

Desarrollo de un laboratorio abierto de enjambres de robots autónomos de limpieza

Laura Pozueco, José Antonio Sánchez, Alejandro G. Tuero, David Melendi, Roberto García, Xabiel G. Pañeda, Noemí Asenjo, Oscar Quintana, Javier Viñuela, Pedro B. López, Adrián Santinho

Departamento de Informática

Universidad de Oviedo

2.7.4, Edificio Polivalente, Campus de Xixón, s/n, Asturias, España

pozuecolaura@uniovi.es, sanchezsjose@uniovi.es, garciatalejandro@uniovi.es, melendi@uniovi.es, garciaroberto@uniovi.es, xabiel@uniovi.es, UO232277@uniovi.es, UO187253@uniovi.es, UO217178@uniovi.es, UO83450@uniovi.es, UO212296@uniovi.es

Resumen- En el corto plazo, la industria del robot autónomo y de los drones será un factor de desarrollo importante. Tanto la fabricación de los propios dispositivos, como del desarrollo del software que los hace funcionar serán actividades empresariales de importancia en el sector TIC. No obstante, el futuro de los robots de trabajo autónomo es colaborar unos con otros y con el entorno que los rodea (adaptación al contexto). Para ello, deben de ser capaces de interactuar con otros equipos mediante protocolos de comunicación y sistemas de razonamiento. Por ello, en este trabajo se presenta una experiencia encaminada a la obtención de un sistema de trabajo colaborativo para que un grupo/enjambre de robots autónomos de limpieza puedan trabajar de forma conjunta. La experiencia se basa en un laboratorio abierto que permite a los alumnos proponer y realizar sus propios desarrollos. En el trabajo se presentan tanto los aspectos metodológicos de la experiencia, como los avances que se han conseguido realizar hasta la fecha.

Palabras Clave- robots autónomos, sistema colaborativo, project oriented learning, laboratorios abiertos

I. INTRODUCCIÓN

Los robots de limpieza suelen trabajar de forma individual. Mediante unos sensores, se desplazan por una superficie que puede tener pequeños obstáculos. Internamente, disponen de una serie de algoritmos que les permiten maximizar el área de limpieza. No obstante, cuando se plantea su despliegue para cubrir una gran superficie, puede llegar a ser necesario utilizar varios de estos robots. En una situación ideal, los robots serían capaces de colaborar unos con otros con la

finalidad de repartirse la superficie a recorrer o trabajar por turnos. El funcionamiento coordinado en forma de enjambre, no solo les permite ser muy eficientes, sino que también viene acompañado de otras ventajas. Algunas pueden ser la fiabilidad, debido a la redundancia, o la capacidad de disponer de un sistema con altas capacidades de cómputo, gracias a la paralelización [1] No obstante, hasta donde los autores pueden saber, no existen productos comerciales de limpieza capaces de operar en forma de enjambre.

Con la finalidad de disponer de robots que operen de forma coordinada, el grupo de investigación DMMS de la Universidad de Oviedo ha diseñado un laboratorio que se centra en el desarrollo de un sistema de trabajo colaborativo para robots autónomos de limpieza comerciales. El desarrollo de este sistema parte de la necesidad de ser capaces de ejercer un control sobre el funcionamiento de los robots que vaya más allá de funcionamiento convencional. Adicionalmente, es necesario incorporar funciones básicas de comunicación. Idealmente, estas comunicaciones serán inalámbricas, utilizando mecanismos ad-hoc y/o elementos de infraestructura. Además, es necesario crear un protocolo de comunicaciones de alto nivel. Este protocolo, sobre la infraestructura inalámbrica subyacente, permitiría el intercambio de mensajes necesario para llevar a cabo el trabajo colaborativo. Finalmente, deberá crearse un sistema de razonamiento que les permita analizar los mensajes recibidos y la información capturada de un conjunto de sensores, para

reaccionar en consecuencia. Todo esto permitiría la implementación de aplicaciones finales.

Toda la experiencia se ha enmarcado en el paradigma de los espacios de trabajo abierto, siguiendo el planteamiento de [2], y el paradigma del *project oriented learning* (POL). De esta forma, se permite a los estudiantes utilizar los materiales del laboratorio para desarrollar sus propios proyectos. La única restricción es que todos deberán tener un objetivo general definido: conseguir que lo que diseñen limpie de la forma más eficiente posible. Cada estudiante o equipo podrá trabajar definiendo su contexto de limpieza, tipo de local o zona, si requieren comunicaciones entre los robots, sensores, inteligencia, etc. De esta forma, el laboratorio del proyecto se ha configurado como un entorno de aprendizaje cooperativo entre los estudiantes, en el que proyectos anteriores podrán ser utilizados como base de conocimiento para los nuevos diseños. Así se definirá un proceso evolutivo dentro del propio laboratorio en el que generación tras generación de proyectos se evolucione la tecnología, experimentando los alumnos la situación real de los desarrollos tecnológicos en la industria. Los aspectos pedagógicos del laboratorio se describen en [3].

El resto del artículo se ha estructurado de la siguiente forma. En la Sección II se describen los aspectos metodológicos del proyecto. En la Sección III se presenta el sistema de base que ha sido utilizado. En la Sección IV se describen los distintos proyectos que han sido completados hasta la fecha. Finalmente, la Sección V presenta las conclusiones y algunos trabajos futuros.

II. ASPECTOS METODOLÓGICOS DEL LABORATORIO

La metodología seguida en el proyecto se ha basado en ciclos evolutivos de creación y evaluación de prototipos centrados en robots autónomos. Es decir, se van construyendo nuevos proyectos tomando como partida los proyectos que han sido desarrollados con anterioridad. La idea es la de crear un prototipo inicial o sistema base, sobre el cual se vayan aplicando diferentes funcionalidades para conseguir una evolución del mismo. Esta evolución puede ir encaminada en la mejora del intercambio de información entre los robots, en el desarrollo de métodos de comunicación o en la obtención de un sistema final.

Desde un punto de vista pedagógico, los alumnos desarrollan sus proyectos partiendo del trabajo de otros alumnos. Por ello, la única restricción que tienen es la de construir módulos que puedan ser reaprovechables. En este sentido, es muy importante el diseño del software desarrollado y la calidad de la documentación generada por los alumnos. Esta forma de trabajar, se ilustra en la Fig. 1.

Además de un correcto diseño y de una buena documentación, los prototipos desarrollados por los alumnos deberían pasar un pequeño control de calidad. Tras las pruebas realizadas en un entorno de desarrollo

convencional, se realizarían pruebas de campo en un laboratorio diseñado al efecto. El laboratorio es un aula vacía en la que se han instalado dos cámaras que permiten la monitorización remota de los experimentos, y su potencial control mediante técnicas de visión por computador. Una cámara fija obtiene un plano general del laboratorio, y la cámara robotizada cuyo interfaz se muestra en la Fig. 2 realiza el seguimiento del robot.

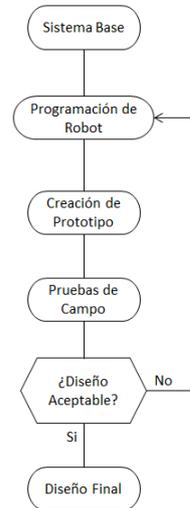


Fig. 1. Flujo de tareas del proyecto.

Para evitar problemas con el seguimiento del robot por parte de la cámara robotizada, se han instalado vinilos en las ventanas del laboratorio. Antes de la instalación, la cámara detectaba movimiento fuera del laboratorio y se perdía el rastro del robot. Adicionalmente, las pruebas siempre se realizan bajo las mismas condiciones de iluminación para evitar problemas con el tratamiento de las imágenes capturadas con la cámara.

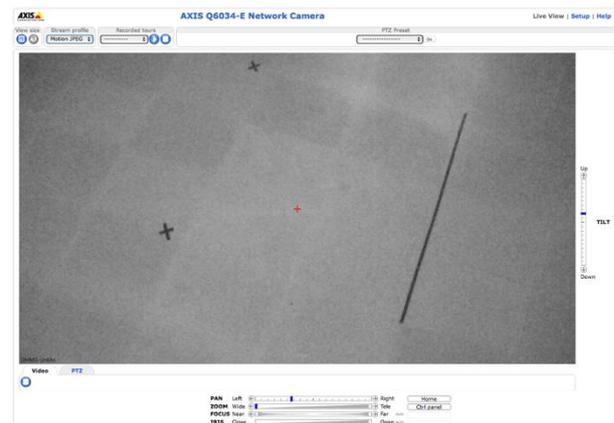


Fig. 2. Interfaz de usuario de la cámara robotizada instalada en el laboratorio de experimentación.

III. DESCRIPCIÓN DEL SISTEMA BASE

El laboratorio ha utilizado como punto de partida el kit de desarrollo que la empresa iRobot (spin-off del MIT) dentro de su programa *Create 2* [4]. Este programa se basa en los robots que fabrica esta

empresa, dentro de la serie Roomba. El kit incluye el robot de prototipado que se muestra en la Fig. 3, y permite la incorporación de nuevo hardware con su software correspondiente. No en vano, los robots Roomba han sido utilizados en entornos educativos desde hace tiempo [5], por tratarse de un recurso de bajo coste aplicable en el aprendizaje y la investigación en el campo de la robótica.



Fig. 3. iRobot Create 2.

Dado que el robot Roomba funciona de forma autónoma e independiente, lograr el trabajo cooperativo implica dotar a este dispositivo de algún tipo de elemento capaz de ejecutar programas y de permitir comunicaciones inalámbricas entre los robots, o con algún tipo de infraestructura. Por lo tanto, los inicios del proyecto consistieron en el establecimiento de un sistema base que contemplase estos requisitos. Para ello, seleccionamos equipos Raspberry Pi debido a su flexibilidad, a su coste reducido y al hecho de que disponen de comunicaciones inalámbricas (integradas en las versiones más recientes y a través de un adaptador USB en las versiones anteriores). Estos aspectos han sido considerados para usos similares en trabajos anteriores como [6].

La interconexión entre el robot y la Raspberry Pi se realiza mediante comunicaciones serie. A través de una conexión de este tipo, la Raspberry Pi puede acceder a la información de los sensores del robot y enviarle comandos para que haga algo (por ejemplo, moverse en una dirección concreta). Esta conexión se realiza utilizando un cable de comunicación serie-USB. El cable se proporciona, dentro del proyecto Create 2, junto con el robot.

Durante el desarrollo del proyecto, también descubrimos que el interfaz serie presente en el robot incorporado en el kit del programa Create 2 también está presente en otros robots del mismo fabricante. Esto nos permitió disponer de dispositivos adicionales, más sencillos de adquirir. En particular, se consiguieron varios robots Roomba modelo 631. Para acceder al interfaz serie, es necesario efectuar un orificio en la cubierta del robot.

Por otro lado, los robots disponen de una batería que les permite trabajar de forma autónoma. No obstante, surge el problema de alimentar eléctricamente la Raspberry Pi en movilidad. Por ello, y dado que el cable provisto en el kit no posee sistema de alimentación, ha sido necesario emplear baterías externas portables que se conectan a la entrada

MicroUSB de las Raspberry Pi. El prototipo completo se muestra en la Fig. 4.

IV. PROYECTOS DEL LABORATORIO

En esta sección se describen algunos trabajos que han sido desarrollados en el marco del laboratorio abierto, siguiendo el marco metodológico expuesto con anterioridad. En particular, han participado alumnos de Grado en Ingeniería, en las disciplinas Industrial/Mecánica, de las Telecomunicaciones y de la Informática.



Fig. 4. Prototipo desarrollado.

A. Desarrollo de un interfaz de control de robots

Más allá de los aspectos físicos del sistema base, era necesario disponer de algún mecanismo que permitiese el control del robot a través de comunicaciones serie. En los primeros prototipos, este control se efectuaba mediante un programa sencillo que se ejecutaba en la Raspberry Pi. Este programa se basaba en la información indicada en la especificación del robot.

La especificación del robot incluye el conjunto de instrucciones que permiten acceder a los valores de los sensores y provocar el desplazamiento del robot [7]. Su funcionamiento se basa en el modelo de estados que se muestra en la Fig. 5.

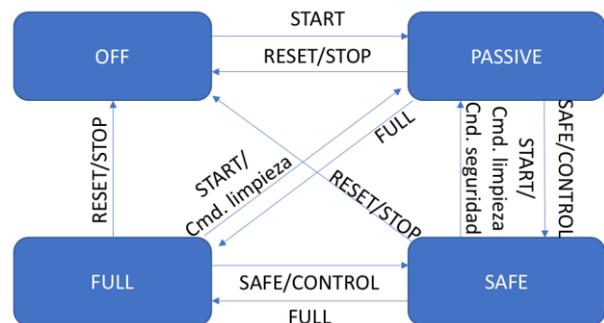


Fig. 5. Modelo de estados del interface del robot.

Los estados se describen a continuación:

- **OFF:** En este modo la interfaz de entrada no está activa hasta que se envía un comando START, tras el cual ya es posible el envío de otros comandos al robot. Por otro lado, en este modo el robot es capaz de enviar información de los eventos que puede detectar, como una pulsación

del botón CLEAN, la separación del robot del suelo, etc.

- PASSIVE: En este modo se entra tras el envío de un comando START o cualquiera de los comandos de limpieza disponibles en la interfaz (CLEAN, SPOT, etc.). No obstante, no es posible cambiar los parámetros de los comandos de actuación (MOTORS, DRIVE, etc.). Para ello es necesario hacer una transición a los modos SAFE o FULL.
- SAFE: En este modo se entra tras un comando SAFE, que proporciona un control total sobre el robot, a excepción de algunas condiciones de seguridad. Algunas de estas condiciones son la detección de algún desnivel durante el movimiento hacia delante o hacia atrás, la detección de las ruedas levantadas o la conexión con la estación de carga. Si ocurre alguna de estas condiciones, el robot realiza una transición al modo PASSIVE.
- FULL: En este modo se entra tras la recepción de un comando FULL, que proporciona un control completo del robot sin ningún tipo de restricción.

Teniendo en cuenta el funcionamiento del robot, se hicieron pruebas sencillas de funcionamiento como las siguientes:

- Con el envío de comandos de actuación al robot se comprobó que se movía hacia delante o atrás, que giraba hacia un lado u otro, etc.
- Con los comandos de limpieza se confirmó que limpiaba, encendía los motores de aspiración, etc.
- Con la recepción de información de los múltiples sensores hubo que estudiar cada uno en particular para interpretar correctamente la información.

Una vez completadas las pruebas con la aplicación de prototipado, la experiencia conseguida permitió el desarrollo de una librería de programación reutilizable. Esta librería encapsulaba todas las funciones de control del robot en unas clases, con la idea de poder ser utilizadas en programas planteados con un nivel de abstracción mayor.

B. Desarrollo de un subsistema de comunicaciones

Adicionalmente, para permitir el trabajo cooperativo, se desarrollaron mecanismos de comunicación basados en redes inalámbricas ad-hoc. Los dispositivos Rapsberry Pi instalados en los robots, son capaces de comunicarse entre sí e intercambiar información sin necesidad de ningún tipo de infraestructura inalámbrica. Además, para permitir la comunicación efectiva entre las aplicaciones a desarrollar, se ha definido un protocolo de comunicaciones de alto nivel. Respetando la filosofía de laboratorio, el diseño se ha planteado de forma que se obtenga un middleware reutilizable que permita

abstraer a los desarrolladores de detalles relacionados con la comunicación, como en [8].

En nuestro caso, el protocolo se basa en documentos XML y datagramas UDP y permite intercambiar información entre los dispositivos respecto a diferentes situaciones que se pueden producir. El protocolo contempla dos tipos de mensajes:

- NetMsg: se utilizan por los procesos de mantenimiento y configuración de la red, pero en ningún caso se proporcionan a la capa de aplicación. Un ejemplo se muestra en la Fig. 6.
- RoomMsg: Contiene información de lo que se desea enviar al resto de robots de la red.
- Ack: Son mensajes de confirmación de mensajes RoomMsg
- Err: Son mensajes de error en el procesamiento de algún mensaje RoomMsg

```
<?xml version="1.0" encoding="UTF-8"?>
<NetMsg len="41" cksum="123" id="42c1304f1">
  <source>0000000042c1304f</source>
  <msg>HI</msg>
</NetMsg>
```

Fig. 6. Ejemplo de mensaje NetMsg.

C. Sistema de inteligencia del robot

El dispositivo Rapsberry Pi también se ha dotado de un programa que hace las funciones de cerebro del sistema. El funcionamiento de este programa se basa en el concepto de "situación". Una situación es una herramienta que permite modelar un escenario determinado en un documento XML. Está formada por uno o varios sensores, una o varias acciones y una o varias señales que deben enviarse al resto de robots. El administrador puede configurar estas situaciones para implementar escenarios concretos. Un ejemplo de situación se muestra en la Fig. 7.

```
<situacion id="robo">
  <sensor name="RIGHT_WHEEL">1</sensor>
  <sensor name="LEFT_WHEEL">1</sensor>
  <externa>LED 0,0,0,1,128,128</externa>
  <actuacion tipo="SONG">40,50</actuacion>
</situacion>
```

Fig. 7. Ejemplo de situación.

El diseño de alto nivel del sistema de inteligencia se muestra en la Fig. 8.

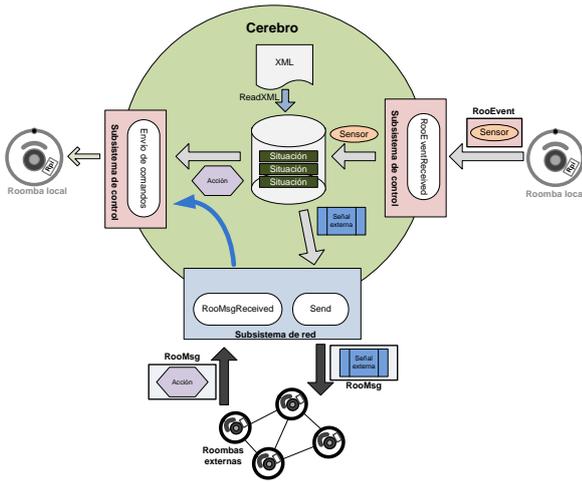


Fig. 8. Cerebro del sistema.

Básicamente, se examina el comportamiento del robot en el que se ejecuta la aplicación, y se generan eventos que contienen la información leída de los sensores. Estos eventos se comparan con la información de la base de datos local, de forma que si alguno de los valores de un sensor, o sensores, coincide con lo establecido en una situación determinada, se obtienen las acciones y señales externas. Las primeras se ejecutan en el robot y las segundas se envían al resto de robots Roomba. En caso de que se reciba un mensaje de otro robot, éste contiene la acción que debe ser realizada.

La combinación de los tres primeros proyectos, es la base para desarrollar nuevas aplicaciones colaborativas. La filosofía es la misma que la de los casos anteriores. Los desarrollos se construyen de forma incremental y se encapsulan en librerías de programación para futuros proyectos.

D. Banco de pruebas para sistemas Roomba

Los robots utilizados en el proyecto se desplazan gracias a una pareja de ruedas motrices que giran de forma independiente. Este diseño permite al robot hacer giros de 360 grados sin necesidad de desplazarse. Debido a este diseño, fue necesario desarrollar un banco de pruebas que permitiese operar el robot en un espacio reducido (por ejemplo, encima de una mesa). Este banco de pruebas consistiría en un sistema de rodillos que permitiría fijar el robot en una posición estable.

El primer prototipo desarrollado para la ejecución de las pruebas se muestra en la Fig. 9. Para implementar el prototipo, se ha partido de un banco de rodillos comercial. Se han aprovechado la parte superior del banco y los rodillos. Como se puede ver en la figura, se cortaron los carriles laterales y se agujerearon de nuevo para introducir los rodillos con la distancia adecuada para las ruedas de los robots. Las piezas fueron montadas sobre una tabla de madera, que actuaba como bastidor.



Fig. 9. Primera versión del banco de pruebas.

Tras las primeras pruebas con el prototipo, quedó de manifiesto que el banco de ensayos podía establecer unas condiciones de funcionamiento muy similares a la realidad. No obstante, se constató que se producían vibraciones en el robot. Los problemas se debían al diseño del soporte central del robot, a la excentricidad de los rodillos y a la falta de rodamientos. Por todo lo anterior, se decidió realizar un nuevo diseño más elaborado. El resultado final se muestra en la Fig. 10.

El banco se complementa con una tapa de acero dotada de una goma sobre la que se deposita el robot. Adicionalmente, tiene dos aberturas que permiten depositar las ruedas del robot sobre los rodillos. Adicionalmente, las instrucciones de desplazamiento y de giro que se dan al robot se pueden controlar realizando mediciones sobre los rodillos. Para ello, el banco se ha equipado con una placa Arduino conectada a un circuito con sensores de efecto Hall. En concreto, para cada pareja de rodillos se instaló un imán y 3 sensores separados 180°.

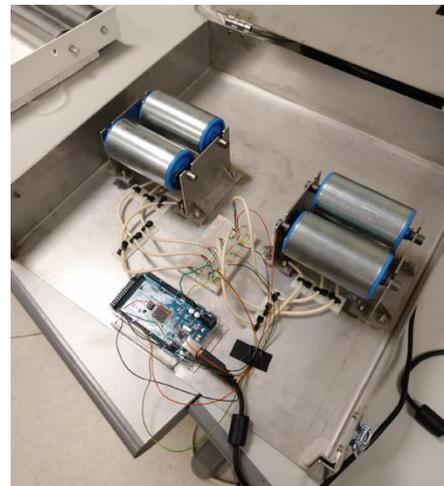


Fig. 10. Banco de pruebas desarrollado.

E. Aplicación RooBra

Esta aplicación es un sencillo sistema de alarma que se activa cuando un robot se separa del suelo. Su funcionamiento se puede observar en la Fig. 11. Cuando la lectura de los sensores del robot indica que éste equipo se ha separado del suelo, se genera una alarma que se propaga hacia el resto de elementos de la red. La propagación se realiza mediante comunicaciones ad-hoc utilizando el paradigma *store*,

carry and forward. Cuando los robots reciben este mensaje de alarma, simplemente encienden un led de color rojo.

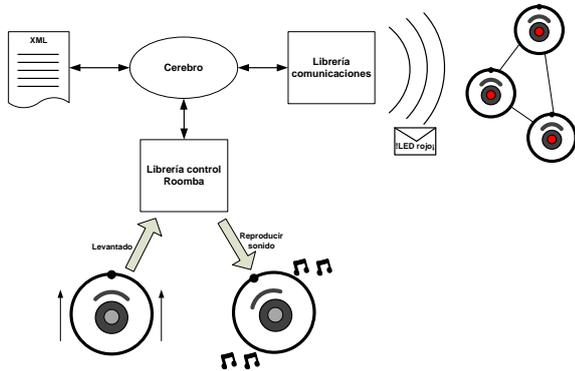


Fig. 11. Aplicación RooBra.

F. Sistema de carga colaborativa

Los robots están equipados con una batería que es necesario recargar de forma periódica. Cuando un robot detecta que su batería está a punto de vaciarse, deja el trabajo de limpieza para dirigirse a un punto de recarga. El problema surge cuando hay varios robots en una misma zona, que comparten una única estación de recarga. Para permitir una utilización solidaria del punto de recarga, esta aplicación permite que varios robots se coordinen. El comportamiento normal del robot es el de permanecer en la estación de carga mientras no se encuentra limpiando. En este contexto sería imposible que varios robots pudiesen usar la misma estación. Esta aplicación permite liberar la estación de carga para que sea utilizada por el robot que tiene menos capacidad almacenada. Un escenario de los producidos en esta aplicación se muestra en la Fig. 12.

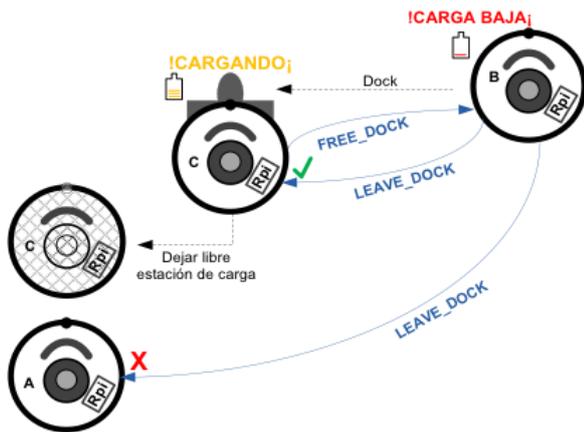


Fig. 12. Aplicación de carga colaborativa.

G. Sistema de trabajo dirigido

Esta aplicación permite dar instrucciones a los robots en función de las imágenes que se toman desde una cámara. Utilizando técnicas de visión por computador y las imágenes capturadas por las cámaras del laboratorio, la aplicación es capaz de detectar puntos de suciedad, tal y como se observa en la Fig. 13. Adicionalmente, la aplicación es capaz de ubicar al

robot, dentro de la zona observada. De esta forma, el sistema de trabajo dirigido es capaz de proporcionar instrucciones de desplazamiento al robot, relativas a su posición actual. De esta forma, se consigue dirigir al robot para limpiar las zonas en las que se ha detectado suciedad.

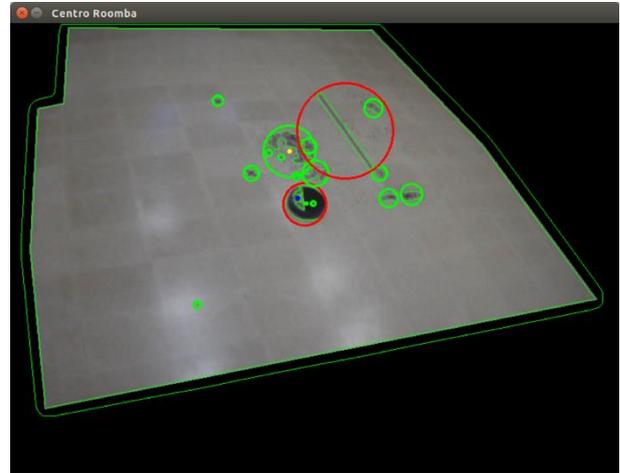


Fig. 13. Sistema de trabajo dirigido.

H. Sistema de control basado en Arduino

En este proyecto, se pretende obtener un sistema equivalente al sistema básico, pero utilizando una placa Arduino. Estos dispositivos son un alternativa a las Raspberry Pi. En concreto, se ha utilizado una placa Arduino UNO.

Tal y como se observa en la Fig. 14, la implementación de la comunicación serie con el robot se ha realizado a través de los pines de entrada salida de la placa, dejando libre el puerto USB para otros usos. Para las tareas de comunicaciones inalámbricas se emplea un chip ESP8266. En este momento se ha implementado un módulo para el control de robot y se han realizado pruebas de concepto del protocolo de comunicaciones, pero no se han iniciado los trabajos de desarrollo del subsistema de inteligencia.

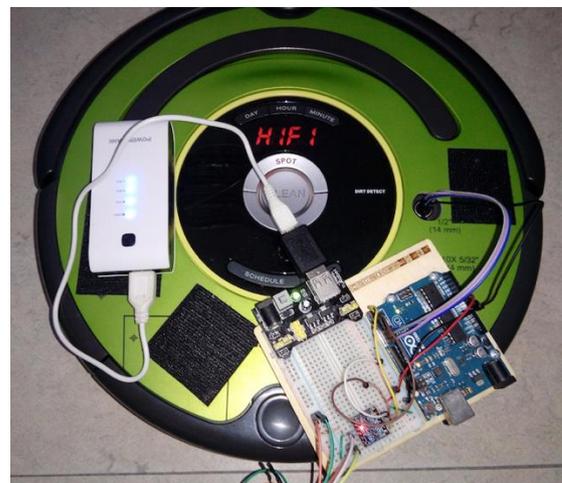


Fig. 14. Prototipo de sistema de control basado en Arduino.

V. CONCLUSIONES

En este artículo se han presentado distintos aspectos y el estado de evolución del proyecto de desarrollo de un laboratorio abierto de enjambres de robots autónomos de limpieza, emprendido por el grupo de investigación DMMS de la Universidad de Oviedo.

En un proceso de evolución continua, se han llevado a cabo distintos proyectos encaminados a la obtención de un sistema colaborativo de robots. Se han creado mecanismos de control de los robots y funciones de comunicaciones que permiten el intercambio de información entre los miembros del enjambre de robots. Por otro lado, los robots son capaces de procesar la información que recogen de sus sensores y combinarla con los mensajes que reciben de otros robots para tomar decisiones.

El enfoque pedagógico que se ha dado al proyecto tampoco está ausente de algunos aspectos innovadores. La concepción del laboratorio de base como un espacio abierto de trabajo, permite a los estudiantes disponer de material y equipos sobre los que implementar sus propios diseños. Al mismo tiempo que se enriquece el proyecto, se crea un espacio para la innovación. Todo esto ha permitido la defensa de distintos trabajos fin de estudios, tanto a nivel de grado como de máster.

El sistema de inteligencia que se ha implementado en los robots, les permite tomar decisiones previstas con anterioridad. No obstante, nos gustaría dotar al sistema de mecanismos de aprendizaje automático que le permitan mejorar sus resultados. Esto podríamos enmarcarlo dentro de los trabajos futuros que se plantea el grupo de investigación. La ubicación en interiores también es algo que necesita cierto desarrollo. Ser capaces de ubicar los distintos robots que operan en un área concreta, nos permitiría distribuir el trabajo de limpieza de forma más eficiente. Finalmente, sería necesario diseñar un sistema que permita alimentar las

Raspberry Pi desde la propia batería del robot Roomba, evitando la necesidad de instalar una batería adicional para esta función.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Instituto Universitario de Tecnología Industrial de Asturias (IUTA) a través del proyecto "Desarrollo de un sistema de trabajo colaborativo de robots autónomos de limpieza".

REFERENCIAS

- [1] Gerardo Beni, "From Swarm Intelligence to Swarm Robotics", en *Lecture Notes in Computer Science*, vol 3342. Springer, 2005.
- [2] University of Dayton, "Open Lab: University of Dayton, Ohio" [En línea]. Disponible en: <https://www.udayton.edu/artsscience/about/casit/labs/index.php> [Accedido por última vez: 10-may-2017].
- [3] José Antonio Sánchez, Laura Pozueco, Alejandro G. Tuero, Noemí Asenjo, David Melendi, Roberto García, Xabiel G. Pañeda y Gabriel D. Orueta, "Laboratorio abierto para el desarrollo de proyectos con robots de limpieza autónomos" En actas del Congreso Tecnología, Aprendizaje y Enseñanza de la Electrónica, pp. 363-368, Sevilla, 2016.
- [4] iRobot, "iRobot-Create 2" [En línea] Disponible en: <http://www.irobot.com/About-iRobot/STEM/Create-2.aspx> [Accedido por última vez: 10-may-2017].
- [5] Ben Tribelhorn y Zachary Dodds, "Evaluating the Roomba: A low-cost, ubiquitous platform for robotics research and education", en actas del IEEE International Conference on Robotics and Automation, Roma, 2007.
- [6] Visislav Celan, Ivo Stancic y Josip Music, "Cleaning up Smart Cities - Localization of Semi-Autonomous Floor Scrubber", en actas del International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, Croacia, 2016.
- [7] iRobot, "iRobot Create 2 Open Interface (OI)". 2015.
- [8] Danilo H. F. Menezes, Marco T. Chella y Hendrik T. Macedo, "A Client/Server Message Oriented Middleware for Mobile Robots", en la revista *Journal of Software*, vol. 7, no. 5, 2012.