

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Integración de algoritmo optimizado con Reinforcement y
Deep Learning con Simuladores físicos**

Protocolo de trabajo de graduación presentado por Marco Antonio
Izeppi Rosales, estudiante de Ingeniería Mecatrónica

Guatemala,

2021

Resumen

Antecedentes

El Megaproyecto Robotat es un proyecto que tenia como base el diseño de robots con aplicaciones de seguimiento de trayectorias y comportamientos de enjambre. A partir de esta idea realizaron el diseño electrónico y mecánico de los "Bitbots". Inicialmente se enfocaron en el diseño de hardware de los "Bitbots" y en un algoritmo de visión por computadora para obtener la posición y orientación de los robots en un plano bidimensional. Mas adelante, se empezaron a refinar detalles de software tales como el protocolo de comunicación y el algoritmo computacional que sera utilizado por los "Bitbots". Para el desarrollo del algoritmo de control para determinar el comportamiento de los robots se fueron desarrollando diversas tesis mostradas a continuación:

0.1. Algoritmo Modificado de Optimización de Enjambre de Partículas

En esta tesis de Aldo Aguilar [1] se busca la implementación de Algoritmo Modificado de Optimización de Enjambre de Partículas con robots diferenciales reales. En el desarrollo de esta tesis se buscaba definir un algoritmo para establecer el comportamiento de los robots para que se comporten como un enjambre con la habilidad de búsqueda de metas. Uno de los principales problemas que se presentaron era el acople directo del movimiento de las partículas del PSO debido a que el movimiento era irregular generando la posibilidad de que se saturen los actuadores del robot.

Para solucionar este problema se planteo que cada robot no iría directamente a la posición dada por el PSO, sino que se utilizaría la posición dada por el PSO como una sugerencia de el punto al cual debe desplazarse el robot. Durante este proceso de resolución se realizaron pruebas con 8 controladores: Transformación de unicycle (TUC), Transformación de unicycle con PID (TUCPID), Controlador simple de pose (SPC), Controlador de pose Lyapunov-estable (LSPC), Controlador de direccionamiento de lazo cerrado (CLSC), Transformación de unicycle con LQR (TUC-LQR), y Transformación de unicycle con LQI (TUC-LQI). Al finalizar las pruebas con estos 8 controladores se determino que los mejores resultados en cuanto a la generación de trayectorias y velocidades continuas reguladas para los robots diferenciales era el controlador TUC-LQI y el TUC-LQR.

0.2. Algoritmo de optimización de enjambre y Artificial Potential Fields

En esta tesis de Juan Cahueque [2] se realiza una implementación de la teoría de enjambre en el área de búsqueda y rescate de personas. Sabiendo que el movimiento de partículas bidimensional se realiza en una superficie tridimensional, siendo la tercera componente, el resultado de la función de costo la cual nos indica cual es el mínimo global y por lo tanto,

hacia que punto convergerán las partículas.

Teniendo esto bien definido, se definen funciones artificiales de potencia de choset y kim, Wang y shin con comportamiento multiplicativo y aditivo. Esto lo que nos dice es que prácticamente es posible modelar una función de costo "personalizada" para establecer a que punto convergerán las partículas. Se realizaron tres modelos, cada uno contaba con una meta y diversos obstáculos, los cuales tenia que evitar el robot para llegar a la meta. Para lograr que estos modelos fueran implementables en robots diferenciales físicos.

Se emplearon los controladores propuestos por Aldo Aguilar [1] con algunas modificaciones para hacerlos compatibles con esta nueva metodología. Realizando diversas simulaciones se llego a determinar los mejores parámetros para los controladores en los robots diferenciales físicos. Cabe destacar que se tuvieron que realizar diversas pruebas con webots ya que los parámetros establecidos en MATLAB eran idealizados y al ser implementados en Webots no brindaban el comportamiento esperado por los mismos.

0.3. Algoritmo PSO modificado con aprendizaje reforzado y aprendizaje profundo

En esta tesis Eduardo Santizo [3] Empleo la metodología de aprendizaje reforzado y aprendizaje profundo para desarrollar un PSO tuner y un generador de trayectorias. Para el desarrollo del PSO tuner, se empleo una red neuronal recurrente que toma diversas métricas propias de las partículas PSO y realiza una predicción de los valores óptimos para ω , ϕ_1 y ϕ_2 . Entrenando las redes neuronales con 7,700 simulaciones de un algoritmo estándar de PSO se generan predicciones de carácter dinámico, Causando que el algoritmo final redujera el tiempo de convergencia y susceptibilidad a mínimos locales del PSO original.

Para el generador de trayectorias se utilizo aprendizaje profundo. En las pruebas iniciales se utilizo una cuadrícula simulando el gridworld de Webots en el cual el robot solo tenia la posibilidad de moverse en 4 direcciones: Arriba, Abajo, Izquierda o Derecha. A el robot se le daban penalizaciones si chocaba contra un obstáculo y se le premiaba si se movía esquivando el obstáculo. Si el robot llegaba a la meta rápidamente recibía un premio aun mayor, usando este sistema de premios y penalizaciones, se entreno al algoritmo para predecir de mejor manera la trayectoria. Luego se modifiko el algoritmo para permitir el movimiento en diagonales a 45° , Por lo que se tienen 8 posibles movimientos y se siguió la misma metodología anteriormente descrita. Obteniendo finalmente el modelo propuesto por Eduardo Santizo [3].

0.4. Optimización de trayectorias de robots móviles en ambientes desconocidos

En este artículo Safa Ziadi, Mohamed Njah y Mohamed Chtourou [4] Utilizaron el enfoque de optimización de enjambre de partículas multiobjetivo (PSO) para optimizar los parámetros del Método Canónico de Campo de Fuerza (CF2) los cuales son P , c, k, Q y ρ_0 .

El cálculo de los parámetros óptimos se reinicia en cada nueva posición del robot y el PSO se utiliza para minimizar la distancia entre la posición y la meta a la cual se desea llegar y también para maximizar la distancia segura entre la posición actual del robot y los obstáculos cercanos. La eficacia del método se demuestra mediante simulaciones en el entorno de Webots. Las simulaciones se llevan a cabo en varios entornos conocidos y desconocidos. En los entornos conocidos, el robot reconoce la posición del obstáculo al comienzo de la navegación y la planificación de la ruta es global. Pero en los entornos desconocidos, la localización del robot se basa en las lecturas de los sensores y la planificación de la ruta es individual para cada robot, dependiendo de las lecturas de los sensores.

Justificación

Objetivos

Objetivo General

Integración de Algoritmo de inteligencia de enjambre Optimizado con Reinforcement Learning y Deep Learning con Plataforma Webots y primeros pasos con ROS.

Objetivos Específicos

- Simulación de comportamiento de robots utilizando el algoritmo de enjambre optimizado y la plataforma webots.
- Modificar el algoritmo para que sea implementable en una simulación en una plataforma virtual.
- Verificar la eficacia del algoritmo en una simulación con condiciones físicas.
- Brindar un primer vistazo de las simulaciones utilizando el Programa Robot Operating System (ROS).

Marco teórico

Particle Swarm Optimization (PSO)

La Optimización por Enjambres de Partículas (conocida como PSO, por sus siglas en inglés, Particle Swarm Optimization) es una técnica de optimización/búsqueda. Este método fue descrito alrededor de 1995 por James Kennedy y Russell C. Eberhart (Kennedy, J. & Eberhart, R. (1995), 'Particle swarm optimization', Neural Networks, 1995. Proceedings., IEEE International Conference), y se inspira en el comportamiento de los enjambres de insectos en la naturaleza. Formalmente hablando, se supone que tenemos una función desconocida, $f(x, y)$, que podemos evaluar en los puntos que queramos, pero a modo de caja negra, por lo que no podemos conocer su expresión. El PSO es fácil de implementar y ha sido ampliamente usado en variadas aplicaciones con resultados excelentes resolviendo problemas reales de optimización. Este algoritmo puede ser computacionalmente ineficiente por

que puede quedar atrapado fácilmente en óptimos locales cuando resuelve problemas cuyo espacio de solución es multimodal; estas debilidades han hecho que el campo de aplicación de la metodología esté un poco restringido [5].

El objetivo es el habitual en optimización, encontrar valores de x e y para los que la función $f(x, y)$ sea máxima (o mínima, o bien verifica alguna relación extremal respecto a alguna otra función). La idea que vamos a seguir en PSO comienza de forma similar, situando partículas al azar en el espacio de búsqueda, pero dándoles la posibilidad de que se muevan a través de él de acuerdo con unas reglas que tienen en cuenta el conocimiento personal de cada partícula y el conocimiento global del enjambre [5].

Cada partícula (individuo) tiene una posición, p (que en 2 dimensiones vendrá determinado por un vector de la forma $[x, y]$), en el espacio de búsqueda y una velocidad, v (que en 2 dimensiones vendrá determinado por un vector de la forma $[v_x, v_y]$), que determina su movimiento a través del espacio. Además, como partículas de un mundo real físico, tienen una cantidad de inercia, que los mantiene en la misma dirección en la que se movían, así como una aceleración (cambio de velocidad), que depende principalmente de dos características:

- Cada partícula es atraída hacia la mejor localización que ella, personalmente, ha encontrado en su historia (mejor personal).
- Cada partícula es atraída hacia la mejor localización que ha sido encontrada por el conjunto de partículas en el espacio de búsqueda (mejor global).

Formalmente, lo podemos escribir como:

$$v_i(t+1) = v_i(t) + C_1 r_1 (p_i^m e j o r - p_i(t)) + c_2 r_2 (p_g^m e j o r - p_i(t)) \quad (1)$$

En donde:

- c_1, c_2 = son las constantes de atracción al mejor personal y el mejor global
- r_1, r_2 = son números aleatorios entre 0 y 1
- $p_i^m e j o r$ = es la mejor posición por la cual ha pasado una partícula
- $p_g^m e j o r$ = es la mejor posición global de todo el sistema de enjambre

Una vez actualizadas las velocidades de todas las partículas, sus posiciones se actualizan siguiendo una ley simple:

$$p_i(t+1) = v_i(t) + p_i(t) \quad (2)$$

un conjunto de partículas representa potenciales soluciones, donde cada partícula i está asociada a dos vectores, el vector de velocidades y a la posición del vector. La velocidad y la posición de cada partícula son inicializadas por vectores aleatorios con sus correspondientes rangos [6].

0.5. Webots

Webots es una Multi-plataforma Open-Source (Código abierto) utilizada para realizar simulaciones de robots en entornos mas cercanos a la realidad. Provee un entorno completo de desarrollo para modelar, programar y simular robots. Esta plataforma fue desarrollada por el Dr. Oliver Michel del Instituto Federal Suizo de Tecnología para fines educativos. Utiliza el Open Dynamics Engine que Permite la simulación de colisiones y la dinámica de cuerpos rígidos creando así un entorno mas real. El programa permite utilizar controladores escritos en C, C++, Java, Python, MATLAB y ROS [7].

0.5.1. Robot Diferencial E-Puck

El E-Puck es un robot diferencial móvil desarrollado por la Escuela Politécnica Federal de Lausana (EPFL). Este robot esta incluido en la librería de Webots para realizar simulaciones. Este robot cuenta con una gran variedad de sensores que lo hacen una buena opción para su implementación con el algoritmo PSO. Es un robot pequeño y su programación es amigable con el usuario, por lo cual ha sido utilizado como herramienta de aprendizaje [8].

0.6. Robot Operating System (ROS)

0.7. Matlab R2019

Metodología

Cronograma de actividades

Índice preliminar

Referencias

- [1] A. Aguilar, «Algoritmo Modificado de Optimización de Enjambre de Partículas (MPSO),» Tesis de mtría., Universidad del Valle de Guatemala, Guatemala, 2019.
- [2] J. Cahueque, «Implementación de enjambre de robots en operaciones de búsqueda y rescate,» Tesis de mtría., Universidad del Valle de Guatemala, Guatemala, 2019.
- [3] E. Santizo, «Aprendizaje Reforzado y Aprendizaje Profundo en Aplicaciones de Robótica de Enjambre,» Tesis de mtría., Universidad del Valle de Guatemala, Guatemala, 2021.
- [4] S. Ziadi, M. Njah y M. Chtourou, «PSO optimization of mobile robot trajectories in unknown environments,» *International Multi-Conference on Systems, Signals & Devices*, vol. 13, págs. 774-782, 2016.
- [5] D. Álvarez, E. Toro y G. Ramón, «Algoritmo de Optimización cumulo de partículas aplicado en la solución de problemas de empaquetamiento óptimo bidimensional con y sin rotación,» *Scientia et Technica*, págs. 10-16, 2009.

- [6] Z. Chi, G. Hai-bing, G. Liang y Z. Wan-guo, «Particle Swarm Optimization(PSO) Algorithm,» Tesis de mtría., Huazhong University of Science & Technology, China, 2010.
- [7] Cyberbotics Ltd, *Webots*, <https://cyberbotics.com>, 2021.
- [8] C. 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,» *International Workshop on Swarm Robotics*, págs. 103-115, 2006.