

## DISEÑO Y DESARROLLO DE UN ROBOT OMNIDIRECCIONAL PARA APLICACIONES COLABORATIVAS\*

**Edgar Granados Osegueda**

Departamento de Sistemas Digitales  
ITAM

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

**Marco Morales**

Departamento de Sistemas Digitales  
ITAM

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

**José Guadalupe Romero**

Departamento de Sistemas Digitales  
ITAM

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

### RESUMEN

*En este artículo se presenta el diseño y construcción de un robot móvil omnidireccional para aplicaciones colaborativas. El modelo cinemático que describe el movimiento omnidireccional del robot es implementado con un algoritmo de control a bordo del robot en una tarjeta de desarrollo MOJO la cual cuenta con una FPGA Spartan 6 XC6SLX9 y un microcontrolador ATmega32U4 y con un programa externo al que se accede desde el Robot Operating System (ROS). Los experimentos realizados ilustran el buen comportamiento del sistema diseñado.*

### INTRODUCCIÓN

Actualmente, los robots móviles son utilizados en un amplio rango de actividades, como exploración (e.g., DEPTHX en exploración acuática, y MER de la Nasa en la exploración de Marte) y transporte. Existen diversos tipos de robots móviles incluyendo los humanoides (e.g, el Nao), los voladores y los que se mueven con ruedas. Estos últimos tienen una estructura mecánica y modelos relativamente sencillos [1]. Por esto, son una muy buena opción para realizar tareas sobre pisos planos. Un caso específico de este tipo de robots es el robot omnidireccional que puede alcanzar cualquier pose en el plano sin realizar maniobras auxiliares. Esto lo hace muy útil en ambientes dinámicos con poco espacio para maniobrar o que requieren

cambios abruptos de dirección como fábricas, hospitales y almacenes. Asimismo, en diversas tareas el trabajo colaborativo de múltiples robots tiene beneficios tales como la distribución de objetos en inventarios o la supervisión o vigilancia en entornos grandes.

En este trabajo presentamos un robot con el que podamos implementar soluciones para aplicaciones colaborativas tanto para investigación como para enseñanza. Nuestro diseño requiere ser versátil para adaptarse fácilmente por lo que un diseño modular es preferible. Una de las aplicaciones específicas de nuestro robot es en la RoboCup Small Size League, por lo que se toman las especificaciones físicas de ésta liga como marco de referencia para el diseño del robot.

Describimos a detalle la arquitectura del sistema propuesto. Presentamos la estructura del robot, incluyendo los mecanismos para movimiento y las como el cómputo interno. El cómputo externo al robot está implementando en ROS para aprovechar la modularidad que éste ofrece, mientras que el cómputo interno utiliza una FPGA Spartan 6 XC6SLX9 y un microcontrolador ATmega32U4. En la sección de algoritmos se describen el modelo cinemático del robot y el algoritmo de control utilizado. Para validar el modelo cinemático y el algoritmo de control, se realizaron experimentos de seguimiento de trayectorias que mostraron un desempeño adecuado.

---

\*SE AGRADECE EL PATROCINIO A ESTA INVESTIGACIÓN DE LA ASOCIACIÓN MEXICANA DE CULTURA A.C.

## MARCO DE REFERENCIA

Un robot omnidireccional se puede mover en un plano entre cualesquiera dos poses arbitrarias  $A$  y  $B$  trasladándose sobre una línea recta mientras gira sobre su propio eje [2]. Las poses  $A$  y  $B$  se caracterizan por una posición  $(x,y)$  y una orientación  $\theta$  con respecto a un sistema de referencia. Se puede lograr movimiento omnidireccional con un robot de ruedas concéntricas que tenga al menos 3 ruedas omnidireccionales colocadas con distintos ángulos de separación [3]. Ruedas adicionales proporcionan redundancia que permite cierta tolerancia a fallas por lo que el robot de 4 ruedas es muy popular.

Existen diversos diseños de rueda omnidireccional. Se pueden clasificar en ortogonales y universales [4]. Las primeras fueron utilizadas brevemente en Robocup SSL por el equipo de la Universidad de Cornell [5] aunque dejaron de utilizarlas debido a que no daban la tracción necesaria para un control adecuado a altas velocidades sustituyéndolas por ruedas universales [6]. Un diseño muy popular es la rueda Mecanum, conformada por rodillos en un ángulo en la periferia de la rueda. Sus principales limitaciones son el diseño complejo (conceptual y físico) así como el gran espacio que ocupan los rodillos. Otro diseño igualmente popular cuenta con rodillos alineados al eje principal de la rueda. La ventaja de ésta rueda radica en su diseño compacto aunque el perfil de la rueda suele no ser perfecto debido a los rodillos. **TODO: MM: perfecto no me parece muy adecuado, ¿a que te refieres exactamente?**

La principal ventaja de los robots omnidireccionales es su maniobrabilidad. Algunos robots que aprovechan esto son: OmniBot de la NASA, el Airtrax ATX-3000 y la OMNI-chair. El primero fue desarrollado como un robot altamente maniobrable para ambientes peligrosos [7]. El segundo es un levantacargas para trabajo en almacenes [8]. El último es una silla de ruedas omnidireccional diseñada para darle mayor capacidad de movimiento a su usuario [9].

El robot presentado en este trabajo se probó inicialmente en el contexto de la *Small Size League* de la RoboCup [?], o SSL, que utiliza robots omnidireccionales como jugadores de fútbol. Los robots deben caber en un cilindro de 18 cm de altura por 15 cm de diámetro y deben moverse sin dañar la carpeta que forma la cancha. La pose de los robots se obtiene con un sistema de visión integrado por cámaras en la parte superior de la cancha además del software provisto por la liga para el procesamiento de imágenes. Esto requiere que los robots tengan un *Patrón Estándar* de colores en su parte superior. En esta competencia, la mayoría de los equipos utiliza un control de velocidad, sin embargo, en 2011, SKUBA presentó un control basado en el torque del motor [10].

Lograr un movimiento correcto en un robot omnidireccional presenta diversos retos. Al utilizar por lo menos tres motores resulta esencial implementar un algoritmo de control eficiente. **TODO: MM: podrías mencionar un par de retos mas?**

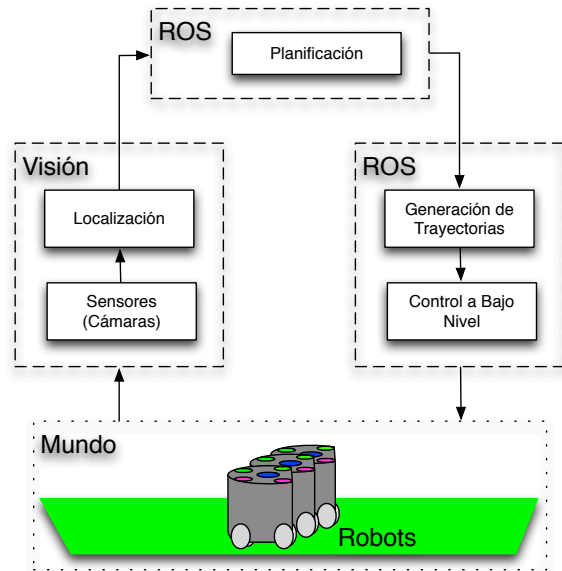


FIGURE 1. Arquitectura del Sistema

## 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. Los componentes se comunican mediante el Robot Operating System (ROS) que permite que el sistema sea modular, facilitando la interacción y reemplazo de los diversos componentes. Aunque la arquitectura presentada es aplicada inicialmente a Robocup SSL, es posible reemplazar componentes para aplicar el sistema a otros problemas. A continuación describimos los componentes de nuestro sistema.

**TODO: MM: si te da tiempo cambia las etiquetas ROS de los módulos de arriba y de la derecha por Planificación (la primera) y Control la Segunda**

### Visión

Este componente proporcionado por RoboCup SSL procesa las imágenes obtenidas de dos cámaras colocadas en la parte superior de la cancha para obtener la pose de cada robot así como la posición de la pelota. Como el componente no está implementando en ROS, se implementó una interfaz para ROS que permite comunicarlo con los demás componentes.

### Planificación

Este componente es el encargado de realizar la planificación de tareas y de movimientos para todos los robots de un equipo. La planificación de tareas se encarga de asignar poses meta a cada robot en base a un rol que tengan asignado. La planificación de movimientos identifica rutas libres de colisión para que cada

robot llegue a la pose asignada por el planificador de tareas.

## Control

El control está dividido en un generador de trayectorias y en un controlador de bajo nivel. Las trayectorias se generan para mover al robot de su pose actual a la deseada en el tiempo establecido por el planificador. De no ser posible generar la trayectoria deseada en el tiempo deseado, se reporta al planificador. La trayectoria resultante se discretiza y se envía al robot en forma de un perfil de velocidades en  $x$ ,  $y$  y  $\theta$ .

## Partes físicas del robot

Las partes físicas del robot incluyen su estructura interna, la electrónica de control y comunicaciones, los motores y las ruedas.

**Estructura** La estructura del robot está limitada por las características definidas por la *Small Size League* de la RoboCup. Debe caber dentro de un cilindro de 18 cm de altura por 15 cm de diámetro, siendo el diseño más eficiente un robot cilíndrico con ruedas concéntricas. Realizando un análisis de los robots de la liga, se estableció la velocidad máxima en 3.5 m/s. Igualmente, la carga que debe soportar se estableció en 2.5 Kg. Por otro lado, 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. Por esto se optó por utilizar el modelado por deposición fundida (impresión 3D) con la ventaja de que este método permite realizar diseños complejos que se pueden probar rápidamente. Se utilizaron dos tipos de plástico: ABS y HIPS. El diseño modular resultante facilita adaptarlo a otros escenarios así como hacer mejoras.

Los principales componentes de la estructura son la base y la carcasa. La base es la pieza en la cual se ensambla el resto de los componentes, mientras que la carcasa protege al robot de impactos. 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. En la parte superior de la carcasa se reservó un espacio donde se coloca el patrón estándar necesario para que la visión detecte la pose del robot.

**Electrónica** Como se observa en la Figura 2, la electrónica del robot se compone de una tarjeta de desarrollo MOJO, un sistema de comunicaciones y circuitos de fuente de energía y potencia para los motores. La tarjeta MOJO cual tiene una FPGA Spartan 6 XC6SLX9 descrita en la Sección y un microcontrolador ATmega32U4 que se describe en la Sección . Para las comunicaciones se utiliza una tarjeta XBee WiFi como se describe en la

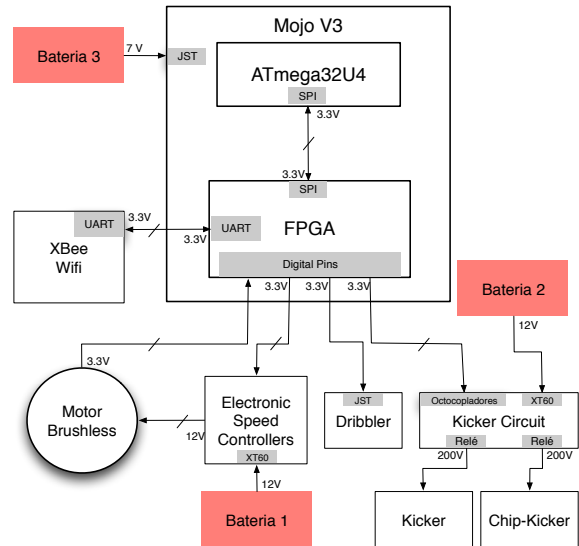


FIGURE 2. Componentes electrónicos

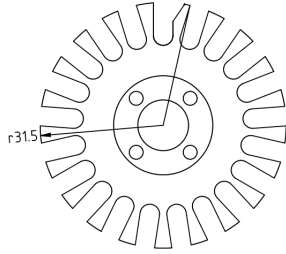
## Sección .

Para dar potencia a cada motor se utiliza un ESC (Electronic Speed Controller) 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 (12V) y transformar la señal de CD que proviene del PWM a una señal trifásica que requieren los motores sin escobillas utilizados. Se utiliza un modelo de ESC dedicado a drones multi-hélice debido a su rapidez de respuesta.

Los mecanismos para el movimiento, motores y ESC son alimentados por una batería de 12 V. 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 sí, para la comunicación entre ellos se utilizan optocopladores.

**Motores y reductores** 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 utilizó 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 y ser resistentes en altas velocidades.

**Ruedas** La mayoría de los diseños de ruedas omnidireccionales comerciales maximizan la eficiencia de la rueda buscando que el perfil de la rueda se perfectamente circular ([2]), sacrificando el espacio utilizado por la rueda. Debido a las re-



**FIGURE 3.** Diagrama de la Rueda Omnidireccional Utilizada

stricciones en las dimensiones del robot, se realizó un diseño propio, delgado y con rodillos perpendiculares al eje principal.

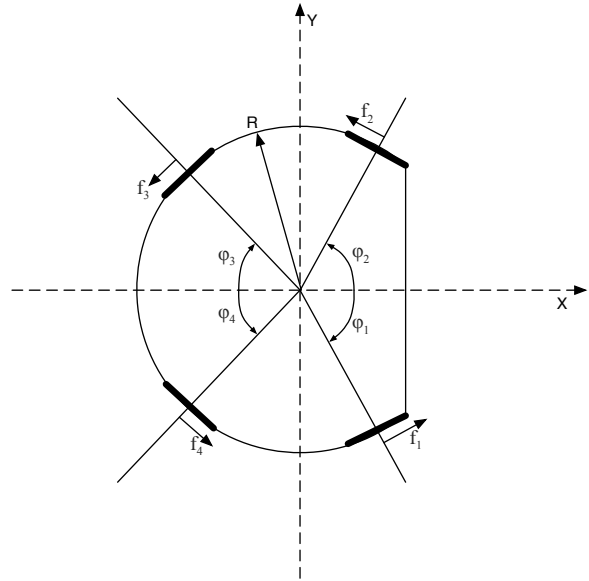
Los factores más importantes en el diseño de la rueda es la facilidad de ser ensamblada y la resistencia del material utilizado. El diseño final utiliza cuatro piezas únicas, de las cuales 2 son de diseño propio: una pieza de plástico ABS (3) y 21 rodillos de aluminio; y dos piezas comerciales: 1 cable de acero y 21 orings. De esta manera, el esqueleto es formado por una pieza lo cual simplifica el ensamblado y la hace más resistente que 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.

**FPGA** Debido a que es necesario controlar los 4 motores así como otros componentes, es necesario atender simultáneamente múltiples entradas y salidas. Una FPGA ofrece esta capacidad además de permitir representar números con mayor resolución que 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 señales de sensores adicionales.

La FPGA se comunica con el Microcontrolador mediante SPI y Memory Mapping. Envía las velocidades deseadas ( $V_x, V_y, \omega$ ) 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 ( $V_x, V_y, \omega$ ) así como las reales de cada motor y se determina una nueva velocidad deseada mediante el modelo omnidireccional. Las velocidades por motor calculadas son transferidas a la FPGA. El modelo cinemático del robot así como el algoritmo de control son implementados en el microcontrolador como se describe en la Sección .



**FIGURE 4.** Distribución de Ángulos y Fuerzas

**Comunicación** Para el control del robot es necesario contar con comunicación inalámbrica. Se eligió WiFi porque se puede utilizar prácticamente cualquier equipo de cómputo para el control del robot. En específico, se utiliza XBee WiFi por su tamaño con voltajes TTL compatibles 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.

## ALGORITMOS

En esta sección se describen los algoritmos implementados en el robot para su control de bajo nivel

### 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.  $F = Ma$  se puede derivar la Eqn. (1) 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. 4, es posible derivar la Eqn. (2) para el motor 1 en X y Eqn. (3) para el motor 1 en Y. Similarmente, se pueden obtener las ecuaciones para los otros motores.

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

$$Ma_{1x} = |f_1| \cos(90 - \varphi_1) = |f_1| \sin(\varphi_1) \quad (2)$$

$$Ma_{1y} = |f_1| \cos(\varphi_1) \quad (3)$$

Para obtener la aceleración angular que aporta cada motor al robot, de la Eqn.  $\dot{\omega} = \frac{Rf}{I}$  se puede derivar la Eqn. (4) específica para el uso de 4 motores. Sustituyendo el momento de inercia Eqn. (5) para el caso de un cilindro con distribución de masa desconocida en (4) se obtiene la Eqn. (6).

$$\dot{\omega} = \frac{R}{I} (f_1 + f_2 + f_3 + f_4) \quad (4)$$

$$I = \alpha MR^2; \quad 0 \leq \alpha \leq 1 \quad (5)$$

$$R\dot{\omega} = \frac{1}{M\alpha} (f_1 + f_2 + f_3 + f_4) \quad (6)$$

De los resultados obtenidos de las aceleraciones traslacionales y angulares, se puede derivar la Eqn. (7). También se puede expresar como  $a = C_\alpha F$ , donde  $C_\alpha$  se conoce como la Matriz de Acoplamiento de Fuerzas.

$$\begin{pmatrix} a_x \\ a_y \\ R\dot{\omega} \end{pmatrix} = \frac{1}{M} \begin{bmatrix} \sin \varphi_1 & -\sin \varphi_2 & -\sin \varphi_3 & \sin \varphi_4 \\ \cos \varphi_1 & \cos \varphi_2 & -\cos \varphi_3 & -\cos \varphi_4 \\ \frac{1}{\alpha} & \frac{1}{\alpha} & \frac{1}{\alpha} & \frac{1}{\alpha} \end{bmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} \quad (7)$$

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  $V'_l = \frac{V_l r}{2\pi r}$ . 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. (8) y su forma reducida  $v_m = DV'_l$  donde a  $D$  se le conoce como la Matriz de Acoplamiento de Velocidades.

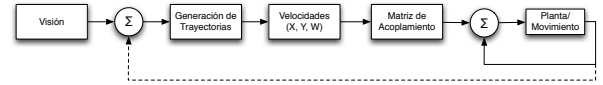


FIGURE 5. Esquema de Control Implementado

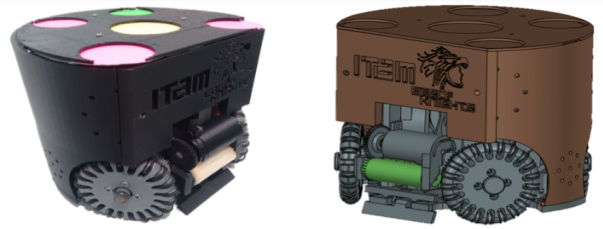


FIGURE 6. Robot Real vs Modelo 3D del Robot

$$v_m = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{bmatrix} \sin \varphi_1 & \cos \varphi_1 & 1 \\ -\sin \varphi_2 & \cos \varphi_2 & 1 \\ -\sin \varphi_3 & -\cos \varphi_3 & 1 \\ \sin \varphi_4 & -\cos \varphi_4 & 1 \end{bmatrix} \begin{pmatrix} v'_x \\ v'_y \\ R\omega' \end{pmatrix} \quad (8)$$

## 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. 5. 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. 7 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. 8 se pueden ver los resultados de 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

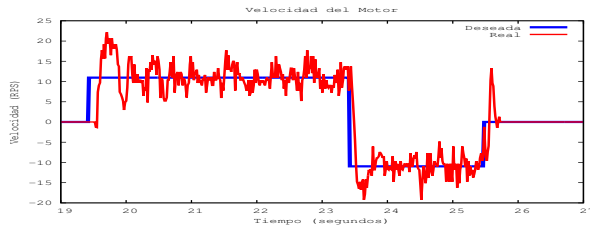


FIGURE 7. Velocidad real vs Velocidad deseada

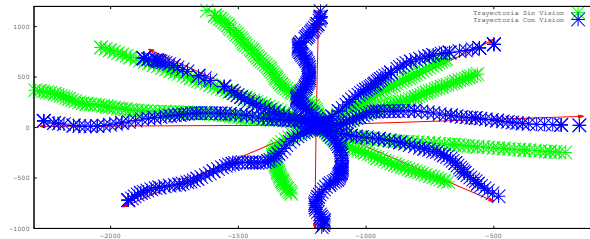


FIGURE 8. Trayectorias con retroalimentación

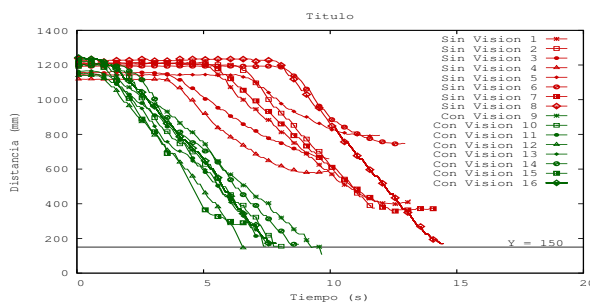


FIGURE 9. Robot Real vs Modelo 3D del Robot

las direcciones en que practicamente 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.

Utilizando los datos del experimento mencionado anteriormente, se puede obtener el error que tiene el robot en el tiempo. Definiendo el error como la diferencia entre la posición deseada y la actual, en la Fig. 9 se puede observar la mejora que representa la retroalimentación de visión. Debido a las dimensiones del robot, se estableció un margen de 150 mm por lo que se detiene al alcanzar esa posición reportada por visión.

## 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. 6 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.

## REFERENCES

- [1] Corke, P., 2011. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg.
- [2] Rojas, R., 2005. "A short history of omnidirectional wheels". See also URL <http://robocup.mi.fu-berlin.de/buch/shortomni.pdf>.
- [3] Rojas, R., and Förster, A. G., 2006. "Holonomic control of a robot with an omnidirectional drive". *KI-Künstliche Intelligenz*, 20(2), pp. 12–17.
- [4] Ashmore, M., and Barnes, N., 2002. "Omni-drive robot motion on curved paths: The fastest path between two points is not a straight-line". In *Australian Joint Conference on Artificial Intelligence*, Springer, pp. 225–236.
- [5] D'Andrea, R., Kalmar-Nagy, T., Ganguly, P., and Babish, M., 2000. "The cornell robocup team". In *Robot Soccer World Cup*, Springer, pp. 41–51.
- [6] Purwin, O., and D'Andrea, R., 2003. "Cornell big red 2003". *Robocup*.
- [7] Houshangi, N., and Lippitt, T., 1999. "Omnibot mobile base for hazardous environment". In *Electrical and Computer Engineering, 1999 IEEE Canadian Conference on*, Vol. 3, IEEE, pp. 1357–1361.
- [8] Adăscăliței, F., and Doroftei, I., 2011. "Practical applications for mobile robots based on mecanum wheels-a systematic survey". *Gh. Asachi Technical University of Iasi, Mechanical Engineering Faculty, Theory of Mechanisms and Robotics Department, B-dul D. Mangeron*, pp. 61–63.
- [9] Borgolte, U., Hoyer, H., Bühler, C., Heck, H., and Hoelper, R., 1998. "Architectural concepts of a semi-autonomous

wheelchair”. *Journal of Intelligent and Robotic Systems*, 22(3-4), pp. 233–253.

- [10] Chaiso, K., and Sukvichai, K. Skuba 2011 extended team description.