

Robot Móvil de Tracción Diferencial con Plataforma de Control Modular para Investigación y Desarrollo Ágil de Proyectos

Pérez Arreguín Jorge Israel, Tovar Arriaga Saúl, Ubaldo Giovanni Villaseñor Carrillo, Gorrostieta Hurtado Efrén, Pedraza Ortega Jesús Carlos, Vargas Soto José Emilio, Ramos Arreguín Juan Manuel y Sotomayor Olmedo Artemio

Universidad Autónoma de Querétaro, Facultad de Informática.
Campus Juriquilla, Avenida de las Ciencias s/n, Juriquilla Querétaro Qro. CP. 76230
israel_1985@live.com.mx

Resumen

En este trabajo se presenta un auto con tracción diferencial diseñado con una plataforma de control modular para ser utilizado en diferentes proyectos que involucren disciplinas como inteligencia artificial, control y robótica. El control del robot es realizado con una plataforma Arduino el cual está diseñado para soportar diferentes módulos como, por ejemplo, el control de motores, comunicación inalámbrica, Ethernet y muchos más. Como caso de estudio, se presenta el sistema con un módulo equipado con una antena XBee para el control inalámbrico del robot. Como sistema de visión, se ha montado un iPod para la transmisión inalámbrica de video por medio de la aplicación FaceTime. Una interface gráfica con diferentes opciones para el control manual así como para la realización de trayectorias predefinidas es presentada. La calidad con que las trayectorias son realizadas en relación a las trayectorias ideales es puesta a prueba.

Palabras clave: Software libre, Plataforma Arduino, Robot móvil con tracción diferencial

1. Introducción

Los robots móviles, en comparación con los robots fijos, abren un gran abanico de posibilidades en la realización de tareas, sea de modo automático o siendo manipulados por medio de teleoperación [1, 2]. Estos sistemas se mueven en ambientes cambiantes llenos de obstáculos que pueden ser aleatorios e inesperados. A diferencia de los robots fijos, los robots móviles cambian con cada movimiento la posición y orientación de su eje coordenadas base. Por lo tanto deben reconocer

constantemente el ambiente donde se mueven para poder adaptarse acordemente. Para que un sistema robótico móvil pueda ser exitoso necesita estar altamente sensorizado y tener cierto grado de autonomía aplicando algunos conceptos de inteligencia artificial.

Para realizar un proyecto de inteligencia artificial o control con un robot móvil se necesita invertir una cantidad considerable de tiempo, dinero y esfuerzo en implementar un sistema funcional. Construir un robot de estas características, aunque parezca relativamente sencillo, implica invertir varias horas de trabajo y se necesitan diferentes tipos de herramientas mecánicas, electrónicas así como informáticas. En ciertas ocasiones, se desearía trabajar directamente en los nuevos conceptos más que en la implementación del prototipo. Una solución para aminorar esfuerzos es adquirir un sistema robótico comercial con las características en tamaño y configuración de tracción más cercanas a las deseadas. El problema de este enfoque es que las características del robot no son siempre las mejores y su forma de programación no es muy flexible. Cabe mencionar que este tipo de sistemas tienen un costo relativamente alto.

Ciertas áreas de la electrónica también se han beneficiado con el concepto de software libre. Un caso muy conocido es el de la plataforma de Arduino [3]. Los desarrolladores de esta plataforma se dieron cuenta de lo engorroso que es realizar un proyecto con microcontroladores empezando desde cero. Para esto, desarrollaron una tarjeta que contiene los elementos electrónicos necesarios para comenzar un proyecto, casi de cualquier tipo. Lo más innovador de este sistema es que puede ser programado con software libre por medio de un cable USB. Esto permite no solamente hacer el proyecto de una manera más rápida, sino que también llega a ser más económico debido a que no hay que comprar licencias de software. Otro concepto muy interesante

que maneja el sistema Arduino es el de modularidad. Su construcción permite que se conecten más tablillas (*Shields*) con componentes electrónicos con cierta facilidad. Este tipo de tablillas permite al Arduino llevar a cabo tareas como mover motores de CD o de pasos, establecer conexión a Ethernet, conectarse con otros dispositivos por medio de circuitos de radiofrecuencia [4, 5], etc. Hoy en día, diversos fabricantes ofrecen varios tipos de *shields* para realizar diferentes tareas. Tal ha sido el éxito del sistema Arduino que hoy en día se pueden ver en YouTube diversos proyectos, algunos de ellos muy interesantes, realizados por estudiantes o amateurs.

En este trabajo se presenta un robot móvil diseñado con una plataforma de control modular para desarrollar proyectos de investigación de una forma rápida y eficiente. Para esto se hace uso de la plataforma Arduino y diferentes *shields* que permitió desarrollar el proyecto de forma sencilla a un costo relativamente bajo. Como caso de estudio, se presenta un sistema telecontrolado desde una computadora por medio de una interface gráfica. El sistema utiliza además un iPod como sistema de video para visualizar remotamente una imagen con perspectiva desde el robot.

2. Materiales y Métodos

2.1 Control Inalambrico

La comunicación inalámbrica se logra con un módulo XBee S2 (Digi International Inc., Minnetonka, Minesota, EUA) el cual está basado en el protocolo inalámbrico ZigBee (IEEE 802.15.4). El módulo XBee puede transmitir a distancias de hasta 30 metros en lugares cerrados y hasta 100 metros en lugares abiertos [5]. El sistema se utiliza para comunicarse con el Arduino como si fuera un puerto serial. A través del puerto se controla el robot por medio de la transferencia de caracteres. En comparación con otros sistemas de radiofrecuencia, el XBee permite la transmisión confiable debido a su manera de encriptación y codificación. Para realizar una conexión modular con el Arduino se utilizó un sistema Arduino XBee Shield (© Libelium Comunicaciones Distribuidas S.L., Zaragoza, España) lo cual evitó que se diseñara una tarjeta dedicada solo a conectar la antena XBee. El protocolo ZigBee usa una conexión serial y puede comunicarse con cualquier conexión UART. Como se mencionó anteriormente, el robot es telecontrolado por medio de una computadora portátil. Ya que

raramente estas computadoras tienen un puerto serial, se optó por utilizar un puerto USB. Sin embargo, es necesario hacer la transición de un puerto serie a un puerto USB por lo cual se utilizó un convertidor XBee Explorer Dongle (SparkFun Electronics, Denver, EUA). En la Fig. 1 se puede apreciar el esquema general de control del robot.

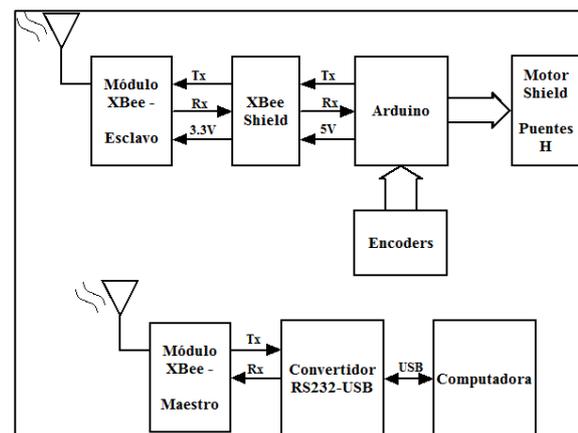


Fig. 1. Arquitectura general del sistema de control

2.2 Control de velocidad

Para lograr que el auto pueda avanzar de manera lineal es necesario ajustar la velocidad de las llantas del robot. Por ejemplo, para que el robot pueda avanzar en línea recta es necesario que ambas ruedas giren a la misma velocidad. Para esto se utilizó un control de lazo abierto que controla la velocidad de cada llanta en forma independiente. Para poder ajustar las variables de control se utilizaron dos salidas de PWM (*Pulse Width Modulator*) del sistema Arduino.

Tomando en cuenta los beneficios del concepto de modularidad, se utilizó para la etapa de potencia un *Motor Shield* especial para Arduino (Adafruit, Nueva York, EUA), el cual tiene cuatro puentes H integrados en dos circuitos electrónicos L293D. El módulo es colocado y conectado encima del Arduino como se puede observar en la Figura 2. Para el fácil manejo de estos circuitos se utilizó una librería de descarga libre (AFMotor.h) ofrecida por el fabricante del *Motor Shield* especial para Arduino. La función *setSpeed()* permite manejar por argumentos, la velocidad desde cero hasta 255, siendo éste último el valor máximo de la velocidad que los motores pueden alcanzar.

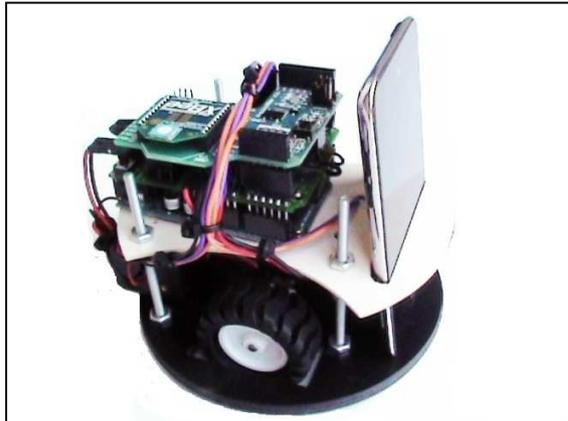


Fig. 2 Robot móvil con plataforma Arduino y un iPod para la transmisión de video inalámbrico por WiFi.

2.3 Tracción

El movimiento del robot está basado en una tracción diferencial, la cual es una configuración muy común para sistemas utilizados en interiores, debido a que permite al robot girar sobre su propio eje [1, 6]. De esta manera, el robot puede moverse en espacios congestionados con cierta facilidad. La tracción diferencial utiliza dos ruedas controladas individualmente con una rueda loca como tercer punto de apoyo. Para actuar las llantas se utilizaron motores de corriente directa con transmisión 30:1 (Pololu Corporation, Las Vegas, Nevada, EUA).

La velocidad y la posición del vehículo se pueden obtener por medio de encoders (Pololu Corporation, Las Vegas, Nevada, EUA) los cuales vienen en un *kit*, en conjunto con las llantas, especial para este tipo de aplicaciones. Los encoders tienen la capacidad de dar 48 cuentas por revolución.

El desplazamiento del robot D a lo largo de la trayectoria está determinado por la siguiente ecuación:

$$D = \frac{D_l + D_r}{2}$$

Donde:

D = Desplazamiento del robot
 D_l = Desplazamiento de la rueda izquierda
 D_r = Desplazamiento de la rueda derecha

De forma similar, la velocidad V del robot está dada por la siguiente ecuación:

$$V = \frac{V_l + V_r}{2}$$

Donde:

V = Desplazamiento del robot
 V_l = Desplazamiento de la rueda izquierda
 V_r = Desplazamiento de la rueda derecha

Así mismo, el ángulo θ se puede obtener mediante la siguiente fórmula (Fig. 3):

$$\theta = \frac{D_l - D_r}{d}$$

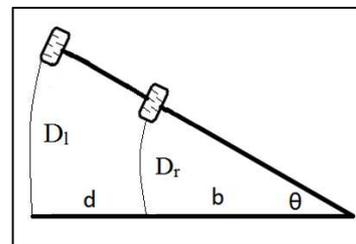


Fig. 3 Los arcos D_l y D_r son trazados por las ruedas izquierda y derecha para el cambio en el avance del robot con ángulo θ .

Note que el cambio de orientación del vehículo es función de los desplazamientos de las llantas. Cabe mencionar que la variable d en el denominador representa una fuente de error significativa debido a las incertidumbres asociadas con el punto de contacto efectivo. Estos errores se deben los desplazamientos indeseados debido a las irregularidades del piso. Adicionalmente, los diámetros de las ruedas pueden variar minúsculamente debido a que las llantas son de goma y por lo tanto son pequeñamente comprimidas por el peso. Todas estas fuentes de error hacen que el vehículo se desvíe de la trayectoria deseada.

2.4 Interface Gráfica

Para el fácil e intuitivo control a distancia del robot se diseñó en Visual Basic una interface gráfica (GUI) con varias opciones de movimiento, Fig. 4.



Fig. 4 Interface gráfica del usuario para el telecontrol del robot móvil.

El usuario puede manipular el robot presionando con el mouse los botones o también tiene la opción de manipular el robot por medio de los cursores del teclado de la computadora. La interface envía información vía puerto serial por medio de la antena XBee. Esta información se recibe en el Arduino del robot, programado para responder de acuerdo a la información que se le envía.



Fig. 5 Computadora para el telecontrol del robot móvil con interfaz gráfica para la recepción de video.

El sistema cuenta con subrutinas previamente programadas en el robot. Al seleccionar una opción el carrito comenzará inmediatamente a ejecutar la trayectoria que dibuje la figura descrita en el texto del botón presionado, las cuales son: Círculo, Cuadrado, Zigzag y Ocho. La Fig. 5 muestra el sistema de control con interface gráfica y un XBee conectado al USB de la computadora.

El robot tiene en su construcción una ranura para poder colocar un sistema de video, en este caso un iPod Touch G4. La transmisión de video lleva a cabo con la aplicación FaceTime de Apple Inc. la cual

transmite y recibe video a un alto nivel de compresión utilizando varios estándares industriales (H.264, ACC, SIP-IETF). Para que la aplicación pueda ser habilitada el iPod se debe encontrar dentro al alcance de una red inalámbrica WiFi con cifrado WEP.

2.5 Trayectorias

Las formas básicas de avance con las que cuenta el robot son la de ir hacia adelante, reversa, girar a la izquierda y a la derecha.

Se implementaron anchos de pulso PWM específicos para cada uno de los motores. Esto es debido a que los dos motores son independientes y por lo tanto no cargan la misma cantidad de peso, lo cual haría impreciso su avance de no ser por el ajuste de valores en estas funciones. Tomando en cuenta lo anterior y debido a la tracción diferencial del robot, se puede deducir que un giro sobre el eje central de rotación del carrito, es posible haciendo que una rueda avance hacia adelante y la otra hacia atrás a la misma velocidad.

El robot fue programado con cierto nivel de autonomía, la cual consiste en seguir rutinas previamente programadas en el Arduino. Estas trayectorias pueden ser modificadas fácilmente para cambiar su dimensión. Las trayectorias programadas son:

Círculo: Para lograr que el robot dibujara un círculo conforme avanzaba fue necesario implementar un PWM con mayor ancho de pulso en uno de los motores, de esta forma una rueda avanza más que la otra y se puede tener el control del diámetro deseado en el círculo, modificando los valores de la función de tracción en cada uno de los motores. De la misma manera es posible controlar la velocidad requerida.

Cuadrado: El robot logra seguir esta trayectoria mediante la implementación de rutinas de avance recto sencillo durante cierto periodo de tiempo y consecutivamente una rutina para girar noventa grados. Repitiendo el avance y el giro cuatro veces, conseguimos que el carrito regrese a la posición en la que inicio su trayectoria.

Zigzag: Esta figura se puede dibujar con ángulos de noventa grados, alternando el lado del giro, una vez que se ha implementado la rutina del cuadrado solo es necesario modificar el segundo y el cuarto giro, invirtiendo su dirección y de esta forma lograr la figura.

Ocho: Así como se utilizó de base la rutina del cuadrado para lograr la figura del Zigzag, aquí se modificó la rutina del círculo, prolongando el periodo de tiempo de movimiento del carrito e invirtiendo los valores PWM de los motores al finalizar el primer círculo, así entonces el robot comenzará a trazar en su trayectoria el segundo círculo complementario para lograr el Ocho.

3. Experimentación y resultados

La primera prueba de avance, se realizó sobre una superficie de azulejo rugoso, en esta prueba se descartó el uso de dos rollcasters. Se removió una, debido a que el robot se atoraba en dicha superficie. De esta manera logramos un avance más eficiente en cuanto a trayectoria y tracción.

Al avanzar, el robot no lograba seguir una línea recta a pesar de que los dos motores habían sido programados para funcionar con la misma tracción. Esto, debido al peso distribuido de manera no uniforme. Para corregir este problema, fue necesaria la calibración de cada uno de los motores por medio del PWM, lo cual permitió emparejar la distancia recorrida en los dos motores.

Debido a que la superficie donde se realizaron las pruebas, no era plana, se construyó una plataforma cuadrada de madera de 2x2m. Una vez sobre la plataforma, el carrito logró seguir de manera más nivelada las trayectorias programadas, disminuyó la diferencia de valores de potencia entre los motores y disminuyó la fricción entre las ruedas y la superficie donde giraban, evitando así el derrape.

El siguiente problema a resolver era que, a pesar de dar los mismos valores de PWM a ambos motores, el carrito no avanzaba en línea recta cuando se le requería. Esto se debía a que un motor hacía girar su rueda más rápido que el otro. Para resolver esto, se asignaron valores que consiguen la misma cantidad de revoluciones por segundo para ambos motores, sin importar el peso que soporten.

Una vez regulados los motores, se procedió a realizar las rutinas con los valores ya modificados para que el carrito avanzara de manera uniforme hacia adelante, hacia atrás e hiciera giros de 90 grados hacia la derecha y hacia la izquierda. Estas últimas son las rutinas básicas de las que se componen todas las demás rutinas conformadas por líneas rectas y vueltas. Para el caso de las figuras circulares, se modificó la tracción de los motores de acuerdo al

diámetro y al tiempo en que se deseaba hacer el recorrido. En las Figuras 5 y 6 se pueden apreciar en diferentes tomas la progresión de la posición del carrito. Dichas fotos fueron extraídas de un vídeo que contiene los recorridos del robot.

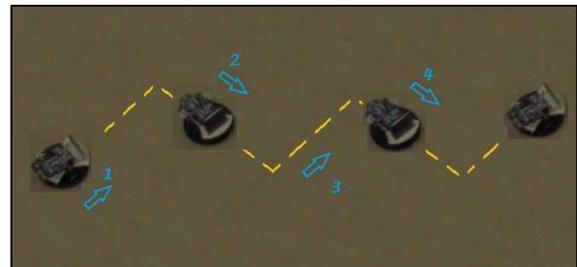


Fig. 5. Trayectoria predefinida en ZigZag la cual fue realizada autónomamente.

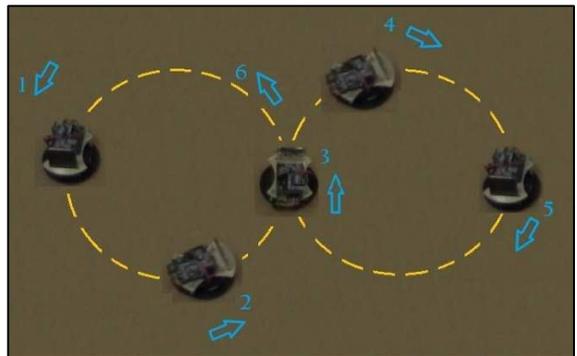


Fig. 6. Trayectoria predefinida en forma de ocho y la cual también fue realizada autónomamente.

4. Discusión

En este artículo se presentó el concepto de plataforma de control modular para el diseño ágil de un robot móvil con tracción diferencial para el desarrollo de proyectos de investigación de telecontrol e inteligencia artificial. Se describe en forma detallada el telecontrol del robot utilizando protocolo ZigBee para la transmisión inalámbrica así como la posibilidad de utilizar sistemas de visión existentes como lo es un dispositivo móvil comercial con la posibilidad de transmitir video en tiempo real dentro de una red WiFi. Haciendo uso del concepto de control modular se emplea una tarjeta comercial la cual es diseñada para mover motores de CD mediante un puente H. Además, se describe la forma de manipulación del robot así como también los beneficios del uso de la tracción diferencial. En este proyecto se utiliza una interface gráfica que permite

mover al robot por medio de botones o en su caso por medio del teclado.

El robot ha sido construido con relativa facilidad, poco tiempo y esfuerzo, y además a un precio más económico a los robots que existen en el mercado. La ventaja más visible, en relación con otros sistemas robóticos comerciales, es que el robot presentado puede ser programado en una plataforma fácil de aprendizaje, con software libre y empleando un estilo de programación que se ha hecho muy popular en los últimos años. Además, el concepto modular permite que se pueda migrar con cierta facilidad a otro tipo de plataformas con más capacidad de procesamiento como pudiera ser un Arduino Mega, Netduino, FEZ Panda II, entre otras.

En esta fase de la investigación el carrito aún no realiza continuamente las trayectorias con el mismo tamaño y se llegar a desviar un ligeramente de la trayectoria deseada. Lo anterior se puede ver cuando el carrito se utiliza en días diferentes, lo cual requiere una calibración del PWM. Esto se debe a que se utiliza control de lazo abierto para realizar el control de la velocidad. Se tiene planeado emplear los encoders para poder hacer control de lazo cerrado utilizando un control tipo proporcional/integral.

Actualmente se están desarrollando más proyectos utilizando los conceptos antes descritos [7, 8], como por ejemplo un robot autónomo para encontrar salidas a laberintos. En este prototipo solo se le agregaron unos sensores ultrasónicos y simplemente se quitó la parte de radio control. Como trabajo futuro se tiene contemplado realizar tareas de sincronización entre más robots construidos bajo el mismo concepto.

5. Referencias

- [1] HR Everett (1995), *Sensors for mobile robots, theory and application*, A K Peters, Ltd., ISBN 1-56881-048-2
- [2] JC Moreno, M Berenguel, F Rodríguez, JF Sarabia, R Garrote, JL Guzmán, O López (2001), *Proyecto de Aplicación de Telerobótica a un Minirobot Móvil*, EIWISA'01, Madrid, 2001.
- [3] www.arduino.cc (última revisión 24 agosto dl 2011).
- [4] Faludi Robert (2011), *Building Wireless Sensor Networks, A Practical Guide to the*

ZigBee Mesh Networking Protocol, O'REILLY, ISBN: 978-0-596-80773-3.

- [5] Shahin Farahani (2008), *ZigBee Wireless Networks and Tranceivers*, Newnes, Elsevier, ISBN: 978-0-7506-8393-7.
- [6] Jesús Manuel Gómez De Gabriel (2006), *Teleoperación y telerobótica*, Prentice Hall, ISBN: 8483222965
- [7] UG Villaseñor-Carrillo, MA González-Aguirre, A Sotomayor-Olmedo, E Gorrostieta-Hurtado, JC Pedraza-Ortega, JE Vargas-Soto, S Tovar-Arriaga (2010), "Desarrollo de un sistema de navegación para robots móviles mediante diferentes patrones de comportamientos", CIINDET 2010, Cuernavaca, Morelos, México.
- [8] UG Villaseñor-Carrillo, A Sotomayor-Olmedo, E Gorrostieta-Hurtado, JC Pedraza-Ortega, MA Aceves-Fernández, S Tovar-Arriaga (2010), *Implementation of Different Pattern Behaviors for Mobile Robots using Reduced Algorithms based on Fuzzy Techniques*, 7th CIIE 2010