

## MiniPI: Plataforma Inteligente de Tamaño Reducido

Alejandro Gastón Alvarez, Maximiliano Marufo da Silva, Fernando Sampieri, Damián Pirlo, Juan Manuel Gabis, Pablo Di Sabato, Nicolas Marcovecchio, Luciano Cismondi, Pablo Folino, Sergio Alberino, Claudio Verrastro † y Juan Carlos Gómez ‡

*Grupo de Inteligencia Artificial y Robótica (GIAR), UTN.BA, Buenos Aires, Argentina.  
†fcverra@cae.cnea.gov.ar, ‡juanca@inti.gov.ar*

**Resumen**— Se presenta una plataforma robótica inteligente, autónoma y de pequeñas dimensiones para ser utilizada con fines de investigación y didácticos. Su bajo costo de fabricación, su fácil replicación y su sistema de comunicación, la convierten en un prototipo ideal para generar unidades funcionales que trabajen en forma cooperativa.

Se trata de una plataforma modular y didáctica, por lo que es altamente personalizable y expandible en cuanto a hardware y software se refiere. Por lo tanto, se pretende que sea utilizada como plataforma de investigación y desarrollo dentro del ámbito educativo, tanto para el aprendizaje de las tecnologías que actualmente emplea como para su transferencia a otros proyectos que surjan a partir del mismo.

**Palabras Clave**— plataforma educativa, robótica colaborativa, modular, MiniPI

### 1. INTRODUCCIÓN

En la actualidad se pueden encontrar varios robots móviles desarrollados por grupos de investigación o incluso empresas privadas. La mayoría de ellos se encuentran orientados a aplicaciones particulares, y si bien es posible adaptarlos para otras actividades, se requiere un tiempo relativamente prolongado y muchas veces es necesario realizar demasiados cambios, desaprovechando tiempo y recursos. Por este motivo se establece como primera premisa para la plataforma MiniPI, que todas sus partes constituyan módulos individuales, relacionados entre sí para conformar un sistema adaptable.

Plataformas similares se utilizan en instituciones educativas de todo el mundo, tanto a nivel universitario como secundario, para la formación de estudiantes en el área de la robótica y la inteligencia artificial [1]. Sin embargo, debido al costo de estos dispositivos fabricados por empresas privadas, en su mayoría extranjeras, es recomendable o incluso necesario desarrollar este tipo de robots a nivel local. Por lo tanto, otra idea fundamental de este proyecto es que los mismos sean de bajo costo para así poder proveer a cada estudiante o grupo reducido de un robot. También se

busca que la plataforma sea lo suficientemente didáctica como para que los alumnos puedan generar distintos proyectos, probar algoritmos o crear sus propias soluciones para distintas tareas en un tiempo relativamente corto.

Por otro lado, desde hace ya varios se vienen desarrollando dentro del Grupo de Inteligencia Artificial y Robótica (GIAR) de la Universidad Tecnológica Nacional - Facultad Regional Buenos Aires, distintos algoritmos para proveer de inteligencia artificial a robots móviles [2,3]. Como consecuencia, surgió la necesidad de desarrollar una nueva plataforma capaz de navegar de forma autónoma dentro de diferentes tipos de ambientes. Estos ambientes se pueden clasificar, ordenados de menos complejos a más complejos, en estructurados, semi-estructurados y no estructurados. A medida que se incrementa el nivel de complejidad de los ambientes, es necesario incorporarle a los robots algoritmos que requieren ejecutar mayor cantidad de instrucciones por segundo, además de poder asegurar la respuesta de los mismos dentro de un lapso determinado.

Si bien ya existen robots similares fabricados por empresas nacionales [4] o desarrollados por grupos de investigación [5], el mercado todavía no está del todo desarrollado en nuestro país. MiniPI es una nueva alternativa que incorpora tecnología más moderna, por lo que puede realizar tareas más complejas que requieran mayor velocidad de procesamiento. Esto posibilita, a su vez, la incorporación de un sistema operativo de tiempo real, como es FreeRTOS, que actualmente se encuentra implementado en la plataforma, y que permite asegurar la respuesta del robot dentro de tiempos preestablecidos. Por otro lado, el robot dispone de un sistema de comunicaciones orientado a facilitar la resolución de tareas en forma colaborativa, por medio de la coordinación de varios robots para resolver un problema en forma conjunta.

La plataforma cuenta con dimensiones físicas compatibles con las competencias locales de robótica móvil, como son “Sumo de Robots”, “Fútbol de Robots”, “Seguidores de línea” y “Resolución de laberintos”. Esto implica que puede ser utilizada en cualquiera de estas actividades sin ser necesario realizar modificación alguna, lo que le otorga un alto grado de versatilidad con respecto a otros robots móviles.

## 2. DESCRIPCIÓN DE LA PLATAFORMA

### 2.1. Descripción general

MiniPI es una plataforma móvil inteligente destinada para aplicaciones tanto de robótica móvil como colaborativa. A su vez es apta para la investigación, especialmente para el uso de algoritmos de inteligencia artificial y control. Su diseño se puede dividir en 3 partes: estructura mecánica, hardware electrónico y desarrollo del software. A continuación se detallarán las mismas.

### 2.2. Estructura mecánica

La estructura mecánica está fabricada íntegramente en acrílico y posee los anclajes necesarios para todas las partes componentes del robot (batería, electrónica, sensores, etc.). Posee ruedas para su desplazamiento, dispuestas en configuración diferencial. La motorización se realiza con motorreductores de corriente continua, con tensión de alimentación de hasta 12V.

Tabla 1: Características Mecánicas

Característica	Descripción
Dimensiones	100 x 100 mm
Peso	500 g
Velocidad Máxima	1,2 m/s
Fuerza de empuje máx. s/ resbalar	400 g
Material de la estructura	Acrílico de alto impacto
Motores	Pololu 47:1 Metal Gearmotor 25Dx52L mm HP con Encoder 48 CPR



Figura 1: Render de MiniPI.

### 2.3. Hardware Electrónico

#### 2.3.1. Placa Base

La electrónica del robot consiste principalmente en una placa base, que contiene los circuitos de potencia para el manejo de los motores y todos los puertos y conectores necesarios para el conexionado de periféricos como ser sensores, alimentación, comunicación, etc. La misma dispone de un zócalo en donde se conecta otra placa que posee el microcontrolador encargado del control del robot. De esta manera, el hardware es independiente del

microcontrolador utilizado, lo cual permite la reutilización de la placa base para otros robots como así también su fácil reemplazo, teniendo en cuenta la rápida evolución de los mismos.

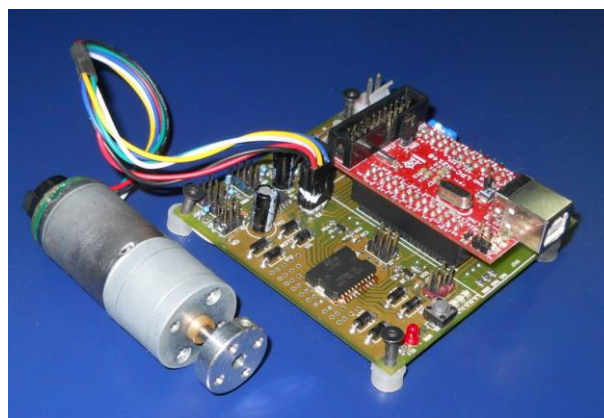


Figura 2: Placa base con la placa de desarrollo y el motor conectados.

#### 2.3.2. Microcontrolador

El microcontrolador utilizado actualmente es un LPC2148, basado en un núcleo ARM 7 de 32 bits, con una frecuencia máxima de ejecución de 60 MHz, lo que permite que el robot pueda ejecutar varias tareas simultáneamente con gran velocidad.

#### 2.3.3. Sensores

El sistema sensorial está compuesto por sensores de línea, sensores infrarrojos (IR), un sensor por ultrasonido (sonar) y sensores de efecto hall utilizados como encoders en los motores para calcular indirectamente la velocidad del robot. Se prevé la incorporación a futuro de un sistema de odometría, basado en un sensor que utilizan los mouse ópticos.

##### 2.3.3.1. Sensores reflectivos

Existen aplicaciones en las cuales se provee de información al robot, generando patrones en la superficie donde se traslada, como ser una línea o zonas delimitadas. Para poder captar esta información es necesario poseer un sensor reflectivo. La plataforma presentada aquí utiliza 4 sensores CNY70 de la marca Vishay. Son sensores de un tamaño reducido, que indican con precisión, a distancias menores a 0,5mm, si hay un objeto que refleja la luz frente a él.

##### 2.3.3.2. Sensores infrarrojos

Como *sensor de proximidad* (detección en distancias cortas), se aplicó el módulo GP2Y0A21 de la marca Sharp. El mismo se caracteriza por medir distancias entre 10cm y 80cm, por tener un bajo consumo y dimensiones reducidas, lo cual lo hace ideal para aplicaciones de esta índole.

##### 2.3.3.3. Sensor por ultrasonido

Se encuentra disponible un sensor por ultrasonido principalmente para funcionar como sonar. Debido a que los sensores infrarrojos tienen un alcance muy reducido, se utiliza este sensor para detectar obstáculos

a distancias mayores. El dispositivo utilizado es el LV-MaxSonar®-EZ1, que puede detectar objetos de 0 a 6,45 metros. El control del dispositivo se realiza mediante dos pines, uno que le indica cuando comenzar una medición, y otro por el cual se detecta un pulso que envía el dispositivo al microcontrolador, cuyo ancho es proporcional a la distancia medida.

#### 2.3.3.4. Encoders

Los motores que utiliza actualmente MiniPI tienen integrados unos encoders en cuadratura que proveen de 1633 cuentas por revolución a la salida de la caja reductora.

También es posible utilizar, en caso de que se emplee otro motor, el encoder magnético rotativo AS5045, que permite la medición precisa del desplazamiento angular sin que haya contacto entre el sensor y el eje de giro. El dispositivo es un *system-on-chip*, que integra elementos de efecto Hall, elementos analógicos y un procesador digital de señales (DSP). A su vez, permite una medición de los 360° y su salida entrega una medición absoluta de 12 bits.

#### 2.3.3.5. Sensor de posicionamiento

El ADNS-2610 es utilizado normalmente como sensor óptico para el posicionamiento en algunos tipos de mouse. Este dispositivo mide los cambios de posición ópticamente mediante la adquisición de imágenes secuenciales (*frames*) de la superficie, y luego, mediante el procesamiento de las mismas, determina la dirección y magnitud del movimiento. El sensor se encuentra en un encapsulado tipo DIP8 y está diseñado para ser utilizado con el lente HDNS-2200. La interfaz de comunicación se realiza mediante dos líneas en serie. La resolución es de 400 cuentas por pulgada (cpi) con velocidades de hasta 12 pulgadas por segundo (ips).



(a) 4 puntos (b) 2 puntos (c) 1 punto

Figura 3: Imágenes de 18x18 píxeles, obtenidas mediante el sensor ADNS-2610 con tres tamaños de fuente distintos (1 punto = 0,35 mm) y 64 niveles de gris.

El uso de este sensor provee de dos características muy interesantes, las cuales pueden además implementarse simultáneamente. La primera es que incorporando al robot dos de estos sensores, se puede implementar un sistema de posicionamiento por odometría [7]. Este método, a diferencia de la sola utilización de encoders en las ruedas, puede detectar resbalones o derrapes de las mismas. La otra característica es la posibilidad de usar las imágenes adquiridas para implementar un algoritmo de localización, como por ejemplo, el de *Grid Localization*.

Si bien estas dos características todavía no están implementadas en la plataforma, están siendo desarrolladas actualmente, habiéndose ya realizado algunas pruebas.

Para su implementación como odómetro, se realizaron trayectorias con los sensores, comparando la distancia real recorrida y la medida. El error obtenido fue de 0,1% en el caso de trayectorias rectas, y de 1,25% para el recorrido de una circunferencia. Se implementó también un algoritmo de localización, el cual contiene la información del mapa donde se encuentra el robot, pero se desconoce su posición inicial, la cual debe ser estimada luego de una serie de movimientos. Para la prueba se utilizó un mapa cuadriculado, de 8x8 casilleros blancos y negros distribuidos aleatoriamente. Se logró conocer la posición del sensor luego de un promedio de 5 movimientos.

#### 2.3.4. Comunicación

La comunicación, que aún no se encuentra integrada a la plataforma, se realiza mediante un XBee/XBee-PRO@DigiMesh 2.4 RF Modules. Consta de un protocolo de comunicación peer-to-peer basado en ZigBee, y tiene como características más destacables *self-discovery*, *network stability*, *dense network operation* y además soporta *sleeping routers*, por lo tanto resulta excelente para aplicaciones donde el ahorro de energía es crucial. Mediante este protocolo se pueden hacer *broadcasts* y comunicaciones entre dispositivos donde todos tienen el mismo nivel de importancia, es decir, no es necesaria la existencia de un Master en la comunicación. Es por eso que se eligió también este protocolo, ya que permite ser utilizado en muchas aplicaciones de robótica colaborativa. Por otro lado, tiene la posibilidad de transmitir mensajes entre dispositivos de manera directa, o mediante el pasaje de información entre dispositivos intermedios que repiten el mensaje para llegar a un receptor que se encuentre muy alejado del emisor.

### 2.4. Desarrollo del Software

#### 2.4.1. Sistema Operativo y Estructura de Software

El robot tiene el sistema operativo de tiempo real de código libre FreeRTOS, lo cual permite una fácil gestión de la ejecución de las diferentes rutinas y procesos que debe realizar. Sobre este sistema operativo se ejecuta el programa principal, el cual podemos dividir en varias capas, listadas a continuación (de alto a bajo nivel):

1. Capa de usuario
2. Capa de gestión del comportamiento
3. Capa de procesos/tareas específicas
4. Capa de drivers y acceso a hardware

El software implementado aprovecha las posibilidades de definición de distintas tareas para dividir en módulos cada funcionalidad del robot. Se logra así independencia entre tareas, solo permitiendo la comunicación entre ellas mediante la intervención del kernel como intermediario.

El sistema ejecuta en forma periódica las tareas de recopilación de datos de los sensores de proximidad y

línea, cada 10ms y 1ms respectivamente, y en forma asincrónica (mediante interrupción) las tareas del sensor de sonido y los encoders. Cada tarea envía por una cola de mensajes múltiple los datos recolectados a la tarea de control correspondiente. Las tareas de control se despiertan únicamente cuando hay datos nuevos disponibles provenientes de alguna tarea de acceso a sensores.

La tarea de control principal recibe la información de todos los sensores, ya que es la encargada de tomar las decisiones y envía un comando a la tarea de control de motores proporcional, integral y derivativo (PID). La tarea del PID recibe los comandos de la tarea de control y de la tarea de los encoders y su función es la de alcanzar y mantener la velocidad de los motores solicitada por la tarea de control.

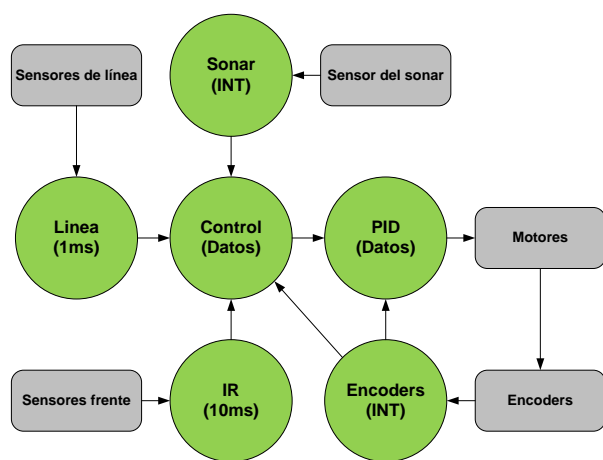


Figura 4: Diagrama de tareas. Se muestra el hardware asociado a cada tarea y se indica entre paréntesis el método utilizado para despertarlas.

#### 2.4.2. Tareas

Las tareas siguen una estructura estándar para facilitar su programación y modularidad. Debido a que FreeRTOS no ofrece independencia completa entre tareas, se adoptó una convención en su programación.

La función de ingreso a las tareas está reservada y se implementa mediante un formato estándar, el cual necesita pocas modificaciones para adaptar el sistema a una nueva implementación. Esta función se encarga de gestionar con el kernel los estados corriendo/bloqueado de cada una de ellas. En su interior se llama a una función principal, cuya implementación es la destinada a contener el código del usuario. De esta manera, el usuario solo debe programar esta función dejando el resto al kernel.

Las prioridades de las tareas se ajustan en función de la implementación que se le dé al robot. Así, por ejemplo, si es utilizado en una competencia de Sumo Robot, se le da mayor prioridad a la tarea que recopila la información de los sensores reflectivos, ya que es crítico que el robot no salga del tatami.

El sistema implementado en el robot consta de las siguientes tareas:

#### 2.4.2.1. PID

Tarea destinada a ejercer el control de velocidad sobre los motores. Recibe comandos de la tarea de control y de la de los encoders para lograr su objetivo. Su funcionamiento se regula mediante la llegada de información a la cola de mensajes. Mientras no exista información nueva, permanece dormida.

Esta tarea consiste en un controlador PID a tiempo discreto, y cuenta además con un filtro de promedio móvil ponderado exponencialmente o, en inglés, Exponencialy Weighted Moving Average (EWMA), que permite obtener un mejor cálculo del término derivativo en presencia de ruido [8]. Dentro del GIAR se han desarrollado otros algoritmos de control dEWMA PID [9] que se encuentran disponibles para su uso.

Para el ajuste de los parámetros del controlador, se utilizaron métodos empíricos. Se aplicó una señal escalón a la entrada del encoder, correspondiente a un cambio brusco en la velocidad. Se probó a su vez un filtro de media móvil para poder comparar el desempeño de ambos filtros, obteniendo similares resultados sobre la señal de salida. Sin embargo, el filtro EWMA es mucho más eficiente en cuanto a la utilización de recursos.

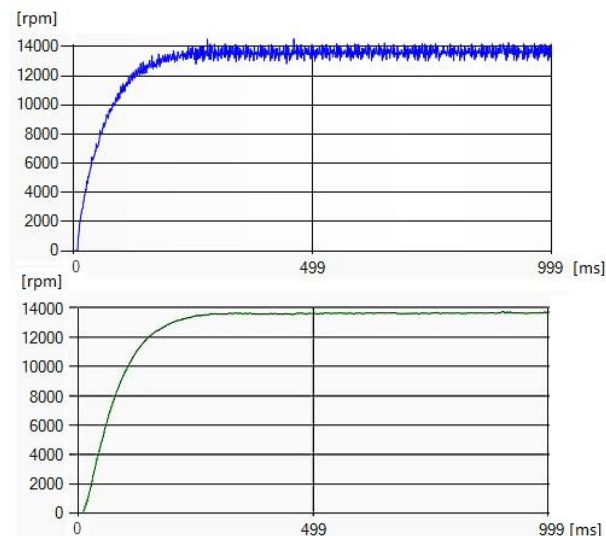


Figura 5: Mediciones a lazo abierto de la velocidad en función del tiempo, sin y con filtro EWMA.

#### 2.4.2.2. Encoders

Tarea destinada a obtener información de los encoders. Envía información a la tarea de control y PID. Se regula su operación por una interrupción de hardware, mediante el uso de un semáforo.

#### 2.4.2.3. Control

Tarea destinada a la lógica y control del sistema. Recibe información de todas las tareas excepto del PID, al cual le envía los resultados de la decisión. Su funcionamiento se regula mediante la llegada de información a la cola de mensajes. Mientras no exista información nueva, permanece dormida. Se implementó un *timer* por software con periodo de 1ms, que se encarga de despertar a la tarea, entre otras cosas.

Para esta tarea se desarrolló un algoritmo de *fuzzy logic*, el cual se puede utilizar, en principio, en competencias de Sumo Robot, pero puede ser adaptado a otras funciones [10]. Este algoritmo de control utiliza la información de los sensores de proximidad como entrada, y las velocidades de los motores como salida. De esta manera, si el sensor izquierdo detecta que el oponente está “muy lejos” y el derecho que está “lejos”, el motor izquierdo “avanza” y el derecho “avanza lento”.

#### 2.4.2.4. Sonar

Tarea destinada a obtener información del sonar. Envía información a la tarea de control. Se regula su operación por medio de una interrupción de hardware y el uso de un semáforo.

#### 2.4.2.5. IR

Tarea destinada a obtener información de los *sensores de proximidad*. Envía información a la tarea de control. Su ejecución es temporizada.

#### 2.4.2.6. Línea

Tarea destinada a obtener información de los *sensores de línea*. Envía información a la tarea de control. Su ejecución es temporizada.

#### 2.4.3. Cola de mensajes múltiples

La cola de mensajes múltiple es una mejora de las herramientas que ofrece el sistema operativo. Su función es permitir la comunicación entre múltiples elementos. La cola de mensajes incluida en el FreeRTOS es bidireccional, pero de solo dos extremos. Si bien es posible acceder a ella desde múltiples tareas, en el momento que una de ellas retire un elemento de la cola de mensajes, este ya no estará disponible para ninguna otra.

La cola de mensajes múltiple es una modificación que permite definir un sistema de comunicaciones entre tareas más eficiente, ya que ofrece la posibilidad de que la información enviada sea recibida en uno o más extremos en forma independiente, es decir, que cada extremo de la cola de mensajes tendrá una copia de la información que le fue enviada para que tenga plena libertad e independencia de operación. El agregado o quitado de extremos de la cola es configurable en *run-time*, es decir, es posible agregarlos o quitarlos en cualquier momento de la ejecución del programa.

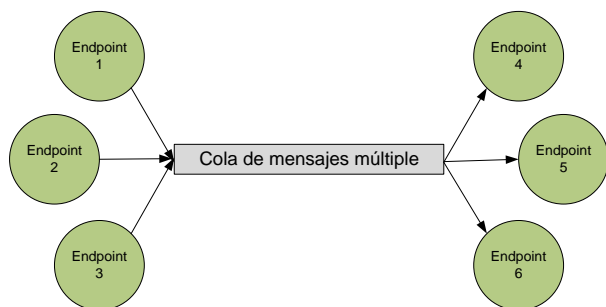


Figura 6: Esquema de la cola de mensajes múltiples agregada al FreeRTOS.

### 3. APLICACIONES

#### 3.1. Sumo de robots

Las competencias de sumo de robots son muy populares entre instituciones educativas y aficionados a la robótica. Consisten en enfrentamientos entre dos robots, que deben empujar a su oponente fuera de un área delimitada por un círculo, denominada tatami.

El desafío para estos robots es poder encontrar a su oponente, usualmente mediante sensores de proximidad o por ultrasonido, y luego empujarlo hasta sacarlo del tatami, evitando abandonar él mismo el área de juego. Estas competencias suelen estar divididas en categorías, en función del peso y tamaño de los robots. MiniPI cumple con las dimensiones y peso exigidos dentro de la categoría mini-sumo. Cabe destacar además que participó de la 9° Competencia Nacional de Robótica, donde recibió una distinción por su diseño.

#### 3.2. Fútbol de robots

Existen en la actualidad distintas competencias de fútbol de robots. El principal objetivo que tienen estas competencias es promover la investigación y el desarrollo de sistemas multi-robots para resolver tareas que requieran de la cooperación de varias unidades. Los problemas implícitos en este tipo de competencias son la formación de equipos de robots, reconocimiento de obstáculos, planeamiento cooperativo, localización y mapeo, distribución de tareas, etc. Algunas de las áreas de la investigación abarcadas son la inteligencia artificial, el procesamiento de imágenes en tiempo real y sistemas de comunicaciones en robótica móvil. Los principales organizadores de este tipo de competencias son FIRA y RoboCup. MiniPI cumple los requisitos para las categorías RoboSot y RoboCupJunior respectivamente.

#### 3.3. Seguidores de línea

Una de las aplicaciones más simples dentro de la robótica móvil es la de seguir una línea pintada o marcada sobre el suelo. Usualmente la política de funcionamiento de los robots es la de moverse hacia adelante mientras detecten la línea, y girar o moverse hacia los costados cuando deje de ser detectada. Otros, en cambio, poseen sensores CMOS o CCD, y se desplazan de manera que la línea se mantenga centrada en la imagen capturada. MiniPI puede actualmente aplicar la primera política mediante los sensores reflectivos, y en un futuro utilizar los sensores de posicionamiento para aplicar la segunda política.

#### 3.4. Resolución de laberintos

Este tipo de competencias implican el uso de algoritmos específicos para su resolución. Los algoritmos de seguidor de paredes, de Pledge o de Trémaux son generalmente utilizados cuando no se cuenta con información previa del laberinto. En cambio, cuando se tiene el mapa del lugar, se utilizan usualmente los algoritmos de relleno de callejones sin salida, o algoritmos de búsqueda como el de búsqueda en anchura o el de A\*(A-estrella) [11]. Los laberintos sin bucles son equivalentes a árboles en teoría de grafos,

y es por eso que esta rama de la inteligencia artificial está asociada con la teoría de grafos. MiniPI participó en la 3° edición de la competencia anual del Club de Robótica de la Facultad de Ingeniería de la UBA, donde obtuvo el tercer puesto en la categoría “laberinto”.

### 3.5. Robótica colaborativa

El principal paradigma de la robótica colaborativa está asociado a la resolución de la coordinación de varios robots para resolver una tarea en forma conjunta. Este enfoque pretende resolver problemas tales como el de inspección de grandes áreas [12], movimiento de materiales, almacenes robóticos, automatización de tareas agrícolas etc., que pueden resolverse en forma costo efectivo con robots simples, adecuadamente coordinados. La tarea que puede resolver el sistema cooperativo debe ser más compleja que la que puede ejecutar cada robot individualmente, por ejemplo el transporte de una carga que por sus dimensiones o por su peso supere la capacidad de un robot individual.

Los sistemas de múltiples robots han atraído recientemente considerable atención por parte de expertos en robótica, ya que los mismos ofrecen la posibilidad de llevar a cabo una tarea que no podría resolver un solo robot por su propia cuenta. El uso de robótica colaborativa permite una solución altamente eficiente, especialmente en tareas de gran dificultad.

## 4. CONCLUSIONES Y FUTUROS TRABAJOS

Se presentó una plataforma robótica inteligente de pequeñas dimensiones, cuyas principales características son su bajo costo, versatilidad, adaptabilidad y posibilidad de expansión. Este robot es apto para ser utilizado en varias competencias de robótica móvil, por lo que tiene un gran alcance dentro del ámbito educativo. A su vez, incorpora hardware y software de avanzada tecnología, siendo también una plataforma de desarrollo para proyectos de investigación sobre inteligencia artificial y robótica.

Actualmente su sistema de comunicación se encuentra en etapa de desarrollo, aunque se espera que esté disponible próximamente. Una vez implementado, se comenzará a trabajar en resolver tareas específicas de la robótica colaborativa. Las aplicaciones de esta rama de la robótica son muy variadas, y van desde tareas de poca dificultad, hasta la resolución de problemas mucho más complejos, que incluso no se podrían resolver mediante robots individuales en tiempos razonables.

## REFERENCIAS

[1] S. Herbrechtsmeier, U. Rückert y J. Sitte, “AMiRo – Autonomous Mini Robot for Research and Education” *Advances in Autonomous Mini Robots: Proceedings of the 6-th AMiRE Symposium*, Bielefeld, Alemania, 2011, pp. 101-112.

[2] A. J. Carimatto, C. Verrastro y J. C. Gómez, “Mapeo y Navegación por nodos de potencial para robótica móvil” *XIII Reunión de Trabajo en*

*Procesamiento de la Información y Control*, Rosario, Argentina, 2009.

- [3] M. Prieto Canalejo, R. Verrastro, L. Di Matteo, J. C. Gómez y C. Verrastro, “Odometría Estereoscópica en Tiempo Real” *IV Seminario de Inteligencia Artificial y Robótica*, Buenos Aires, Argentina, 2012.
- [4] M. J. Paves, “KIT RobotGroup para enseñanza de tecnología y robótica” *Catálogo de Proyectos del Concurso Nacional de Innovaciones*, 7ma edición, p. 291, 2011.
- [5] P. De Cristóforis, S. Pedre y J. Santos, “ExaBot: un robot para divulgación, docencia e investigación” *V Jornadas Argentinas de Robótica (JAR'08)*, Bahía Blanca, Argentina, 2008.
- [6] D. Gaydou, G. Perez Paina, J. Salomone y G. Steiner, “Plataforma móvil de arquitectura abierta” *V Jornadas Argentinas de Robótica (JAR'08)*, Bahía Blanca, Argentina, 2008.
- [7] A. Bonarini, M. Matteucci, y M. Restelli, “Automatic error detection and reduction for an odometric sensor based on two optical mice” *IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, España, 2005, pp. 1675-1680.
- [8] C. Verrastro, S. Alberino, P. Folino y J. C. Gómez, “Fast self tuning PID controller specially suited for mini robots” *AMIRE: 4th International Symposium on Autonomous Minirobots for Research and Edutainment*, Buenos Aires, Argentina, 2007, pp. 223-230.
- [9] S. Alberino, P. Folino, C. Verrastro y J. C. Gómez, “Criterios de Sintonización y Estabilidad del Control dEWMA-PID” *XIV Reunión de Trabajo en Procesamiento de la Información y Control*, Entre Ríos, Argentina, 2011, pp. 96-101.
- [10] R. Cignoli, “Control Difuso para Sumo de Robots” *Simposio Argentino de Sistemas Embebidos 2012*, Buenos Aires, Argentina, 2012
- [11] F. Carmona, D. Lopez Amado, D. De Biase, S. Verrastro, C. Foti, L. Di Matteo, A. C. Mangone, R. Barneda y C. Verrastro, *Aplicación de Técnicas de Robótica e Inteligencia Artificial sobre un robot móvil utilizando el entorno de programación Visual Studio.Net*. Buenos Aires, Argentina: Microsoft Academic Alliance Repository, 2006.
- [12] W. Burgard, M. Moors, D. Fox, R. Simmons y S. Thrun, “Collabortive multi-robot exploration” *IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, Estados Unidos, 2000, pp. 476-481.