ISSN 1870-4069

Selección de acciones para la navegación de un robot móvil basada en fuzzy Q-learning

Elizabeth López Lozada, Elsa Rubio Espino, Juan Humberto Sossa Azuela, Víctor Hugo Ponce Ponce

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

elizabeth.l.lozada@gmail.com,
{erubio, hsossa}@cic.ipn.mx

Resumen. Los robots móviles deben contar con la capacidad de moverse dentro de un entorno de trabajo bajo la menor intervención de un supervisor humano, para ello se les provee de algoritmos y herramientas que les permiten tomar decisiones de acuerdo a su estado actual y a las variables de su entorno. En este artículo, se describe una arquitectura para la selección de acciones con el método de Fuzzy Q-Learning (FQL) con el fin de que un robot móvil pueda tomar decisiones con base en el nivel de carga de batería y pueda escoger entre desplazarse hacia su objetivo principal, ir a una estación de carga de baterías o detenerse. Adicionalmente, como entrada se toman en consideración las distancias que existen entre el robot y los posibles destinos. Para lograrlo, en la arquitectura propuesta se integró un módulo de planificación de ruta que utiliza el método de campos potenciales artificiales, para que un robot pueda navegar de forma reactiva sobre su entorno de trabajo.

Palabras clave: Robots Móviles Navegación Planificación de Ruta Fuzzy Q-Learning Campos Potenciales Artificiales.

Selection of Actions for the Navigation of a Mobile Robot Based on Fuzzy Q-Learning

Abstract. Mobile robots must have the ability to move within a work environment under the least intervention of a human supervisor, for this they are provided with algorithms and tools that allow them to make decisions according to their current state and the variables of its surroundings. In this article, an architecture for the selection of actions with the Fuzzy Q-Learning (FQL) method is described so that a mobile robot can make decisions based on the battery charge level and can choose between moving towards your main objective, going to a battery charging station or stopping. Additionally, the distances between the robot and the possible destinations are taken into consideration as input. To achieve this, a route planning module was integrated into the proposed architecture that uses the artificial

potential fields method, so that a robot can reactively navigate its working environment.

Keywords: Mobile robots, Navigation, Route planning, Fuzzy Q-learning, Artificial potential fields.

1. Introducción

Los robots móviles se usan en una gran cantidad de aplicaciones, por ejemplo, en vigilancia, exploración, operaciones de búsqueda y rescate, limpieza, automatización industrial, construcción o en museos. Para cumplir con sus tareas, deben contar con algoritmos robustos que les permitan navegar a través de su entorno y evadir los obstáculos que se encuentren en el camino. La navegación para los robots móviles incluye todas las acciones que llevan a que un robot se desplace desde su posición actual hasta el destino y se puede dividir en dos tipos: global y local [1,3].

Una de las tareas esenciales de la navegación en robots móviles es la planificación de rutas, cuyo objetivo es generar las entradas de referencia que determinen una ruta libre de colisiones, para que un robot ejecute la trayectoria planeada [3]. Para determinar el desplazamiento del robot y que pueda evadir obstáculos se han usado sistemas de inferencia difusos [12,13], aprendizaje reforzado [4, 6, 16], campos potenciales artificiales [7, 8, 9, 10, 14] y redes neuronales [15]. Cada método tiene sus ventajas y desventajas, por ejemplo, el método de campos potenciales artificiales se usa para la planificación de rutas, ya que es simple y tiene un costo computacional bajo [10].

En todos estos casos, los autores sólo consideran la información proveniente de los sensores de distancia, sin embargo, no consideran la descarga eléctrica de la batería que ocurre durante la operación del robot. En [11], determinan que el robot hace una planificación de la trayectoria donde hay un punto de partida, cinco nodos que el robot tiene que visitar, y el último nodo corresponde a una estación de servicio donde el robot puede reponer su batería.Estos robots deben contar con dos características importantes: autonomía y autosuficiencia. Con esto, surge la pregunta ¿puede un robot aprender a seleccionar acciones de forma autónoma? La respuesta es sí, de manera que este trabajo busca mostrar una forma en que un robot podría aprender a tomar decisiones en base a prueba y error. Es por ello, que en este documento se presenta una arquitectura basada en el método de Fuzzy Q-Learning (FQL) junto con un módulo para la planificación de rutas, a fin de que un robot tenga la capacidad para tomar decisiones de forma independiente en base al nivel de batería y seleccione entre ir a la estación de carga de baterías, ir a un destino predeterminado o esperar.

Este artículo está organizado como sigue: en la sección 2, se presentan los conceptos y las ecuaciones básicas del método de Fuzzy Q-Learning; en la sección 3, se presenta el método de campos potenciales que se utiliza para la planificación de la trayectoria del robot hacia el destino seleccionado; en la sección 4, se describe la arquitectura propuesta en este trabajo; en la sección 5 se presentan las simulaciones y los resultados obtenidos; y finalmente en la sección 6 se dan las conclusiones obtenidas durante el desarrollo de este trabajo.

2. Fuzzy Q-Learning

El método de Fuzzy Q-Learning (FQL) [2] puede verse como una extensión de los sistemas de inferencia difusos (FIS), en donde las reglas difusas definen el estado del agente de aprendizaje. Cuando las entradas de las reglas son convertidas en valores difusos, caen en una regla difusa que corresponde a un estado; la fuerza de la regla difusa α_i, ayuda a definir el grado de que el agente se encuentre en un estado. Con esto el sistema de aprendizaje escoge una acción a_i del conjunto de acciones A en cada regla, se llama $a[i,J]$ a la $j-$ésima acción posible en la regla i y $q[i,J]$ a su correspondiente valor q. Con esto, el FIS se construye de la siguiente forma:

Si x es S_i **entonces** $a[i,1]$con $q[i,1]$**o** $\dots$**o** $a[i,J]$con $q[i,J]$.

Al final, el agente de aprendizaje debe encontrar la mejor solución para cada regla, es decir, la acción con el mejor valor q.

El valor q se obtiene de una tabla que contiene $i \times j$ valores q, las dimensiones de la tabla corresponden al número de reglas por cantidad de acciones. Las acciones se seleccionan con una política de exploración-explotación, que está basada en la calidad del par estado-acción. Se propone que la probabilidad de exploración-explotación esté dada por $\varepsilon = 10/(10+T)$,en donde T corresponde al número del paso. Esta probabilidad se usa para compensar entre la exploración y el control del algoritmo, y eliminar la exploración de forma gradual. Para calcular la acción inferida y el valor q, se ocupan las ecuaciones en (1):

$$a(x) = \frac{\sum_{i=1}^{N} \alpha_i \times a(i,i^o)}{\sum_{i=1}^{N} \alpha_i(x)},\ Q(x,a) = \frac{\sum_{i=1}^{N} \alpha_i \times q(i,i^o)}{\sum_{i=1}^{N} \alpha_i(x)}, \tag{1}$$

en donde $Q(x,a)$ es una función evaluada en x y a; x es el estado o regla y a es la acción inferida; i es el estado o regla en el que se encuentra el agente; $i°$ es el índice de la acción seleccionada; α_i es la fuerza de la regla; y N es un número positivo, $N \in \mathbb{N}^+$ y corresponde al número total de reglas. La función de valor-estado se calcula con la ecuación (2):

$$V(x,a) = \frac{\sum_{i=1}^{N} \alpha_i(x) \times q(i,i^*)}{\sum_{i=1}^{N} \alpha_i(x)}, \tag{2}$$

en donde i^* corresponde al índice de la acción óptima, es decir, la acción que tiene el valor q más alto y x es un estado. Y la derivada de la función Q se define como en la ecuación (3), en donde r corresponde a la recompensa y γ es el factor de descuento:

$$\Delta Q = r + \gamma V(x,a) - Q(x,a). \tag{3}$$

Con la ecuación (4) se obtiene un valor de elegibilidad e[i, j], que se usa durante la actualización del valor q en donde γ es un parámetro, $0 \leq \gamma \leq 1$, llamado factor de descuento, λ es el parámetro de decaimiento, en donde $\lambda \in [0,1]$, i es el número de la regla y j es la acción seleccionada:

$$e[i,j] = \begin{cases} \lambda\gamma e[i,j] + \dfrac{\alpha_i(x)}{\sum_{i=1}^{N}\alpha_i(x)}, & \text{si } j = i^{\circ}, \\ \lambda\gamma e[i,j], & \text{en otro caso.} \end{cases} \quad (4)$$

Finalmente, para actualizar el valor q se utiliza el valor de elegibilidad e[i, j], la derivada de la función Q y una pequeña fracción positiva $\epsilon \in (0, 1]$, la cual influye en la tasa de aprendizaje. En la ecuación (5) se muestra la función de actualización del valor q:

$$\Delta q[i,j] = \epsilon \times \Delta Q \times e[i,j]. \quad (5)$$

3. Campos potenciales artificiales

El uso de los campos potenciales en la navegación de robots fue propuesto por Khatib [5] en 1986, en su trabajo presentó el concepto de campos potenciales artificiales para que robots móviles evadan obstáculos en tiempo real. Su teoría se fundamenta en el uso de fuerzas de atracción para alcanzar el objetivo y en fuerzas de repulsión para evitar colisionar con los obstáculos, y que pueden ser calculadas a través de un conjunto de ecuaciones sencillas. La fuerza de atracción es definida en la ecuación (6) como una función relativa de distancia entre el robot y el objetivo, en donde básicamente el robot es atraído hacia el objetivo por el campo de atracción que este tiene [7]. La fuerza de repulsión, en la ecuación (7), es la que indica al robot la presencia de obstáculos en un umbral definido y lo empuja lejos de los obstáculos. Finalmente, se calcula una fuerza resultante que equivale a la suma de las fuerzas de atracción y repulsión:

$$F_{attr}(q,g) = -\xi p(q,g), \quad (6)$$

en donde ξ es un factor de escala positivo; $p\ (q,g)$ es la distancia euclidiana entre el robot y el objetivo; q es la posición del robot; y g es la posición del objetivo:

$$F_{rep}(q) = \begin{cases} \eta\left(\dfrac{1}{p\left(q_i,p_{o_j}\right)} - \dfrac{1}{p_{o_j}}\right)\dfrac{p^2\left(q_i,p_{o_j}\right)}{p\left(q_i,p_{o_j}\right)}, & \text{si } d < p_{o_j}, \\ -\dfrac{\eta}{p_{o_j}}, & \text{si } d = p_{o_j}, \\ 0, & \text{si otro caso,} \end{cases} \quad (7)$$

en donde η es el factor de repulsión; $p\left(q_i,p_{o_j}\right)$ es la distancia entre el robot y el obstáculo; p_{o_j} es el umbral entre el robot y el radio del obstáculo; y d es la distancia medida.

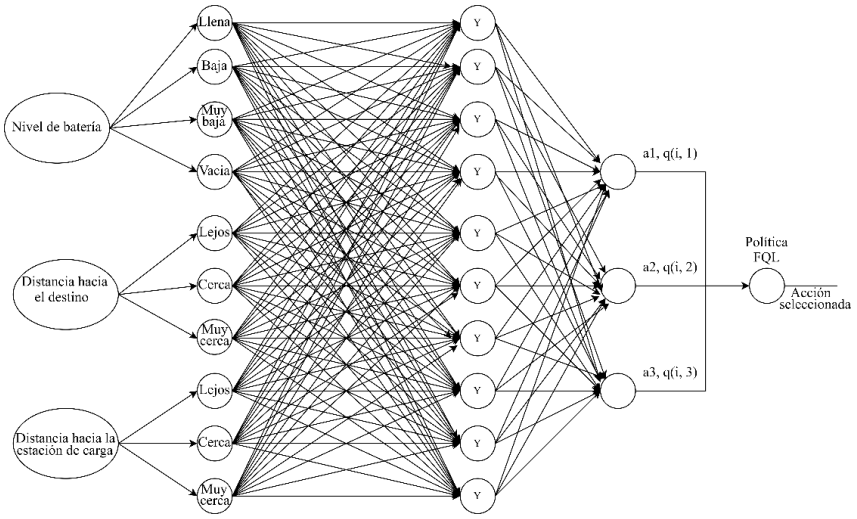


Fig. 1. Arquitectura del sistema de Fuzzy Q-Learning.

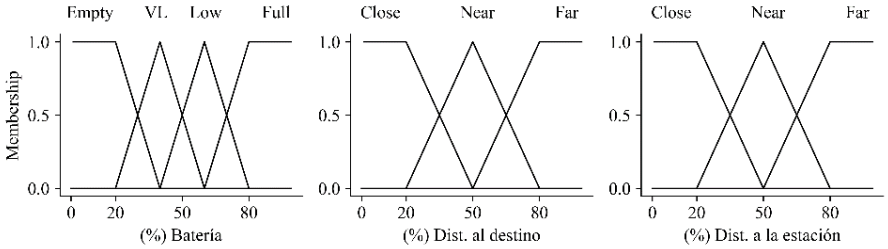


Fig. 2. Funciones de membresía.

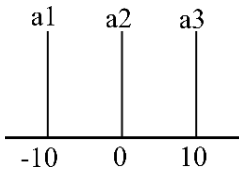


Fig. 3. Funciones singleton de las salidas.

4. Arquitectura propuesta

La arquitectura propuesta se representa en el diagrama de la Figura 1. Las entradas están definidas como el nivel de batería, la distancia hacia el objetivo y la distancia hacia la estación de carga de la batería, las cuales están conectadas a las funciones de membresía correspondientes.

Para la primera entrada, están definidos cuatro conjuntos difusos que corresponden al nivel de la batería llena, baja, muy baja y vacía; la segunda y tercera entrada se conectan a las funciones de membresía lejos, cerca y muy cerca. En total se tienen 36 reglas difusas y hay tres posibles acciones que están asociadas a un valor numérico q, que pueden ser seleccionadas de acuerdo con la política de exploración-explotación descrita en la sección 2.

Algoritmo 1. Navegación en base al nivel de batería.

Entrada:
robot ← Posición del robot
objetivo ← Posición del objetivo
estación ← Posición de la estación de carga
obstáculos ← Posición de los obstáculos
γ ← Factor de descuento
α ← Tasa de aprendizaje
Inicialización del sistema.
destino ← objetivo.
mientras robot != destino **hacer**
estado ← Obtener el estado actual.
acción ← Seleccionar una acción
destino ← Actualizar el destino
salida ← Calcular la salida global usando la ecuación (1)
q ← Calcular el valor Q con la ecuación (2)
Ejecutar la acción
nuevo estado ← Obtener el nuevo estado.
recompensa ← Obtener la recompensa con la función (9)
valor estados ← Calcular con la ecuación (3)
ΔQ ← Calcular delta Q con la ecuación (6)
elegibilidad ← obtener con la ecuación (5)
Δq[i, j] ← ϵ × ΔQ× elegibilidad
Actualizar el valor Q

En la Figura 2 se muestran los conjuntos que forman las funciones de membresía para cada entrada. La figura izquierda corresponde a la función de membresía para el nivel de carga de batería, la figura central es para la entrada de la distancia hacia el destino y la figura derecha es para la distancia hacia la estación de carga.

Las posibles salidas están definidas por las funciones de tipo singleton, que permiten asociar un valor numérico a una salida puntual, en este caso a cada una de las acciones que puede ejecutar el sistema, en la Figura 3 se muestra el conjunto de funciones que se utilizan.

Mientras que la función de recompensas está dada por la expresión (8), con $\gamma = 0.5$ y $\alpha = 0.01$.

$$\mathrm{r}(t) = \begin{cases} +10, \text{ si el robot esta cerca del objetivo,} \\ +5, \text{ si el robot esta cerca de la estación,} \\ -20, \text{ si el robot esta alejado y el nivel de batería es muy bajo,} \\ 1, \text{ en otro caso.} \end{cases} \quad (8)$$

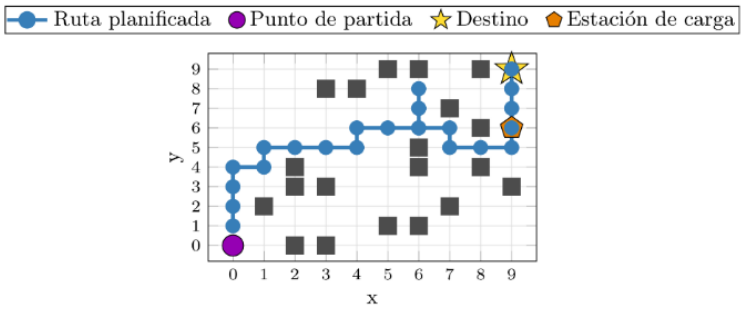


Fig. 7. Ruta que toma el agente para llegar al destino.

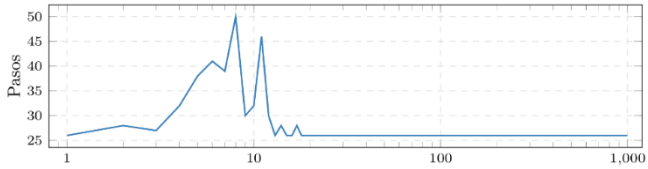
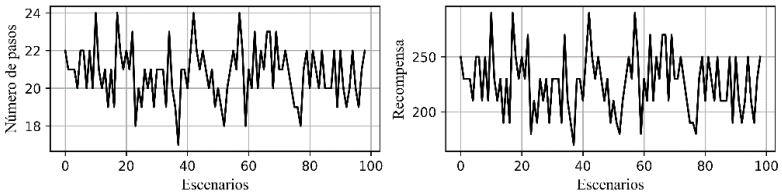


Fig. 8. Pasos para que el robot complete la tarea durante cada época.



(a) Número de pasos por escenario (b) Recompensa acumulada por escenario

Fig. 9. Comportamiento del robot en diferentes escenarios.

Finalmente, el Algoritmo 1 muestra el conjunto de pasos que se siguen para llegar al destino usando FQL y el módulo de campos potenciales.

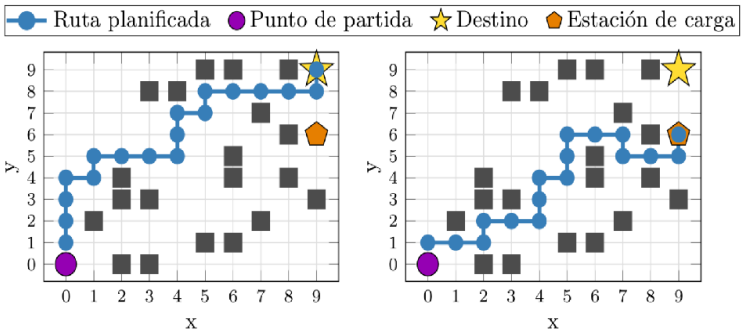
5. Simulaciones y resultados

A continuación, se colocan los resultados obtenidos de las simulaciones en un escenario que cuenta con 20 obstáculos, un objetivo y una estación de carga, los cuales están representados por cuadrados, una estrella y un pentágono respectivamente.

Para fines ilustrativos, las figuras que se muestran en esta sección corresponden a uno de los escenarios en donde el sistema fue simulado.

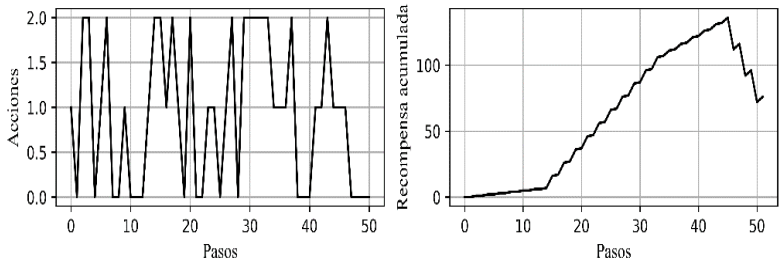
La generación de ruta es uno de los primeros pasos que se realiza, por ello en la Figura 4a se muestra la ruta generada para llegar a la posición objetivo y en la Figura 4b la ruta generada para llegar a la estación de carga.

En la Figura 5 se muestran las acciones seleccionadas y la recompensa acumulada durante el desplazamiento, en donde se observa una curva ascendente hasta que llega al



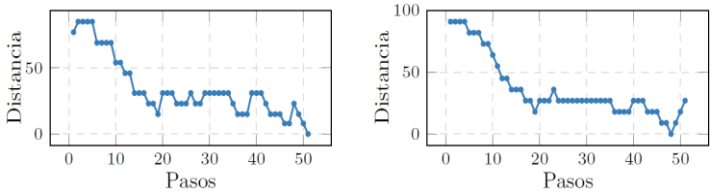
(a) Ruta hacia el destino (b) Ruta hacia la estación de carga

Fig. 4. Rutas generadas.



(a) Acciones seleccionadas (b) Recompensa acumulada

Fig. 5. Comportamiento del robot durante la simulación.



(a) Distancia hacia el destino (b) Distancia hacia la estación de carga

Fig. 6. Comportamiento de las distancias durante la simulación.

paso 45. A partir de este paso la recompensada asignada es negativa, lo cual se debe a que el nivel de batería es bajo porque el agente no ha logrado llegar a su destino.

El comportamiento de las distancias durante la simulación se muestra en la Figura 6 y se observa como disminuye la distancia entre el destino y el robot durante cada paso de la simulación.

A partir del paso 45, se observa que la distancia entre el robot y la estación de carga se acorta, mientras que la distancia entre el robot y el objetivo se incrementa, sin embargo, después de que el robot llega la estación de carga continua su camino hasta llegar al objetivo.

En la Figura 7 se muestra la ruta que tomó el robot hasta llegar al objetivo, la cual es diferente a la mostrada en la Figura 4a causa de las acciones seleccionadas por el módulo de FQL. Se aprecia que el agente llegó a la estación de carga de batería y después siguió su camino hacia el objetivo.

Y en la Figura 8 se muestran los resultados de la simulación después de que el destino fue alcanzado durante 1000 épocas. Mientras el robot aprende la cantidad de pasos es variante y a partir de la época número 18, los pasos que le toma al robot completar su tarea se estabilizan en 26 pasos.

Para finalizar, en la Figura 9 se muestra el comportamiento del sistema bajo 100 escenarios distintos con un total de 20 obstáculos en cada escenario. En la Figura 9a, se visualiza el número de pasos que le tomó al sistema llegar al destino después de un entrenamiento de 1000 épocas, en cada uno de los 100 escenarios.

Mientras que, en la Figura 9b se muestra la recompensa acumulada durante la última época de entrenamiento de cada escenario.

6. Discusiones

Este trabajo presentó un sistema de navegación para robots móviles basado en una arquitectura FQL, que permite a un robot tomar decisiones de forma autónoma en base a su nivel de carga de batería mientras se mueve de un punto de partida a un punto de destino. Utilizando FQL, el sistema aprende a través del ensayo-error con un paradigma de aprendizaje de refuerzo, en el que la definición de la función de recompensa tuvo un papel importante en el proceso de aprendizaje del sistema.

Se probaron diferentes funciones hasta que se eligió la función (8). En los escenarios usados los obstáculos y los posibles destinos se colocaron en diferentes posiciones. Las variaciones que se presentan de la ruta que se genera con método de planificación de ruta. En algunos casos cuando el robot se encuentra muy cerca de la estación de carga y el valor difuso del nivel de batería cae en un conjunto diferente a lleno, el robot sigue la ruta hacia la estación de carga.

Entre todas las ventajas que se encontraron al seleccionar esta arquitectura, se distingue la posibilidad de que el experto pueda definir el número de estados en los que puede caer el sistema, así como el número de reglas por las que se compone el FIS. Si se utiliza un Q-learning clásico, el número de estados podría ampliarse al intervalo de mediciones del voltaje de la batería, es decir, 100 estados si se usa una escala de nivel porcentual entera de 0 a 100, o incluso más estados, si el dispositivo de medición de la carga dispone de una escala de milivoltios y se ocupan los milivoltios como estados.

Entre las desventajas, el sistema podría no elegir necesariamente el camino más corto en todo momento, ya que en algunos pasos el agente de aprendizaje puede seleccionar permanecer en modo de espera mientras se determinaba el destino a seguir. Sin embargo, con el aprendizaje adquirido, el sistema logra seleccionar las acciones que le permiten completar su tarea.

7. Conclusiones

La arquitectura propuesta para la toma de decisiones ayuda a que un robot móvil sea capaz de seleccionar entre un conjunto de acciones, las cuales le permiten cumplir su objetivo, en este caso el de llegar a un destino predeterminado. Al usar el método de fuzzy Q-Learning la complejidad del sistema es asignada por el número de reglas que se definieron.

A diferencia del método de Q-Learning clásico, en donde los estados corresponderían al total de posibles mediciones del nivel de batería, con el método propuesto se limita la cantidad de estados a los rangos asignados con las funciones de membresía. Se muestra que el método propuesto ayuda a que un robot aprenda a seleccionar tareas de forma autónoma y con esto complete su tarea. En trabajos futuros el número de acciones que tiene el sistema pueden ser aumentadas en caso de que el robot tenga que cumplir con otras tareas.

Agradecimientos. Se agradece el apoyo prestado para desarrollar este proyecto al Instituto Politécnico Nacional (IPN), al Consejo Nacional de Ciencia y Tecnología (CONACYT), y a la Secretaría de Investigación y Posgrado (SIP) a través de los proyectos 20180943, 20190007, 20195835, 20200630 y 20201397

Referencias

1. Agarwal, D., Bharti, P.S.: Nature inspired evolutionary approaches for robot navigation: survey. Journal of Information and Optimization Sciences, 2(41), pp. 421–436 (2020)
2. Glorennec, P.Y., Jouffe, L.: Fuzzy q-learning. In: Proceedings of 6th International Fuzzy Systems Conference, IEEE Barcelona, 2(4864), pp. 659–662 (1997)
3. Gul, F., Rahiman, W., Nazli-Alhady, S.S.: A comprehensive study for robot navigation techniques. Cogent Engineering, 6, pp. 1–25 (2019)
4. Jiang, L., Huang, H., Ding, Z.: Path planning for intelligent robots based on deep q-learning with experience replay and heuristic knowledge. Journal of Automatica Sinica, pp. 1–11 (2019)
5. Kathib, O.: Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research, 5(1), pp. 90–98 (1986)
6. Liu, J., Qi, W., Lu, X.: Multi-step reinforcement learning algorithm of mobile robot path planning based on virtual potential field. In: Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators ICPCSEE´17: Data Science (2017)
7. Lyu, H., Yin, Y.: COLREGS-Constrained real-time path planning for autonomous ships using modified artificial potential fields. The Journay of Navigation, 72(3), pp. 588–608 (2019)
8. Matoui, F., Boussaid, B., Metoui, B., Frej, G.B., Abdelkrim, M.N.: Path planning of a group of robots with potential field approach: decentralized architecture. IFAC-PapersOnLine, 50(1), pp. 11473–11478 (2017)
9. Park, J.W., Kwak, H.J., Kang, Y.C., Kim, D.W.: Advenced fuzzy potential field method for mobile robot obstacle avoidance. Computational Intelligence and Neuroscience, pp. 1–13 (2016)
10. Rostami, S.M.H., Sangiaiah, A.K., Wang, J., Liu, X.: Obstacle avoidance of mobile robots using modified artificial potential field algorithm. EURASIP Journal on Wireless Communications and Networking, 70(1), pp. 1–19 (2019)

11. Shidujaman, M., Samani, H., Raayatpanah, M.A., Mi, H., Premachandra, C.: Towards deploying the wireless charging robots in smart environments. In: International Conference on System Science and Engineering (ICSSE´18), pp. 1–6 (2018)
12. Subbash, P., Chong, K.T.: Adaptive network fuzzy inference system based navigation controller for mobile robot. Front Inform Technol Electron Eng., 20(2), pp. 141–151 (2019)
13. Tiwari, A.K., Guha, A., Pandey, A.: Dynamic motion planning for autonomous wheeled robot using minimum fuzzy rule based controller with avoidance of moving obstacles. International Journal of Innovative Technology and Exploring Engineering, 9(1), pp. 4192–4198 (2019)
14. Tuazon, J.P.C., Prado, K.G.V., Cabial, N.J.A., Enriquez, R.L., Rivera, F.L.C., Serrano, K.K.D.: An improved collision avoidance scheme using artificial potential field with fuzzy logic. In: Tencon´16, IEEE Region 10 Conference, Singapore (2016)
15. Wei, P.A., Tsai, C.C., Tai, F.C.: Autonomous navigation of an indoor mecanum wheeled omnidirectional robot using segnet. In: Proceedings of National Symposium on System Science and Engineering, Taiwan (2019)
16. Zhang, J.: A hybrid reactive navigation strategy for a non-holonomic mobile robot in cluttered environments. In: Proceedings of the 38th Chinese Control Conference (CCC) (2019)