

Hermes II: Robot Educativo Holonómico para la Enseñanza en Ingeniería

Rubén E. Ayme¹, Orlando Micolini¹, Luis O. Ventre¹, Ana B. García Cabral¹ y Sergio J. Sagripanti¹

¹Laboratorio de Arquitectura de Computadoras, Facultad de Ciencias Exactas Físicas y Naturales
Universidad Nacional de Córdoba, Córdoba, Argentina.

Fecha de recepción del manuscrito: 04/02/2020

Fecha de aceptación del manuscrito: 30/10/2020

Fecha de publicación: 31/10/2020

Resumen— El presente trabajo detalla el proceso de diseño y construcción de un robot holonómico con cuatro ruedas omnidireccionales con fines educativos, de bajo costo y escalable. Con el objetivo de que cada alumno construya y programe su propio robot visto como una herramienta cognitiva que le permita construir y reconstruir el mundo percibido por medio de la experiencia personal, logrando una dinámica incremental en la construcción del conocimiento.

En cuanto al sistema de control del robot, se encuentra embebido en una placa de desarrollo que soporta un sistema operativo libre, el cual permite la programación concurrente haciendo uso de hilos en diferentes lenguajes de programación, esto admite desarrollar el software de control orientado a componentes, facilitando la incorporación de mejoras futuras sobre el software, los sistemas actuadores y sensores. Se priorizaron los criterios de funcionalidad, costo, simplicidad y duración para el diseño y la selección de los elementos constitutivos. Como resultado de este trabajo se obtuvo un robot con las particularidades de poder ser construido y programado por los alumnos, previendo las facilidades para su escalabilidad e incorporación de nuevas características tanto de software como de hardware.

Palabras clave— robot omnidireccional, programación concurrente, educación universitaria.

Abstract— This paper details the design and construction process of a holonomic robot with four omnidirectional wheels for educational, low-cost, and scalable purposes. Aiming for each student to be able to build and program their own robot, this is seen as a cognitive tool that allows them to build and rebuild the perceived world through personal experience, achieving an incremental dynamic in the construction of knowledge.

As for the robot's control system, it is embedded in a development board that supports a free OS, which allows concurrent programming using threads in different programming languages, which supports developing component-oriented control software, facilitating the incorporation of future improvements on the software, actuator systems, and sensors controlled by it. The criteria of functionality, cost, simplicity, and duration were prioritized for the design and selection of components. As a result of this work, a robot was obtained with the peculiarities of being able to be built and programmed by the students, providing the facilities for its scalability and incorporation of new features of both software and hardware.

Keywords— omnidirectional robot, concurrent programming, university education

INTRODUCCIÓN

Los robots se están convirtiendo en un componente integral de nuestra sociedad y tienen gran potencial para ser utilizados como tecnología educativa. Los robots educativos son un subconjunto de la tecnología educativa, utilizándose para facilitar el aprendizaje y mejorar el rendimiento educativo de los estudiantes. La teoría del constructivismo establece que el conocimiento aprendido está conformado por lo que los alumnos saben y experimentan. Papert se suma a esto al introducir la noción de construcciónismo, que establece que el aprendizaje ocurre cuando un estudiante construye un artefacto físico y

reflexiona sobre su experiencia de resolución de problemas basada en la motivación para construir el artefacto (Mubin et al., 2013).

El conocimiento no es simplemente un artículo para transferir, codificar, retener y volver a aplicar, sino una experiencia personal que se elabora. Los alumnos como constructores de sus propias herramientas cognitivas, así como de sus realidades externas, construyen y reconstruyen constantemente el mundo a través de la experiencia personal. El conocimiento y el mundo percibido por el alumno ganan existencia y forma a través de la interacción del uno con el otro. (Ackermann, 2010)

Para alcanzar una dinámica incremental de la construcción del conocimiento, el alumno debe estar inmerso en los procesos constructivos por los cuales supera sus puntos de vista actuales del mundo, y elabora comprensiones más profundas sobre sí mismo y su entorno (Ackermann, 2010)

Dirección de contacto: Ing. Ventre, Luis O., Av. Velez Sarfield 1611, Ciudad Universitaria, X5016 CGA. Tel: 5353800 interno 29000.

Correo Electrónico: luis.ventre@unc.edu.ar

En este trabajo se propone la construcción y programación de robots como una forma de materialización del objeto de estudio con la capacidad de agregar interacción social al contexto de aprendizaje (estar situado, conectado y sensible a las variaciones del entorno).

Esta iniciativa tiene como objetivo ir más allá del aprendizaje puramente basado en software y/o la simulación (Mubin et al., 2013).

El objetivo común es resaltar los procesos de construcción y programación de un robot holonómico para que los alumnos reflexionen por medio de la experimentación, con el fin de elaborar comprensiones más profundas sobre lo estudiado.

A nivel universitario, en particular en las carreras de ingeniería, la robótica requiere integración de conocimientos de diferentes áreas, en particular en el diseño y construcción de un robot convergen: la electrónica, la programación, los sistemas de control, las comunicaciones, la mecánica, etc. conocidos como Cyber-Physical-Systems (Song et al., 2016). Además, esta iniciativa permitirá integrar e impulsar la metodología STEM, por sus siglas en inglés Science, Technology, Engineering and Mathematics. Está tiene como objetivo el aprendizaje integrado de todas las disciplinas científico-tecnológicas y se caracteriza por ser una metodología activa, manipulativa, constructivista y por descubrimiento (Bybee, 2010, Sanders, 2008).

Para lograr una real integración de los conocimientos por parte de los alumnos es deseable que el robot esté al alcance de todos, que el robot sea de construcción simple, de costo accesible y propiedad de cada estudiante tal que puedan interactuar libremente con él, con la premisa que lo utilicen a lo largo de la carrera; para lo cual debe ser fácilmente escalable, permitiendo la incorporación de crecientes prestaciones que acompañen al estudiante en el uso de tecnologías y herramientas cada vez más complejas y actuales, en la medida que avanza en su carrera. Más aún motivar a que cada estudiante extienda el uso, la aplicación, y experimentación libremente en aprendizajes no formales como: concursos, ferias, exposiciones, juegos, etc..

El criterio adoptado para el diseño de este robot es el incremental, dado que serán los alumnos de Ingeniería en Computación de la Facultad de Ciencias Exactas Físicas y Naturales de la Universidad Nacional de Córdoba; quienes realizarán este desarrollo a lo largo de su carrera. La problemática de la construcción del robot se evaluó solicitando a alumnos avanzados de la carrera la construcción e integración de distintas partes de este robot.

Como resultado de este trabajo se obtuvo un robot con capacidad de ser programado y construido a un bajo costo por los alumnos, con posibilidades de escalabilidad para la incorporación de nuevas características.

Este documento ha sido estructurado de la siguiente manera: inicialmente se analizan los antecedentes de robots para la educación, con estos antecedentes se determinan las características que debe tener el robot a desarrollar. Seguidamente se definen los requerimientos funcionales y no funcionales estableciendo la arquitectura de alto nivel del robot. A continuación se desarrollan los componentes de la arquitectura: el sistema mecánico de locomoción del robot, conjuntamente con los actuadores; los sensores; la forma de comunicación y finalmente la arquitectura de hardware del sistema de control. Posteriormente se

describen las 6 iteraciones, del desarrollo incremental, hasta llegar al modelo final con todas las funcionalidades incorporadas y probadas. Finalmente se describen los resultados y las conclusiones obtenidas.

ANTECEDENTES

Esta propuesta toma la experiencia adquirida del desarrollo del robot móvil Hermes (Lichtensztein et al., 2014), el cual fue implementado como parte del proyecto de investigación “Diseño e Implementación de Software y Hardware Optimizados para Sistemas de Computación Paralelos en Ingeniería”. Como así también, el trabajo de investigación realizado en (Mondada et al., 2009) en el cual se realiza un exhaustivo análisis de los robots comerciales concluyendo que: la mayoría de los robots existentes en el mercado son exclusivamente robots móviles eficientes o buenas herramientas educativas, pero no ambos a la vez. Los criterios de análisis para esta conclusión son:

1) Criterios para ser un robot móvil eficiente

Bajo costo: los cursos de ingeniería son numerosos por lo que se requiere de una gran cantidad de robots, tal que cada alumno acceda a uno; sumado a que el presupuesto de muchas instituciones es escaso o está disminuyendo, llevan a la necesidad de reducir el costo individual de cada robot.

Escalable: un diseño por componentes (Siewert, 2016) permite un creciente número de prestaciones con mínimas modificaciones a nivel de interfaces.

Mantenible: su construcción robusta debe permitir un bajo costo de mantenimiento.

2) Criterios para que un robot sea una buena herramienta educativa

Tamaño de escritorio: Un robot que puede evolucionar en el escritorio cerca de la computadora mejora drásticamente la eficiencia del estudiante durante la experimentación. Se considera que para una buena movilidad el espacio de experimentación debe ser entre 7 y 10 veces el diámetro del robot.

Amplio abanico de posibilidades desde el punto de vista educativo y de ingeniería: Para utilizar esta herramienta en varios campos de la ingeniería, como el procesamiento de señales, el control automático, la programación integrada o el diseño de sistemas inteligentes distribuidos, el robot debe proporcionar un amplio conjunto de funcionalidades.

Fácil de usar: Las interfaces de usuario deben ser las mismas que el alumno utiliza en el desarrollo de programas y sistemas.

Información abierta: El robot tiene que ser compartido entre profesores, laboratorios, escuelas y universidades. Un modelo de desarrollo de hardware / software de código abierto es una forma efectiva de lograr este objetivo.

Considerando ambos antecedentes en la nueva versión, denominada Hermes II, se rediseñó el sistema mecánico de locomoción utilizando ruedas omnidireccionales, el sistema de control se encuentra embebido en una placa de desarrollo que soporta un sistema operativo (SO) libre, el cual permite la programación concurrente haciendo uso de hilos en diferentes lenguajes de programación, esto admite desarrollar el software de control orientado a componentes (Siewert, 2016), facilitando la incorporación de mejoras futuras sobre el software, los sistemas actuadores y sensores

controlados por el mismo. Se priorizó el criterio de funcionalidad, costo y simplicidad para el diseño y la selección de las partes.

MÉTODOS

Se optó por emplear un proceso de desarrollo iterativo con entregas incrementales ya que permite satisfacer los requerimientos más críticos al inicio del desarrollo disminuyendo así el riesgo total.

Un aspecto fundamental para guiar el desarrollo incremental es la priorización de los requerimientos y/o objetivos en función del valor que aportan al usuario final (Sommerville, 2016). De esta manera se van añadiendo nuevos requerimientos o mejorando los que ya se completaron. Al finalizar cada iteración se obtiene un prototipo funcional.

Partiendo de los objetivos planteados se desarrolla una primera aproximación del sistema a construir: Debe ser un robot con capacidad de movimiento con tres grados de libertad, donde dos son de desplazamiento y uno de rotación sobre un eje perpendicular al plano de desplazamiento. Requiere de sensores para reconocer el ambiente y tener parámetros para decidir los movimientos a realizar. Finalmente ha de ser capaz de mantener comunicaciones bidireccionales con otros dispositivos (Lichtensztein et al., 2014). De esto se desprenden los *requerimientos funcionales* para el robot:

- El robot debe poder moverse de forma holonómica en todas las direcciones sobre el plano.
- El usuario debe poder comunicarse con el robot para configurarlo y controlarlo.
- El robot debe poder reconocer el ambiente mediante el uso de sensores.
- El robot debe soportar la programación de hilos para ser utilizado en la enseñanza de programación concurrente

Mientras que los *requerimientos no funcionales* serán los siguientes:

Generales

- Fácil armado siguiendo un instructivo (aproximadamente seis horas).
- El máximo desvío de su trayectoria no debe superar los 2cm en 1m recorrido.
- El costo del robot debe ser accesible
- Permitir su manejo desde múltiples plataformas, como: PC, Smartphone, Tablet, joystick.
- Diseño estructural robusto.
- Que sea capaz de proponer problemas de concurrencia

Software:

- Programable en Java, C, C++ y Python.
- Implementado con bibliotecas abiertas.
- Uso de un Sistema Operativo libre.

Hardware:

- Hardware libre.
- Partes con abundante documentación.
- Microcontrolador de simple programación.
- Uso de múltiples sensores.
- Múltiples entradas para sensores.
- Múltiples salidas para actuadores

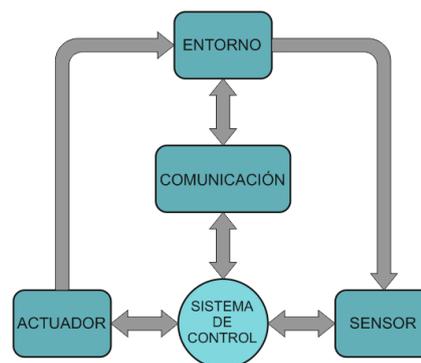


Fig. 1: arquitectura de alto nivel del sistema

Luego de estudiar los requerimientos y los objetivos del proyecto se diseña la arquitectura de alto nivel del sistema que se observa en la Fig. 1.

La arquitectura propuesta consta de diferentes subsistemas, los cuales permiten realizar un diseño modular. Se ha considerado que cada subsistema sea autónomo, para facilitar la incorporación, actualización y eliminación de subsistemas sin la necesidad de modificar el sistema completo.

MATERIALES

Sistema Mecánico de Locomoción

El sistema de locomoción es el responsable de la traslación del robot. Las configuraciones más comunes son las siguientes: Diferencial, Síncrono, Triciclo, Ackerman y Omnidireccional (Baturone, 2005).

Considerando el requerimiento funcional del movimiento holonómico en todas las direcciones sobre el plano se opta por el sistema mecánico de locomoción omnidireccional, el cual permite mayor libertad de movimiento que los sistemas de ruedas clásicas. Los robots que implementan este sistema pueden moverse en cualquier dirección sobre el plano y en cualquier momento sin la necesidad de hacer movimientos previos para modificar su trayectoria. Requiere ruedas que permitan movimiento en más de una dirección. Este sistema puede ser implementado con tres o cuatro ruedas omnidireccionales, ver Fig. 2.

Las ruedas omnidireccionales ruedan en el sentido de avance, pero también se pueden desplazar lateralmente con gran facilidad como se observa en la Fig. 3.

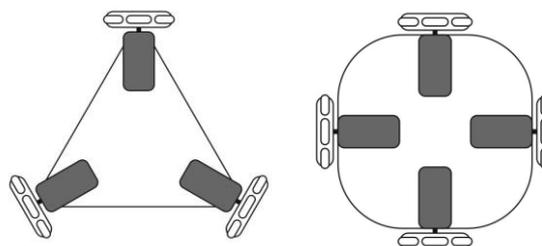


Fig. 2: Sistema holonómico con tres y cuatro ruedas

El inconveniente surgido en esta etapa ha sido el acoplamiento de las ruedas omnidireccionales a los ejes de los motores. Para realizar el acoplamiento se maquinó un buje de aluminio el que se enclavo al eje, dilatándolo térmicamente, esto se muestra en la Fig. 3.



Fig. 3: Rueda omnidireccional

Sistema de Control

“Se representan las velocidades del robot deseadas como el vector $\{\dot{x}, \dot{y}, \dot{\theta}\}$ en coordenadas cartesianas, y luego se relaciona estas con los ángulos de las ruedas $\{\phi_1, \phi_2, \phi_3, \phi_4\}$ para obtener la velocidad angular de cada una de ellas” (John, 1989), esto se muestra en la Fig. 4.

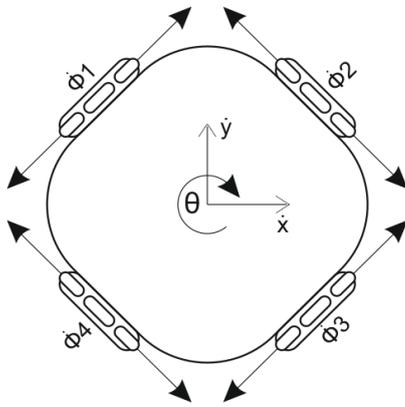


Fig. 4: Sistema de referencia cinemático del robot omnidireccional

Luego de un desarrollo matemático que se puede profundizar en la referencia (John, 1989), se obtiene la matriz de transformación (1) para el sistema de control de un robot omnidireccional de cuatro ruedas fijas.

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin\alpha_1 & \cos\alpha_1 & L \\ -\sin\alpha_2 & \cos\alpha_2 & L \\ -\sin\alpha_3 & \cos\alpha_3 & L \\ -\sin\alpha_4 & \cos\alpha_4 & L \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} \quad (1)$$

Por lo tanto, dada la velocidad lineal $v = [\dot{x}, \dot{y}]$ y angular $\dot{\theta}$ deseadas, las velocidades angulares requeridas de las ruedas $\{\phi_1, \phi_2, \phi_3, \phi_4\}$ pueden ser determinadas por el sistema de ecuaciones (1).

Aquí la problemática se presenta cuando una rueda patina y se desfasa con respecto al valor propuesto. Este problema ha sido solucionado realimentando el control con las mediciones de los sensores de proximidad como error.

Actuadores

Los actuadores tienen por misión generar el movimiento de los elementos del robot según las órdenes dadas por la unidad de control. De manera general, los

actuadores utilizados en robótica pueden emplear energía neumática, hidráulica o eléctrica.

Las características de control, sencillez y precisión de los accionamientos han hecho que los actuadores eléctricos sean los más usados. (Baturone, 2005). Dentro de los actuadores eléctricos pueden distinguirse tres tipos diferentes:

Motores de corriente continua (DC):

- _ Controlados por inducción
- _ Controlados por excitación

Motores de corriente alterna (AC):

- _ Síncronos
- _ Asíncronos

Motores paso a paso

Se opta por motores de corriente continua, como el que se muestra en la Fig. 5, pues el torque generado es proporcional a la diferencia de potencial aplicado a los terminales de alimentación y el sentido de giro depende de la polaridad; facilitando de esta manera el control, (Baturone, 2005).



Fig. 5: Motor de corriente continua

Este tipo de motores, pequeños y de bajo costo, generalmente presentan como inconveniente que no publican sus curvas. Para resolver este inconveniente se realizaron distintos experimentos. Esencialmente se construyó un montaje que consta de una polea con un peso conocido, que no es más que un recipiente con agua como muestra la Fig. 6, la polea se fija al eje del motor para realizar los experimentos.

Se realizaron distintas corridas variando tensión y corriente, se tomaron mediciones de velocidad y torque. Con estos datos se construyeron las curvas de rpm-torque y corriente-torque, como se muestra en la Tabla 1.

TABLA 1: RELACIÓN TENSIÓN, VELOCIDAD, TORQUE

Tensión [V]	Velocidad [RPM]	Torque [Kg*cm]
3 V	80 RPM	0,3 Kg*cm
6 V	200 RPM	1 Kg*cm
12 V	400 RPM	2 Kg*cm

con estos experimentos. También se midió la respuesta del sistema a un escalón.



Fig. 6 Montaje para medir torque

Sensores

Un sensor es un dispositivo eléctrico y/o mecánico capaz de convertir magnitudes físicas, como la luz, velocidad, aceleración, presión, temperatura, etc, en otra magnitud, normalmente eléctrica, que sea posible manipular y cuantificar, (Baturone, 2005).

Sensor de Distancia

Los sensores de distancia y transductores de distancia, están pensados para realizar la medición de distancia o desplazamiento lineal, de una forma automatizada, y que proporcionan una señal eléctrica según la variación física, en este caso la variación física es la distancia. Emiten pulsos de sonido y se mide el tiempo que demora en retornar una vez que ha sido reflejado por un objeto. De esta forma, teniendo en cuenta la velocidad de propagación del sonido, se determina la distancia. Conviene poner de manifiesto que las características de la superficie que refleja la onda y el ángulo de incidencia tienen una notable influencia en la precisión de estos sensores, Fig. 7.

Estos sensores presentan errores cuando las superficies reflectantes están oblicuas con respecto a la dirección de la onda sonora, por lo que devuelven una distancia media. Para obtener mejor precisión es necesario combinarlos con otros sensores.



Fig. 7: Sensor de distancia

Sensor Óptico

Es un detector óptico, en el cual se emplean diodos emisores de luz y fotodetectores tales como fotorresistores, fototransistores o fotodiodos. Los fotodiodos poseen una mayor sensibilidad y producen una señal lineal en un rango muy amplio de niveles de intensidad. Sin embargo, su salida necesita ser amplificada. (Baturone, 2005). Se emplean también sensores ópticos en los que el emisor y el receptor se montan sobre un mismo encapsulado, utilizados como detector de presencia por reflexión como muestra la Fig. 8.

En este caso particular, se utilizan sensores infrarrojos que son sensibles a la longitud de onda, tales como 880nm inmediatamente inferiores a la visible. Sin embargo, conviene poner de manifiesto que, en principio, no suministran ninguna medida de distancia, sino tan solo una señal binaria indicando si existe o no un objeto próximo en un rango de distancia característico del sensor, (Baturone, 2005)

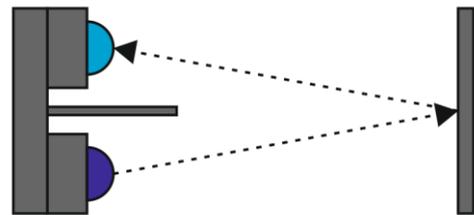


Fig. 8: Sensor óptico

Se utilizan dos sensores infrarrojos TCRT5000 como el de la

Fig. 9, para diseñar un detector para implementar un seguidor de línea

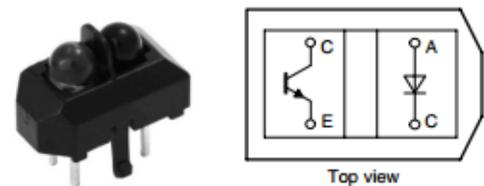


Fig. 9: Sensor óptico TCRT5000

Medición de velocidad de motores

El objetivo es construir un sistema de lazo cerrado que controle la velocidad de los motores, con el fin de aproximar la trayectoria calculada.

Para medir la velocidad, en revoluciones por minuto (RPM), se colocó, en el eje del motor, una rueda con un patrón impreso el cual es detectado por un sensor infrarrojo, como se ve en la Fig. 10. El patrón impreso consta de marcas blancas sobre un fondo negro, con el fin de generar interrupciones. Se realizaron 12 marcas en cada rueda para simplificar el cálculo.



Fig. 10: Rueda con patrón impreso para medir RPM

Para realizar la medición de RPM se experimentaron tres métodos distintos:

Primer método, generando una interrupción con el flanco de la señal obtenida del sensor infrarrojo. Las RPM se determinan en función de la cantidad de interrupciones contadas en el periodo de una base de tiempo implementada para este fin.

Segundo método ensayado, ha sido tomar el tiempo del sistema entre dos interrupciones alternativas, y luego calcular las RPM.

Tercer método ensayado, consistió en medir la cantidad de pulsos, generada por una base de tiempo externa, entre dos interrupciones. Para lo cual se toma la lectura de un contador en la primera interrupción y se toma la lectura nuevamente en la interrupción siguiente. Con la diferencia entre las lecturas se calculan las RPM.

Se debe destacar que es necesario filtrar el rebote de los flancos de subida y bajada de cada interrupción.

Si bien el método que se implementó es el primero, generó una gran sobrecarga en el procesador. Esto es debido a que la cantidad de ciclos necesarios para atender una interrupción oscilan entre 30 y 60, por lo tanto si se requiere medir 1000 RPM se requieren 200 pulsos por segundo, es decir una pérdida de 12000 ciclos por segundo, más la rutina que procesa la interrupción esto por cada rueda, esta sobrecarga es un gran limitante en sistemas de tiempo real. Si el caso es un sistema de tiempo real, la solución recomendable es medir la RPM con un hardware dedicado.

Comunicación

Se analizaron tres formas de comunicación inalámbrica para el robot, Wi Fi, Bluetooth y Zigbee.

Luego de estudiar las tres posibilidades se optó por usar Bluetooth, en particular el módulo HC-05, debido a su bajo costo y alta disponibilidad en dispositivos existentes (Lichtensztein et al., 2014). Este módulo brinda la posibilidad de incluir en los experimentos una comunicación serial TTL.

Si bien algunos modelos de placas de control tienen Wi-Fi incluido, se ha supuesto que no está incluido debido al requerimiento de bajo costo, de cualquier manera, las ventajas que este dispositivo brinda, en cuanto a alta velocidad y cifrado, no eran necesarias en este proyecto. Por otra parte, Zigbee no fue elegido por el poco uso en dispositivos domésticos, es decir es necesario un adaptador para permitir a una PC utilizar la comunicación, y es complejo para una Tablet o un Smartphone lo posean.

Arquitectura de Hardware del Sistema

Para el estudio y selección de la placa de control se tuvieron en cuenta los requerimientos de software y hardware planteados. Se analizaron placas que se

encuentran disponibles actualmente en el mercado: Arduino Mega 2560, Arduino Due, Raspberry Pi B+, Freescale FRDM-K64 y CubieBoard.

Analizadas las ventajas y desventajas de cada placa, se selecciona la placa de desarrollo Raspberry Pi B+ la cual permite trabajar con distintos Sistemas Operativos open source, admite comunicación I2C, comunicación serial, dispone de numerosos puertos GPIO para el uso de sensores y puede ser programada en diferentes lenguajes y entornos de programación. El inconveniente de esta placa es que no dispone de la cantidad suficiente de puertos PWM para el control de los actuadores, para suplir esta deficiencia se incorpora un microcontrolador que dispone de más puertos PWM, tal como la placa de desarrollo Arduino Pro Mini, con un microcontrolador ATMEGA328. Esta placa dispone de 6 puertos de PWM y comunicación I2C.

La arquitectura de hardware de alto nivel planteada se presenta en la Fig. 11. Se diseñó una arquitectura modular con cuatro subsistemas: comunicación, sensores, actuadores y placa de control. La placa de control (subsistema principal), interactúa con los otros subsistemas a través de sus correspondientes interfaces. El subsistema comunicación es la interfaz entre el usuario y la placa de control. El subsistema sensores está formado por los sensores que son los encargados de captar las condiciones del entorno y enviar datos a la placa principal para su proceso. El subsistema actuadores recibe información de la placa de control para ejecutar las acciones programadas. En este caso los actuadores son cuatro motores que reciben de la placa de control información de velocidad y sentido de giro.

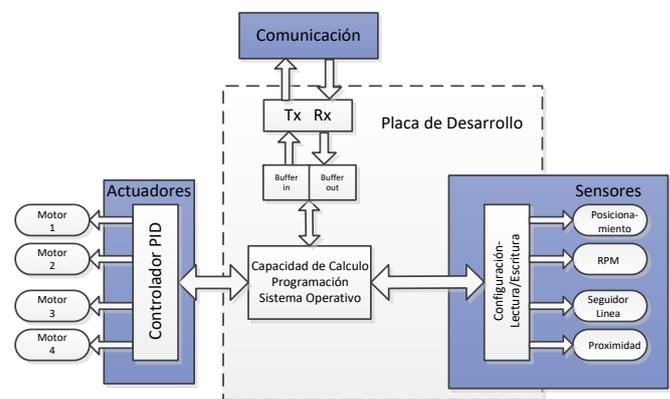


Fig. 11: Arquitectura de hardware

DESARROLLO

Seguidamente una breve síntesis de cada iteración en el desarrollo del prototipo:

Iteración 1: en esta iteración se debe seleccionar la placa de control, que según lo expresado en el apartado anterior, se seleccionó la placa de control Raspberry pi B+.

Iteración 2: se desarrolla el sistema mecánico de locomoción que consta de cuatro ruedas omnidireccionales, conjuntamente con el chasis, inicialmente se construyó una versión de prueba, donde se montará el robot con todos sus componentes. Se seleccionaron los motores de corriente



Fig. 12: Montaje actuador en chasis

continúa que funcionarán como los actuadores eléctricos con el suficiente torque para mover el robot de su condición de reposo.

En la Fig. 12 se puede observar el montaje en el chasis del actuador eléctrico con el sistema mecánico de locomoción: ruedas omnidireccionales.

Iteración 3: En esta iteración se trabaja sobre la placa de desarrollo Raspberry Pi B+ seleccionada en la iteración 1. Se escoge y embebe un sistema operativo con el fin que el robot pueda ser programado en los lenguajes de programación planteados en los requerimientos de software del proyecto., en la bibliografía (Fernandes, 2018) se encuentra todos los pasos necesarios para realizar estas tareas. Se configuran los parámetros para hacer uso de los distintos protocolos de comunicación de los que dispone Raspberry Pi, con la placa Bluetooth (Fig. 13) y por medio de la comunicación I2C con la placa Arduino (Fig. 13) que comanda los actuadores.

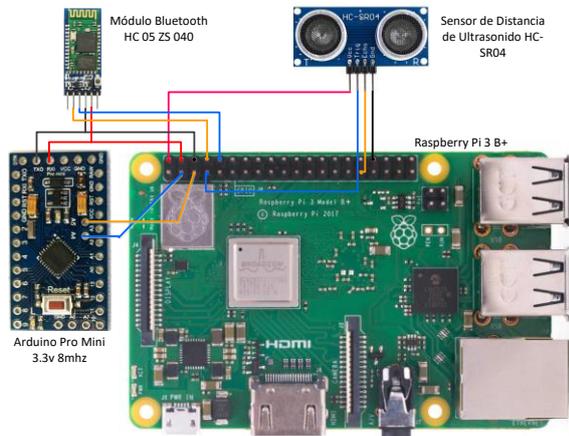


Fig. 13: Conexiones

Iteración 4

En esta iteración se seleccionó el puente H para el control de los motores, la fuente de alimentación, la batería y el cargador de batería para el sistema. Esto es posible de realizar puesto que se conocen todos los consumos y cargas del sistema, solamente se ha estimado la carga que representa la plataforma.

Puesto que se conoce el consumo máximo de los motores se ha seleccionado un puente H con capacidad para controlarlos de 12 V y 2 A como carga máxima. La fuente de alimentación, convierte DC a DC, de 12 V a 5 V que es el requerimiento de las placas de control. Para las baterías se ha estimado una autonomía de 2 hs 30 minutos, dado el

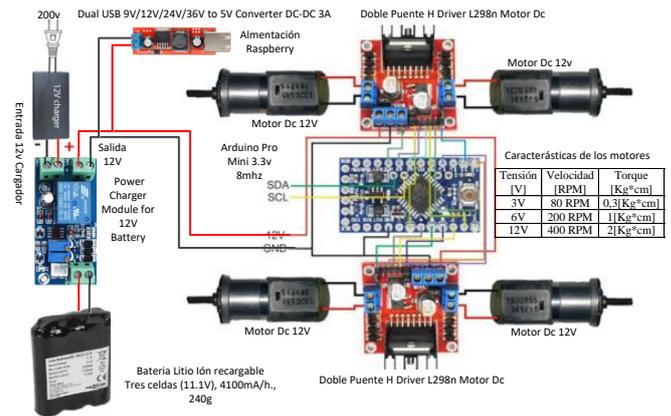


Fig. 14: Interconexión de sistema de alimentación, motores DC, puente H y Arduino

consumo del sistema se requiere una unidad de 4100 mAh, la tensión de la batería está determinada por la tensión de los motores seleccionados, se optó por una batería de 11,1V.

En la

Fig. 14 se muestra la disposición e interconexión del sistema de alimentación, cargador de baterías y de los motores con los puentes H (Train's., 1996).

Iteración 5: En esta iteración se realizó un rediseño del chasis para incorporar todos los componentes seleccionados en las iteraciones previas, Fig. 15.

Se arma un nuevo prototipo con los sensores para ser usados en una iteración posterior. Además se desarrolla el hardware y software relacionado al funcionamiento de los sensores. En cuanto al software, se diseña y desarrolla orientado a componentes para facilitar la incorporación de nuevos módulos al sistema

Iteración 6: En esta iteración se realiza una integración de todos los componentes del sistema Hermes II y se analizan los posibles escenarios que se pueden afrontar.

En el comienzo del proyecto se diseñó la arquitectura de Hardware y posteriormente la arquitectura de Software orientada a componentes. En la Fig. 16 se presenta el diseño de la arquitectura del sistema integrado

RESULTADOS

El sistema final obtenido es un robot omnidireccional de mecánica simple con cuatro motores de corriente continua anclados al chasis para la tracción y dirección del mismo. Para permitir el movimiento de forma holonómica, se

utilizaron cuatro ruedas omnidireccionales posicionadas de forma perpendicular a los ejes del plano cartesiano, disminuyendo las partes móviles del sistema.

Hermes II dispone de un sensor de distancia y dos de proximidad para reconocimiento de ambiente, con capacidad tanto de hardware como de software para incorporar otros sensores.

Debido al diseño de piezas encastrables, el robot puede ser construido de forma fácil e intuitiva con materiales de bajo costo. Esto permite a los estudiantes participar desde la etapa de construcción mecánica y electrónica hasta la programación del robot.

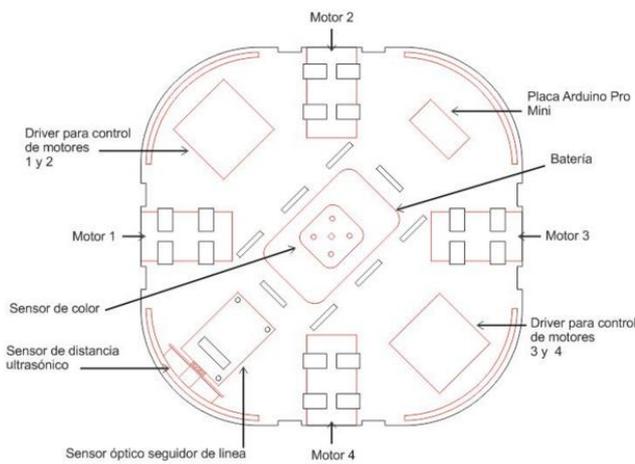


Fig. 15: Chasis con las partes distribuidas

Se desarrolló un software en lenguaje Java, con arquitectura basada en componentes, para facilitar la incorporación de nuevos sensores, actuadores y sistemas de comunicación. Los componentes desarrollados para el robot pueden ser instanciados por el estudiante en la clase “Hermes” según sus necesidades.

Dichos componentes permiten modelar el funcionamiento de todos los elementos del sistema. Posee una clase central, la cual contiene los comandos de movimiento disponibles para el robot: avanzar, retroceder, giro horario y giro anti horario. Con lo cual se pueden componer todos los movimientos posibles para un vehículo terrestre. Se crearon clases para los sensores infrarrojos y de distancia; la instanciación de un objeto, de estas clases, permite controlar el sensor deseado. Mediante el uso de los métodos es posible configurar y realizar mediciones con cualquier sensor.

Se desarrolló la documentación y ejemplificación del uso de esta biblioteca. Como así también, los planos del robot y el manual con las piezas numeradas para facilitar el armado del mismo. También se incluyen los archivos de las piezas para que puedan ser construidas con sistema de corte láser, de bajo costo de proceso y materiales. Esta información está disponible en <https://github.com/omicolini/HERMES II>.

En cuanto al costo, se obtuvo una versión básica del robot que cumple con el requerimiento de competitividad en el mercado, contando con la posibilidad de incorporación de nuevos sensores y actuadores para su ampliación, asegurando su escalabilidad.

El costo de los componentes, por unidad, suma un total de 260 dólares. Es posible parcializar la compra, de los componentes, en la medida que se requiera para los experimentos a lo largo de la carrera.

CONCLUSIONES

En el presente trabajo se diseñó y construyó un robot holonómico de cuatro ruedas omnidireccionales. El sistema de locomoción resultante está conformado por un número reducido de piezas móviles de simple construcción, esto mejora la confiabilidad mecánica y prolonga su vida útil lo que permite su utilización a lo largo de la carrera.

Las ruedas omnidireccionales utilizadas fueron el elemento disruptivo del proyecto, ya que permiten modificar la trayectoria del robot sin modificar la orientación de las mismas. Este sistema de movimiento

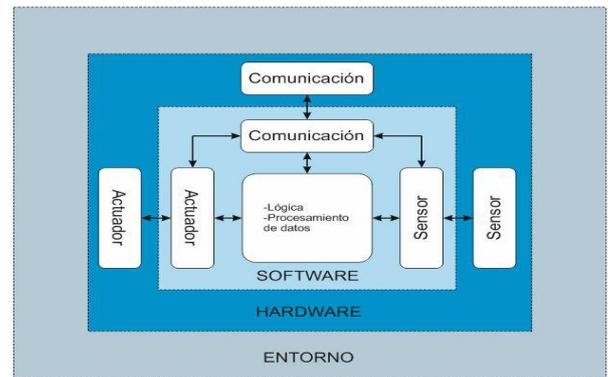


Fig. 16: Arquitectura del sistema

motivo a los alumnos a interactuar tanto con el modelo físico como matemático del sistema para lograr las trayectorias deseadas.

Hermes II posee un SO abierto que permite al usuario desarrollar programación concurrente en distintos lenguajes. Cabe destacar que para la versión actual se han desarrollado todas las bibliotecas en Java.

Hermes II fue presentado en eventos públicos como Expocarrera, Edutech y semana TIC de Córdoba, Argentina despertando curiosidad e interés tanto en los docentes como alumnos presentes. Además se expuso en la academia de ciencias de Córdoba durante treinta días, donde fue exhibido y utilizado por numerosos estudiantes y público en general. Se observó que la propuesta de construcción y programación de robots, como forma de materialización del objeto de estudio, motivó a los alumnos que por medio de la experimentación elaboraron comprensiones más profundas sobre los experimentos presentados.

Este proyecto integra distintas disciplinas, como ciencias, tecnologías, ingeniería y matemática -STEM, la codificación, el pensamiento computacional y el aprendizaje de competencias para ingeniería.

El robot ha mostrado ser una herramienta educativa de aprendizaje tecnológico y además ha motivado la integración social por medio del aprendizaje y la experimentación.

Finalmente se destaca que este robot es una herramienta educativa innovadora para los estudiantes de ingeniería en un sentido amplio. Por sus prestaciones y costo permite su uso en una amplia gama de tópicos. Al igual que la naturaleza de código abierto mejora la calidad del apoyo y acceso a los estudiantes.

REFERENCIAS

- [1] ACKERMANN, E. 2010. Piaget's Constructivism, Papert's Constructionism: What's the difference? . *MIT learning media communications*.
- [2] BATURONE, A. O. 2005. *Robótica: manipuladores y robots móviles*, Marcombo.
- [3] BYBEE, R. W. 2010. What is STEM education? : American Association for the Advancement of Science.
- [4] FERNANDES, S. L. 2018. *Raspberry Pi 3 cookbook for Python programmers: unleash the potential of Raspberry Pi 3 with over 100 recipes*, Packt Publishing Ltd.
- [5] JOHN, J. C. 1989. Introduction to robotics: mechanics and control. *Reading: Addison-Wesley*.
- [6] LICHTENSZTEIN, L. F., MICOLINI, O. & CEBOLLADA, M. “Hermes”: Sistema robótico embebido para la educación. Biennial Congress of Argentina (ARGENCON), 2014 IEEE, 2014. IEEE, 310-315.

- [7] MONDADA, F., BONANI, M., RAEMY, X., PUGH, J., CIANCI, C., KLAPTOCZ, A., MAGNENAT, S., ZUFFEREY, J.-C., FLOREANO, D. & MARTINOLI, A. The e-puck, a robot designed for education in engineering. Proceedings of the 9th conference on autonomous robot systems and competitions, 2009. IPCB: Instituto Politécnico de Castelo Branco, 59-65.
- [8] MUBIN, O., STEVENS, C. J., SHAHID, S., AL MAHMUD, A. & DONG, J.-J. 2013. A review of the applicability of robots in education. *Journal of Technology in Education and Learning*, 1, 13.
- [9] SANDERS, M. E. 2008. Stem, stem education, stemmania.
- [10] SIEWERT, S. 2016. *Real time embedded components and systems*, Cengage Learning.
- [11] SOMMERVILLE, I. 2016. Software Engineering. International computer science series, Tenth Edition. *ed: Addison Wesley*.
- [12] SONG, H., RAWAT, D. B., JESCHKE, S. & BRECHER, C. 2016. *Cyber-physical systems: foundations, principles and applications*, Morgan Kaufmann.
- [13] TRAIN'S., T. A. 1996. A study guide to dc motor controls. *Solid state motor controls training series*.