots.
- Para el control de los robots en una implementación física se recomienda comenzar por un PID clásico con filtro de hard stops. Si se desea, se recomienda usar posteriormente un controlador Linear Quadratic Regulator.
- En una futura implementación física, se recomienda realizar una dispersión aleatoria de los agentes sobre el campo antes de aplicar el modelo de campos artificiales de potencial con el algoritmo de Particle Swarm Optimization.

CAPÍTULO 12

Bibliografía

[1] J. Kennedy y R. Eberhart, "Particle swarm optimization (PSO)", en *Proc. IEEE International Conference on Neural Networks, Perth, Australia*, 1995, págs. 1942-1948.

[2] D. P. Stormont y M. D. Berkemeier, "Blue Swarm 2.5: A Step Toward an Autonomous Swarm of Search and Rescue Robots.", en *AAAI Mobile Robot Competition*, 2003, págs. 36-40.

[3] J. León, G. A. Cardona, A. Botello y J. M. Calderón, "Robot swarms theory applicable to seek and rescue operation", en *International Conference on Intelligent Systems Design and Applications*, Springer, 2016, págs. 1061-1070.

[4] R. Poli, "Analysis of the Publications on the Applications of Particle Swarm Optimisation", *J. Artif. Evol. App.*, vol. 2008, 4:1-4:10, ene. de 2008, ISSN: 1687-6229. DOI: 10.1155/2008/685175. dirección: https://doi.org/10.1155/2008/685175.

[5] R. Berganza, "¿Se repetirá la tragedia de El Cambray II", *Revista Cientıfica CONRED*, vol. 1, n.º 1, págs. 49-54, 2016.

[6] B. Benito, E. Molina y L. Laın, "METODOLOGÍA PARA ESTUDIO DE AMENAZA SÍSMICA EN GUATEMALA APLICACIÓN AL DISEÑO SISMORRESISTENTE", *Reporte de investigación*, 2001.

[7] J. Morales y C. Izquierdo, "Estructuras Colapsadas", en *Manual de estructuras colapsadas*. Grupo Tragsa y CEIS Guadalaja, 2015, cap. 3, págs. 242-290.

[8] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki y S. Thrun, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[9] D. H. Kim, H. Wang y S. Shin, "Decentralized control of autonomous swarm systems using artificial potential functions: Analytical design guidelines", *Journal of Intelligent and Robotic Systems*, vol. 45, n.º 4, págs. 369-394, 2006.

[10] S. Chowdhury, W. Tong, A. Messac y J. Zhang, "A mixed-discrete particle swarm optimization algorithm with explicit diversity-preservation", *Structural and Multidisciplinary Optimization*, vol. 47, n.º 3, págs. 367-388, 2013.

[11] C. M. Cianci, X. Raemy, J. Pugh y A. Martinoli, "Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics", en *International Workshop on Swarm Robotics*, Springer, 2006, págs. 103-115.

[12] P. Gonçalves, P. Torres, C. Alves, F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano y A. Martinoli, "The e-puck, a Robot Designed for Education in Engineering", *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, vol. 1, ene. de 2009.

[13] A. Aguilar, "Algoritmo Modificado de Optimización de Enjambre de Partículas (MPSO)", unpublished, N.D.

CAPÍTULO 13

Anexos

13.1. Código de simulaciones en MATLAB

El código usado para generar las simulaciones en condiciones ideales se puede encontrar en el siguiente enlace: *https://github.com/jpcahueque/PSO/tree/master/matlabSimulation* .

En este repositorio se pueden encontrar las siguientes siete funciones:

- *mainPSO.m*: es la función en donde se definen los parámetros tanto del modelo de Artificial Potential Fields de Choset como del modelo de Kim, Wang y Shin. Además, se definen los parámetros de la simulación como el tiempo inicial, el tiempo final y el cambio de tiempo por iteración. En esta función se debe configurar en 1 la variable *behaviour* si se desea un comportamiento multiplicativo o 0 si se desea un comportamiento aditivo, la variable *choset* debe ser 1 si se desea usar el modelo de Choset o 0 si se desea usar el modelo de Kim, Wang y Shin, la variable *showAPF* en 1 si se desea mostrar las gráficas de los Artificial Potential Fields o en 0 si no, y la variable *artificialMin* en 1 si se desea usar la idea de mínimos artificiales en el caso A. Este archivo es el único al que se le necesita dar RUN en MATLAB.

- *CasoA.m*: Es la función que modela el caso A. Recibe como parámetros las variables *gridsize,step,rounding,zeta,eta,dstar,Qi,*c_g*,*l_g*,*c_o*,*l_o*,choset,behaviour y showAPF* y devuelve *Utot* o la función de potencial generada.

- *CasoB.m*: Es la función que modela el caso B. Recibe como parámetros las variables *gridsize,step,rounding,zeta,eta,dstar,Qi,*c_g*,*l_g*,*c_o*,*l_o*,choset,behaviour y showAPF* y devuelve *Utot* o la función de potencial generada

- *CasoC.m*: Es la función que modela el caso C. Recibe como parámetros las variables *gridsize,step,rounding,zeta,eta,dstar,Qi,*c_g*,*l_g*,*c_o*,*l_o*,choset,behaviour y showAPF* y devuelve *Utot* o la función de potencial generada.

- *simulacion.m*: es la función encargada de generar una simulación del algoritmo PSO en el caso generado. Recibe como parámetros las variables *gridsize,initsize,step ,N,T,dt,w, alpha,beta,gamma,K,Utot,artifcialMin* y devuelve *histPos y histPot* que son matrices de historial de posición y de potencial de cada agente implementado.

- *mapNtoX*: es una función interna y sirve para transformar la posición de un agente en la matriz de potencial usada a la posición en coordenadas continuas.

- *mapXtoN*: es una función interna y sirve para transformar la posición en coordenadas continuas de un agente a la posición en la matriz de potencial.

El código fue generado en la versión de MATLAB R2017a, por lo que se garantiza su funcionamiento en esta versión.

13.2. Código de simulaciones en Webots

El código usado para generar las simulaciones en condiciones ideales se puede encontrar en el siguiente enlace: *https://github.com/jpcahueque/PSO/tree/master/webotsSimulation.*

En este repositorio se pueden encontrar tres carpetas: una con cada situación simulada para los casos A, B y C. Cada carpeta cuenta con su controlador respectivo, por lo que únicamente es necesario seguir los siguientes pasos para realizar una simulación.

- Descargar los archivos del repositorio mencionado.

- Abrir el programa Webots R2019b.

- Hacer clic en el menú File y luego en Open World

- Seleccionar el archivo con extensión wbt dentro de una de las carpetas CasoA, CasoB o CasoC.

- Correr la simulación en tiempo real o al doble de velocidad.

Para cambiar de controlador es necesario configurar las variables PID_CONTROLLER, NKC_CONTROLLER_1,NKC_CONTROLLER_2, NKC_CONTROLLER_3, LQR_CONTROLLER, HARDSTOP_FILTER en 0 o en 1 para seleccionarlos. La variable NKC_CONTROLLER_1 selecciona el controlador de pose simple, la variable NKC_CONTROLLER_2 selecciona el controlador de Lyapunov, la variable NKC_CONTROLLER_3 selecciona el controlador de lazo cerrado y la variable HARDSTOP_FILTER selecciona el filtro hard stop disponible únicamente con el controlador PID. Finalmente se debe seleccionar el parámetro USE_BEARING si se usa el controlador PID.

13.3. Velocidades de los robots obtenidas de las simulaciones en Webots

A continuación se presentan las velocidades de uno de los agentes en la simulación del caso A, con los parámetros de PSO óptimos e implementando el parámetro de diversidad.

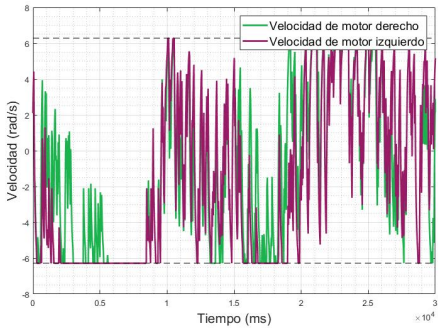


Figura 116: Velocidad de los motores implementando controlador PID y parámetros óptimos de PSO

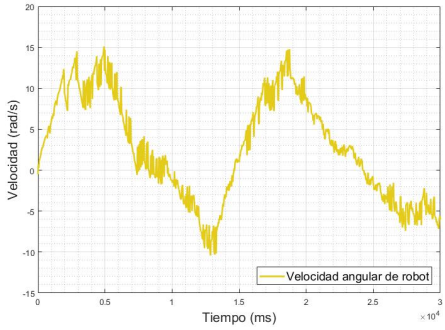


Figura 117: Velocidad anngular del robot implementando controlador PID y parámetros óptimos de PSO

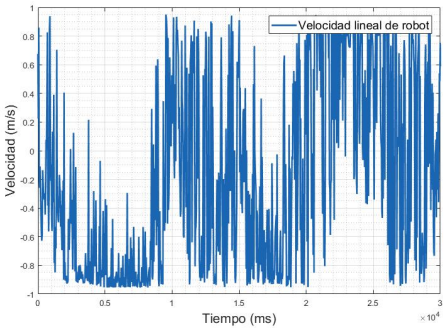


Figura 118: Velocidad lineal del robot implementando controlador PID y parámetros óptimos de PSO

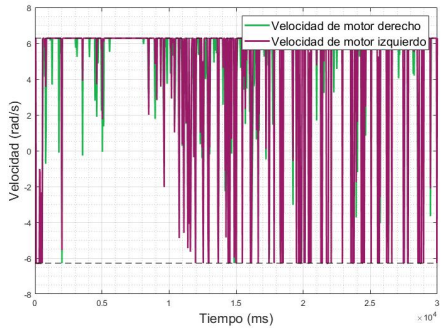


Figura 119: Velocidad de los motores implementando controlador LQR y parámetros óptimos de PSO

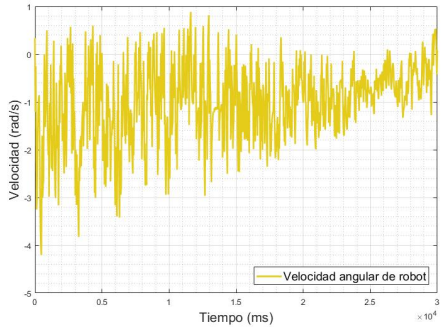


Figura 120: Velocidad anngular del robot implementando controlador LQR y parámetros óptimos de PSO

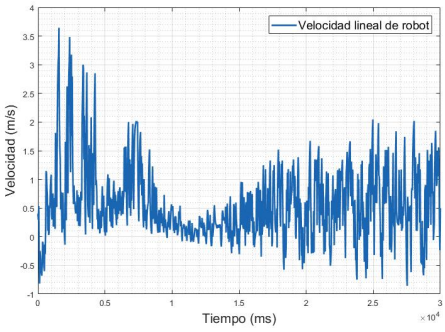


Figura 121: Velocidad lineal del robot implementando controlador LQR y parámetros óptimos de PSO

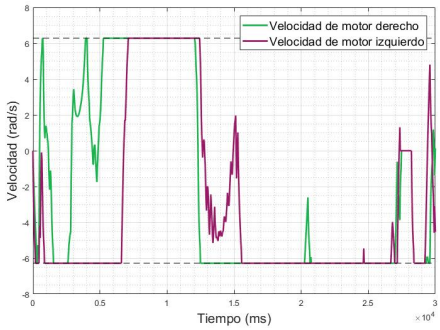


Figura 122: Velocidad de los motores implementando controlador PID y parámetro de diversidad en PSO

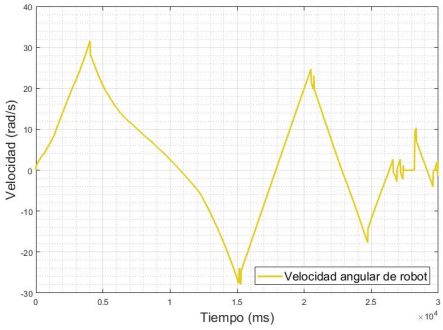


Figura 123: Velocidad anngular del robot implementando controlador PID y parámetro de diversidad en PSO

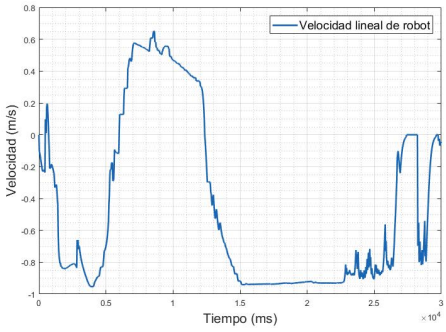


Figura 124: Velocidad lineal del robot implementando controlador PID y parámetro de diversidad en PSO

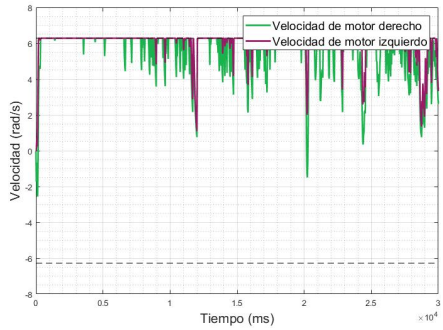


Figura 125: Velocidad de los motores implementando controlador LQR y parámetro de diversidad en PSO

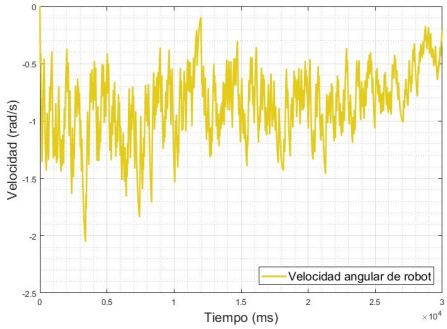


Figura 126: Velocidad anngular del robot implementando controlador LQR y parámetro de diversidad en PSO

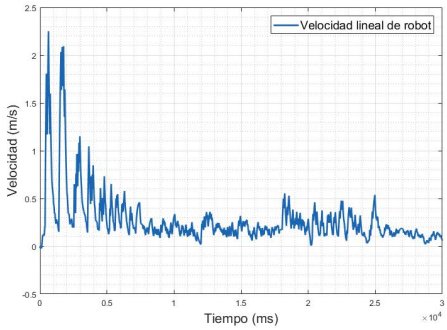


Figura 127: Velocidad lineal del robot implementando controlador LQR y parámetro de diversidad en PSO

13.4. Capturas de pantalla de simulaciones en Webots

A continuación se presentan capturas de pantalla de los agentes convergiendo a la meta en las simulaciones de Webots.

Figura 128: Agentes convergiendo en caso A

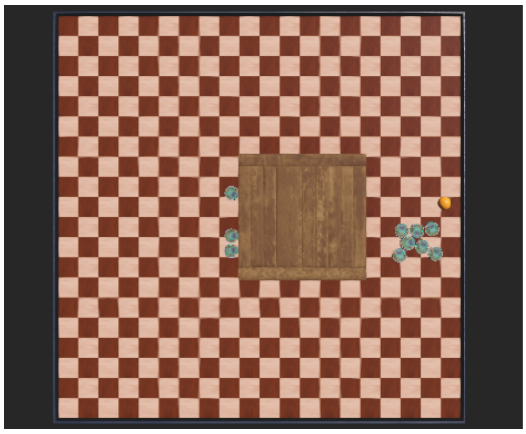


Figura 129: Agentes convergiendo en caso B

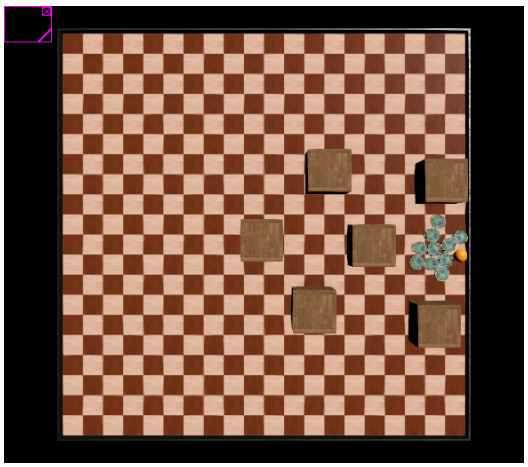


Figura 130: Agentes convergiendo en caso C