

DISEÑO Y DESARROLLO DE UN ROBOT OMNIDIRECCIONAL PARA APLICACIONES COLABORATIVAS

Edgar Granados Osegueda
Department of Digital Systems
ITAM

Rio Hondo 1, Ciudad de México, 01080
edgar.granados@itam.mx

Marco Morales Aguirre
Department of Digital Systems
ITAM

Rio Hondo 1, Ciudad de México, 01080
marco.morales@itam.mx

José Guadalupe Romero Velazquez
Department of Digital Systems
ITAM

Rio Hondo 1, Ciudad de México, 01080
jose.romerovelazquez@itam.mx

RESUMEN

Este artículo describe el diseño de un robot móvil omnidireccional para Robocup Small Size League, así como su construcción. Igualmente, se describen los algoritmos utilizados para lograr el movimiento omnidireccional así como el algoritmo de control utilizado. Se detalla la arquitectura del sistema, en el cual se utiliza ROS. Como resultados, se muestra el movimiento obtenido con la solución.

INTRODUCCIÓN

Con la finalidad de contar con un sistema para realizar investigación en robótica colaborativa multi-agente, se presenta el diseño de un robot móvil omnidireccional. Se seleccionó éste tipo de robot debido a las características de movimiento que ofrece. El diseño es modular para favorecer la adaptación a distintos escenarios de prueba. El sistema se compone de un equipo homogéneo de robots omnidireccionales y un controlador implementado con ROS para facilitar la modularidad y el mantenimiento del código. Con la finalidad de acotar el problema de diseño físico además de tener una plataforma de prueba, se toma como base las especificaciones y plataforma de Robocup Small Size League.

MARCO DE REFERENCIA

Un robot omnidireccional es aquel que puede moverse de un punto A a un punto B en línea recta mientras gira sobre

su propio eje para llegar en una orientación dada [1]. Para obtener un movimiento omnidireccional con un robot de ruedas concéntricas, es necesario contar con por lo menos 3 ruedas omnidireccionales colocadas con distintos ángulos de separación [2]. Un número de ruedas adicional proporciona redundancia, siendo 4 ruedas el diseño mas popular y el seleccionado.

Específicamente, un robot de SSL debe caber dentro de un cilindro de 18 cm de altura por 15 cm de diámetro, siendo el diseño más eficiente (y el utilizado por todos los equipos) un robot cilíndrico con ruedas concéntricas. La liga cuenta con un sistema de visión el cual da la posición de cada robot y la pelota utilizada, para lo cual el robot debe cumplir con el *patrón estándar* en la parte superior [3]. Igualmente, es necesario contar con mecanismos para la pelota, los cuales son los encargados de mantener la pelota cerca del robot (*dribbler*) y poder patear *kickers*.

ARQUITECTURA DEL SISTEMA

La arquitectura propuesta se muestra en la Fig. 1. Se utiliza la visión proporcionada por SSL, la cual reporta la pose de cada robot. Para los distintos componentes que se implementan se utiliza ROS. La utilización de ROS permite tener un sistema modular, facilitando la interacción y reemplazo de los diversos componentes. Aunque la arquitectura presentada está pensada en Robocup SSL, es posible reemplazar componentes para enfocar el sistema a otro problema.

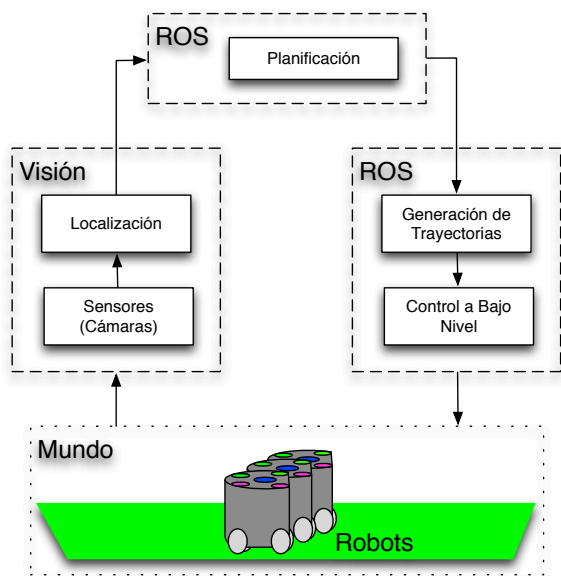


FIGURE 1. Diagrama del Sistema de ROS implementado

VISIÓN

Este es un componente proporcionado por SSL. El componente utiliza las imágenes obtenidas de dos cámaras colocadas en la parte superior de la cancha, las procesa y obtiene la posición y orientación de cada robot así como la posición de la pelota. El componente no está implementando en ROS pero sus salidas son adaptadas a ROS para poder utilizarlo.

PLANIFICACIÓN

Este componente es el encargado de realizar la planeación de tareas para todos los robots del equipo que se encuentren en el juego. Obtiene las posiciones de todos los elementos del juego de la visión así como las posiciones esperadas en un futuro de cada nodo de trayectorias.

TRAYECTORIAS

Cada robot cuenta con su nodo de trayectorias. Éste nodo recibe la posición actual, la deseada y el tiempo deseado del nodo de estrategias y se encarga de generar la trayectoria necesaria. De no ser posible generar la trayectoria deseada en el tiempo deseado, se reporta al nodo de Estrategias.

COMUNICACIÓN

Este nodo recibe la velocidad del robot deseada en cada ciclo y la envía al robot.

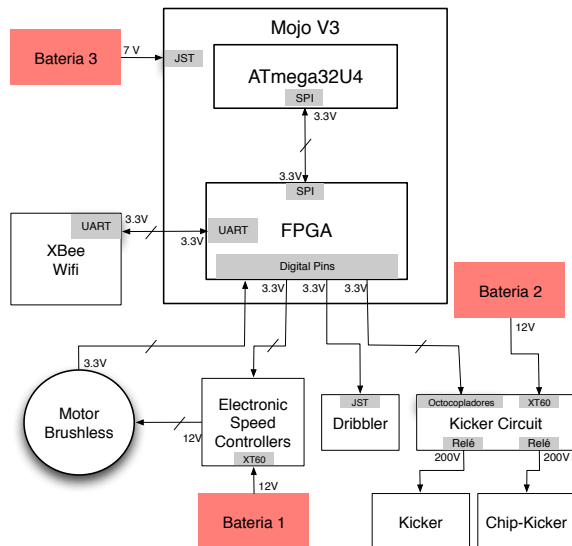


FIGURE 2. Componentes de Electrónica

ESTRUCTURA DEL ROBOT

Realizando un análisis de los robots de la liga, se determinó que la velocidad máxima deseada del robot se estableciera en 3.5 m/s. Igualmente, la carga que el robot debe soportar se estableció en 2.5 Kg. Debido a la velocidad del juego, es común que ocurran colisiones entre ellos por lo que el exterior del robot debe ser de un material resistente además contar con piezas fácilmente reemplazables.

Para poder realizar numerosas pruebas de las distintas partes que conforman al robot, se optó por utilizar el modelado por deposición fundida (impresión 3D). Adicionalmente, este método permite realizar diseños más complejos además de probarlos y validarlos rápidamente. Se utilizaron dos tipos de plástico: ABS y HIPS. Se realizó un diseño modular para facilitar futuras mejoras, así como volverlo fácilmente adaptable a otros escenarios.

La electrónica del robot se compone de una tarjeta de desarrollo MOJO la cual tiene una FPGA Spartan 6 XC6SLX9 y un microcontrolador ATmega32U4. Para la comunicación, se utiliza una tarjeta XBee WiFi. Para el control de los motores, se utiliza un ESC (Electronic Speed Controller) por motor. La interacción de los componentes se muestra en 2.

CARCASA Y BASE

La base es la pieza en la cual se ensamblan el resto, mientras que la carcasa protege al robot de posibles impactos durante su utilización. Estas piezas constituyen las piezas de mayores dimensiones en el diseño por lo que son manufacturadas usando plástico HIPS al ofrecer mejores resultados en piezas de grandes dimensiones. La carcasa está formada por 4 piezas únicas. Como parte de la carcasa se incluye el patrón necesario para que la

visión de la liga proporcione la posición del robot.

ENERGÍA

Los componentes que requieren de electricidad para funcionar son tres: mecanismos para el movimiento, para la pelota y el cómputo. Cada uno utiliza su propia batería LiPo para tenerlos aislados eléctricamente entre si, para la comunicación entre ellos se utilizan optocopladores. Los mecanismos para el movimiento, motores y ESC son alimentados por una batería de 12 V. Los mecanismos para la pelota, Kickers y dribbler, son alimentados por una batería de 12 V. El componente de cómputo es alimentado por una batería de 7 V.

MECANISMOS PARA MOVIMIENTO DEL ROBOT

RUEDA La mayoría de los diseños de ruedas omnidireccionales comerciales están enfocados en aumentar la eficiencia de la rueda buscando que el perfil de la rueda se perfectamente circular ([1]), sacrificando el espacio utilizado por la rueda. Debido a las restricciones en las dimensiones del robot, se realizó un diseño propio. La rueda omnidireccional más común dentro de la liga consiste en una rueda con rodillos perpendiculares al eje principal.

Los factores más importantes en el diseño de la rueda es la facilidad de ser ensamblada y la resistencia del material utilizado. El diseño final utilizado utiliza cuatro piezas únicas, de las cuales 2 son de diseño propio: una pieza de plástico ABS (3) y 21 rodillos de aluminio; y dos piezas comerciales: 1 cable de acero y 21 orings.

Como solamente una pieza conforma el esqueleto de la rueda, se simplifica el ensablado además de hacer ser más resistente comparado con otras versiones que utilizan varias piezas como esqueleto. Adicionalmente, el radio de la rueda así como el número de rodillos utilizados favorece un funcionamiento eficiente de la rueda comparado con versiones de menor tamaño o menor número de rodillos.

REDUCTOR El motor utilizado es Maxon 200142. Este motor tiene una velocidad sin carga de 4370 RPM (nominal de 2940 RPM). Dados los requerimientos de velocidad y peso establecidos, es necesario utilizar un reductor de velocidad de 2.5. Se utiliza un reductor conformado por engranes cilíndricos rectos de metal. Se utilizan engranes comerciales debido a la necesidad de contar con alta eficiencia en la transmisión de la energía además de tener que ser resistentes en altas velocidades.

ELECTRONIC SPEED CONTROLLER (ESC) Se utiliza un ESC por motor que sirve de interfaz entre la FPGA que envía una señal PWM y el motor. Este componente permite aislar eléctricamente la FPGA (que utiliza voltaje TTL) de los motores (que utilizan 12V). Se utiliza un modelo de ESC dedicado

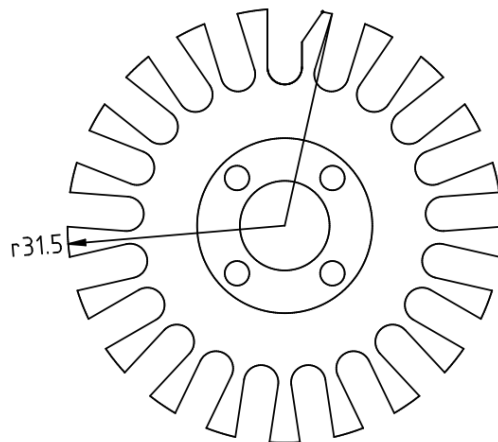


FIGURE 3. Diagrama de la Rueda Omnidireccional Utilizada

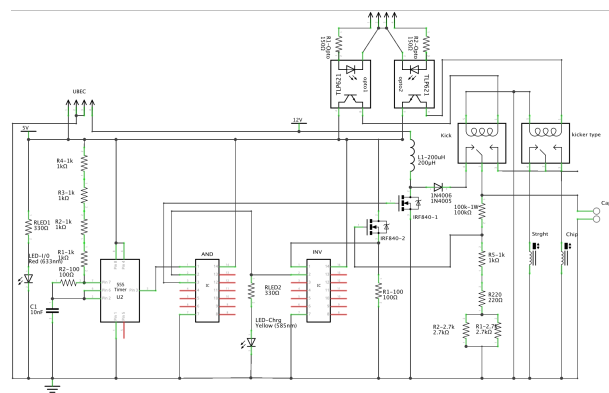


FIGURE 4. Circuito para el Kicker

a drones multihélice debido a la rapidez de respuesta que estos ofrecen y que resulta necesaria para el correcto funcionamiento de cada motor.

MECANISMOS PARA LA PELOTA

En esta sección se presentan los mecanismos diseñados para tener control de la pelota (*dribbler*) así como el mecanismo de pateo (*Kicker*).

KICKERS Se utiliza un seleonoide el cual es alimentado con 200 V cuando se da la señal para patear. En la Fig. 4 se muestra el circuito utilizado para llevar la entrada de 12 V a una salida de 200 V. Dentro del circuito, se utiliza un *Universal Battery Elimination Circuit (UBEC)* para alimentar con 5 V los circuitos integrados a partir de los 12 V de la batería.

DRIBBLER El mecanismo del dribbler consiste en un motor DC que funciona entre 3 y 5 V así como un sistema de engranes de plástico para la transmisión de energía y una espuma en el punto de contacto con la pelota. Para el control del dribbler se utiliza un Puente H en modo unidireccional.

CÓMPUTO

En esta sección se detallan los componentes de cómputo utilizados. Estos componentes mandan las señales necesarias a cada actuador además de recibir las señales de los sensores utilizados. Realizan los cálculos necesarios para el movimiento así como la comunicación con el resto del sistema.

FPGA Debido a que es necesario controlar los 4 motores así como otros componentes, es necesario contar con la capacidad de atender múltiples entradas y salidas al mismo tiempo. Una FPGA ofrece esta capacidad además de tener mayor resolución respecto a un microcontrolador. En el sistema implementado, la FPGA tiene seis funciones:

1. Recibir los datos del XBee.
2. Calcular la velocidad de cada motor mediante la señal de cada sensor hall.
3. Mandar la velocidad deseada a cada ESC mediante PWM.
4. Activar el dribbler.
5. Activar el kicker.
6. Recibir y procesar la señal de sensores adicionales.

La FPGA se comunica con el Microcontrolador mediante SPI y Memory Mapping. Envía las velocidades deseadas X, Y, ω así como las velocidades reales de cada motor y recibe las velocidades deseadas para cada motor.

MICROCONTROLADOR En cada ciclo del microcontrolador, se reciben las velocidades deseadas X, Y, ω así como las reales de cada motor y se determina una nueva velocidad deseada de motor mediante el modelo omnidireccional así como aplicando un algoritmo de control PI. Las velocidades por motor calculadas son transferidas a la FPGA.

COMUNICACIÓN Para el control del robot es necesario contar con comunicación inalámbrica (únicamente esta prohibido usar bluetooth). Se optó por utilizar Wifi debido a la gran adopción que tiene esta tecnología permitiendo utilizar prácticamente cualquier equipo de cómputo para el control del robot. En específico, se utiliza XBee Wifi por su tamaño y utilizar voltaje TTL siendo compatible con la FPGA. Se utiliza direccionamiento estático para tener control de las IPs de cada robot, además de facilitar la conexión inicial con el access point. Para favorecer un diseño modular, el resto de los componentes son agnósticos a la tecnología utilizada para la comunicación.

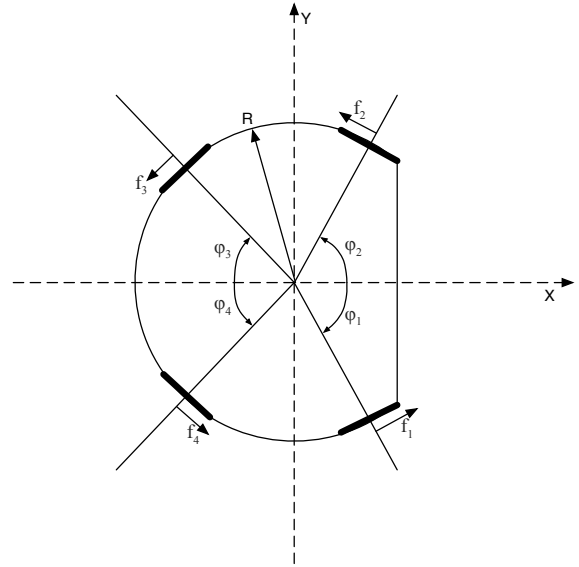


FIGURE 5. Distribución de Ángulos y Fuerzas

MOVIMIENTO OMNIDIRECCIONAL

A partir de un movimiento deseado, expresado en el vector de velocidades deseadas del robot $V_l = (v_x \ v_y \ \omega)^T$, es necesario descomponerlo en velocidades de motores deseadas $V_m = (m_1 \ m_2 \ m_3 \ m_4)^T$.

A partir de la ecuación general de fuerza Eqn. (1) se puede derivar la Eqn. (2) específica para el caso de cuatro motores. Debido a las ruedas omnidireccionales, cada motor aporta en los componentes de X y Y. Utilizando el diagrama de la Fig. 5, es posible derivar la Eqn. (3) para el motor 1 en X y Eqn. (4) para el motor 1 en Y. Similarmente, se pueden obtener las ecuaciones para los otros motores.

$$F = Ma \quad (1)$$

$$a = \frac{1}{M} (F_1 + F_2 + F_3 + F_4) \quad (2)$$

$$Ma_{1x} = |f_1| \cos(90 - \phi_1) = |f_1| \sin(\phi_1) \quad (3)$$

$$Ma_{1y} = |f_1| \cos(\phi_1) \quad (4)$$

Para obtener la aceleración angular que aporta cada motor al robot, de la Eqn. (5) donde R es el radio de acuerdo a la Fig. 3

se puede derivar la Eqn. (6) específica para el uso de 4 motores. Sustituyendo el momento de inercia Eqn. (7) para el caso de un cilindro con distribución de masa desconocida en (6) se obtiene la Eqn. (8).

$$\dot{\omega} = \frac{Rf}{I} \quad (5)$$

$$\dot{\omega} = \frac{R}{I} (f_1 + f_2 + f_3 + f_4) \quad (6)$$

$$I = \alpha MR^2; \quad 0 \leq \alpha \leq 1 \quad (7)$$

$$R\dot{\omega} = \frac{1}{M\alpha} (f_1 + f_2 + f_3 + f_4) \quad (8)$$

A partir de los resultados obtenidos de las aceleraciones traslacionales y angulares, se puede derivar la Eqn. (9). También se puede expresar como en la Eqn. (10), donde C_α se conoce como la Matriz de Acoplamiento de Fuerzas.

$$\begin{pmatrix} a_x \\ a_y \\ R\dot{\omega} \end{pmatrix} = \frac{1}{M} \begin{bmatrix} \sin \varphi_1 & -\sin \varphi_2 & -\sin \varphi_3 & \sin \varphi_4 \\ \cos \varphi_1 & \cos \varphi_2 & -\cos \varphi_3 & -\cos \varphi_4 \\ \frac{1}{\alpha} & \frac{1}{\alpha} & \frac{1}{\alpha} & \frac{1}{\alpha} \end{bmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} \quad (9)$$

$$a = C_\alpha F \quad (10)$$

Para poder realizar la transformación de la velocidad deseada en el espacio a velocidades de cada motor, es necesario considerar tanto el perímetro de la rueda como el factor de reducción mediante la Eqn. (11). A partir de esta, se pueden obtener las ecuaciones de cada motor definiendo el vector $v_m = (v_1 \ v_2 \ v_3 \ v_4)^T$ mediante la Eqn. (12) y su forma reducida, la Eqn. (13) donde a D se le conoce como la Matriz de Acoplamiento de Velocidades.

$$V_l' = \frac{V_l \cdot e}{2\pi r} \quad (11)$$

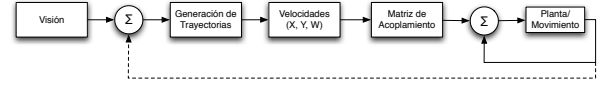


FIGURE 6. Esquema de Control Implementado

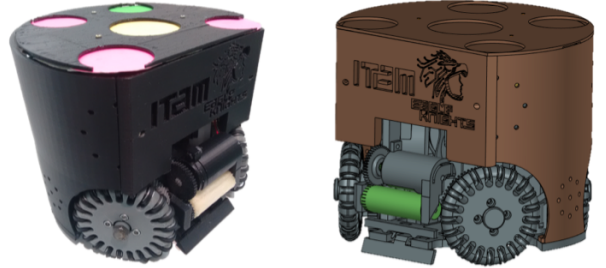


FIGURE 7. Robot Real vs Modelo 3D del Robot

$$v_m = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{bmatrix} \sin \varphi_1 & \cos \varphi_1 & 1 \\ -\sin \varphi_2 & \cos \varphi_2 & 1 \\ -\sin \varphi_3 & -\cos \varphi_3 & 1 \\ \sin \varphi_4 & -\cos \varphi_4 & 1 \end{bmatrix} \begin{pmatrix} v_x' \\ v_y' \\ R\omega' \end{pmatrix} \quad (12)$$

$$v_m = DV_l' \quad (13)$$

CONTROL

Para el control a bajo nivel, utilizando la retroalimentación que otorga cada motor, se implementó un control PI a nivel motor. El esquema general del algoritmo de control utilizado se muestra en la Fig. 6. La sintonización de las variables se realiza a mano siguiendo el método de Ziegler - Nicholson.

Adicional al control PI por motor, se cuenta con la retroalimentación de la visión gracias a la cual se puede calcular y corregir la trayectoria del robot al enviar continuamente actualizaciones de la velocidad deseada (30 Hz).

EXPERIMENTACIÓN y RESULTADOS

En la Fig. 8 se puede ver la respuesta de un motor ante una velocidad deseada con el algoritmo de control PI implementado. Aunque las oscilaciones no se eliminan, se reducen rápidamente, siendo la respuesta rápida uno de los factores más importantes debido al dinamismo del sistema. Igualmente, el motor es capaz de cambiar de velocidad, incluso a velocidades negativas (dirección contraria) rápidamente.

Estableciendo 8 direcciones en el plano XY, se probó la respuesta del robot ante cada dirección con control PI sin visión y con visión. En la Fig. 9 se pueden ver los resultados de

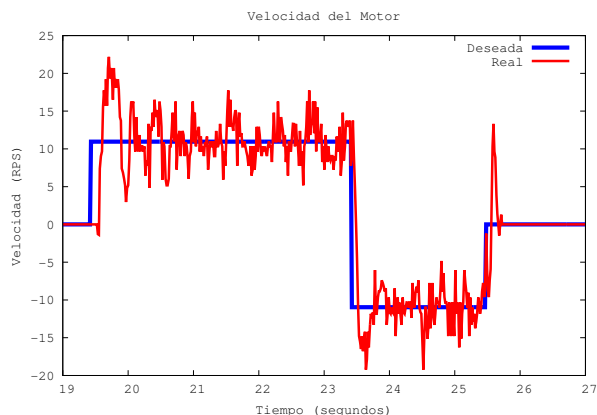


FIGURE 8. Velocidad real vs Velocidad deseada

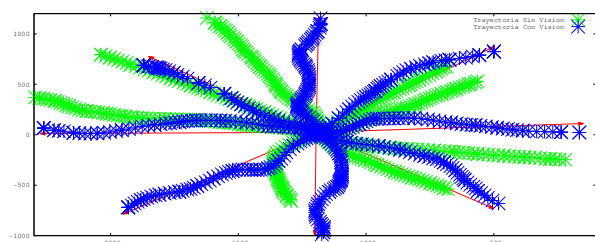


FIGURE 9. Trayectorias con retroalimentación

estas pruebas. Para el caso del robot sin retroalimentación de visión, existen algunas direcciones en las cuales el movimiento y su posición final es cercano al deseado, específicamente en las direcciones en que prácticamente no se utilizan dos ruedas. En cambio en las otras direcciones, se tiene mayor error en el movimiento del robot. Para el caso del movimiento con retroalimentación de visión, aunque en algunos casos existe un movimiento errático, la posición final es muy cercana a la deseada gracias a la corrección que realiza el sistema ante los errores en el movimiento del robot.

CONCLUSIÓN

Con el diseño propuesto se logra que el robot se mueva aproximadamente a la velocidad propuesta, manteniendo la integridad de sus componentes y con un peso menor que el contemplado inicialmente, aunque siendo capaz de cargar los 2.5 Kg propuestos. Tanto la carcasa como la base y las ruedas han sido capaces de resistir golpes entre los robots a las velocidades normales de movimiento y la visión reconoce el patrón estándar incorporado en la carcasa. En la Fig. 7 se muestra una comparación entre el diseño modelado en 3D y la construcción final del robot.

La utilización de ROS para el sistema así como el funcionamiento de la arquitectura propuesta ha sido satisfactorio.

La modularidad entre sus componentes permite intercambiarlos si se requiere una función distinta sin afectar el comportamiento del resto del sistema.

Como trabajo futuro se propone un método de sintonización automático para el sistema de control. Adicionalmente, es necesario implementar un algoritmo de evasión de obstáculos así como un módulo asignación de tareas.

ACKNOWLEDGMENT

Gracias al ITAM, específicamente al Departamento de Sistemas Digitales por el apoyo brindado al Laboratorio de Robótica.

REFERENCES

- [1] Rojas, R. A Short History of Omnidirectional Wheels. <http://ftp.inf.fu-berlin.de/pub/Rojas/omniwheel/shortomni.pdf>.
- [2] Rojas, R., and Förster, G., 2006. "Holonomic control of a robot with an omnidirectional drive." *Künstliche Intelligenz*, Böttcher IIT Verlag.
- [3] Committee, S. S. L. T., 2016. Laws of the robocup small size league 2016. http://wiki.robocup.org/images/1/18/Small_Size_League_-_Rules_2016.pdf, April.