

Potential Field Algorithm Optimization Using PSO

1st Daniel Felipe Ardila Ariza
Facultad de Ingeniería Electrónica)
Universidad Santo Tomás)
Bogotá D.C., Colombia

2nd Corredor Cely Jorge Luis
Facultad de Ingeniería Electrónica
Universidad Santo Tomás
Bogotá D.C., Colombia

3rd García Carrillo Juan Camilo
Facultad de Ingeniería Electrónica
Universidad Santo Tomás
Bogotá D.C., Colombia

Abstract—Unmanned aerial vehicles (UAVs) are research objects of great importance, due to the advantages that they offer. However, current technologies need to be improved for higher efficiency in different scenarios. One of the most important fields is to detect and avoid obstacles. For this field, several navigation algorithms have been proposed, such as the potential fields algorithm. The objective of this article is to optimize this algorithm using the PSO method.

Index Terms—Potencial fields, PSO, Navigation

I. INTRODUCCIÓN

El uso de robots para minimizar el trabajo humano se ha convertido en un campo de investigación muy importante. Desde el uso de robots móviles para la limpieza, hasta robots militares, la robótica se ha convertido en un campo importante para el desarrollo de la humanidad. Sin embargo, un campo importante a tratar son los vehículos no tripulados los cuales necesitan un sistema que detecte y evite obstáculos. El algoritmo Potential field es uno de los algoritmos más importantes y tradicionales para solucionar este tipo de problemas. Este algoritmo depende de diferentes parámetros, los cuales se pueden definir de diferentes maneras. Este documento propone la obtención de estos parámetros a través del uso de un algoritmo de optimización denominado PSO.

La Optimización por Enjambre de Partículas, conocida como PSO (Particle Swarm Optimization) es una técnica de búsqueda y optimización, en donde cada partícula tiene una posición en el espacio de búsqueda, y una velocidad que determina el movimiento de las mismas a través del espacio, en donde cada partícula es atraída hacia la mejor localización que se encuentra basado en su época. La fuerza con la que estas partículas son empujadas en cada una de las direcciones depende de dos parámetros que pueden ajustarse de forma que, a medida que las partículas se alejan de estas localizaciones mejores, la fuerza de atracción es mayor. También se suele incluir un factor aleatorio que influye en cómo las partículas son empujadas hacia estas localizaciones.

II. PROCEDIMIENTO

A. Algoritmo Potential field

Un campo potencial es cualquier campo físico que obedece la ecuación de Laplace. Algunos ejemplos de estos campos potenciales, se encuentran en los campos eléctricos, magnéticos y gravitacionales. Un algoritmo de campo potencial utiliza el campo potencial artificial para regular el comportamiento del robot en un espacio determinado. En nuestro caso, al usar el

software de CoppeliaSim Edu cada espacio se divide en una cuadrícula.

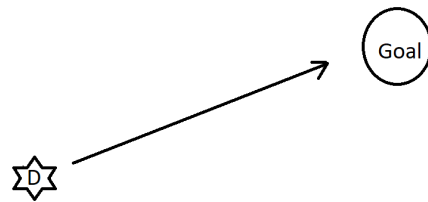


Figure 1. Ejemplo campo potencial

El algoritmo asigna un campo potencial artificial cada punto del mundo. El dron simula desde el potencial más alto, hasta el potencial más bajo. Por lo tanto, se puede decir que el dron al acercarse al objetivo reduce su velocidad.

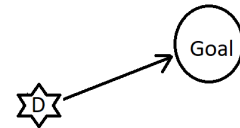


Figure 2. Reducción de atracción

Ahora al considerar un campo con obstáculos, el algoritmo debe considerar una fuerza de repulsión hacia estos, con el fin de evitar colisiones. Un ejemplo de esto, se muestra en la siguiente figura.

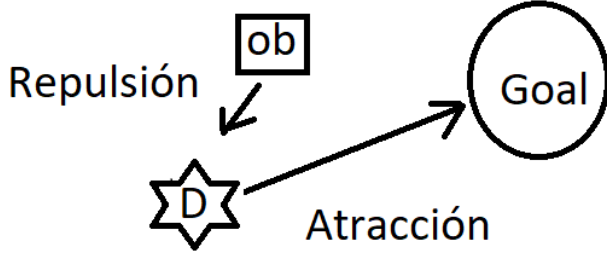


Figure 3. Considerando obstaculos

1) *Optimización por enjambre de partículas*: PSO, Es un método de optimización inspirado en la habilidad de organismos sociales, con el fin de resolver problemas complejos, involucrando las capacidades individuales de cada miembro y haciendo uso de las dinámicas sociales de un grupo. Un grupo de individuos, los cuales tienen un poco de inteligencia cada uno, pueden crear una gran inteligencia social haciendo uso de unas simples reglas de interacción.

En PSO, una partícula tiene información de sí misma, y de su grupo. La mejor solución de grupo, es a su vez, la mejor solución global. La mejor solución global, es usada por cada partícula, para ajustar su dirección de desplazamiento. El desplazamiento de cada partícula obedece las siguientes ecuaciones.

$$V_{i+1} = wV_i + c1 * (x_{ibest} - x_i) + c2 * (x_{Gbest} - x_i) \quad (1)$$

$$x_{i+1} = x_i + V_{i+1} \quad (2)$$

En la ecuación 2 está compuesta por tres terminos, el primero w es la inercia. La cual modera la velocidad actual de la partícula. El segundo termino $c1$, se refiere al componente cognitivo individual de la partícula. Y el último termino $c2$ representa el componente cognitivo grupal de cada partícula.

2) *Aplicación de PSO al problema*: El sistema debe ajustar los vectores atracción y repulsión, con el fin de encontrar los parámetros que establezcan una navegación segura y lo más rápida posible.

$$V_{Dron} = Kp_{obs} * V_{Repulsión} + Kp_{goal} * V_{Atracción} \quad (3)$$

Los rangos de los parámetros a ajustar son los siguientes:

$$Kp_{obs_{min}}, Kp_{obs_{max}} = 0.25, 0.3 \quad (4)$$

$$Kp_{goal_{min}}, Kp_{goal_{max}} = 0.2, 0.3 \quad (5)$$

Estos valores de rangos, se definieron a partir de diferentes pruebas que se hicieron en el sistema. En las diferentes pruebas se evidencio que las mejores posiciones de las partículas siempre se establecian en estos rangos.

A partir de los anteriores rangos, se estableció una velocidad por época igual a 0.01, y la cantidad de épocas se definió en 15.

3) *Entorno de simulación y resultados*: A continuación se presenta el entorno de simulación propuesto.

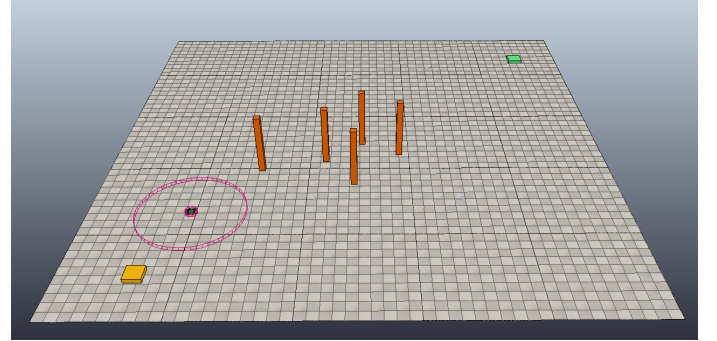


Figure 4. Entorno de simulación

En la anterior figura se observa el dron al cual se le va a implementar el algoritmo de navegación. Este dron, cuenta con un sensor de proximidad con un rango de detección de 2 metros. Además de esto se puede observar las marcas de inicio y de final. Y por último se mencionan los obstaculos, los cuales son 5 y cada uno tiene una altura lo suficientemente alta, para que afecten la trayectoria del dron.

A continuación, se muestra los diferentes tiempos que se tarda el dron a través de las épocas, con la implementación del algoritmo de optimización PSO.

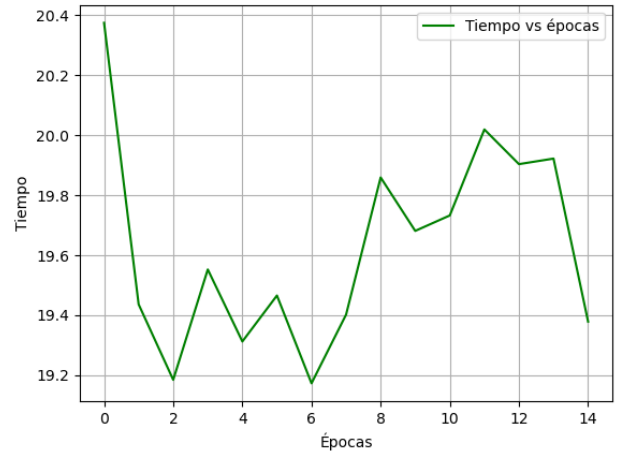


Figure 5. Tiempo vs Épocas

Como se puede observar en la anterior figura, el mejor tiempo se obtuvo en la época 6. En esta época se obtuvo un tiempo de recorrido igual a 14.17 segundos. Y los parámetros de esta época son los siguientes:

$$Kp_{obs}, Kp_{goal} = 0.267, 0.291 \quad (6)$$

Ahora se muestran algunos ejemplos en los que se evidencia la navegación del dron.

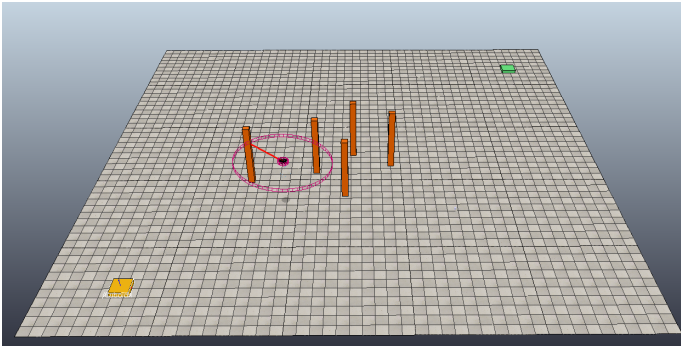


Figure 6. Ejemplo de navegación 1

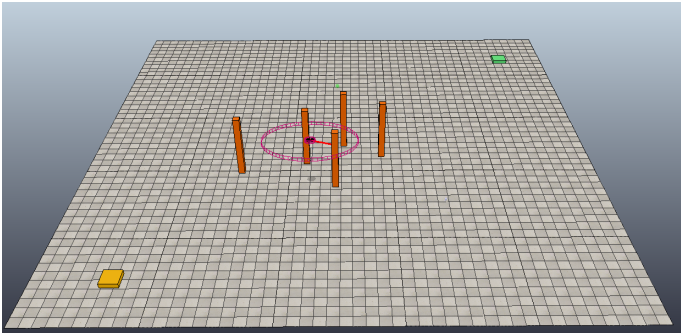


Figure 7. Ejemplo de navegación 2

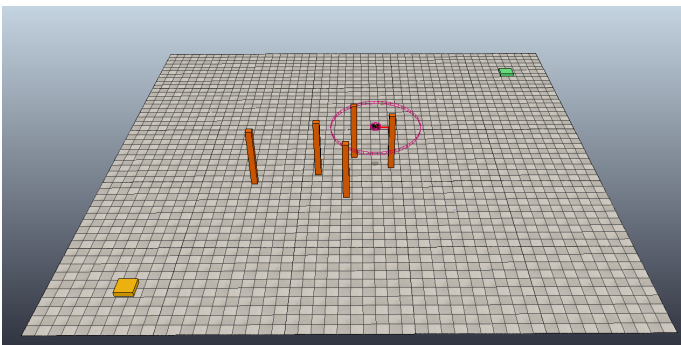


Figure 8. Ejemplo de navegación 3

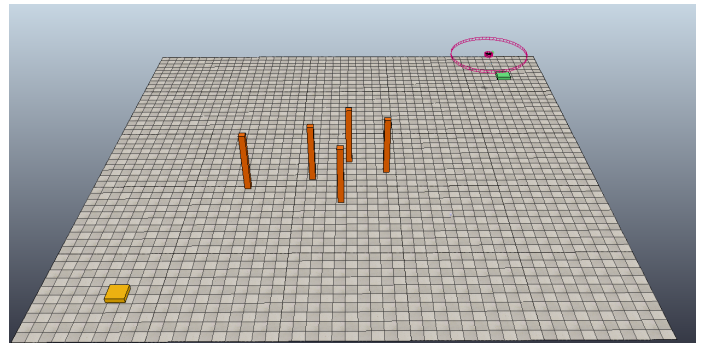


Figure 9. Ejemplo de navegación 4

III. CONCLUSIONES

- La determinación de los parámetros es muy importante para la correcta implementación del algoritmo de optimización PSO
- Se determina que a través del paso de las épocas, el error se reduce dado el comportamiento del algoritmo del PSO.
- Se evidencia que el algoritmo de navegación por campos potenciales funciona de manera adecuada, ya que permite al dron desplazarse hasta el punto de meta evitando los obstáculos.

REFERENCES

- [1] Nizar Rokbani¹, Aijth Abraham², Ade1.M Alimil., "Fuzzy Ant Supervised by PSO and Simplified Ant Supervised PSO Applied to TSP " 2014.
- [2] Shayeghi, H., Jalili, A., Shayanfar, H. A. (2008). Multi-stage fuzzy load frequency control using PSO. Energy conversion and management, 49(10), 2570-2580.