

IMPLEMENTACIÓN DEL PVA PARA LA PLANIFICACIÓN DE TRAYECTORIAS CON OBSTÁCULOS DE UN ROBOT MÓVIL EN UN MÓDULO DE MICROCONTROLADOR

J. A. Rojas Estrada, P. Quintero Álvarez, L.A. Montoya Soto, H.A. Villarreal Hdez.
 Posgrado en Ingeniería Mecatrónica, División de Estudios de Posgrado e Investigación
 Instituto Tecnológico de Nuevo León
 Ave. Eloy Cavazos 2001, Col. Tolteca, Guadalupe, N.L. C.P. 67170, México
 Tel. (81) 81570500 ext. 137 Fax (81) 81570505
 jarojas2001@yahoo.com.mx, patyqar@yahoo.com

RESUMEN

El presente trabajo se enfoca en el problema de la implementación de los algoritmos de control del PVA a un robot diferencial en el robot mismo. Se utiliza un robot móvil armado con dos motores de corriente directa controlados por un Micro controlador Flyport Wi-Fi. Resultado de las ventajosas características del módulo Wi-Fi se pueden utilizar distintos protocolos de comunicación como son TCP/IP o UDP para la comunicación con el robot. Por medio de una aplicación desarrollada en MATLAB que determina la posición y orientación exacta del robot, se dispone de manera continua de la información. Se puede realizar la comunicación entre el sistema de control (CPU), donde se ejecuta MATLAB y el robot, como recibir los parámetros de posición y orientación del mismo, obtenidos desde nuestro sistema de posicionamiento, así como establecer los parámetros de la posición final. Los resultados obtenidos son alentadores.

1. INTRODUCCION

1.1 Robot Diferencial

El vector de estado: $\mathbf{q} = [x \ y \ \theta]^T$ donde (x, y) representan la posición y θ la orientación del robot con respecto al eje X. Representa la configuración cinemática del robot diferencial mostrado en la figura 1. La figura solo solo representa un esquema del robot real, el cual es un robot diferencial armado y que fue construido para implementar diversos procedimientos relacionados con las técnicas del Polígono de Velocidades Admisibles (PVA) para un proceso de navegación con obstáculos [1], [2] y [3].

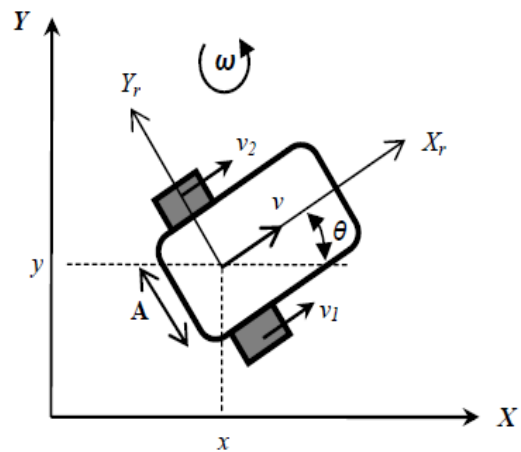


Figura 1. Robot diferencial

El modelo cinemático para este robot está dado por la expresión:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos q & 0 \\ \sin q & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

la cual se puede poner como $\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})\mathbf{v}$, donde $\mathbf{J}(\mathbf{q})$ es el Jacobiano del robot y \mathbf{v} es el vector de control compuesto por la velocidad lineal y la velocidad angular.

Los sistemas no holonómicos tienen restricciones de velocidad no integrables, tales restricciones se representan por la siguiente ecuación:

$$\dot{y} - x \tan q = 0$$

1.2 El Polígono de velocidades admisibles PVA

El Polígono de Velocidades Admisibles es un método para planificar trayectorias llamado PVA y se considera un método local reactivo, gracias a su algoritmo es posible que el robot móvil reaccione a la presencia de obstáculos estáticos o en movimiento, y gracias a su ley de control mediante la regulación de las variables de velocidad logra ser muy estable. El PVA está conformado por dos secciones:

La primera contiene una Ley de Control con un optimizador de velocidad. Su objetivo es de calcular una velocidad de referencia que permita al robot alcanzar una posición fija a pesar de su propiedad no holonómica.

Si $\mathbf{q}_f = [x_f \ y_f \ 0]^T$ es la posición deseada y

$\mathbf{q} = [x \ y \ \theta]^T$ es la posición inicial.

Se usan coordenadas polares (a, α) para obtener el vector de error $\mathbf{z}(t)$ que se muestra en la figura 2.

Donde

$$\mathbf{z}(t) = \begin{bmatrix} a \\ \alpha \end{bmatrix} \quad \text{tal que} \quad a = \sqrt{x_e^2 + y_e^2} \quad y$$

$$\alpha = a \tan 2(y_e, x_e) - \theta$$

con $\alpha \in [-\pi, \pi]$

La ley de control de la velocidad lineal y angular que converge exponencialmente está dada por

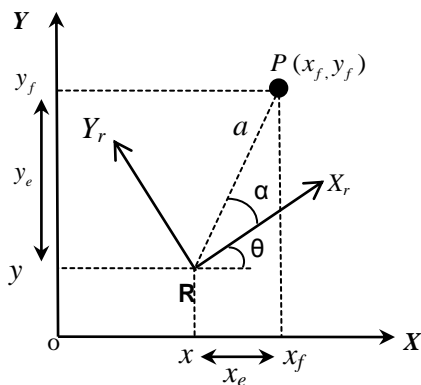


Figura 2. Cálculo de las variables de error

$$v = k_1 a \cos \alpha$$

$$\omega = k_2 \alpha + k_1 \sin \alpha \cos \alpha$$

donde $k_1, k_2 > 0$ y v_1 y v_2 son las velocidades en cada rueda.

El optimizador de velocidad se usa para evadir obstáculos y se deriva de la formulación hecha por Faverjon y Tournassaud, mencionado en [1]. Este módulo a través del cálculo de la mínima distancia encuentra el vector velocidad (velocidad lineal y angular) más cercano a la velocidad de referencia que garantiza que el robot no colisione formando así el polígono de velocidades admisibles. [4]

El segundo módulo utiliza el PVA para lograr que el robot evada las singularidades tales como el caso en el que el robot tome velocidades

($v = 0$ y $\omega = 0$) que puedan aparecer en su trayectoria o cuando la posición de los obstáculos hace el punto de meta inalcanzable.

1.3 Flyport Wi-Fi (Microcontrolador)

Flyport Wi-Fi [5] es un dispositivo inalámbrico el cual contiene un Microcontrolador de la gama alta de Microchip PIC24FJ y un transmisor Wi-Fi. [6] y [7] Tiene un conector de 26 pines para comunicarse con dispositivos electrónicos externos.

Este Microcontrolador tiene las siguientes características:

- WIFI 802.11 b/g/n
- Microchip PIC 24F 16 bit processor
- Wi-Fi Module Microchip MRF24WB0MA/RM con Antena Pcb
- Desarrollo sencillo gracias al IDE OpenPicus.
- Webserver (configurable)
 - TCP Socket
 - UDP Socket
 - SNTP
 - SMTP
- Alimentación de 5V o 3.3V.
- Pines re-mapeables en run-time.
- Entradas y salidas digitales
- Entradas Analógicas.
- 4 UARTs, 1 SPI, 1 I2C
- 35*48*15mm (11 gr).

1.4 Comunicación Wi-Fi

El Wi-Fi es un tipo de comunicación inalámbrica la cual tiene un alcance aproximado de 30 mts. Este medio de comunicación trabaja con diferentes protocolos como el TCP/IP y el UDP.

TCP, (Transmission Control Protocol), es un protocolo orientado a paquetes de datos. se puede

usar para transmitir y recibir paquetes a través de la red entre cliente y servidor. UDP es un protocolo de comunicación orientado a paquetes de datos muy similar al TCP, a diferencia del TCP el UDP no incluye un control del flujo de datos. En este protocolo de comunicación es menor la cantidad de bytes en los paquetes de datos comparado con TCP

2. DESARROLLO

Nuestro trabajo se enfoca en la implementación de los algoritmos de control del PVA a un robot diferencial, con dos motores de corriente directa controlados por un Microcontrolador Flyport Wi-Fi. Gracias a su módulo Wi-Fi podemos utilizar distintos protocolos de comunicación como son TCP/IP o UDP para comunicarnos con el robot. Por medio de una aplicación desarrollada en MATLAB podemos realizar la comunicación entre el sistema de control (CPU) y el robot. Como recibir los parámetros de posición y orientación del mismo, obtenidos desde nuestro sistema de posicionamiento, así como establecer los parámetros de la posición final. La planificación de trayectorias mediante el PVA se realiza mediante mediciones de distancia entre el robot y el obstáculo obtenidos mediante sensor de distancia IR colocado en una de las entradas del convertidor análogo digital de nuestro Microcontrolador, después estas distancias son enviadas por el Microcontrolador al sistema de control por medio del protocolo de comunicación UDP con la ayuda de funciones predefinidas para el envío y recepción de datos. Estas distancias nos permiten calcular el conjunto de velocidades contenidas en el polígono, por otro lado el sistema de posicionamiento del robot debe ser lo bastante preciso para realizar los cálculos de velocidades en la ley de control.

2.1 Características del Robot diferencial.

Este robot diferencial cuenta con dos motores de corriente continua con sistema de transmisión y encoder colocado en las llantas. Las llantas están ubicadas en el centro del robot móvil, y existen dos ruedas locas colocadas en la parte posterior. Los motores son controlados por el Microcontrolador Flyport Wi-Fi.

Un sistema de posicionamiento local para interiores por medio de procesamiento de imágenes fue desarrollado. Este sistema de posicionamiento nos permite saber las coordenadas exactas de posición (x , y) y la orientación (θ). Mediante una cámara digital colocada en el techo del área de trabajo se envían imágenes y se procesan con la

ayuda de una función programada en MATLAB en tiempo real, dicha función es llamada desde el sistema de control cuando es necesario obtener los datos de posición y orientación actuales. El programa identifica dos cuadros de diferente color ubicados en la parte superior del robot móvil, uno colocado en el baricentro y otro en la punta del robot móvil. Se calcula el baricentro de cada cuadro y se obtiene la posición de los píxeles en los que se encuentran dichos baricentros en la imagen. Después de obtener las coordenadas de los dos cuadros, por medio de la ecuación de la pendiente se obtiene el ángulo de orientación del robot respecto al eje X del espacio de trabajo, en este caso la imagen capturada por la cámara. Así como la posición del robot (x , y), en este caso las coordenadas del cuadro colocado en el baricentro del robot.

El robot cuenta también con dos sensores de distancia IR 2D120X, los cuales detectan objetos estáticos o en movimiento que se presentan en la trayectoria del robot. Estos transductores regresan un dato en voltaje proporcional a la distancia a la que se encuentra el objeto y pueden detectar objetos ubicados hasta 30 cm del robot. Estos sensores son conectados al convertidor análogo digital del Microcontrolador del robot móvil. El Microcontrolador se encarga de procesar los datos digitalizados de voltaje y los convierten en valores de distancia real y los envía por medio de Wi-Fi con el protocolo de comunicación UDP al sistema de control.

En la figura 3 se muestra un diagrama del sistema completo.

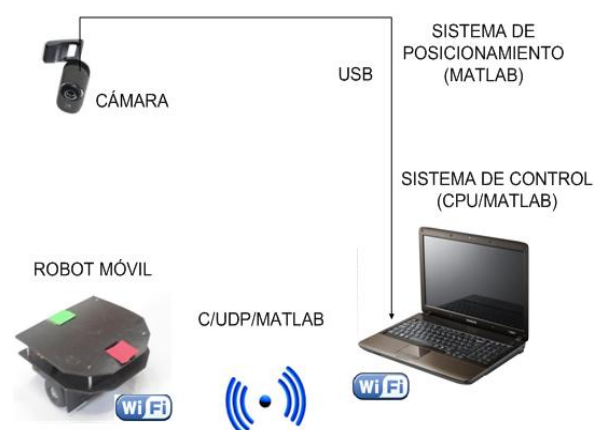


Fig 3 .- Flujo de datos del sistema.

2. 2 Programación del controlador

Se desarrolló una aplicación de MATLAB la cual contiene diferentes secciones; interface, algoritmo de control, una función para obtener los datos del sistema de posicionamiento y otra función para la comunicación UDP entre el sistema de control y el robot móvil.

La interface HMI muestra las coordenadas de posición, la orientación, la velocidad lineal y la velocidad angular actuales del robot móvil. En la Figura 4 se muestra esta interface donde el usuario puede ingresar las coordenadas finales a las cuales se desea que llegue el robot móvil, así como detenerlo en cualquier momento.

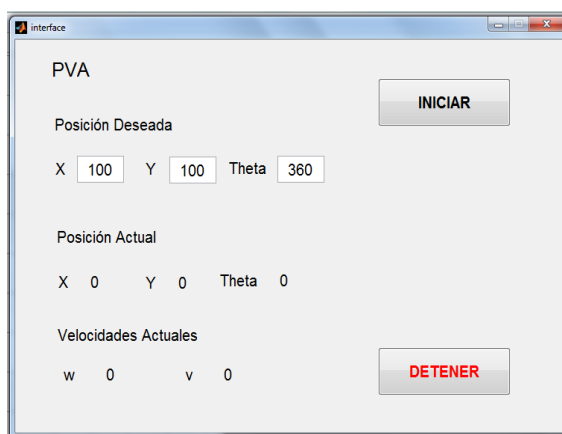


Fig 4.- Interface gráfica de usuario del PVA.

El algoritmo de control del PVA, recibe datos de otras secciones del programa como la posición y orientación actuales y finales del robot móvil. Este algoritmo de control regresa el vector de posición $[v \ \omega]^T$, el cual es enviado por la sección de comunicación del programa.

La parte de comunicación se realiza mediante el protocolo UDP, se envían y reciben datos desde el robot, y desde el sistema de posicionamiento. A continuación se muestran los fragmentos de código de la clase para el envío y recepción de datos mediante este protocolo.

3. RESULTADOS Y CONCLUSIONES

Los módulos de distancia y posicionamiento funcionan correctamente, se obtienen valores de distancia, posicionamiento y orientación muy precisos.

Aunque fue difícil crear una administración del envío de datos mediante el protocolo UDP, se pudo lograr una comunicación entre los dos dispositivos CPU y Microcontrolador (robot móvil). La comunicación UDP utilizada fue por medio de WiFi, con una dirección IP y un puerto de comunicación del dispositivo remoto, en este caso el robot con el módulo WiFi.

El procedimiento para el monitoreo de posición y orientación presentado en este artículo tiene la ventaja de que es un proceso continuo, limitado solamente al área de alcance de la cámara web. Básicamente el CPU (programa en Matlab) representa el sistema de control, pero quien lleva a cabo toda la ejecución del robot es el Microcontrolador que reside en el robot. Como trabajo futuro se menciona que se desarrollan los módulos para la implementación de restricciones cuando el robot navegue en ambientes de obstáculos. El principal problema que se tenía, conocimiento de la posición y orientación de manera continua además de la distancia a los obstáculos [4], fue solucionado gracias a los resultados de este trabajo.

4. REFERENCIAS

- [1] Quintero Álvarez P., Ramirez, G., Zeghloul, S., *A collision free path planning method for articulated mobile robot*, Applied Bionics and Biomechanics, 4:2, pp.71-81, 2007
- [2] Quintero Álvarez P., Rojas Estrada J. A., Fernández Ramírez A. A., Ramírez Torres J. G. *Técnicas para evasión de obstáculos en robótica móvil*, CISC 2010, Orlando Florida, 2010, pp-75-80, Vol. I.
- [3] Martínez Soto M. A., Quintero Álvarez P., Rojas Estrada J. A., Fernández Ramírez A. A., *Planificación de trayectorias libres de colisión en robots comerciales*, Electro 2010, Chihuahua, Chihuahua, página 226-230, Vol. XXXII.
- [4] J. A. Rojas Estrada, P. Quintero Álvarez, M.A. Martínez Soto, A.A. Fernández Ramírez., I. Z. Ochoa, *Medición de distancia para la aplicación del PVA en robots comerciales* Electro 2011, Chihuahua, Chihuahua, página 124-129 Vol XXXIII.
- [5] Flyport IDE 2.2 User guide OpenPicus. <http://www.openpicus.com>
- [6] PIC24FJ256 datasheet MicroChip. <http://www.microchip.com>
- [7] FLYPORT - Programmer's Guide (IDE 2.2) rev 1.0 OpenPicus. <http://www.openpicus.com>