

SIMULACIÓN DE ROBOT MÓVIL CON ALGORITMO GENÉTICO

M. en T. C. José Eduardo Cardoza Plata
cardozakhan@hotmail.com
Estudiante de maestría Alejandro Benítez
Hernández
alienxss.hernandez@gmail.com
Dr. Mauricio Olguín Carbajal
molguinc@ipn.mx
Dr. Juan Carlos Herrera Lozada
jcrls.ipn@gmail.com
Dr. Jacobo Sandoval Gutiérrez
jacobosandoval@hotmail.com

Instituto Politécnico Nacional
Centro de Innovación y Desarrollo Tecnológico en
Cómputo

Boletín No. 87
1o. de noviembre de 2021

Resumen

Los robots móviles son máquinas capaces de desplazarse de un lugar a otro, ya sea volando, rodando, saltando, combinando movimientos, etc. En la búsqueda de una locomoción autónoma se han desarrollado diferentes algoritmos, como el algoritmo genético que evalúa diversos conjuntos de movimientos simultáneamente en busca del óptimo a través de las generaciones. Dicha búsqueda implica un desgaste físico significativo para cualquier robot, por lo que realizar una simulación es de gran utilidad para la reducción del desgaste y el tiempo que toma evaluar cada conjunto de movimientos, en la búsqueda de la locomoción autónoma óptima.

I. Introducción

A medida que la robótica avanza, la variedad tecnológica de vehículos no tripulados crece con el desarrollo de sistemas robóticos terrestres y aéreos, estos últimos suelen ser más veloces aunque requieren un mayor esfuerzo en su control, por su parte los vehículos terrestres pueden desplazarse mejor en espacios reducidos. La investigación de robots móviles conduce al desarrollo de algoritmos con la finalidad de que sus movimientos sean autónomos. El Algoritmo aquí presentado, explora una amplia gama de conjuntos de movimientos simultáneamente buscando el óptimo. Dicha exploración alude a un desgaste significativo del robot físico, es por ello, que se hace uso de un simulador, en donde se puede ejecutar y optimizar el algoritmo las veces que sean necesarias.

II. Desarrollo

En este trabajo se utilizó Gazebo un simulador de robótica en 3D de código abierto que utiliza un motor físico de alto rendimiento para simular un modelo basado en el robot tuk tuk, véase en la figura 1, una plataforma robótica de bajo costo fabricada por Parrot. En cada simulación el robot se mueve sobre un escenario con obstáculos.

El programa desarrollado tiene como finalidad dotar al robot de un conjunto de movimientos aleatorios, realizar la simulación y evaluar la posición final del robot con respecto a una meta establecida sobre el escenario, a dicha evaluación la llamaremos aptitud la cual será mayor mientras más cerca de la meta se encuentre el robot. En conjunto con nuestro programa se utilizó un micro-algoritmo genético, como el que se muestra en la figura 2, el cual explora varios conjuntos de movimientos simultáneamente buscando el óptimo.



Figura 1. Robot tuk tuk de parrot.

Al principio del algoritmo se genera una población aleatoria de 5 conjuntos de movimientos a los cuales llamaremos individuos, dicha población es evaluada con base en su aptitud, entre más cerca de la meta mayor será la aptitud, posteriormente se realiza un proceso de cruce, donde los individuos más aptos se reproducirán más veces, también aportarán los alelos dominantes en la recombinación genética a sus progenitores, conformando tanto padres como hijos una nueva población, la cual es reevaluada, seleccionando a los dos mejores individuos (selección natural). Estos individuos junto con otros 3 seleccionados aleatoriamente (deriva genética), serán los sobrevivientes a la autorregulación donde la población regresa a su tamaño original. El proceso se repite cíclicamente hasta alcanzar la convergencia nominal, en donde, mediante un proceso de elitismo se seleccionan a los dos mejores individuos de la última población, los cuales junto con 3 nuevos individuos generados aleatoriamente integran la nueva población que entrará al principio del algoritmo conformando así un segundo bucle con una condición de parada establecida en la convergencia general, al término de la cual tendremos al mejor individuo posible, resultado de la herencia genética a través de varias generaciones, y por lo tanto, la mejor solución posible. En la última etapa donde se introducen nuevos individuos totalmente aleatorios, se le conoce como ruido estocástico, el cual ayuda a mantener la diversidad en la población, escapando así de una convergencia prematura y evitando quedar atrapados en óptimos locales, mientras que el elitismo nos permite escalar sobre las mejores soluciones en búsqueda un óptimo

global. Se debe mantener un balance entre la diversidad proporcionada, es decir, la exploración y la explotación del micro-algoritmo. La explotación es el proceso de usar la información obtenida de las soluciones exploradas previamente para determinar hacia cuales es más conveniente continuar avanzando, mientras que, la exploración es el proceso de visitar nuevas regiones del espacio de búsqueda (nuevas soluciones) para ver si se puede encontrar algo prometedor.

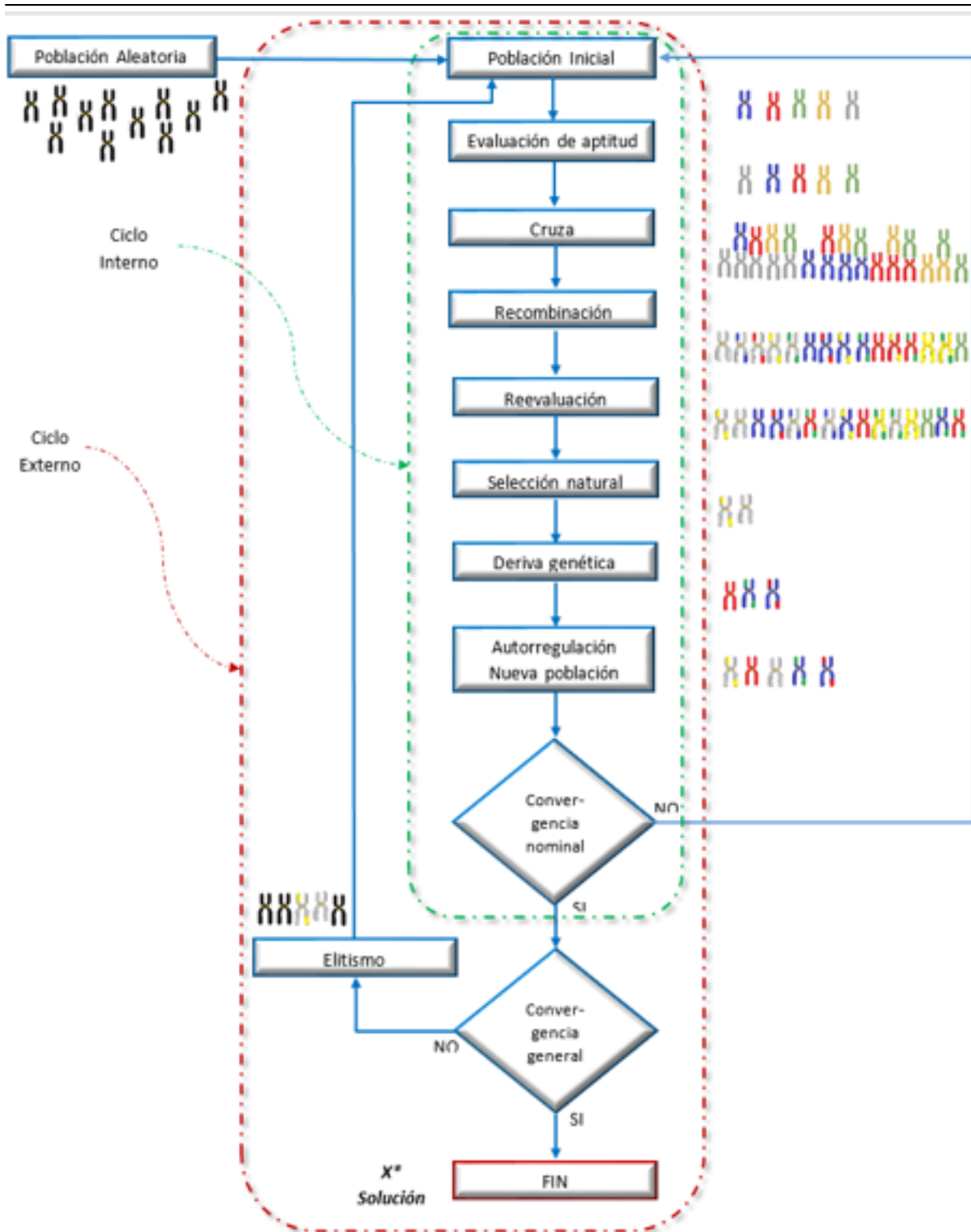


Figura 2. Micro-algoritmo Genético.

La integración del micro-algoritmo genético con el simulador, véase la figura 4, se realizó a través de un programa en lenguaje C ejecutado en Ubuntu, mediante el cual se leen y escriben archivos world.

Un archivo world contiene todos los elementos necesarios para ejecutar una simulación en Gazebo, para cada conjunto de movimientos se crea un archivo world el cual es simulado desde la terminal con el comando gzserver, el cual nos permite realizar una simulación sin emplear la interfaz gráfica. Dicho comando es ejecutado desde el programa en C mediante el uso de la función Fork.

La función fork crea un proceso en este caso gzserver, permitiéndonos determinar la duración del mismo, cada simulación tiene un tiempo en función al número de movimientos del robot. Una vez finalizado el proceso de simulación, Gazebo guardara un registro de esta por lo que, desde el programa en C ejecutamos el comando gzlog, el cual nos permite interpretar el registro mencionado, así como filtrar la información del mismo, recordando que es de nuestro interés la posición final del robot con respecto a la meta establecida para evaluar la aptitud del conjunto de movimientos simulado.

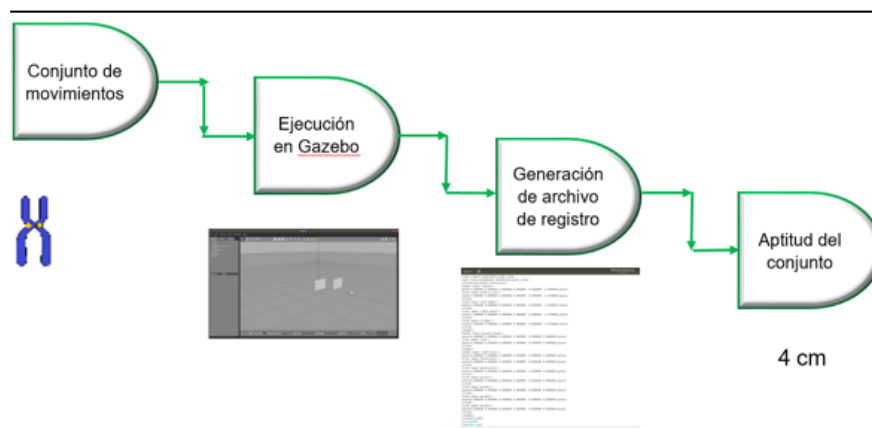


Figura 3. . Integración del micro-algoritmo genético con el simulador.

III. Pruebas y resultados

Para la fase de experimentación, se utilizó el micro-algoritmo genético en representación binaria, con poblaciones de 5 individuos. Cada individuo es una cadena binaria de 144 bits.

El robot puede realizar giros de 360° a la derecha o izquierda, utilizando 14 bits para cada giro, también puede avanzar 50cm hacia adelante o atrás, utilizando 10 bits.

Por lo que cada individuo podrá realizar 6 giros y 6 avances, intercalados entre sí, es decir después de un giro se realiza un avance y después de un avance se realiza un giro. Se implementó una mutación a cada bit de $(1/\text{longitud de la cadena}) * 10$.

Se implementó un proceso de cruce dinámica la cual varía a lo largo de las generaciones entre 3 y 23 bits por fragmento de cadena, cada fragmento es de 24 bits representando un giro y un avance, cruzando dos de ellos aleatoriamente en cada proceso.

EL robot inicio en el punto (0,0) en un laberinto cerrado con 2 obstáculos con meta 2 metros adelante con centro en el punto (2,0) y un área de 20 cm. Podemos observar los resultados después de 300 generaciones en la figura 4, donde la mejor aptitud es de 7cm de distancia con el centro de la meta encontrando al robot dentro del área deseada.

La búsqueda de la mejor solución o una suficientemente buena de una amplia gama de soluciones potenciales, resulta interesante para los algoritmos bioinspirados, siendo el micro-algoritmo genético presentado una técnica heurística eficiente ante este tipo de problemas de locomoción de un robot móvil, siendo de gran utilidad el uso de un simulador el cual nos permite evaluar las múltiples soluciones sin desgastar físicamente el robot, además de realizarlo en un menor tiempo.

IV. Conclusiones

La búsqueda de la mejor solución o una suficientemente buena de una amplia gama de soluciones potenciales, resulta interesante para los algoritmos bioinspirados, siendo el micro-algoritmo genético presentado una técnica heurística eficiente ante este tipo de problemas de locomoción de un robot móvil, siendo de gran utilidad el uso de un simulador el cual nos permite evaluar las múltiples soluciones sin desgastar físicamente el robot, además de realizarlo en un menor tiempo.

V. Referencias

1. J.C. Herrera (2011). *"Sistema Inmune Artificial con Población Reducida para Optimización Numérica"*, Centro de Investigación en Computación, Instituto Politécnico Nacional, Distrito Federal, México.
2. A. Leon (2009). *"Diseño e implementación en hardware de un algoritmo bio-inspirado"* Centro de Investigación en Computación, Instituto Politécnico Nacional, Distrito Federal, México.
3. N. Cruz. (2004). *"Sistema inmune artificial para solucionar problemas de optimización"* Tesis Doctoral. CINVESTAV- IPN. México.