PROPUESTA DE APLICACIÓN DEL PROBLEMA DEL PRODUCTOR-CONSUMIDOR EN LA ROBÓTICA

Rodrigo Vázquez López, M. en C.
Esther Viridiana Vázquez Carmona, M. en C.
Luis Alberto Flores Montaño, M. en C.
Juan Carlos Herrera Lozada, Dr.
Jacobo Sandoval Gutiérrez, Dr.

Instituto Politécnico Nacional
CIDETEC

Universidad Autónoma Metropolitana Unidad Lerma
Departamento de Procesos Productivos

rodrigo_em2@hotmail.com ev.vazquezc@gmail.com lfloresm1703@alumno.ipn.mx
jcrls.ipn@gmail.com j.sandoval@correo.ler.uam.mx

Resumen

En la robótica, existen diversos problemas que ha sido estudiados en los últimos años, sin embargo, se requiere desarrollar algoritmos simples pero efectivos que puedan ser capaces de ejecutarse en sistemas de cómputo con poca capacidad. En este trabajo se analizó y se propuso aplicar el problema del Productor-Consumidor en la robótica. Para ello, se utilizó una arquitectura maestro-esclavo para el intercambio de mensajes entre dos robots. El objetivo de la tarea es que el un robot conozca la trayectoria de otro en tiempo real. Los resultados se validaron experimentalmente utilizando un robot LegoEV3 y una computadora.

1. Introducción

La robótica ha evolucionado en los últimos años. Actualmente, los robots están teniendo un impacto considerable en muchos aspectos de la vida moderna que incluyen desde la fabricación industrial hasta la atención médica (Siciliano & Khatib, 2007).

Dentro de la robótica existen diversos problemas que ha sido estudiados en los últimos años. Dependiendo del área y del problema existen diversas soluciones, sin embargo, algunas de ellas requieren realizar tareas que consumen un con alto costo computacional (Patrón et al., 2013). No obstante, cuando los sistemas cuentan con recursos de hardware limitado, dichas soluciones no logran el mismo desempeño de un sistema de mayor escala debido a la cantidad de tareas internas que realizan los robots.

Ante este desafío, se requiere de soluciones que reduzcan al mínimo el costo computacional. Por ello, es necesario desarrollar algoritmos simples pero efectivos con la finalidad de que puedan ejecutarse y obtener mejores rendimientos en sistemas con hardware reducido tales como los sistemas embebidos y los diversos controladores presentes en los sistemas robóticos.

Por otra parte, uno de los problemas ampliamente conocidos en las ciencias de la computación es el del productor-consumidor. Dicho problema ha sido estudiado durante años, sin embargo, Dijkstra propuso una solución que consideró simple pero efectiva (Dijkstra, 1972). Las aplicaciones más importantes de este problema se encuentran en el intercambio de mensajes entre hilos (threads) en los sistemas operativos multitarea (Stallings, 2012), los sistemas distribuidos y procesos concurrentes (Coulouris et al., 2005).

Adicionalmente, es posible encontrar analogías entre problemas de la robótica con el del productor-consumidor y las arquitecturas robóticas maestro-esclavo. Dichas arquitecturas se utilizan para realizar tareas de cooperatividad entre robots (Miki et al., 2019), seguimiento de vehículos autónomos (Xing et al., 2020), e intercepción de navegación (Marzoughi & Savkin, 2021). Una de las tareas a considerar es el intercambio de las coordenadas de posición entre los robots presentes en el espacio. Conocer en tiempo real la trayectoria de los robots dentro del espacio permite que puedan realizar tareas de seguimiento o evasión de obstáculos.

El objetivo de este trabajo es aplicar la solución del problema productor-consumidor en sistemas robóticos como una alternativa efectiva para la comunicación y paso de mensajes. Se implementó del problema en una arquitectura maestro-esclavo con dos robots con el objetivo de compartir la ubicación del robot maestro en el espacio.

2. Problema del Productor-Consumidor

El problema del productor-consumidor considera dos procesos funcionando en paralelo los cuales intercambian información por medio de un espacio de memoria independiente (buffer) tal y como se observa en la figura 1.

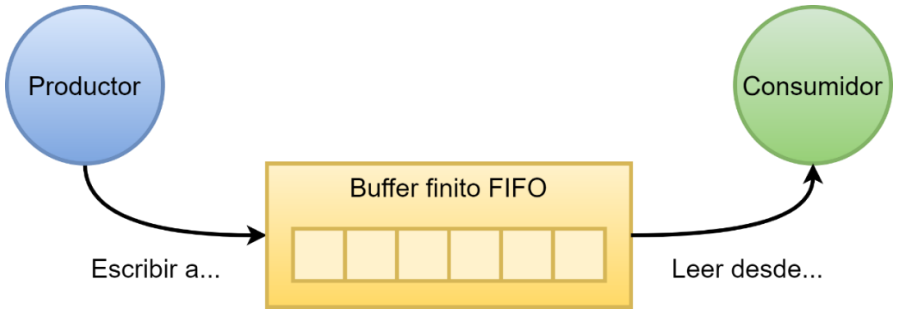

Figura 1. Diagrama de funcionamiento del problema del productor-consumidor

El proceso denominado productor se encarga de generar datos e insertarlos en el buffer mientras que el consumidor extrae los datos del buffer para utilizarlos. Debido a que el buffer es finito, ambos procesos deben procurar que no se sature ni se vacíe completamente el buffer hasta que las tareas finalicen. El algoritmo del productor-consumidor se puede observar en el diagrama de flujo de la figura 2.

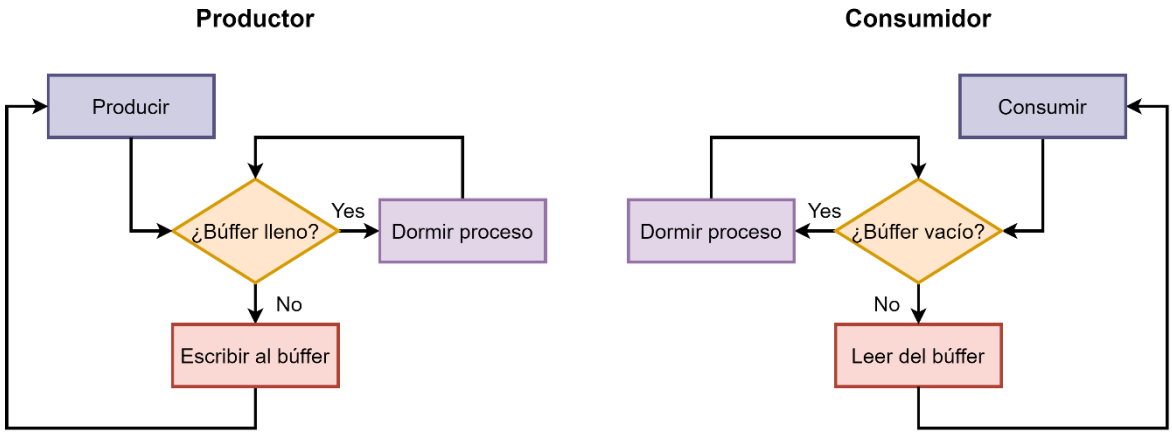

Figura 2. Diagrama de flujo del productor-consumidor.

3. Desarrollo

Para ese trabajo se planteó un escenario con dos robots M (maestro) y E (esclavo) ubicados en el espacio $W \subset \mathbb{R}^2$. El robot E deberá recibir en tiempo real la trayectoria realizada por M en el espacio W compuesta por los puntos $p_i, .., p_f$, tal y como se ilustra en la figura 3.

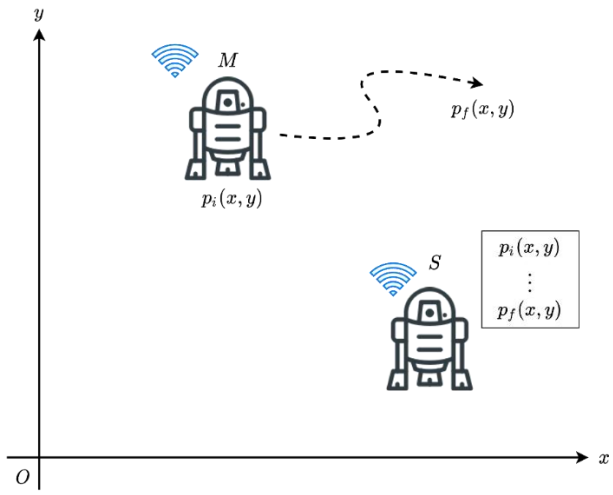

Figura 3. Escenario propuesto para la aplicación del productor-consumidor en la robótica.

3.1. Implementación

Se utilizó un robot móvil LEGO EV3 como robot maestro navegando dentro de un espacio de 2x2 m. Se simuló el robot el esclavo utilizando una PC con sistema operativo Windows. La comunicación entre robot maestro y esclavo se realizó utilizando el protocolo MQTT y comunicación Wi-Fi tal y como se observa en la figura 4.

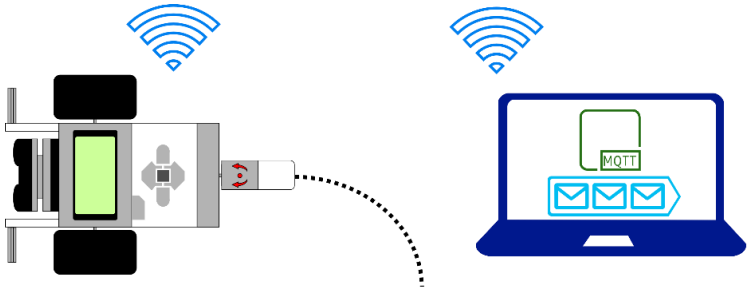

Figura 4. Elementos utilizados para la implementación del problema productor-consumidor.

El robot EV3 es un robot de configuración diferencial que cuenta con giroscopio con precisión de ±3 grados. Los motores de las ruedas incluyen un encoder óptico con exactitud de ±1° para medir el desplazamiento de las ruedas. El EV3 contiene un procesador ARM 9 y 64MB de RAM así como un lector de tarjetas SD en el cual se instaló el Sistema Operativo (SO) EV3dev basado en GNU/Linux. Para realizar el cómputo de la odometría se utilizaron bibliotecas incluidas en el SO que utilizan las ecuaciones presentes en (Feng et al., 1994).

En la figura 5 se muestra el diagrama de flujo de los procesos productor y consumidor que se propusieron. El productor se encarga de obtener la posición del robot EV3 y escribirla en el buffer. Por otra parte, el consumidor se encarga de extraer los datos del buffer y simular que el robot E se desplaza a los puntos de la trayectoria realizada por M.

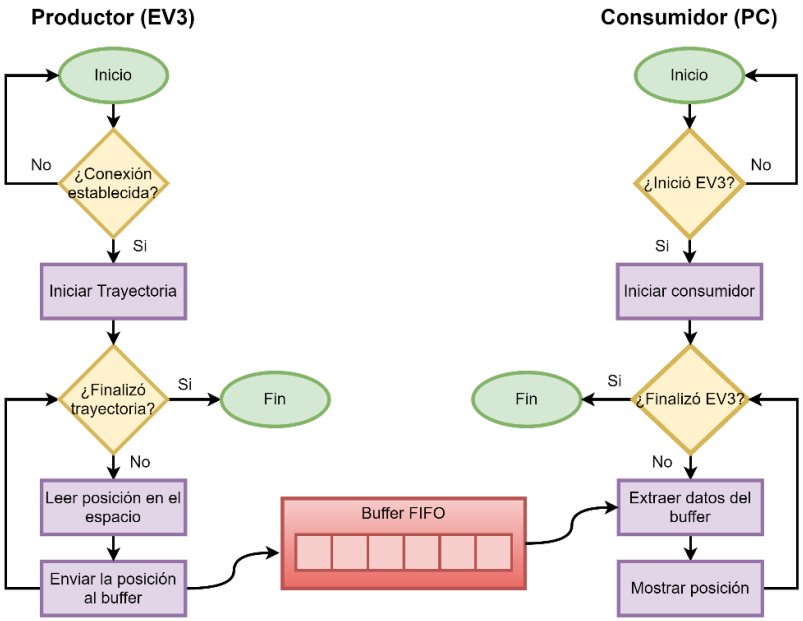

Figura 5. Diagrama de flujo del programa implementado

4. Resultados

En la figura 3 se observan los resultados de la implementación y ejecución de ambos programas. A la derecha se muestran los resultados de la ejecución del programa productor en el robot EV3. A la derecha se muestran los resultados del programa consumidor el cual se ejecutó en la PC.

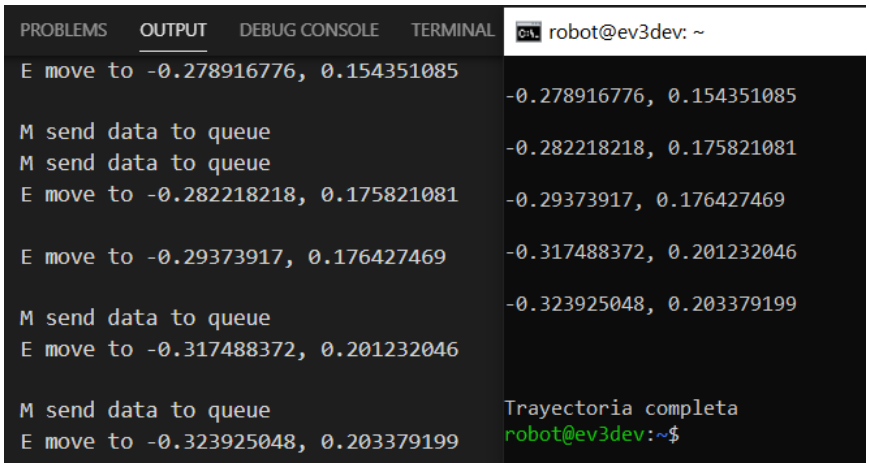

Figura 6. Resultados de la ejecución del programa.

5. Conclusiones

Se analizó y se propuso aplicar el problema del productor-consumidor para el intercambio de mensajes en una arquitectura robótica. Para ello se desarrolló un algoritmo basado en la solución a dicho problema. El algoritmo que se obtuvo es simple pero efectivo y se validó implementándolo en una arquitectura compuesta por dos robots. Los resultados obtenidos

permitieron demostrar la viabilidad de la solución en sistemas con poder de cómputo limitado como el robot EV3.

El algoritmo desarrollado es ligero, por lo que el robot EV3 pudo ejecutarlo sin tener que comprometer el rendimiento del cómputo de las tareas de odometría.

Como trabajo futuro se plantea utilizar el algoritmo en un entorno mejorado para realizar seguimiento e intercepción utilizando vehículos aéreos y terrestres. Adicionalmente, se plantea aplicación del productor-consumidor en otros escenarios de la robótica para probar su eficiencia.

Referencias

Coulouris, G. F., Dollimore, J., & Kindberg, T. (2005). Distributed systems: concepts and design. pearson education.

Dijkstra, E. W. (1972). Information streams sharing a finite buffer. Information Processing Letters, 1(5), 179–180. https://doi.org/10.1016/0020-0190(72)90034-8

Feng, L., Borenstein, J., & Everett, H. R. (1994). Where am i: Sensors and methods for mobile robot positioning. University of Michigan.

Marzoughi, A., & Savkin, A. V. (2021). Autonomous navigation of a team of unmanned surface vehicles for intercepting intruders on a region boundary. Sensors (Switzerland), 21(1), 1–16. https://doi.org/10.3390/s21010297

Miki, T., Khrapchenkov, P., & Hori, K. (2019). UAV/UGV autonomous cooperation: UAV assists UGV to climb a cliff by attaching a tether. Proceedings - IEEE International Conference on Robotics and Automation, 2019-May, 8041–8047. https://doi.org/10.1109/ICRA.2019.8794265

Patrón, J. S., Díaz, L. V., & Solano, M. V. (2013). Diseño e implementación de algoritmo para el procesamiento de imágenes en sistemas embebidos. Ingeniería y Región, 10, 41–53.

Siciliano, B., & Khatib, O. (2007). Springer Handbook of Robotics. Springer-Verlag.

Stallings, W. (2012). Operating systems: internals and design principles. Boston: Prentice Hall.

Xing, H., Ploeg, J., & Nijmeijer, H. (2020). Compensation of Communication Delays in a Cooperative ACC System. IEEE Transactions on Vehicular Technology, 69(2), 1177–1189. https://doi.org/10.1109/TVT.2019.2960114