

Control de Ejecución de Trayectorias de un Robot Diferencial Utilizando el Templado Simulado.

Luis Eduardo Sierra Castillo (1), Dr. Víctor Ayala Ramírez (2)

1 Ingeniería Mecatrónica, Universidad Santo Tomás | Dirección de correo electrónico: led_sierra@hotmail.com

2 Ingeniería Electrónica, División de Ingenierías, Salamanca, Universidad de Guanajuato | Dirección de correo electrónico: ayalav@ugto.mx

Resumen

Este trabajo presenta la implementación del algoritmo de templado simulado para corregir los errores generados en la diferencia de radios en las ruedas de un robot móvil diferencial. Por simplicidad en la investigación se supone que el robot conoce su posición en todo momento y las trayectorias a seguir son rectas. La corrección de los errores de ejecución de trayectoria es importante, ya que son la causa de un mal posicionamiento en una trayectoria dada al robot. Se propone un método de control de ejecución de trayectorias basado en el templado simulado. Se implementó el algoritmo propuesto y se probó en un entorno de simulación robótica desarrollado tanto en Matlab como en C. El método propuesto identifica los parámetros del robot y corrige las velocidades de las ruedas de un robot diferencial. El método desarrollado reduce hasta en un 95 % el error de posicionamiento final del robot móvil para el caso cuando la orientación inicial es similar a la final.

Abstract

This paper presents the implementation of simulated annealing algorithm capable to correct the errors generated by the difference between of the radius in the robot wheels. We assume in this research that the robot knows the position at any time and that the movement primitives are straights. These mistakes are important to correct, because these are the cause of the error when a path is given to the differential robot in an open loop. A simulated annealing algorithm was proposed to control the execution path. For the algorithm proposed we tested it in a robotic simulation environment developed in Matlab and C. This method it's able to identify approximately the kinematics model of the robot and to correct the velocities of the robot's wheels. Also, this method reduces the error up to 95% of the final positioning when the orientation of the robot it's almost equal to the final orientation.

Palabras Clave

Robótica móvil; Robot diferencial; Control de ejecución de trayectorias; Templado simulado.

INTRODUCCIÓN

Un robot móvil necesita tener la característica de autonomía a fin de poder ser utilizado en entornos industriales. Para que un robot realice un movimiento, se debe conocer la trayectoria que el robot debe seguir. El éxito del seguimiento ideal depende de variables que afectan la trayectoria del robot y del mecanismo de control que se implemente para compensar el error. Estas variables pueden incluir las imperfecciones mecánicas del robot o los efectos de la interacción del robot con el escenario. El control deberá servir

al robot a actualizar su conocimiento propio y a identificar el contexto en el cual opera.

Uno de los trabajos encontrados sobre esta temática utiliza control basado en inteligencia artificial para cumplir con el seguimiento de la trayectoria. Medina, Camas, Vazquez, Hernández y Mota [1] abordaron el tema la navegación de un robot móvil en un terreno con complicaciones. El control para la navegación se realizó aplicando redes neuronales entrenadas con anterioridad utilizando un algoritmo de retro-propagación, con el fin de brindar al robot la capacidad de realizar decisiones sobre cuál es el mejor camino a recorrer a partir del conocimiento generado por el entrenamiento de las redes neuronales.

Por otro lado, en el trabajo de Ghommam, Mahmoud, y Saad [2], se utilizan tanto la cinemática como la dinámica del robot en un espacio de estados para realizar un control robusto. En esta aplicación se consideran incertidumbres y perturbaciones en el ambiente del robot. El objetivo de este control es guiar a tres robots en una formación triangular. Esto implica que dichos robots deben cooperar entre sí para dar una ubicación relativa entre todos, y por esta razón que la aplicación requiere un control robusto.

Rossomando, Soria, y Carelli [3] trabajaron en el control de seguimiento de trayectoria de un robot por medio de un control por medio de redes neuronales de base radial. Para este problema se tomó en cuenta la dinámica del robot causada por las velocidades ejercidas en la cinemática.

En este trabajo se diseñó un algoritmo que controla el seguimiento de una trayectoria recta de un robot diferencial haciendo uso del algoritmo templado simulado, el cual se encarga de encontrar el modelo cinemático del robot aproximado al real y las velocidades de las ruedas para mantener el mínimo error. También se contempla que la posición del robot es conocida en todo momento para efectos de simplicidad en el desarrollo de la problemática. En particular, sólo se consideró que los radios en las ruedas del robot no son ideales. Es decir, el robot presenta diferentes radios debido a incertidumbres inherentes a su fabricación o inducidos por el desgaste debido a su uso.

MATERIALES Y MÉTODOS

Robots Móviles de Manejo Diferencial

En el mercado actual existen varios tipos de robots móviles. Cada uno de ellos, implicará diferentes modelos y tipos de control. En este trabajo se utilizan robots móviles de ruedas y basado en el trabajo de Lui y Jiang [4] y en los robots móviles disponibles en el laboratorio LaViRIA de la Universidad de Guanajuato sede Salamanca, se trabaja con un robot de dos ruedas diferenciales.

Cinemática del robot diferencial

Basado en los autores mencionados anteriormente, se aplicó solo la cinemática del robot para hallar las variables necesarias para el control de seguimiento de trayectoria, ya que no se consideraran fuerzas en el modelo del robot. Como siguiente paso, se planteó el modelo del robot, tomando en cuenta las variables presentes en el movimiento de éste (ver Imagen 1).

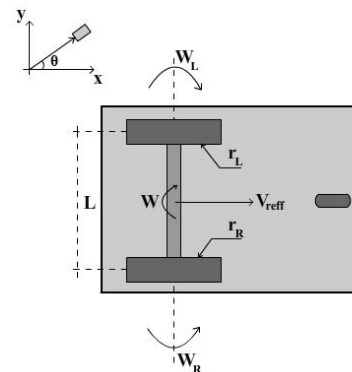


IMAGEN 1: Variables presentes en el movimiento del robot.

Donde L , es la distancias de las ruedas, W_L es la velocidad angular de la rueda izquierda, W_R es la velocidad angular de la rueda derecha, W es la velocidad angular del robot, V_{reff} es la velocidad lineal del robot, θ es la orientación del robot, r_L es el radio de la rueda izquierda y r_R es el radio de la rueda derecha. Al plantear las variables del robot, se pudo obtener la siguiente ecuación que describe la cinemática de este:

$$\begin{bmatrix} \dot{X}(t) \\ \dot{Y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_{reff} \\ \frac{W_R(t) r_R - W_L(t) r_L}{L} \end{bmatrix} \quad (1)$$

Donde V_{reff} se puede calcular con la Ecuación 2:

$$V_{reff} = \frac{W_R(t) r_R + W_L(t) r_L}{2} \quad (2)$$

A partir de la Ecuación 1, se puede conocer los valores de la posición en X y Y y la orientación θ . Para conocer los valores de las posiciones finales y velocidades en cada instante, se consideran constantes los valores del estado actual del robot obteniendo la Ecuación 3. Esta ecuación es clave para realizar el control de trayectoria.

$$\begin{bmatrix} \theta(t) \\ X(t) \\ Y(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \cos(\theta) \\ 0 & \sin(\theta) \end{bmatrix} \begin{bmatrix} W_R r_R - W_L r_L \\ L \\ V_{eff} \end{bmatrix} t + \begin{bmatrix} \theta_0 \\ X_0 \\ Y_0 \end{bmatrix} \quad (3)$$

Dado al continuo uso de los robots móviles, se puede presentar un desgaste en las ruedas que causa que su radio disminuya con respecto al radio ideal, puede que este desgaste sea uniforme y de igual forma en ambas ruedas, pero si no lo es, esta diferencia de radios genera una velocidad angular en el robot, lo que implica que su modelo cinemático es diferente al ideal y como consecuencia, los resultados de la ejecución de una primitiva de movimiento en el robot real pueden ser diferentes a los predichos por el modelo.

En este trabajo se plantea el caso donde el desgaste no es igual para ambas ruedas. Los trabajos encontrados sobre esta temática realizan un control basados en la cinemática del robot, por ejemplo Ribeiro y Lima [5] explican que uno de los problemas de los robots móviles radica en el desarrollo de la cinemática y en la dinámica de este, explicando la diferencia entre estos conceptos y el desarrollo de la cinemática en un robot diferencial. Por otro lado Bonilla, Reyes y Mendoza [6] utilizan la cinemática y la dinámica en un robot móvil para considerar los torques ejercidos por los motores, haciendo un control robusto.

Control de Ejecución de Trayectorias

El control de ejecución de trayectoria de un robot móvil es el problema que trata de que el robot móvil ejecute fielmente la trayectoria que le fue comandada. Las trayectorias planificadas generalmente son calculadas usando modelos ideales de los robots. Sin embargo, el modelo sólo es una aproximación de la realidad, por lo que la ejecución en lazo abierto de esta trayectoria generalmente resulta en errores de posicionamiento del robot móvil. El uso de un lazo de control permite que el robot se ajuste a las condiciones del entorno y a sus propias imperfecciones para corregir su posicionamiento en el momento de ejecución de la trayectoria y ajustarse con fidelidad a la primitiva comandada. Existen diferentes enfoques de control que pueden ser utilizados. Desde aquéllos que utilizan las

técnicas clásicas de control como el control PID hasta aquéllas que basan la estrategia en inteligencia artificial, usando lógica difusa o redes neuronales, entre otras. Nuestro enfoque se basa en aplicar el algoritmo de optimización conocido como templado simulado.

Templado Simulado

El algoritmo de templado simulado sirve para encontrar el óptimo global de una función haciendo una analogía computacional del proceso de templado de un material [7].

En nuestro problema, la energía del templado simulado se asimiló a la suma de la distancia entre la posición real del robot y la ideal, y la diferencia entre las orientaciones real e ideal del robot. Esta energía es la función objetivo que se busca minimizar.

El algoritmo general para usar el templado simulado en el control de ejecución de trayectorias de robot se muestra en la Imagen 2 y se inspira del pseudocódigo presentado por Jones [8].

En el subproceso del templado simulado de la Imagen 2, se realizan dos templados simulados. Uno de ellos es para el reconocimiento del modelo cinemático, pues al reconocer que hay un error mayor, busca otro modelo cinemático que tenga un menor error al actual, ya que el modelo ideal no funciona en las condiciones que se dieron. El segundo templado simulado se realiza para encontrar las velocidades de las ruedas derecha e izquierda que corrijan el error en la orientación que se causó inicialmente. El propósito de implementar estos dos templados simulados es corregir el error significativamente.

RESULTADOS Y DISCUSIÓN

El uso del algoritmo propuesto fue comparado contra la ejecución de la primitiva en lazo abierto. La corrección de la trayectoria se puede ver en la Imagen 3. El cambio es notorio, pues corrige la trayectoria del robot aunque no perfectamente. Esto se debe a que el consumo computacional se incrementa a medida que se busca más precisión en el algoritmo.

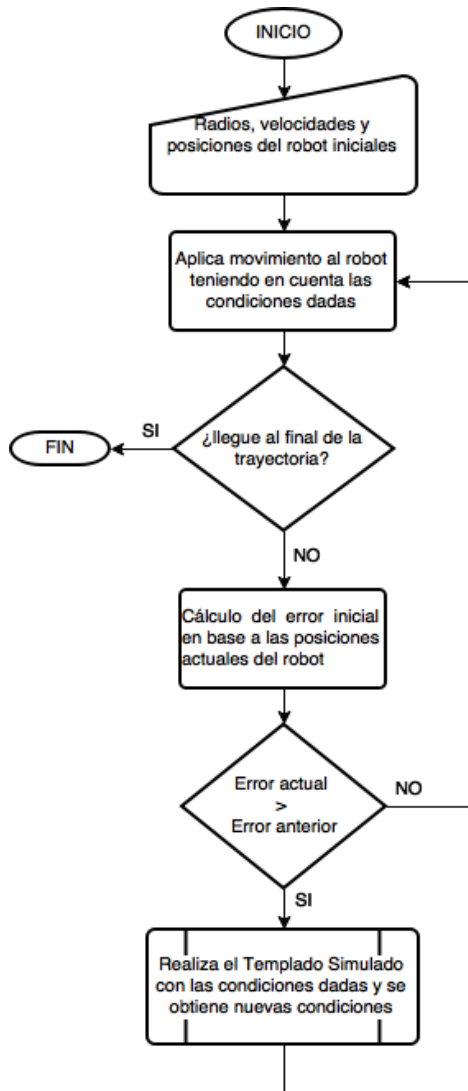


IMAGEN 2: Diagrama de flujo para el control de trayectoria.

Actualmente el algoritmo realiza 20 comparaciones de modelos diferentes al que se empieza, si uno de estos tiene menor error que los otros 19 se elige como el final, de no ser así se realizan otras 20 comparaciones o hasta que la temperatura del templado simulado llegue a la temperatura mínima. En las trayectorias rectas, también se tiene contemplado las diagonales como se muestra en la Imagen 4,

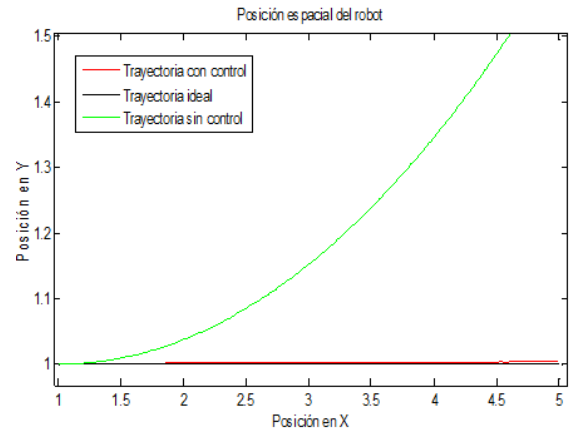


IMAGEN 3: gráfica de posición cartesiana del robot.

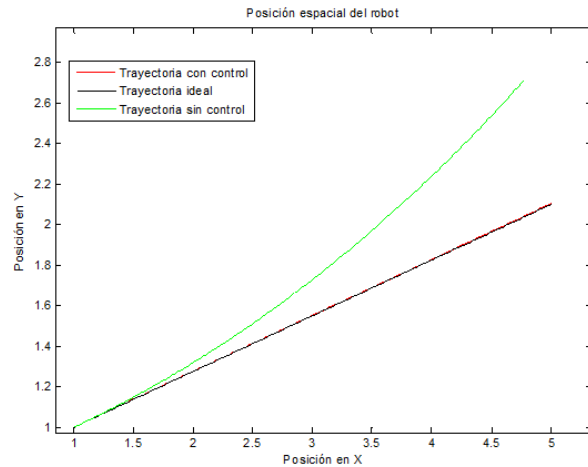


IMAGEN 4: gráfica de posición cartesiana del robot con una trayectoria diagonal.

Como se puede observar, las anteriores pruebas se realizaron cuando el robot se encontraba con la misma orientación inicial a la ideal, si la orientación es diferente como se muestra en la Imagen 5, se observa que le toma más tiempo encontrar estabilizarse en un punto donde no genere la mayor cantidad de error posible.

Este problema se podría solucionar permitiendo que el algoritmo pueda seccionar los movimientos primitivos, rotaciones y traslaciones, por separado. Para este caso se empezaría por la primitiva de la rotación hasta que la orientación sea igual o aproximada a la deseada, terminando por la traslación al punto dado.

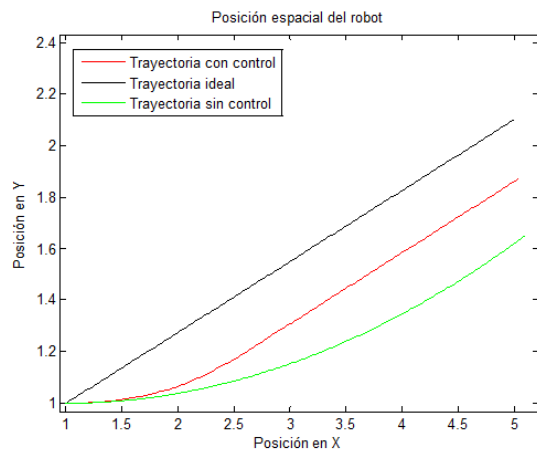


IMAGEN 5: gráfica de posición cartesiana del robot con orientación diferente.

La corrección exitosa de la trayectoria mediante el algoritmo de templado simulado depende del parámetro de desviación estándar s del algoritmo. Este parámetro tiene relación con el espacio de búsqueda de alternativas para identificar los parámetros reales del robot (radios de las ruedas) y las velocidades corregidas a aplicar al robot.

Se ha encontrado que una desviación estándar mayor de 0.05 permite la corrección exitosa de la trayectoria para unos radios desgastados con una desviación en su dispersión normal del 0.01 al radio ideal. El uso de un valor adecuado de s permite cubrir la incertidumbre con redundancia y variedad de valores, evitando caer en un mínimo local.

El algoritmo propuesto se implementó en un entorno de simulación robótica desarrollado en Matlab y en C. La ventaja de la implementación en C es la velocidad de procesamiento, mientras que Matlab nos provee de un entorno de prototipado rápido para la implementación del método.

CONCLUSIONES

Se logró implementar el algoritmo de templado simulado para el problema del control de ejecución de trayectorias de un robot móvil. El método propuesto funciona para primitivas de avanzar en línea recta en cualquier ángulo de orientación cuando el robot tiene una orientación inicial semejante a la de la primitiva por ejecutar. Cuando no se está en este caso, se ha propuesto una

estrategia alternativa. Se ha encontrado que el buen funcionamiento del algoritmo depende de una selección adecuada del parámetro que controla la exploración del espacio de búsqueda del templado simulado. Se ha validado la importancia del uso de la simulación en el desarrollo de funcionalidades robóticas.

AGRADECIMIENTOS

Se agradece a la Universidad de Guanajuato por la beca brindada apoyando el trabajo presente. Así mismo, al Dr. Víctor Ayala y a los integrantes del laboratorio LAVIRIA por compartir las experiencias y conocimientos en la investigación. También a la Universidad Santo Tomás sede Bucaramanga, por apoyarme en económicamente y por brindarme la oportunidad de estar en esta estadía.

REFERENCIAS

- [1] Medina, A., Camas, J. L., Vazquez, J. A., Hernández, H. R., & Mota, R. (2014). Neural Control System in Obstacle Avoidance in Mobile Robots Using Ultrasonic Sensors. *Journal of Applied Research and Technology*, 12 (1), 104-110.
- [2] Ghommam, J., Mahmoud, M. S., & Saad, M. (2013). Robust cooperative control for a group of mobile robots with quantized information exchange. *Journal of the Franklin Institute*, 350 (8), 2291-2321.
- [3] Rossamando, F. G., Soria, C., & Carelli, R. (2010). Control de Robots Móviles con Incertidumbres Dinámicas usando Redes de Base Radial. *Revista Iberoamericana de Automática e Informática Industrial*, 7 (4), 28-55.
- [4] Lui, T., & Jiang, Z. P. (2012). Distributed formation Control of nonholonomic mobile robots without global position measurements. *Automatica*, 49 (2), 592-600.
- [5] Ribeiro, M. I., y Lima, P. (2002). Kinematics model of mobile robots. *Mobile robots course*. Recuperado: <http://users.isr.ist.utl.pt/~mir/cadeiras/robmovel/Kinematics.pdf>
- [6] Bonilla, I., Reyes, F., y Mendoza, M. (2005). Modelling and Simulation of a Wheeled Mobile Robot in Configuration Classical Tricycle. En WSEAS (World Scientific and Engineering Academy and Society). 5th WSEAS International Conference on Instrumentation, Measurement, Control, Circuits and Systems. Cancún, México.
- [7] Theodoridis, S., Koutroumbas, K. (2003). *Pattern recognition* (2nd ed.) San Diego: Elsevier.
- [8] Jones, M. T. (2008). *Artificial Intelligence A System Approach* Massachusetts: Infinity Science Press LLC.