Tesis de Máster

Sistema de Negociación Orientado a Grupos de Agentes Autónomos en Entornos con Comunicaciones Limitadas Inherentes

Manuel Martín Ortiz

Directores: Félix de la Paz, Javier de Lope

Madrid, Septiembre 2012

Máster en Inteligencia Artificial: Sistemas Inteligentes de
Diagnóstico, Planificación y Control. UNED

A mi familia,
especialmente a Petate

Índice general

Appendices

Índice de figuras

Resumen

En esta memoria se expone de forma detallada todo el trabajo desarrollado para la asignatura Trabajo Fin de Máster del Máster en Inteligencia Artificial: Sistemas Inteligentes de Diagnóstico, Planificación y Control impartido por la Universidad Nacional de Educación a Distancia y cursado entre los años 2009 y 2012.

Este trabajo pretende contribuir a la compleja tarea que supone la coordinación de sistemas multi-robot autónomos. Para ello se expone una revisión bibliográfica actualizada a su fecha de defensa, la exposición de una hipótesis basada en un sencillo principio matemático y la valoración de los resultados obtenidos tras la realización de diferentes experimentos que pretenden evaluar la hipótesis inicial dentro del marco de la robótica autónoma.

Capítulo 1

Introducción

Los últimos 20 años han supuesto una gran revolución en el mundo de la robótica, que ha visto colmadas muchas de las perspectivas que se plantearon los primeros investigadores de una incipiente inteligencia artificial en la década de los años 40 del siglo XX. Desde principios de los 90 hemos sido testigos de grandes avances en las dos grandes áreas que componen la robótica, y es que tanto la inteligencia artificial como los tecnología asociada a la robótica se han beneficiado de grandes y significativos avances.

Hasta ese momento, la robótica industrial avanzó a pasos de gigantes, mientras que la robótica autónoma debía aún luchar con las muchas barreras que supone un entorno no controlado. Los paradigmas de control de los robots requerían un profundo conocimiento del entorno y se veían lastrados por la posibilidad de encontrar elementos imprevistos. Fue entonces cuando Brooks y sus colaboradores [5, 6] propusieron un nuevo paradigma de control reactivo (y sus respectivas variaciones híbridas) que rompía con un enfoque clásico orientado a ciclos de percepción, deliberación y actuación.

Este enfoque clásico sigue estando vigente y se aplica con un éxito rotundo en aquellos entornos y tareas en los que se tiene bajo control la mayor parte de las variables, como puede ocurrir en las cadenas producción. El nuevo enfoque reactivo, aun siendo menos preciso en la consecución de tareas, presenta grandes beneficios en entornos variables donde apenas se puede predecir o intuir el medio que rodea al robot. Esta nueva corriente sentó la bases ideológicas y de control que ha permitido a otros investigadores diseñar y construir vehículos capaces de conducir de forma completamente autónoma [58] o explorar a nuestro inhóspito vecino Marte [37, 57].

El uso de varios robots de forma simultánea fue otro de los movimientos que nacieron y se expandieron a lo largo de esta década. Arkin [4, 3] o Kube y Zhang [26, 25] presentaron trabajos inspirados en la naturaleza donde se emplean varios robots, que bien pueden se heterogéneos, para alcanzar una meta común por medio de una relación basada en la colaboración, la competencia o una mezcla de ambas. Así nació un nuevo campo de la robótica conocido como "*swarm robotics*" y que, junto a los humanoides, está siendo objetivo de grandes esfuerzos en recursos económicos y humanos. La Unión Europea, a través de su Programa Marco de colaboración, ha financiado proyectos como SYMBRION[1] o REPLICATOR[2] cuyo objetivo es estudiar y desarrollar principios de evolución y adaptación para comunidades de robots homogéneos.

Este auge en inteligencia artificial está fuertemente ligado al avance de las tecnología de computación, que ha proporcionado al mercado dispositivos generales de procesamiento cada vez más potentes, pequeños y de menor consumo. Algo que ha propiciado la aparición de numerosos robots educativos como el ePuck [40], Alice [7] o el LEGO Next, robots orientados a la investigación como iRobot o Qbo y otros orientados a un mercado de consumo que día a día reclama más este tipo de máquinas. Desarrollar tu propio robot no es algo que esté ya solo al alcance de grandes compañías y universidades. Es algo al alcance de cualquier persona y ello está haciendo que la robótica sea un campo que cuenta cada día con más entusiastas, aficionados y profesionales; recibiendo así el beneficio que todas sus aportaciones suponen.

Este mismo avance tecnológico también ha permitido y facilitado el desarrollo de nuevos sensores que proporcionan una adquisición de datos cada vez más completa, compleja y rápida. Gracias a sensores como el desarrollado por PrimeSense para reconstrucciones de imágenes en 3D (utilizado por la cámara Kinect de Microsoft) o sensores CCD, CMOS o sCMOS que permiten una captura de imágenes rápida y de gran calidad, hemos sido testigos de robots capaces de identificar objetos, seguirlos, reconocer rostros humanos e incluso, de reconocerse a si mismos en un espejo[3]. La inteligencia artificial aplicada a la robótica requiere de sensores precisos y rápidos que le permitan llevar a cabo elaborados algoritmos para una mejor interacción con el entorno, y todos estos nuevos sensores están facilitando enormemente la tarea a investigadores de todo el mundo que pueden

---

[1] `http://www.symbrion.eu`

[2] `http://www.replicators.eu`

[3] QBO Robot in front of a mirror - `http://youtu.be/TphFUYRAx_c`

centrarse de forma más cómoda en el desarrollo de nuevas técnicas y algoritmos.

Mientras que uno de los grandes beneficiarios de los avances de la tecnología han sido los robots de pequeño tamaño, otros actores de la industria nos están presentando nuevas máquinas que hasta hace poco tiempo sólo podían encontrarse en la literatura. En los últimos años hemos visto robots humanoides capaces de correr o subir escaleras[4], cuadrúpedos con una gran capacidad de equilibrio y velocidad en carrera[5], cuadricópteros autónomos con una gran precisión de movimiento que llevan a cabo construcciones de forma coordinada[6] o los silenciosos e invisibles robots que pueblan internet y que día a día recaban información, la organizan, analizan y aprenden de ella para proporcionar al usuario un acceso más granulado y enfocado a sus intereses y aficiones.

Tal vez sea demasiado arriesgado afirmar que la robótica está sufriendo su particular revolución industrial, pero es innegable que estos últimos años han permitido desarrollar de manera imparable las bases que se sentaron a lo largo de los años 50 y 60. La aplicación de todos aquellos principios, el avance de la tecnología y la popularidad que está alcanzando la robótica a nivel doméstico tanto desde el punto de vista lúdico como de consumo, auguran un gran futuro para una de las áreas industriales y de investigación más prometedoras del panorama actual. Sin duda, dentro de esta pequeña revolución o simplemente gran evolución, la robótica colaborativa tendrá un papel importante, y es aquí donde el trabajo expuesto a continuación enfoca su investigación y aplicación.

La memoria del trabajo desarrollado queda dividida de la siguiente forma: En el siguiente capítulo se recaban diferentes trabajos científicos que han servido de base o apoyo al aquí expuesto. A continuación, en el capítulo 3, se detalla el marco en el que se engloba, el objetivo marcado para esta contribución, la metodología planeada para su validación y los resultados esperados. Tras ello, en el capítulo 4 se pasa a describir de una forma más completa los experimentos diseñados, donde se explican las pruebas efectuadas y se presentan los resultados obtenidos. Este trabajo finaliza en el capítulo 6 con una exposición de las conclusiones a las que los autores han llegado y las posibles líneas de investigación y trabajo futuro a las que puede dar pie. Todo el trabajo desarrollado y aquí presentado ha dado como fruto la publicación de artículos científicos en diferentes congresos y revistas, los cuales son añadidos como apéndices al final de esta obra.

---

[4] Highlights of ASIMO at the 2011 Sundance Film Festival - `http://youtu.be/0tRo6a4VhvU`

[5] Cheetah Robot Gallops at 18 mph - `http://youtu.be/83ULlgpT1UQ`

[6] Flying robots, the builders of tomorrow - `http://youtu.be/xvN9Ri1GmuY`

Capítulo 2

Revisión de la bibliografía

El ser humano ha tomado siempre la naturaleza como fuente de inspiración. En ingeniería, uno de los ejemplos más claros tal vez los tengamos en los diseños de Leonardo DaVinci, en trajes que permitiesen al ser humano emular el vuelo de los pájaros. En el caso de la robótica de enjambre, la inspiración ha nacido de los grupos de animales con organizaciones aparentemente sencillas pero robustas. A principios de los años 90, Dorigo [8] publicó el que tal vez sea el primer trabajo científico centrado en la aplicación de la organización de las colonias de hormigas en el ámbito de la ingeniería. Más tarde, la inspiración que reflejan los trabajos de Kennedy y Eberhart [23] nacería de los bancos de peces y las bandadas de pájaros.

A raíz de estos trabajos, Ali Jadbabaie [21] y Reza Olfati-Saber [44] más recientemente, han propuestos mecanismos de organización para grupos de robots. Sostienen sus teorías en los mismos principios que permite a peces y pájaros conservar la estructura de grupo según se desplazan es su entorno. Para la organización de grupos de robots se han propuesto otras soluciones, la gran parte son inspiradas en la naturaleza al igual que las anteriormente citadas, y comprenden aspectos como el reencuentro y agrupamiento de sus integrantes [10, 16], mecanismos de control para el mantenimiento de las formaciones [12, 22] o del despliegue del grupo sobre el terreno [9, 35].

Más en línea con el tema tratado en esta memoria, la coordinación del grupo por medio del consenso, podemos citar a Luc Moreu [42], quien propone un modelo de consenso en el cual los agentes realizan las actualizaciones en función de la proximidad de sus vecinos. En uno de sus trabajos, Olfati-Saber [46] analiza la problemática de alcanzar un consenso en grupo cuando existen variaciones en los tiempos de comunicación y la topología del grupo, un área en la que podemos encontrar varios trabajos [49].

A finales de la década de los noventa se organizó uno de los eventos más populares en la difusión de la robótica: la RoboCup [24]. De forma anual esta convención reúne a grupos de investigación de todo el mundo para poner a prueba, y en igualdad de condiciones, las teorías y el trabajo desarrollado por investigadores de todo el mundo bajo un marco de competición y deportividad. A pesar de existir numerosos grupos y centros de investigación que no participan en la Robocup como escaparate para su trabajo, esta competición se ha convertido con el paso de los años en uno de los referentes para comprobar el resultados de las últimas investigaciones en identificación de objetos, coordinación de grupos y elaboración de estrategias en equipo.

Sin embargo el suelo no es el único entorno en la que se están desplegando unidades autónomas en forma de equipo. El año pasado se presentaron los resultados que están dando el trabajo realizado por Leven [27, 28], quien emplea varios planeadores autónomos, energéticamente independientes, para establecer una red de comunicaciones en caso de emergencia. Esta red móvil pretende ser un apoyo en situaciones de emergencia donde las comunicaciones previas han caído y se necesita una red que de cobertura a un área determinada. Los planeadores reconocen el espacio aéreo y lo dividen en zonas, de tal forma que las sobrevuelan mientras conservan el rango máximo que les permite establecer comunicaciones.

En el otro tipo de entornos, como el agua, se está llevando a cabo un proyecto financiado por la Unión Europea dentro de su programa marco de colaboración (FP7) e identificado con el nombre de CoCoRo [53]. Su principal objetivo es emplear robots submarinos capaces de balancear dinámicamente las tareas pendientes y realizar una vigilancia ecológica, buscando, manteniendo, explorando y cultivando recursos en entornos submarinos. Este tipo de proyectos ambiciosos prometen contribuir enormemente a un campo, la exploración marítima, que sin duda se beneficiará enormemente de los avances en robótica autónoma. Como hemos visto hasta ahora, el nexo común entre estos proyectos, y que se corresponde con el núcleo de la robótica de enjambres, es la coordinación, cooperación y consenso de todos los agentes miembros del grupo.

Generalmente, las técnicas de negociación empleadas para la coordinación y consenso del grupo están inspiradas en las teorías enunciadas por John Von Neumann y Morgenstern y que publicaron en su obra *Theory of games and economic behavior* [60] a mediados del siglo XX. Su trabajo ha modelado nuestro presente a través de teorías económicas y de mercado, tal y como señala Alvin E. Roth [51], además de inspirar a una comunidad tradicionalmente alejada de las teorías económicas en el desarrollo de una tecnología que probablemente sea el futuro industrial de nuestra sociedad.

Por ejemplo, la obra de John Von Newman y Morgenstern, ha inspirado a otros investigadores del campo de la robótica para alcanzar el consenso de grupos de agentes por medio del empleo de subastas [18]. El empleo de estas técnicas no solo se reduce a la forma en la que los agentes han de desplazarse, sino que también han permitido alcanzar consensos cuando la intención del grupo ha sido la identificación y el seguimiento de objetivos precisos [55, 56]. Estas sencillas reglas aparecen en innumerables trabajos, siendo reseñable que otras disciplinas, como la medicina [15], la biología [52] o las ciencias sociales [19, 54] se han visto beneficiadas de unas teorías enunciadas a mediados del siglo XX.

Tal y como analiza W. Ren [50], los agentes comparten la misma información en el momento de la negociación, lo cual les permite alcanzar rápidamente un acuerdo con la seguridad de que ha entrado en juego toda la información posible. A pesar de ser una técnica totalmente válida como han demostrado Olfati-Saber [45] o T. Namerikawa [43], no se han encontrado trabajos donde los sistemas de consenso del grupo contemplen la posibilidad de que la información que los agentes poseen sea incompleta o esté obsoleta.

Generalmente, los mecanismos de coordinación se emplean dentro del grupo de agentes para determinar los objetivos a alcanzar. De esta forma, podemos encontrar un grupo formado por N agentes que emplearán sistemas de consenso con el propósito de establecer los objetivos individuales, o bien el objetivo común, dentro del abanico de posibilidades del que dispongan, en función de la situación particular de cada uno de los integrantes del grupo. Una vez la investigación se encuentra en un punto lo suficientemente maduro como para abandonar los entornos simulados y se da el paso a los entornos reales, existen diferentes aspectos ajenos a los mecanismos de coordinación y consenso que hemos visto hasta ahora: la percepción del entorno y el uso de esta información para interactuar con él.

Tras la percepción del entorno, el agente necesita desplazarse por él para conocerlo, ubicarse e interactuar. Los sistemas de navegación actuales que se emplean en robótica están generalmente basados en técnicas de SLAM. Existen diferentes formas de procesar la información percibida por los sensores para tomar una decisión, por ejemplo, S.K. Pradhan [47] o R. Huq [20] emplean lógica difusa para determinar la ruta que el agente debe seguir en base a la percepción del láser y el mapa que el agente elabora de su entorno. Otras alternativas [30, 29] sugieren el empleo de redes neuronales para este mismo fin.

En nuestro caso particular, hemos optado por técnicas que permitan la navegación en entornos no estructurados y dinámicos basado en el cómputo en tiempo real de las áreas más próximas al robot. Los trabajos basados en este método de centro de área [1], y desarrollados dentro del departamento de Inteligencia Artificial de la UNED, han demostrado la fiabilidad de este sistema en entornos reales y simulados, ofreciéndonos un algoritmo eficiente que ser empleado con robots reales. Este algoritmo ha sido probado en robots Pionner, que gracias al láser que incorpora permite lecturas rápidas y precisas del entorno percibido. Sin embargo, emplear varios robots de este tamaño requiere un gran espacio del que no siempre es posible disponer.

Gracias al avance de la electrónica en los últimos años, hemos sido testigos de múltiples robots de tamaño reducido, como el famoso Khepera [41], el Alice [7], Kobot [59] o el más moderno ePuck [40], para usarlos tanto a nivel doméstico como en laboratorios de investigación. El tamaño de estos robots los hace ideales para experimentos en robótica de enjambre, sin embargo la capacidad de sus sensores limita los experimentos que se pueden llevar a cabo. Otros robots, como la plataforma iRobot sobre la que se montan las famosas aspiradoras y fregonas Roomba y Scooba, disponen de sensores al mismo nivel que los incorporados por el Pionner pero a un precio y un volumen mucho más asequible para los presupuestos y espacios de los laboratorios de investigación. A pesar de ello, su tamaño aún puede imposibilitar el empleo de una decena de estos robots para experimentos complejos en robótica de enjambre. Lamentablemente, no se ha encontrado una solución válida que permita emplear complejos algoritmos de percepción y acción con el entorno empleando pequeños robots.

Capítulo 3

Descripción del Marco Teórico

Uno de los aspectos que suelen asumirse a la hora de diseñar, implementar o presentar una arquitectura destinada al control de un grupo de agentes es la infalibilidad de las comunicaciones entre los miembros del grupo. Gracias a la tecnología actual, las comunicaciones por radio-frecuencia (bien sea usando la familia de protocolos estandarizados 802.11[1], 802.15[2] o similares) proporcionan una gran transparencia en cuanto a los fallos de transmisión a nivel físico se refiere. Los dispositivos que se emplean se encargan de crear el enlace y mantenerlo, conectar de nuevo en caso de pérdida y evitar que el usuario note esos posibles pequeños cortes.

Sin embargo la naturaleza del entorno en la que se encuentra el dispositivo de radio-frecuencia puede impedir que se pueda establecer un enlace. Esto hace que los agentes que integren uno de estos dispositivos para mantener la comunicación con sus compañeros queden aislados durante un tiempo indeterminado. Tiempo que puede ser vital para el desempeño de la tarea. Asumir esa infalibilidad en la cobertura de las comunicaciones en cualquier entorno puede acarrear problemas en grupos de robots si el sistema de negociación o coordinación asume que en todo momento se podrá obtener enlace y comunicación con todos y cada uno de los miembros.

La fiabilidad del hardware es otro de los aspectos que normalmente se asumen. Las probabilidades de un fallo general del robot que lo haga inservible son muy pequeñas, especialmente en entornos controlados, pero en otro tipo de lugares se

---

[1] `http://standards.ieee.org/develop/wg/WG802.11.html`

[2] `http://standards.ieee.org/develop/wg/WG802.15.html`

convierte en algo muy probable. Desplegar un conjunto de robots en estas zonas normalmente asegura que la tarea se completará en algún momento. Y por ello el sistema de negociación y coordinación empleado por el equipo debe contemplar tanto la posibilidad de pérdida completa de un miembro como la incapacidad de crear un enlace que permita conocer el estado actual, posición o evolución de cada uno de los miembros.

De forma paralela a esta corriente que asume una total fiabilidad en comunicaciones y hardware, sería deseable disponer de un sistema que permita al grupo de agentes sobreponerse tanto a la incapacidad de establecer comunicación como a errores irrecuperables del hardware, logrando, eventualmente, la consecución de un objetivo común siempre y cuando quede algún agente del grupo para completarla. Diseñar un sistema que cumpla este requisito ha sido el objetivo que nos hemos marcado. Y para llevarlo a cabo hemos establecido un sistema de negociación propio orientado en esta dirección.

El esquema general de trabajo del equipo no difiere en exceso del que normalmente se encuentra en la literatura: El objetivo común se descompone en tareas, que en número pueden ser superiores a los agentes y solaparse entre ellas, y se lleva a cabo un proceso de asignación para establecer que tareas quedan asignadas a los agentes. Como el número de agentes puede variar en el tiempo tanto en sentido positivo como negativo, no se debería dotar de exclusividad a la asignación tareas, de tal forma que una tarea pueda ser asignada a diferentes agentes a lo largo del tiempo. Este esquema, que disminuye el grado de eficacia en, aumenta sin embargo la robustez del sistema.

La arquitectura propuesta puede ser definida como sigue: *Sistema de negociación para un grupo de agentes de tamaño variable que llevan a cabo una tarea en un entorno donde el establecimiento y cobertura del sistema de comunicaciones no está garantizado.*

Esto nos lleva a debatir otro de los aspectos de la arquitectura: el sistema de negociación. Su existencia implica la necesidad de competencia por recursos, que en este caso serán las tareas en las que se descompone el objetivo global. Los agentes competirán entre ellos para lograr que su tarea sólo sea desempeñada por ellos y no tenga solapamiento con los demás. Para mediar en esta competencia se ha optado por un sistema de subastas.

La subasta es un mecanismo que se ha empleado con éxito en otros ámbitos. El más claro de ellos es la sociedad humana, donde ha demostrado ser una valiosa herramienta para la ubicación eficiente de recursos. Una descripción detallada de los diferentes tipos de subastas y sus motivaciones puede encontrarse en el trabajo de R. Preston McAfee [38]. Las subastas se han empleado en diversas disciplinas, como las comunicaciones inalámbricas [39], aunque sus principales aplicaciones las encontramos en economía.

Sin embargo, el uso de subastas en sistemas multi-robot no es nuevo y Gerkey [18] ya propuso este mismo sistema para coordinar equipos. Conociendo la eficacia de los sistemas de subasta en este marco, hemos querido comprobar si esta misma eficacia seguía vigente cuando los resultados de las negociaciones podían verse afectados por la carencia de participación de todos los involucrados. Es decir, hemos aplicado un sistema de subasta para coordinar a un equipo que carece de la capacidad de comunicarse siempre que lo desee. Nuestro objetivo es comprobar si este sistema de negociación es válido en este tipo de entorno.

El funcionamiento interno de la subasta es sencillo. Una vez se ha cerrado el proceso de aceptación de participantes, un árbitro valora la puja hecha por cada jugador. Las pujas aceptables son aquellas que no solapan con ninguna otra. Este mecanismo lo hemos dividido en dos variantes para contrastar el efecto en el resultado final. Ambas variantes se describen a continuación:

- *No bloqueante*: Cuando una puja es considerada aceptable (no solapa con ninguna otra) se le comunica de forma inmediata al jugador correspondiente y este sale del proceso de subasta, de tal forma que el resto de pujas ya no compiten con esta.
- *Bloqueante*: Cuando una puja es considerada aceptable (no solapa con ninguna otra) se marca como válida pero no se comunica de forma inmediata al jugador correspondiente. De esta forma todas las pujas se comparan entre si.

Hemos establecido una serie de experimentos para comprobar si puede aplicarse un sistema de subasta en entornos como el descrito anteriormente e incide de forma positiva en las tareas que deben desempeñar los agentes. Esperamos que los resultados avalen esta preposición y podamos afirmar que esta arquitectura presenta una ventaja. En el siguiente apartado describimos en que consisten estos experimentos.

Capítulo 4

Marco Experimental

4.1. Descripción

Para comprobar la validez de nuestra arquitectura y la consecución del objetivo enunciado en el apartado anterior, se han diseñado unos experimentos en los cuales un equipo de agentes autónomos debe lograr alcanzar un objetivo común, empleando para ello la arquitectura aquí propuesta. El objetivo global es el siguiente:

Un grupo de agentes autónomos debe visitar, al menos una vez, las aristas que forman parte de un grafo que representa un mapa, conocido previamente por todos ellos. Los agentes podrán hacer uso de las comunicaciones siempre que el entorno lo permita.

Este problema, conocido como el problema del viajante [13], ha sido tratado desde muchos y variados puntos de vista, mediante algoritmos genéticos [11, 14] o redes de neuronas [17] entre otros. Sin embargo, es importante recordar que no es aportar una nueva solución a este problema lo que buscamos con estos experimentos, sino validar nuestra arquitectura empleando un problema real.

Los experimentos han sido divididos en dos fases: simulada y real. Para los experimentos simulados se ha diseñado y construido un simulador sobre el que ejecutar los algoritmos, cuyo objetivo es valorar los resultados de la arquitectura y comprobar cual de sus dos versiones, **bloqueante** o **no bloqueante**, es más apropiada. Los experimentos reales se han llevado a cabo en un entorno construido a tal efecto y en él comprobaremos si los resultados teóricos de las simulaciones se corresponden con un grupo de robots reales.

Todos los experimentos comparten una serie de elementos comunes:

- Agente: Su ciclo de ejecución es independiente del resto del equipo y su comunicación se realiza por medio de un servidor central.
- Servidor Central: Este elemento software ofrece un espacio donde llevar a cabo las negociaciones del equipo y compartir información sobre el estado de las tareas asignadas a cada miembro.

```
while not task_achieved:
  if communication_available:
    update_global_knowledge()
    calculate_best_route()
    join_to_auction_process()
    inspect_my_winner_route()
  else:
    calculate_best_route()
    inspect_my_provisional_route()
```

Figura 4.1: Pseudo-código perteneciente al algoritmo global de ejecución

A continuación, algunos términos que serán empleados a lo largo de los experimentos y que corresponden a estructuras o conceptos:

- Mapa: Se corresponde con una abstracción del entorno en el cual los agentes van a desempeñar su labor. Es usado por los agentes para conocer su posición y elaborar un plan de acción acorde a sus intereses. Su modelo computacional es un grafo, los nodos se corresponden con puntos del entorno desde los cuales se pueden tomar más de una dirección y las aristas son los corredores que comunican dos nodos.
- Ruta: Si visitar todo el mapa es el objetivo común de los agentes, las posibles rutas que se pueden seguir desde la posición actual son las tareas en las que se puede descomponer. Al representarse el mapa como un grafo, una ruta es un *camino*, que de acuerdo con [36] es “una secuencia de arcos

donde todos los arcos están dirigidos en el mismo sentido, es decir, el final de un arco coincide con el inicio del siguiente"[1].

- Profundidad de ruta: Hace referencia al número de aristas o arcos que tiene una ruta. Cuanto menor sea la profundidad mayor serán el número de rutas necesarias para cubrir todo el mapa.
- Tamaño de Ventana: Unidad que mide el tiempo que se espera para iniciar un proceso de subasta desde el momento en el que el primer participante está disponible.

```
while not task_achieved:
  calculate_best_route()

  if communication_possible:
    join_to_auction_process()

    while i_have_possible_routes:
      propose_route()
      if not winner:
        update_knowledge()
        calculate_best_route()

    if not i_have_possible_routes()
        calculate_long_route()

    cover_the_route()

  else:
    go_first_stage()
```

Figura 4.2: Pseudo-código de los agentes

El ciclo de ejecución principal de los experimentos se muestra en la figura 4.1. Tanto para los experimentos llevados a cabo en el simulador como los del entorno real, los agentes se comportan de acuerdo al algoritmo detallado en la figura 4.2, mientras que las negociaciones se ejecutan según el pseudo-código de la figura 4.3 si se comporta de una forma bloqueante o el de la figura 4.4 si su ejecución contempla la variante no bloqueante.

---

[1] Libre traducción del original

```
wait_for_new_players()
while not all_bids_are_winners:
    calculate_winners()
    communicate_results_to_losers()
    receive_new_bids_from_losers()
```

Figura 4.3: Proceso de subasta **bloqueante**

```
wait_for_new_players()
calculate_winners()
communicate_results()
```

Figura 4.4: Proceso de subasta **no bloqueante**

4.2. Simulador

El objetivo del simulador es comprobar la validez de la arquitectura presentada en un entorno controlado, compara las dos versiones propuestas anteriormente y facilitar un entorno en el que, visualmente, poder seguir la evolución de los experimentos. Proporciona dos modos de ejecución: (1) Como sistema desatendido capaz de ejecutar un gran número de experimentos diferentes y guardar estadísticas de sus resultados y (2) como entorno visual en el que poder examinar el devenir de un solo experimento.

Siguiendo una filosofía cercana al mundo Unix para procesos desatendidos, ambos modos de funcionamiento disponen de ficheros de texto para su configuración. La figura 4.5 ofrece una idea del aspecto que ofrece el simulador durante la ejecución de un experimento. En las imágenes se puede apreciar el mundo en forma de grafo, a los agentes representados por robots y las rutas planeadas por estos como líneas naranjas que terminan en su nodo final. Las baldosas de los pasillos marcadas en negro son aquellas que aún no han sido visitadas mientras que las grises son las que, al menos una vez, han sido visitadas por un agente cualquiera.

El simulador contempla la *probabilidad de comunicación* y *probabilidad de fallo hardware* como variables a tener en cuenta en las travesías y negociaciones que llevan a cabo los agentes. Estas variables forman parte del problema que nuestra arquitectura pretende minimizar y por lo tanto es imperiosa su aparición

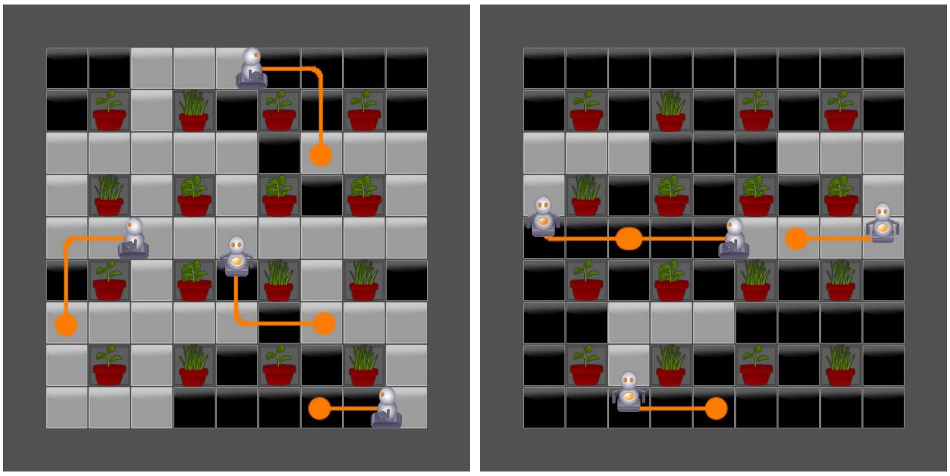


Figura 4.5: Aspecto del simulador en su modo gráfico

en las simulaciones.

En Internet hay alojado un vídeo[2] en el que se aprecia un experimento completo. Este vídeo fue presentado en el *International Work-conference on the Interplay between Natural and Artificial Computation* (IWINAC 2011) como parte de la presentación del artículo *Selective method based on auctions for map inspection by robotic teams* [32] y que está presente como apéndice al final de esta memoria.

4.3. Entorno Real

Los experimentos en un entorno real se han llevado a cabo para comprobar la validez de las simulaciones en un entorno real, en el que no sea necesario aplicar variables estadísticas para los fallos hardware de los robots. Sí se ha conservado la incertidumbre en el establecimiento de comunicaciones debido a la dificultad de generarla de forma natural en un laboratorio.

El entorno real se compone de una mesa con superficie suficiente para albergar varios robots, un grupo de robots ePuck [40] como el de la figura 4.6, una cámara Kinect situada en posición cenital sobre la mesa con una doble función que será detallada más adelante y diferentes obstáculos que los robots deben evitar. El aspecto que presenta este entorno queda ilustrado en la fotografía de

---

[2] `http://youtu.be/E7-w_w-fE60`

la figura 4.7.

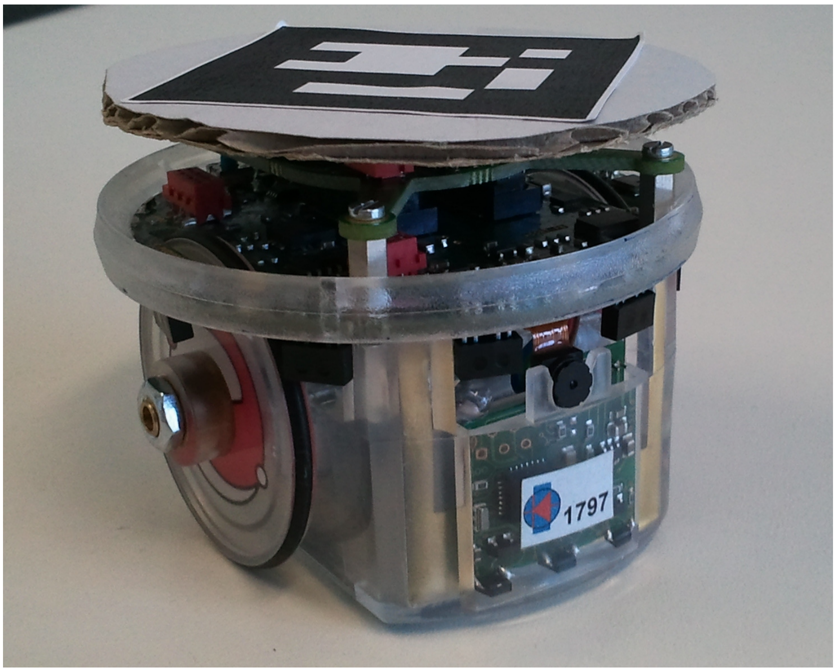


Figura 4.6: Uno de los ePuck empleados en los experimentos con su marca correspondiente

El uso de robots reales presenta ciertos inconvenientes en comparación con la simulación. El principal escoyo es la total ausencia de navegación en las simulaciones, donde se emplean variables de tiempo para representar los tránsitos entre nodos. Un sistema de navegación eficaz necesita de sensores precisos y potentes que lamentablemente el robot ePuck no implementa. Por lo tanto fue necesario recurrir al diseño y construcción de un sensor virtual que facilite la información necesaria.

Es en esta parte donde entra en juego la cámara cenital. Esta cámara se encargará de (1) captar imágenes sobre las que se identificarán los diferentes robots y su entorno, para facilitar la misma información que haría un láser, y (2) permitirá grabar vídeos de los experimentos para su posterior análisis, presentación o retransmisión por Internet.

Para la identificación de los robots se ha hecho uso de una librería implementada por la Universidad de Córdoba y denominada ArUcO[3]. Esta librería hace uso

---

[3] `http://www.uco.es/investiga/grupos/ava/node/26`

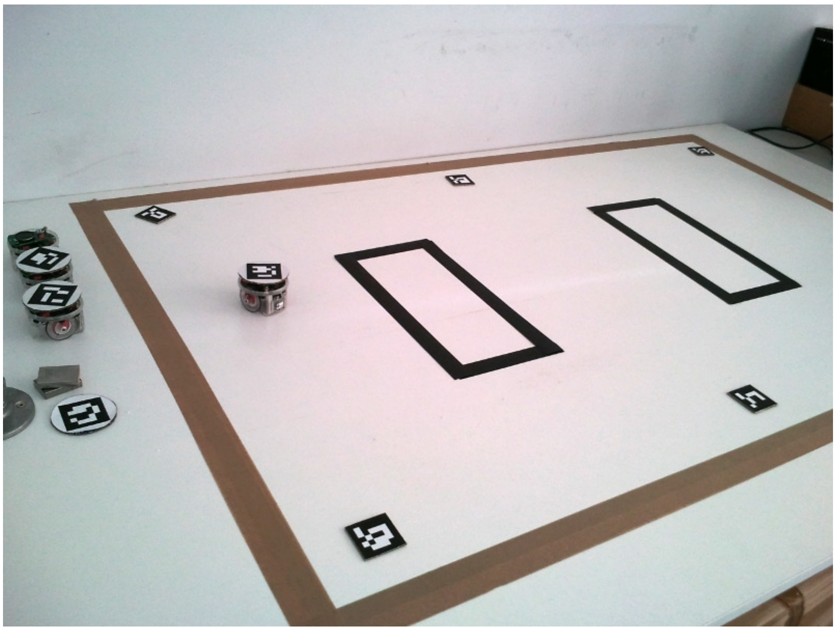


Figura 4.7: Entorno real en el que se han realizado los experimentos

de la técnica de Rekimoto [48] para encontrar en la imagen patrones y determinar su posición relativa y su orientación. Gracias al uso de matrices, las marcas empleadas son únicas, por lo que basta emplear etiquetas diferentes sobre cada robot para conocer su orientación.

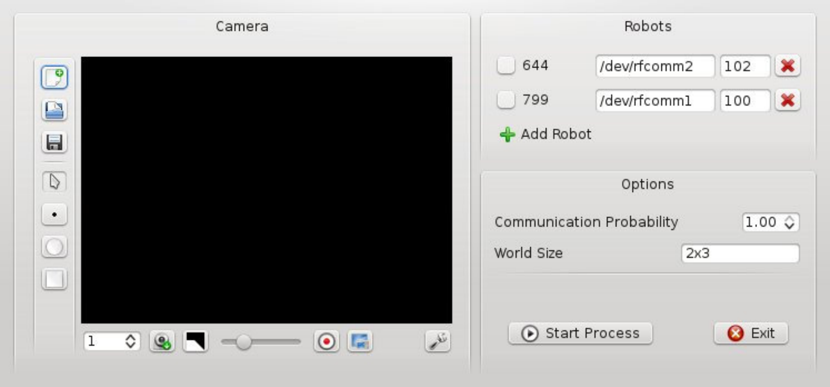
Mediante el uso de una imagen binaria y conociendo el punto desde el que se emite el láser, es sencillo establecer la longitud que los diferentes haces del láser tienen al incidir sobre los diferentes objetos del entorno. Empleando esta misma técnica se han desarrollado otros dos sensores virtuales: un sensor de proximidad a puntos de interés que emula en cierta forma el comportamiento de los chips RFID y un sensor brújula que indica a los robots su orientación relativa a un *Norte* preestablecido.

Para ajustar los diferentes valores de estos sensores virtuales se ha desarrollado una biblioteca en C++ usando el framework Qt que facilita (1) su integración con las librerías de control remoto para el ePuck que el autor de esta memoria desarrolló con anterioridad como parte de su colaboración con el proyecto Avisa2[4], (2) con la aplicación que controla la ejecución de los diferentes agentes, la situación de los nodos en el entorno y las estadísticas, y (3) con el sistema de navegación del centro de área [1, 2] que es el que emplean los robots para

---

[4] AVISA2 (TIN2007 - 67586 - C02 -01)

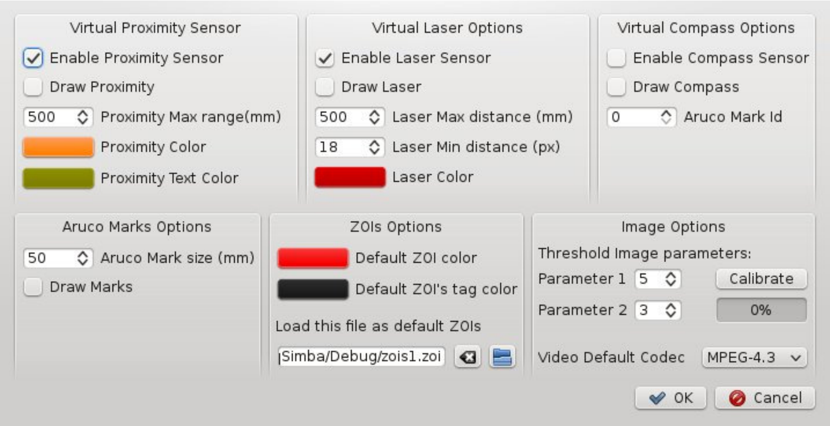
desplazarse por su entorno, evitando tanto obstáculos pasivos como dinámicos. Esta aplicación, junto con el panel que permite configurar los sensores y otros aspectos puede verse en las figuras 4.8 y 4.9.



☐ 644 /dev/rfcomm2 102
☐ 799 /dev/rfcomm1 100
Communication Probability 1.00
World Size 2x3

Figura 4.8: Interfaz de la aplicación principal que aúna todos los componentes

Todo el trabajo desarrollado en este sensor virtual ha sido publicado de forma más detallada en el artículo *Hardware And Software Infrastructure To Provide To Small-Medium Sized Robots With Advanced Sensors* publicado en el *XIII Workshop de Agentes Físicos* (WAF 2012) celebrado a primeros de Septiembre en la ciudad de Santiago de Compostela y del que aún no se dispone de la referencia del editor. Este artículo está presente como apéndice al final de la presente memoria.



☑ Enable Proximity Sensor
☐ Draw Proximity
500 Proximity Max range(mm)
Proximity Color
Proximity Text Color

☑ Enable Laser Sensor
☐ Draw Laser
500 Laser Max distance (mm)
18 Laser Min distance (px)
Laser Color

☐ Enable Compass Sensor
☐ Draw Compass
0 Aruco Mark Id

50 Aruco Mark size (mm)
☐ Draw Marks

Default ZOI color
Default ZOI's tag color
Load this file as default ZOIs
Simba/Debug/zois1.zoi

Threshold Image parameters:
Parameter 1 5
Parameter 2 3
0%
Video Default Codec MPEG-4.3

Figura 4.9: Diálogo de configuración de la aplicación principal

Toda esta infraestructura, formada por la mesa, los robots, la cámara cenital y el sensor virtual, serán empleados en un proyecto educativo del Departamento de Inteligencia Artificial de la ETSI de la UNED para los experimentos prácticos que los alumnos desarrollarán en sus clases de robótica impartidas dentro de los cursos de grado.

Capítulo 5

Resultados experimentales

A continuación se presentan y comentan los resultados obtenidos de los experimentos simulados y teóricos. En la primera parte se describe la configuración de las simulaciones y se comentan los resultados obtenidos. A continuación se hace lo propio con los experimentos llevados a cabo en un entorno real.

5.1. Experimentos simulados

Mediante estos experimentos se han comprobado dos cosas: (1) cual de las dos versiones (*bloqueante* y *no bloqueante*) de nuestra arquitectura es más robusta y (2) que incluso la peor versión de nuestra arquitectura es siempre mejor que una selección aleatoria de objetivos en un grupo de agentes que carecen de mecanismo de coordinación.

Presentamos los resultados en dos apartados. En la primero, el apartado 5.1.1, muestra los resultados obtenidos al comparar ambas versiones de nuestra arquitectura. A continuación, en el apartado 5.1.2 se ha enfrentado un sistema aleatorio de asignación de tareas con nuestra versión *no bloqueante*.

Ambas pruebas comparten los siguientes parámetros de configuración:

- Número de agentes: Desde 1 hasta 20 agentes

- Probabilidad de comunicación: Una probabilidad de $0{,}9999$ junto con un rango comprendido entre $0{,}9$ hasta $0{,}1$ a intervalos de $0{,}1$. Indica la probabilidad de establecer comunicación con el resto de agentes desde la ubicación actual.

- Probabilidad de fallo Hardware: Indica la probabilidad con la que el robot sufrirá un error Hardware irrecuperable que le impedirá continuar con su labor. Se ha empleado un valor muy pequeño, de $0{,}001$ para ajustarnos a la realidad de los robots empleados en el modelo real.

- Tamaño de ventana: Se ha empleado un mecanismo software para excluir de forma aleatoria algunos agentes si estos han participado de forma sucesiva en varios procesos de negociación.

- Profundidad de ruta: Se han realizado experimentos con diferentes profundidad de ruta, variando desde 2 hasta 4 aristas.

Los mapas empleados en los experimentos son los representados por los grafos de las figuras 5.1 y 5.2. Estos grafos están compuestos por 100 nodos de 180 y 351 aristas respectivamente.

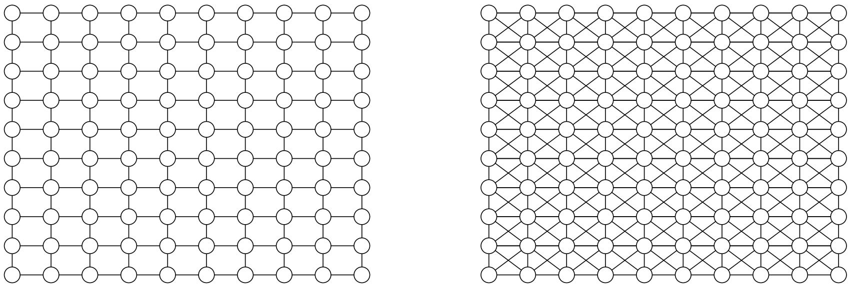


Figura 5.1: Mapa 1

Figura 5.2: Mapa 2

Figura 5.3: Mapas empleados en los experimentos simulados

Para comparar los diferentes resultados se ha tenido en cuenta el número de aristas que se recorren. Este concepto aparece en las gráficas y en resto de la memoria definido como *paso* o *secuencia*. Debe entenderse que en el ámbito de estos experimentos una secuencia equivale a un paso, y este a viajar de un nodo a otro. Otros aspectos, como la profundidad de ruta o el tamaño de ventana se han empleado para comprobar si su variación afecta de modo alguno a los pasos necesarios para completar la inspección.

5.1.1. Arquitectura Bloqueante contra Arquitectura No Bloqueante. Resultados

En cuanto a los resultados obtenidos empleando el *Mapa 1* (figura 5.1), se presentan dos gráficas que reflejan los resultados conseguidos en equipos de diferente tamaño (eje X) respecto al número de pasos que han necesitado para visitar todo el mapa (eje Y).

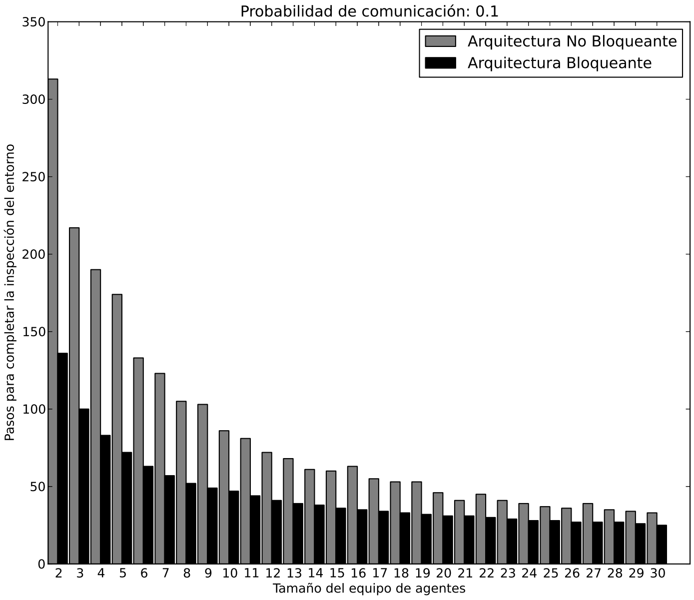


Figura 5.4: Comparación entre arquitecturas *bloqueante* y *no bloqueante*. Baja probabilidad de comunicación. Mapa 1

La primera de ellas, la gráfica 5.4, presenta los resultados de un entorno donde establecer contacto con el resto de agentes es complicado (9 de cada 10 intentos de comunicación fallan). Ante estas condiciones, ambas arquitecturas presentan resultados muy dispares para pequeños equipos. Los resultados obtenidos por una arquitectura bloqueante reducen a la mitad los conseguidos por una arquitectura no bloqueante es esos grupos pequeños, de hasta 10 agentes aproximadamente. A partir de ahí, ambas arquitecturas se acercan en resultados, posiblemente debido a la gran cantidad de agentes en un espacio demasiado reducido.

Si las condiciones de comunicación son mejores, como sucede en la gráfica 5.5, las distancias entre ambas arquitecturas aún se conservan, lo cual de-

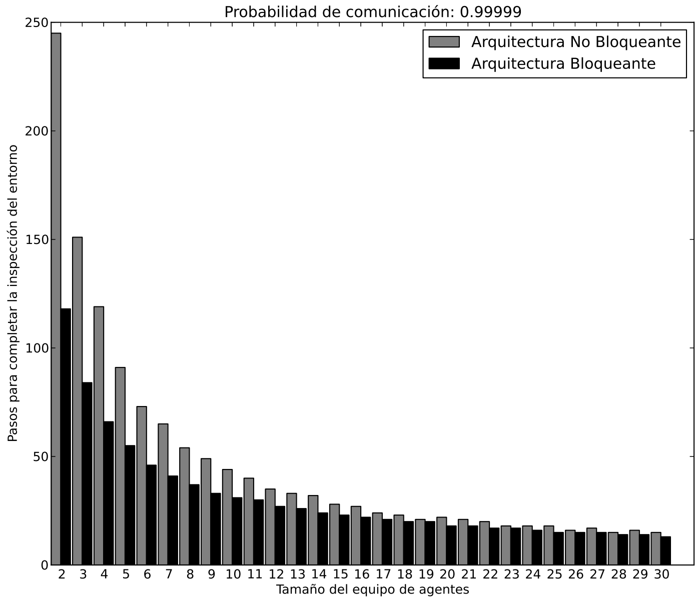


Figura 5.5: Comparación entre arquitecturas *bloqueante* y *no bloqueante*. Alta probabilidad de comunicación. Mapa 1

muestra la solidez de las metodologías y su independencia de la probabilidad de comunicaciones. Sea cual sea la probabilidad de comunicación entre los agentes, la proporción se conserva entre ambas arquitecturas.

Observando una última vez al *Mapa 1*, si comparamos los pasos que se han necesitado para diferentes probabilidades, como es ilustrado en la figura 5.6, se puede observar lo que defendíamos en el párrafo anterior: la relación de mejora en los resultados se conserva y, además, los resultados obtenidos en entornos con buena comunicación (línea naranja punteada) es muy similar a entornos donde esta falla 1 de cada 2 veces (línea verde de guiones). La mejora de los resultados de un experimento con malas comunicaciones en relación con uno en el que existen buenas comunicaciones oscila entre un 20 % y 25 % para equipos superiores a 5 agentes.

Los resultados obtenidos cuando se emplea el *Mapa 2* como entorno de experimentos ofrece la misma línea de resultados pero las diferencias son mucho menores. Este hecho, que puede verse presentado en las figuras 5.7 y 5.8 para unas condiciones de comunicación deficientes y apropiadas respectivamente, se debe a una mayor complejidad del grafo.

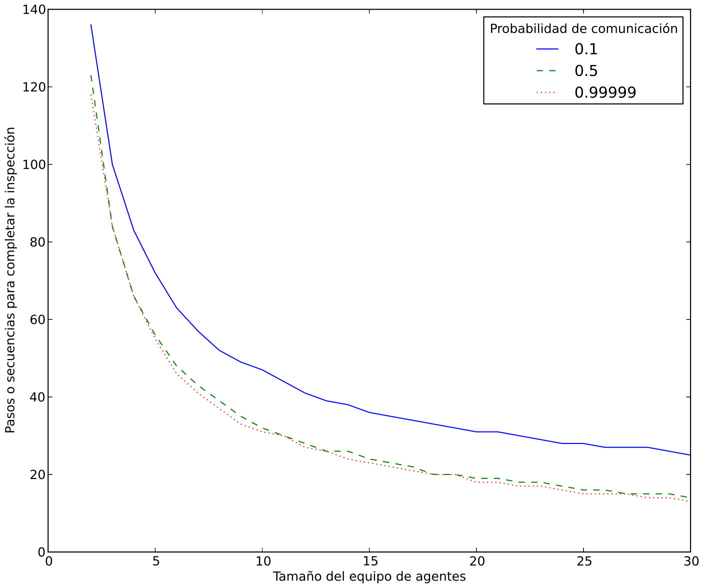


Figura 5.6: Comparación entre diferentes probabilidades de comunicación. Mapa 1

En cuanto a la relación de mejora en los resultados acorde a la probabilidad de comunicación, la figura 5.9 muestra como la relación de mejora en los resultados se mantiene dentro de una arquitectura bloqueante, que en este caso de nuevo ha presentado resultados, si bien la mejora no es tan acusada, si continúa siendo significante.

Como puede verse claramente en los resultados, el empleo de la arquitectura bloqueante, donde los participantes en las negociaciones han de permanecer activos hasta que estas finalizan, ofrece mejores resultados para ambos mapas. Si bien la mejora no es tan significativa en mapas de alta complejidad, si existe dicha mejora en todos los casos. Gran parte de estos resultados han formado parte del artículo *Inspection method based on multi-agent auction for graphic-like maps* [31] que se publicó en el *2011 Nature and Biologically Inspired Computing* (NABIC 2011) celebrado en Salamanca.

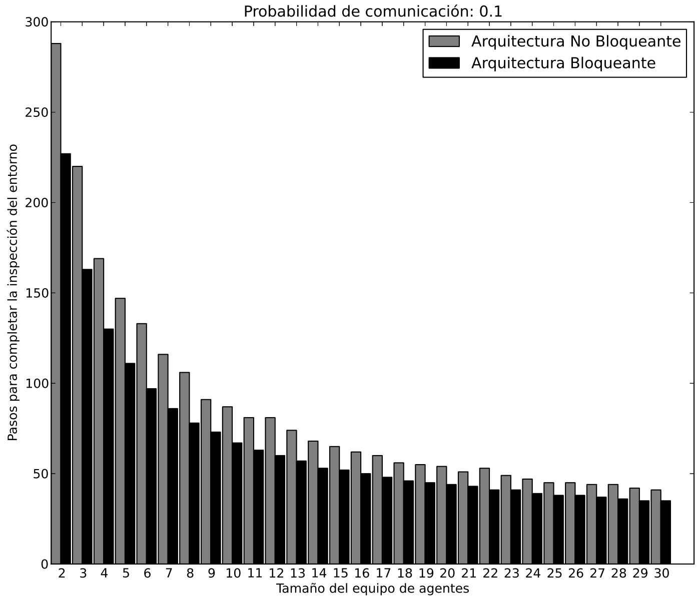


Figura 5.7: Comparación entre arquitecturas *bloqueante* y *no bloqueante*. Baja probabilidad de comunicación. Mapa 2

5.1.2. Arquitectura No Bloqueante contra Arquitectura Aleatoria. Resultados

En el apartado anterior hemos demostrado que la arquitectura *Bloqueante* ofrece mejores resultados en todos los casos. Para poder saber si, aún así, nuestra propuesta puede ser mejor que la carencia completa de negociaciones, hemos enfrentado la arquitectura *No Bloqueante*, que ha demostrado tener peores resultados, contra una inspección de aristas en las que los agentes han decidido la dirección a seguir de forma aleatoria entre todas las posibles aristas no visitadas.

Tal y como se indicó al principio de este capítulo, la configuración del simulador continúa siendo la misma. Sin embargo, por motivos de claridad los resultados presentados no contemplan equipos de 11 o más agentes. A partir de grupos de dicho tamaño la información recogida conserva la tendencia de los datos anteriores. En la figura 5.12 muestra las gráficas que contemplan los datos recogidos al efectuar los experimentos en dos entornos, el primero con una baja probabilidad de establecer comunicación con el resto del equipo (figura 5.10), en el segundo entorno el establecimiento de comunicaciones es más probable

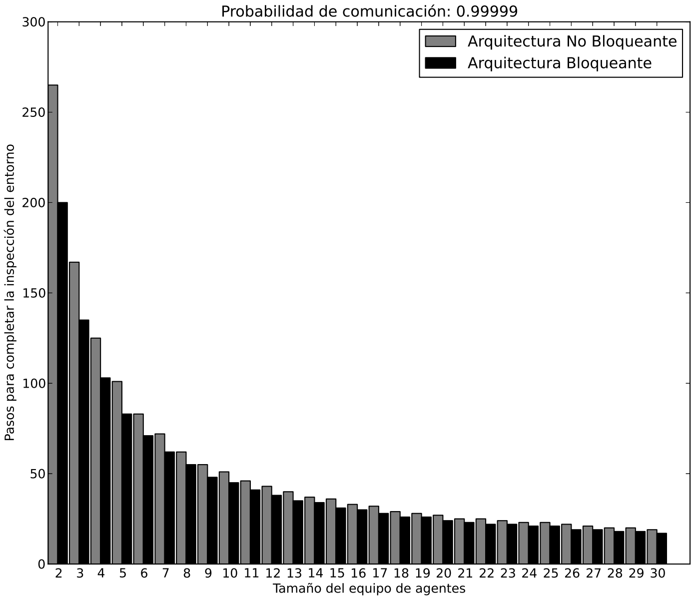


Figura 5.8: Comparación entre arquitecturas *bloqueante* y *no bloqueante*. Alta probabilidad de comunicación. Mapa 1

(figura 5.10).

A pesar de que los resultados de la arquitectura *No Bloqueante* son los peores de las dos alternativas presentadas, estos siguen siendo mejores que la falta de coordinación. Estos mismos resultados se repiten cuando se usa el Mapa 2 como escenario de los experimentos, tal y como puede apreciarse en las dos gráficas de la figura 5.15, subfigura 5.13 para un entorno donde difícilmente se establecerán comunicaciones y subfigura 5.14 para un entorno donde prácticamente siempre existirán comunicaciones entre los miembros del grupo.

5.2. Experimentos reales

El objetivo de estos experimentos es, una vez comprobado que la arquitectura **bloqueante** propuesta presenta un buen comportamiento en las simulaciones, estos resultados se mantienen en un entorno con robots reales. Con respecto al entorno simulado, el entorno real ha contado con los siguientes parámetros:

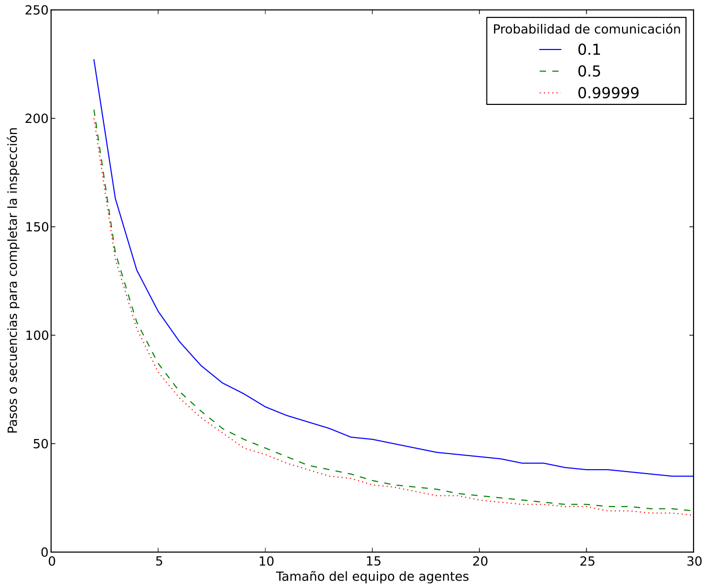


Figura 5.9: Comparación entre diferentes probabilidades de comunicación. Mapa 2

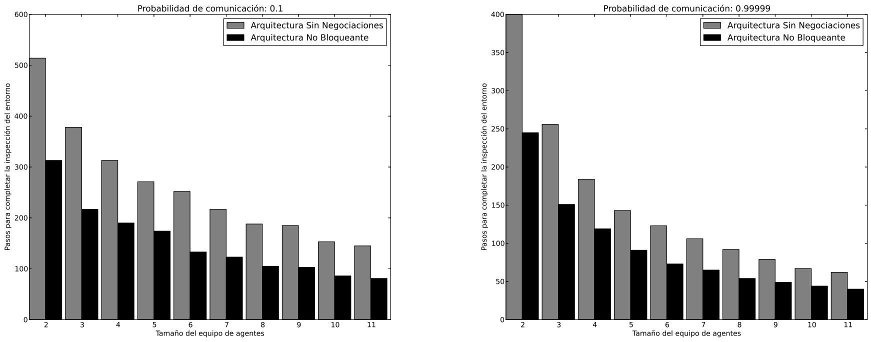


Figura 5.10: Malas comunicaciones Figura 5.11: Buenas comunicaciones

Figura 5.12: Comparación entre arquitectura No Bloqueante y arquitectura sin negociaciones en el Mapa 1

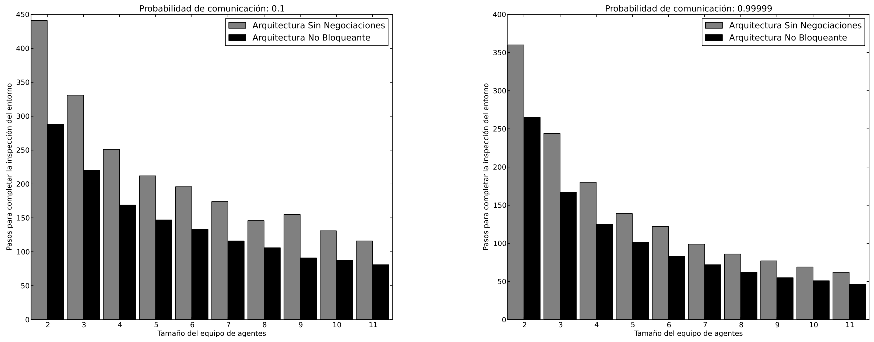


Figura 5.13: Baja Prob. de comunicaciones

Figura 5.14: Alta Prob. de comunicaciones

Figura 5.15: Comparación entre arquitectura No Bloqueante y arquitectura sin negociaciones en el Mapa 2

- Número de agentes: Desde 2 hasta 4 robots ePuck
- Probabilidad de comunicación: $0{,}5$
- Tamaño de ventana: 2 segundos
- Profundidad de ruta: 2 aristas

El mapa del mundo es un grafo compuesto por 12 nodos y distribuido acorde a la figura 5.16. Las posiciones iniciales de los robots son establecidas de forma aleatoria para cada experimento.

Una descripción detallada del funcionamiento de los diferentes componentes del entorno y los argumentos que motivaron su elección puede encontrarse en el capítulo 4.3.

5.2.1. Resultados

Se ha empleado la misma escala de medidas que en los experimentos reales para poder hacer un contraste entre simulaciones y entorno real. Los resultados de cada uno de los experimentos se pueden apreciar en la figura 5.17, donde se han superpuesto los valores recogidos para grupos de 2, 3 y 4 robots.

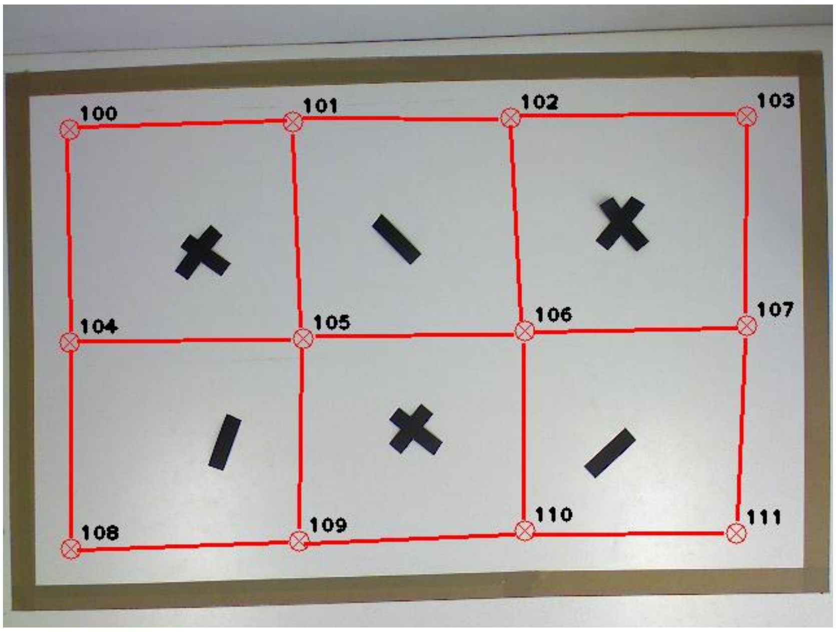


Figura 5.16: Entorno real para los experimentos. Sobreimpreso en rojo está representado el mapa de los robots

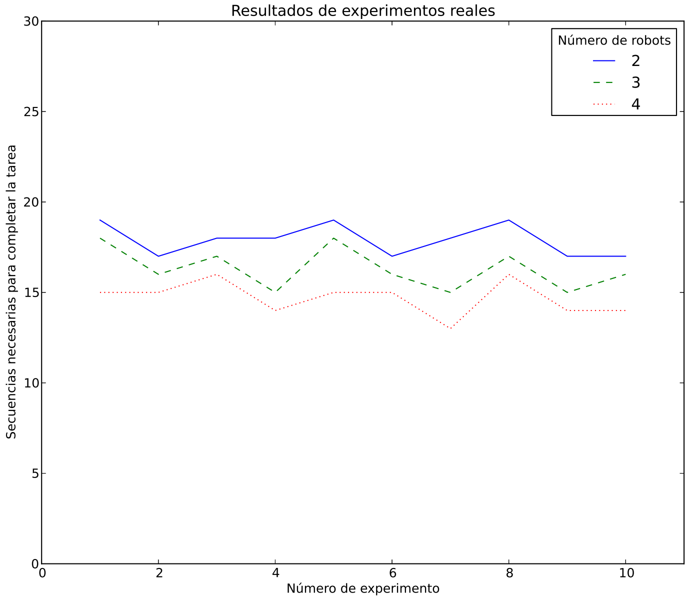


Figura 5.17: Resultados obtenidos en los experimentos reales

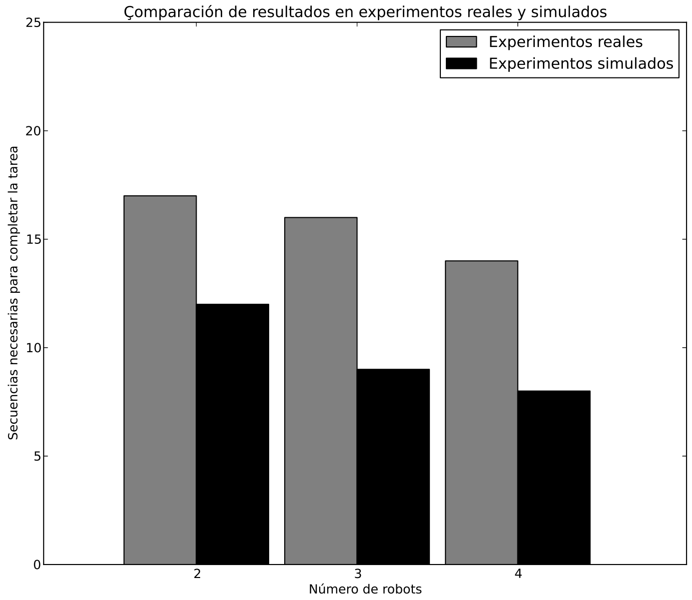


Figura 5.18: Comparación de resultados obtenidos en simulación y entorno real

Estos resultados muestran como se mantiene la reducción de pasos necesaria acorde aumenta el número de robots, lo cual implica que el espacio recorrido para completar la tarea ha sido también menor.

Al recuperar los resultados de la simulación para un número igual de agentes en un mapa idéntico esperamos que la diferencia entre ambos exista pero sea pequeña. No podemos esperar unos resultados iguales o mejores puesto que los sistemas de navegación empleados en robots no son tan eficientes como el simple hecho de viajar de un punto A hasta un punto B. el tiempo requerido es variable y esto hace que los robots pierdan mucho tiempo en los desplazamientos, perdiendo la oportunidad de participar en subastas.

La figura 5.18 muestra la relación que guardan estos experimentos reales con su equivalencia en el simulador. Como preveíamos, la diferencia existe. El aumento de esta distancia según aumenta el número de robots, deja en evidencia como estos se estorban mutuamente, ampliando el tiempo requerido para viajar de un lugar a otro y reduciendo sus posibilidades de participar en subastas con más robots.

Capítulo 6

Conclusiones y trabajo futuro

En la presente memoria se encuentra reflejado todo el trabajo realizado para el trabajo fin de Máster, partiendo de una revisión bibliográfica actualizada de las áreas abarcadas por este trabajo y finalizando con los resultados alcanzados por los diferentes experimentos.

El objetivo marcado ha sido comprobar la validez de un sistema de subastas cuando es empleado para coordinar a un grupo de robots en lugares donde las comunicaciones no son fiables y por ende muchos participantes quedan excluidos de las negociaciones.

Para comprobarlo se han propuesto dos metodologías diferentes para las subastas y se han llevado a cabo los experimentos necesarios, descritos en el capítulo 4.1, para verificar que propuesta es la más acertada. También se han propuesto dos escenarios diferentes para los experimentos: un simulador y un entorno real, descritos de forma separada en los apartados 5.1 y 5.2.

Los experimentos han demostrado la validez de nuestra arquitectura, en cualquiera de sus dos variantes *bloqueante* y *no bloqueante*, demostrando que es mejor el empleo de un sistema de coordinación que la carencia completa de él. Finalmente, comparamos los resultados obtenidos empleando la arquitectura de coordinación propuesta entre nuestro simulador y un entorno con robots reales. Los resultados obtenidos han demostrado que existe una diferencia provocada por los problemas propios de la navegación.

Las investigaciones llevadas a cabo durante este trabajo han dado como fruto la publicación de 4 trabajos repartidos en 3 congresos de ámbito internacional: (1) *International Work-conference on the Interplay between Natural and Artificial Computation* (IWINAC 2011) [32], (2) *2011 Nature and Biologically Inspired Computing* (NABIC 2011) [31], (3) *Workshop of Physical Agents* (WAF 2012) [34] y un artículo en el número especial de la revista *Robotics and Autonomous Systems* [33]. Estos cuatro trabajos se encuentran adjuntos como anexos de la memoria.

Como fruto del trabajo llevado a cabo para la construcción de un entorno real, se han desarrollado dos librerías bajo licencia libre (en C++ y Python) para el control remoto del robot ePuck. Así como la librería (programada en C++) y también publicada bajo una licencia libre que obtiene las imágenes cenitales del entorno y elabora diferentes sensores virtuales para los robots. Todo este software, junto con la estructura física y el trabajo fin de carrera realizado por Roberto García Misis, será empleado por la UNED para proporcionar a los alumnos de la asignatura de Robótica Autónoma de Cuarto curso del Grado de Informática un entorno real para llevar a cabo sus prácticas de forma remota.

El futuro de la línea de trabajo seguida hasta ahora se separa en diferentes ramas. Por un lado se abre la posibilidad de descentralizar el sistema de subastas, creando una red de comunicaciones P2P y un mecanismo de control que permita a los agentes llevar a cabo las negociaciones de forma descentralizada. También existe la posibilidad de expandir este sistema de comunicaciones a un entorno de exploración, de tal forma que los miembros del equipo deban explorar zonas desconocidas, usando las comunicaciones en la medida de lo posible para reducir el tiempo necesario que esta tarea requiere. En el ámbito más académico, emplear la arquitectura completa para las clases de Robótica Autónoma del Grado de Informática, permitirá a otros alumnos ampliar y mejorar para beneficio común las librerías que se han desarrollado durante este trabajo, añadiendo sensores o robots, y facilitando la tarea de aquellos compañeros que decidan llevar a cabo trabajos fin de ciclo en el ámbito de la robótica autónoma.

Bibliografía

[1] José Ramón Álvarez Sánchez, Félix de la Paz López, José Manuel Cuadra Troncoso, and Daniel de Santos Sierra. Reactive Navigation in Real Environments Using Partial Center of Area Method. *Robotics and Autonomous Systems*, pages 1231–1237, 2010. In press: doi 10.1016/j.robot.2010.05.009.

[2] José Ramón Álvarez Sánchez, Félix de la Paz López, José Manuel Cuadra Troncoso, and José Ignacio Rosado Sánchez. Partial Center of Area Method Used for Reactive Autonomous Robot Navigation. In José Mira, José Manuel Ferrández, José R. Álvarez, Félix de la Paz, and F. Javier Toledo, editors, *Bioinspired Applications in Artificial and Natural Computation*, volume 5602 of *LNCS*, pages 408–418. Springer Verlag, June 2009.

[3] R.C. Arkin, T. Balch, and E. Nitz. Communication of behavorial state in multi-agent retrieval tasks. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 588–594. IEEE, 1993.

[4] Ronald C. Arkin. Cooperation without communication: Multiagent schema-based robot navigation. *Journal of Robotic Systems*, 9:351–364, 1992.

[5] R. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14–23, 1986.

[6] R. Brooks. Asynchronous distributed control system for a mobile robot. In *SPIE Conference on Mobile Robots*, pages 77–84, 1986.

[7] G. Caprari and R. Siegwart. Mobile micro-robots ready to use: Alice. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3295–3300. IEEE, 2005.

[8] A. Colorni, M. Dorigo, V. Maniezzo, et al. Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life*, volume 142, pages 134–142, 1991.

[9] J. Cortés and F. Bullo. Coordination and geometric optimization via distributed dynamical systems. *Arxiv preprint math/0305433*, 2003.

[10] J. Cortés, S. Martinez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *Automatic Control, IEEE Transactions on*, 51(8):1289–1298, 2006.

[11] M. Dorigo and L.M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 1(1):53–66, 1997.

[12] M. Egerstedt and X. Hu. Formation constrained multi-agent control. *Robotics and Automation, IEEE Transactions on*, 17(6):947–951, 2001.

[13] M.M. Flood. The traveling-salesman problem. *Operations Research*, 4(1):61–75, 1956.

[14] B. Freisleben and P. Merz. New genetic local search operators for the traveling salesman problem. *Parallel Problem Solving from Nature—PPSN IV*, pages 890–899, 1996.

[15] R.A. Gatenby and T.L. Vincent. Application of quantitative models from population biology and evolutionary game theory to tumor therapeutic strategies. *Molecular cancer therapeutics*, 2(9):919, 2003.

[16] V. Gazi and K.M. Passino. Stability analysis of swarms. *Automatic Control, IEEE Transactions on*, 48(4):692–697, 2003.

[17] E. Gelenbe, V. Koubi, and F. Pekergin. Dynamical random neural network approach to the traveling salesman problem. In *Systems, Man and Cybernetics, 1993.'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on*, pages 630–635. IEEE, 1993.

[18] B.P. Gerkey and M.J. Mataric. Sold!: Auction methods for multirobot coordination. *Robotics and Automation, IEEE Transactions on*, 18(5):758–768, 2002.

[19] E.H. Hagen and P. Hammerstein. Game theory and human evolution: A critique of some recent interpretations of experimental games. *Theoretical Population Biology*, 69(3):339–348, 2006.

[20] R. Huq, G.K.I. Mann, and R.G. Gosine. Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation. *Applied Soft Computing*, 8(1):422–436, 2008.

[21] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48(6):988–1001, 2003.

[22] EW Justh and PS Krishnaprasad. Equilibria and steering laws for planar formations. *Systems & Control Letters*, 52(1):25–38, 2004.

[23] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.

[24] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara. RoboCup: A challenge problem for AI. *AI magazine*, 18(1):73, 1997.

[25] C.R. Kube and H. Zhang. Collective robotic intelligence. pages 460–468, 1992.

[26] C.R. Kube and H. Zhang. Collective robotics: From social insects to robots. *Adaptive Behavior*, 2(2):189–218, 1993.

[27] S. LEvEN. *Enabling Large-Scale Collective Systems in Outdoor Aerial Robotics*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2011.

[28] S. Leven, J.C. Zufferey, and D. Floreano. Dealing with midair collisions in dense collective aerial systems. *Journal of Field Robotics*, 2011.

[29] H. Li, S.X. Yang, and M.L. Seto. Neural-network-based path planning for a multirobot system with moving obstacles. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 39(4):410–419, 2009.

[30] C. Luo and S.X. Yang. A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. *Neural Networks, IEEE Transactions on*, 19(7):1279–1298, 2008.

[31] M. Martin-Ortiz, J. Pereda, J. de Lope, and F. de la Paz. Inspection method based on multi-agent auction for graph-like maps. In *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, pages 317–322. IEEE, 2011.

[32] M. Martín-Ortiz, J. Pereda, J. de Lope, and F. de la Paz. Selective method based on auctions for map inspection by robotic teams. *Foundations on Natural and Artificial Computation*, pages 175–184, 2011.

[33] Manuel Martín-Ortiz, Javier de Lope, and Félix de la Paz. Auction based method for graphic-like maps inspection by multi-robot system in simulated and real environments. *Robotics and Autonomous Systems*, (0):–, 2012.

[34] Manuel Martín-Ortiz, Javier de Lope, and Félix de la Paz. Hardware and software infrastructure to provide to small-medium sized robots with advanced sensors. pages 155–158, 2012.

[35] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.

[36] P. Mathis. *Graphs and Networks: Multilevel Modeling (Geographical Information Systems Series)*. 2007.

[37] M. Maurette. Mars rover autonomous navigation. *Autonomous Robots*, 14(2):199–208, 2003.

[38] R.P. McAfee. Auctions and Bidding. *Journal of Economic Literature*, 25(2):699–738, 1987.

[39] T. Melodia, D. Pompili, V.C. Gungor, and I.F. Akyildiz. A distributed coordination framework for wireless sensor and actor networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 99–110. ACM, 2005.

[40] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65, 2009.

[41] F. Mondada, E. Franzi, and A. Guignard. The development of khepera. In *Proceedings of the 1st international Khepera workshop*, volume 64, pages 7–13, 1999.

[42] L. Moreau. Stability of multiagent systems with time-dependent communication links. *Automatic Control, IEEE Transactions on*, 50(2):169–182, 2005.

[43] T. Namerikawa and C. Yoshioka. Consensus control of observer-based multi-agent system with communication delay. In *SICE Annual Conference, 2008*, pages 2414–2419. IEEE, 2008.

[44] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *Automatic Control, IEEE Transactions on*, 51(3):401–420, 2006.

[45] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[46] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, 2004.

[47] S.K. Pradhan, D.R. Parhi, and A.K. Panda. Navigation of multiple mobile robots using rule-based-neuro-fuzzy technique. *International Journal of Computational Intelligence*, 3(2):1304–2386, 2006.

[48] J. Rekimoto. Matrix: A realtime object identification and registration method for augmented reality. In *Computer Human Interaction, 1998. Proceedings. 3rd Asia Pacific*, pages 63–68. IEEE, 1998.

[49] W. Ren and R.W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *Automatic Control, IEEE Transactions on*, 50(5):655–661, 2005.

[50] W. Ren, R.W. Beard, and E.M. Atkins. A survey of consensus problems in multi-agent coordination. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1859–1864. IEEE, 2005.

[51] A.E. Roth. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica*, 70(4):1341–1378, 2002.

[52] L. Samuelson. Evolution and game theory. *The Journal of Economic Perspectives*, 16(2):47–66, 2002.

[53] T. Schmickl, R. Thenius, C. Moslinger, J. Timmis, A. Tyrrell, M. Read, J. Hilder, J. Halloy, A. Campo, C. Stefanini, et al. CoCoRo–The Self-Aware Underwater Swarm. In *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2011 Fifth IEEE Conference on*, pages 120–126. IEEE, 2011.

[54] M. Shubik. Game theory in the social sciences: concepts and solutions. 2006.

[55] K. Skrzypczyk. Game theory based target following by a team of robots. In *Robot Motion and Control, 2004. RoMoCo'04. Proceedings of the Fourth International Workshop on*, pages 91–96. IEEE, 2004.

[56] B. Song, C. Soto, A.K. Roy-Chowdhury, and J.A. Farrell. Decentralized camera network control using game theory. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–8. IEEE, 2008.

[57] S.W. Squyres, A.H. Knoll, R.E. Arvidson, J.W. Ashley, J.F. Bell III, W.M. Calvin, P.R. Christensen, B.C. Clark, B.A. Cohen, PA de Souza Jr, et al. Exploration of Victoria crater by the Mars rover Opportunity. *Science*, 324(5930):1058–1061, 2009.

[58] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Winning the DARPA Grand Challenge. *Journal of Field Robotics*, 2006. accepted for publication.

[59] A.E. Turgut, F. Gokce, H. Celikkanat, L. Bayindir, and E. Sahin. Kobot: A mobile robot designed specifically for swarm robotics research. *Department of Computer Engineering, Middle East Technical University, Ankara, Turkey, Tech. Rep*, 2007.

[60] J. Von Neumann, O. Morgenstern, A. Rubinstein, and H.W. Kuhn. *Theory of games and economic behavior*. Princeton Univ Pr, 2007.

Apéndice A

Artículo Iwinac 2011

Selective method based on auctions for map inspection by robotic teams

Manuel Martín-Ortiz[1], Juan Pereda[1], Javier de Lope[2], Félix de la Paz[3]

[1]ITRB Labs
Research, Technology Development and Innovation, S.L
[2]Computational Cognitive Robotics
Universidad Politécnica de Madrid
[3]Dept. Artificial Intelligence
UNED
`manuel.martin@itrblabs.eu,juan.pereda@itrblabs.eu,javier.delope@upm.es,`
`delapaz@dia.uned.es`

Abstract. In the inspection of a known environment by a team of robots, communication problems may exists between members of the team, even, due to the hostile environment these members can be damaged. In this paper, a redundant, robust and fault tolerant method to cover a known environment using a multi-agent system and where the communications are not guaranteed is presented. Through a simple auction system for cooperation and coordination, the aim of this method is to provide an effective way to solve communication or hardware failures problems in the inspection task of a known environment. We have conducted several experiments in order to verify and validate the proposed approach. The results are commented and compared to other methods.

1 Introduction

Nowadays, the autonomous robotics field covers a wide range of projects, from UAV (Unmanned Aerial Vehicles) [6,10,3] to humanoids with social skills [14,2]. This wide range of works shows the great progress that is been achieved in the last decade and opens a promising path for future research. The main topic of the autonomous robotics focuses on techniques that allow robots to collaborate between themselves in order to achieve a specific task [8] as it can be the exploration of surroundings [4,11]. Multi-robot systems and collaborative robotics [7] are becoming one of the biggest and most exciting challenges in this field.

One of the best applications that could be find in collaborative robotics, is the assistance to human beings in hostile environments. Pipes, sewers or abandoned mines are examples of this type of hostile environments. Projects like Makro [15] or MOIRA [13] and authors like Thrun [18] have already dealt with these environments, marking the lines to follow and emphasizing the importance of inspection tasks in these places. Kawaguchi [9] and Zhang [19] design robotic systems for pipe inspection.

The inspection task can be dealt with solutions based on A* algorithm, like Learning Real-Time A* (LRTA*) [16]. This algorithm used in robot teams needs a centralized server for planning the inspection routes. Likewise, a stable and reliable communications system is needed.

One of the handicaps in this type of underground environments is the difficulty to keep continuous communication among the members that conform the team. This requires (1) high degree of autonomy for each member of the group and (2) all the members must share all the possible information when the environment allows the communication between themselves. Despite new techniques are growing [1,5], empirical evaluations [17] shown that, for example, the soil composite or oil present are a decisive aspect.

In this paper, we introduce a redundant, robust and fault-tolerant method, which is able to perform inspection tasks on environments described above. Inspection tasks assume knowledge about the environment. We model this knowledge like a connected graph to represent the infrastructure of the sewer or pipe network. The agents of the team compete against each other through auctions to prevail the best travel routes and then complete the main task.

The paper is structured as follow. First, we describe how the proposed method works and the requirements that must fulfilled. Second, we expose the different experiments in a brief comparative with other methods. Finally, we comment the conclusions and possible improvements in our method as well as some proposals for future works.

2 Description of the proposed method

2.1 Objective

The aim of the proposed method is to inspect, in a finite time, a pipe or sewer environment modeled as a graph by the collaboration of a team of agents. We consider the inspection task as the action to visit each of the edge that comprise the graph by some of the team's members of the multi-robot team. To accomplish this, agents can make use of communications for coordination with the rest of the team when the circumstances allow it. The method proposed is able to overcome failures in the communication or in the robots' hardware. In the worst case, a single agent should be able to, eventually, visit all edges of the graph.

2.2 Prerequisites

Modeling a pipe network or a sewerage system like a graph allows us to work with powerful data structures and make use of graph theory. Thus, nodes represent intersections and edges represent corridors or tunnels.

When an agent reaches a node, it knows which node is. If an agent wants to go from the node A to the node B, eventually it will arrive at node B directly from node A, i.e. an agent placed in the node A knows which edge has to be selected to travel to the node B.

One characteristic of our method is that each agent could start to work at any moment, it is not needed that the agents start altogether. Furthermore, it is possible to lose agents during the inspection task.

2.3 Elements and structures

In the sequel we enumerate the structures and elements used in our method:

- Graph: The graph is the representation of the real environment where nodes represent intersections and edges represent corridors or tunnels. Isolated nodes are not present in the graph neither small groups of disassociates nodes or islands. Thus, we are working with connected graphs.
- Route: It is a *path* of the graph, where a *path* is *"a sequence of arcs where all the arcs are directed in the same way, i.e. the end of an arc coincides with the origin of the following one"* [12]. The first node of a path always is the agent's current node. The routes represent a set of nodes (and therefore a set of edges) that the agent decides to visit.
- Agent: An agent is the representation for a robot, it will select the routes to be visited. The agent will negotiate the routes with other agents in an auction through a central server.
- Central Server: The aim of this element is to be a shared memory. When agents can communicate with the central server, they will be able to know which edges have been visited for other agents. Likewise, this will be the place where agents bid for routes during the negotiation.

2.4 How it works

As each member of the team as the central server work independently. An individual member of the team negotiates with other teammates through the central server to accomplish the main task: inspect all edges of the graph. The central server works as a shared memory that allow the team to share knowledge about the graph state and make auctions to make the appropriate decisions.

The following sections describe in more detail the role played by each element.

Central Server This component has two main tasks: it develops the role of shared memory for all agents and it is also a place where the agents perform the auctions. As shared memory provides to the agents the necessary information to keep their own maps updated with the progress on the global task.

When an agent wants to start an auction, it sends a message to the central server for starting the auction process. Others agents can submit theirs own routes and when finish the time for the auction, the central server proceed to make an election and communicate to each agent if its route is the winner. An agent with a winner route could do the trip through the nodes of the route.

The routes are compared in function of the shared stretchs (edges between two nodes). In terms of set theory, if each route is a set of stretchs to be cover, the central server compares all routes that intersect. The routes that not intersect with any other are selected as winners (see eq. 2.1). For the routes that intersect with others, the central server will choose as the winner to the route with more exclusive stretchs (see eq. 2.2), or a random of them in case of equality.

Thus, if Σ is the set of proposed routes by agents, the route a is declared winner in an auction if and only if one of the following equations is satisfied:

$$\exists a\epsilon\Sigma,\ \forall b\epsilon\Sigma,\ a\neq b\wedge(a\cap b)=\phi \tag{2.1}$$

$$\exists\, a\epsilon\Sigma,\ \forall b\epsilon\Sigma,\ card(a-C_a)\geq card(b-C_b) \tag{2.2}$$

where

$$C_a\ =\cup\, c\epsilon\Sigma,\ c\neq a \tag{2.3}$$

$$C_b\ =\cup\, c\epsilon\Sigma,\ c\neq b \tag{2.4}$$

Fig. 2.1 shows an example of route selection in terms of set theory. Each proposed route is denoted with a letter (*A, B, C,...*), each route is composed by stretchs denoted with numbers (*1,2,3,...*). In the situation showed in this figure, routes *A* and *E* will be winners and agents that proposed routes *B, C* and *D* will have to propose a new routes and start a new auction process.

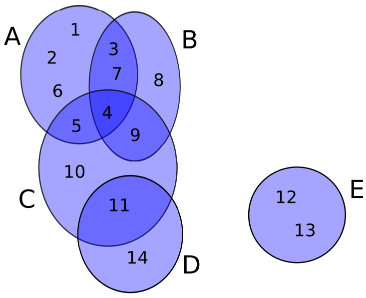


Fig. 2.1. Route selection process

Agent The aim of the agents is to achieve that all the edges of the graph will be visited. To accomplish this objective, all members of the team negotiate the routes that best fit to reduce the number of times an edge is inspected.

Throughout the execution of its algorithm (described in Fig. 2.2), the agent checks if it has a precalculated route. If the route does not exists, then it will calculate the best fit route to its current position. This route selection process is described later.

If a route exists, then the agent checks if this route is confirmed. We employ the term *confirmed route* to call the routes which have been won by the agents in a previous negotiation with other teammates. Normally, these routes can be executed by the robots without the need of communicate with the central server.

If the route is not confirmed and communication with central server is possible, then an auction process will start. In this auction all routes presented compete. If an agent wins an auction, its route is confirmed and it can execute it. Agents with not winner routes should calculate its best next route and start a new auction process.

On the other hand, if the route is not confirmed and the communication is not possible, the agent will execute the first step of its personal trip, trying to contact with the central server in the next node to negotiate this route.

The route selection process is as follow. The agent selects all possible paths starting from its current position on the map. The agent only considers paths with a length of three nodes. This arbitrary number is empirically determined and can be tunned according to size and topology of the map. The criterium followed to establish the length of the paths is to avoid combinatorial explosion in the route search process. Then, preselected routes are evaluated according to the edges: a visited edge is more interesting that a not visited edge. Finally, the selected route for auction process is the most valued route between all the preselected. In case of equality, the agent selects a random path.

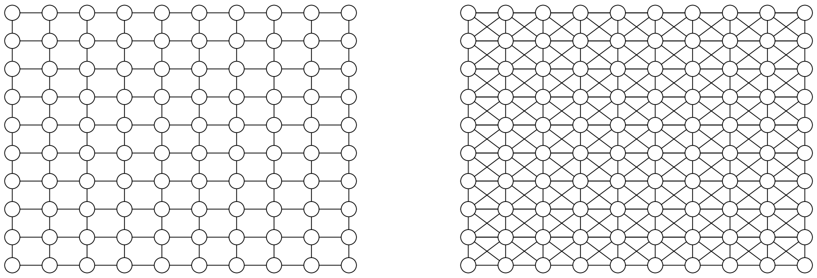
```
route := []
route_confirmed := False
while not is_all_map_inspected():
    if route != []:
        if route_confirmed:
            go_to_next_node(route)
        else:
            if is_communication_possible():
                auction_result := start_auction(route)
                route_confirmed := auction_result
            else:
                go_to_next_node(route)
    else:
        route := select_best_route_from_current_position()
```

Fig. 2.2. Pseudo-code for robot-agent

3 Experimental results

We have conducted several experiments to check the worth of the proposed method. The results are compared to the obtained by teams of robots with a random behavior. The criterium followed by these teams is completely random, i.e. each agent randomly selects a not visited edge from its current position and when if all the edges are been visited, a random edge is selected.

The objective to achieve is that all edges of the graph will be visited with the least number of steps. We assume a *step* as the action of *"go from a node A to a node B"*. We will show the results of the experiments for two kind of teams formed by agents, each with a different behavior (one for random behavior and the other with our method). Each behavior is executed by an incremental group of agents, starting with one agent and finishing with teams of thirty agents working together to achieve the main objective. The communication failure probabilities ranges between 0.9 and 0.00001. A total number of 100 tests are carried out for each particular experiment.



(a) Map denominated as *Map 1* (b) Map denominated as *Map 2*

Fig. 3.1. Maps designed for experiments

Using a map as the showed in Fig. 3.1 (a) (here in after *Map1*) with a high and a low probability of communication failure (exactly 0.9 and 0.00001 respectively) were collected the results showed in Fig. 3.2. Grey bars show the steps needed by the different groups of agents executing a random behavior, black bars are the steps needed by the groups executing the selective method.

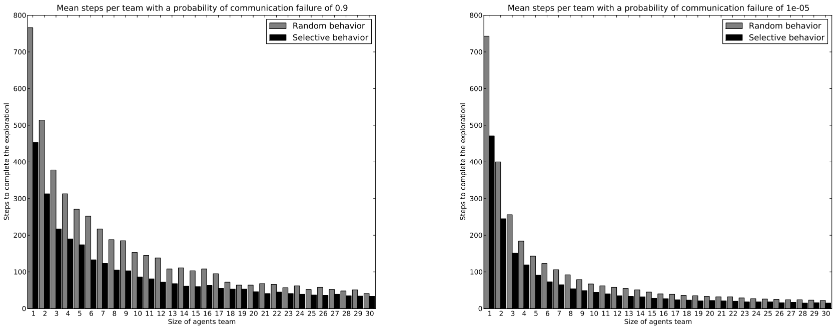
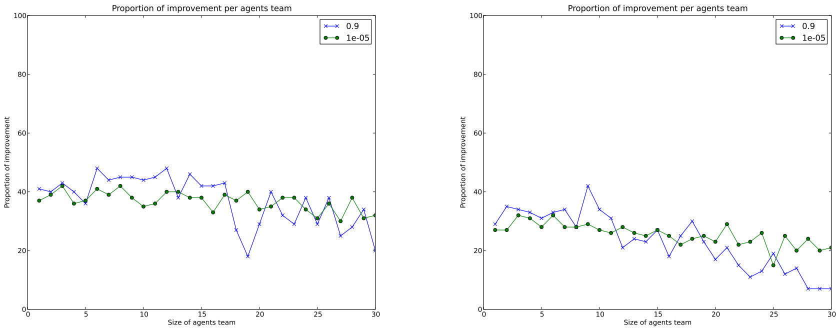


Fig. 3.2. Experiments results relative to mean steps

Fig. 3.3 (a) depicts the curve that represent the improvement of our method over the random behavior for both probabilities shown previously. The range of this improvement goes from 50% to 30% independently the size of agent team.



(a) *Map 1* (b) *Map 2*

Fig. 3.3. Improvement of our method over the random behavior

The same experiments are performed over the map as showed in Fig. 3.1 (b) (referenced as *Map2*). This map, with a high connectivity in all nodes, presents a great number of routes on which to choose as best candidate. Fig. 3.4 shows the results of the inspection task in the *Map 2* by the random and the selective behaviors with a high and a low probability of communication failure (0.9 and 0.0001 respectively).

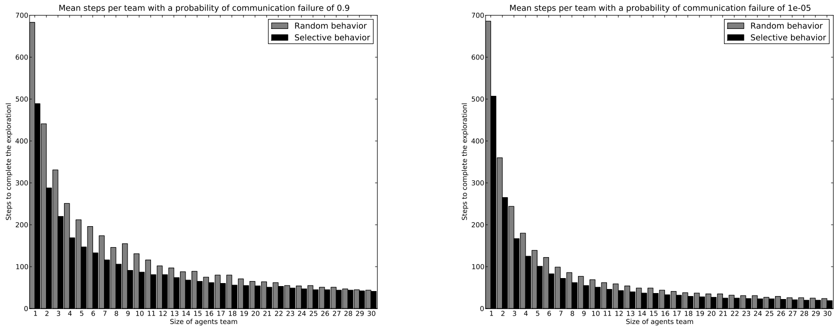


Fig. 3.4. Experiments results relative to mean steps

In this case, the improvement over the random behavior is not so good as in previous experiments. Now, as is showed in Fig. 3.3 (b), it ranges 40% to 20% (10% for a high number of agents per team and a high probability of communication failure).

If we compare our method in both maps (Fig. 3.5), it can be appreciate as the edge increase does not significantly affect the steps necessary for complete inspection of the environment, i.e. our method is quite robust against the map topology.

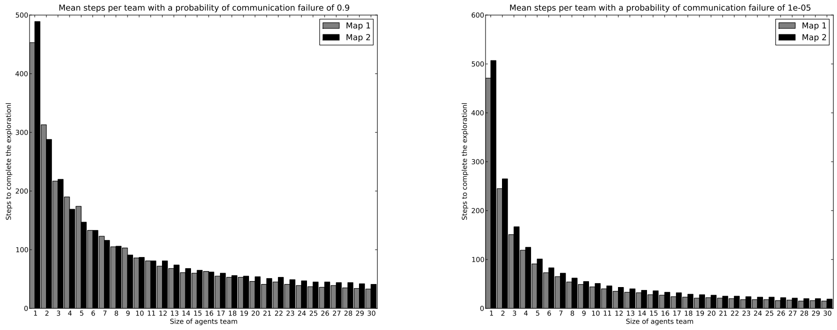
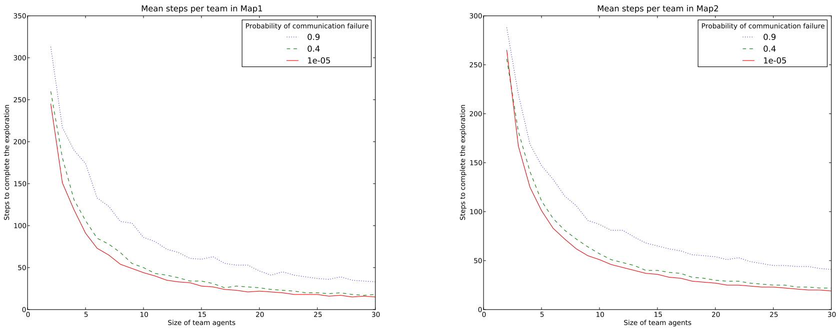


Fig. 3.5. Comparative between the two maps and the method presented

Finally, Fig. 3.6 displays how the probability of communication failure affects the results. It can be noticed that the average steps needed for a complete inspection process when the communication failure probability is intermediate are very close to the steps needed when the probability of communication failure is very small, i.e. our method is tolerant to communication failure.



(a) *Map 1* (b) *Map 2*

Fig. 3.6. Comparative between the two maps and different probability of communication failure

4 Conclusions and future work

A novel, selective method based on auctions for map inspection by teams of robotic agents has been proposed. The experiments conducted show that the selective method improves a random behavior, which could be very useful in places with communication problems. The method is able to inspect in finite time, pipe or sewer environments modeled as a graph. A provided framework for the coordination of members of a team of robots. We have emphasized on to solve the communication problems, which are inherent to this kind of environments.

Currently, we are working on the improvement of route selection method, seeking for optimal routes that will be centered in complete an intelligent exploration of the immediate environment. Likewise, the inclusion of other aspects for the planned agent's route evaluation could be a good first step for future versions of this method. We are also addressing the exploration of unknown environments with this method.

References

1. Akyildiz, I., Sun, Z., Vuran, M.: Signal propagation techniques for wireless underground communication networks. Physical Communication 2(3), 167–183 (2009)
2. Breazeal, C.: Emotion and sociable humanoid robots. International Journal of Human-Computer Studies 59(1-2), 119–155 (2003)

3. Burdakov, O., Doherty, P., Holmberg, K., Kvarnström, J., Olsson, P.: Positioning unmanned aerial vehicles as communication relays for surveillance tasks. In: Proceedings of the 5th Robotics: Science and Systems Conference (RSS), Seattle, WA. Citeseer (2009)
4. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. Robotics, IEEE Transactions on 21(3), 376–386 (2005)
5. Chehri, A., Fortier, P., Tardif, P.: Application of Ad-hoc sensor networks for localization in underground mines. In: Wireless and Microwave Technology Conference, 2006. WAMICON'06. IEEE Annual. pp. 1–4. IEEE (2007)
6. Dasgupta, P.: A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 38(3), 549–563 (2008)
7. Duro, R., Graña, M., Lope, D.: On the potential contributions of hybrid intelligent approaches to multicomponent robotic system development. Information Sciences 180(14), 2635–2648 (2010)
8. Emery, R., Sikorski, K., Balch, T.: Protocols for collaboration, coordination and dynamic role assignment in a robot team. In: Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on. vol. 3, pp. 3008–3015. IEEE (2002)
9. Kawaguchi, Y., Yoshida, I., Kurumatani, H., Kikuta, T., Yamada, Y.: Internal pipe inspection robot. In: Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on. vol. 1, pp. 857–862. IEEE (2002)
10. Kurnaz, S., Cetin, O., Kaynak, O.: Fuzzy logic based approach to design of flight control and navigation tasks for autonomous unmanned aerial vehicles. Journal of Intelligent and Robotic Systems 54(1), 229–244 (2009)
11. Maio, D., Rizzi, S.: A multi-agent approach to environment exploration. IEEE Transactions on Pattern Analysis Machine Intelligence 18(11) (1996)
12. Mathis, P.: Graphs and Networks: Multilevel Modeling (Geographical Information Systems Series) (2007)
13. Osuka, K., Kitajima, H.: Development of mobile inspection robot for rescue activities: MOIRA. In: Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on. vol. 4, pp. 3373–3377. IEEE (2003)
14. Robins, B., Dautenhahn, K., Boekhorst, R., Billard, A.: Robotic assistants in therapy and education of children with autism: Can a small humanoid robot help encourage social interaction skills? Universal Access in the Information Society 4(2), 105–120 (2005)
15. Rome, E., Hertzberg, J., Kirchner, F., Licht, U., Christaller, T.: Towards autonomous sewer robots: the MAKRO project. Urban Water 1(1), 57–70 (1999)
16. Shoham, Y., Leyton-Brown, K.: Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge Univ Pr (2008)
17. Silva, A., Vuran, M.: Empirical evaluation of wireless underground-to-underground communication in wireless underground sensor networks. Distributed Computing in Sensor Systems pp. 231–244 (2009)
18. Thrun, S., Thayer, S., Whittaker, W., Baker, C., Burgard, W., Ferguson, D., Hahnel, D., Montemerlo, D., Morris, A., Omohundro, Z., et al.: Autonomous exploration and mapping of abandoned mines. Robotics & Automation Magazine, IEEE 11(4), 79–91 (2005)
19. Zhang, Y., Yan, G.: In-pipe inspection robot with active pipe-diameter adaptability and automatic tractive force adjusting. Mechanism and Machine Theory 42(12), 1618–1631 (2007)

Apéndice B

Artículo Nabic 2011

Inspection method based on multi-agent auction for graph-like maps

Manuel Martín-Ortiz*, Juan Pereda*, Javier de Lope† and Félix de la Paz‡
*ITRB Labs Research, Technology Development and Innovation, S.L. Madrid
†Computational Cognitive Robotics, Universidad Politécnica de Madrid
‡Dept. Artificial Intelligence, UNED

***Abstract*—Nature teaches us how the collaboration between the members of a herd is an important aspect to ensure their survival as a group. Collaboration requires communication, but it may happen that it could not be established for some reason. In this context, members of the herd have to make decisions on its own, trying to do the best for the group. Inspired by this principle, we have designed a method to overcome the communication difficulties of the environment. The team members compete using their own made decisions, this competition affects on the best possible way to the team benefit.**

I. INTRODUCTION

When groups of animals are observed, it could seen how they collaborate to achieve tasks that are important to the group. However, when individuals are observed, they show selfish behaviors, like if the group interest would not be as important as the group behavior proves. Probably the most representative example of these facts are the Caribou migrations. When the herd is attacked, each member shows a selfish behavior, trying to save itself instead of organizing a collaborative defense. However, this selfish behavior is the best option for the group at the end, ensuring the group survival.

Obviously, nature is a very good model for inspiration, as evidenced works on robot construction [1], emulation in biological sensors [2], human navigation systems [3] or human cognitive processes [4], [5]. But, as some studies shown [6], remains much to learn from nature.

Following this nature-inspired line, the robotic community has developed very important works, like the *Mataric's Nerd Herd* [7] or the experiments conducted by Anderson and Denath [8]. Focusing our attention on the collaboration between robots, one of the most interesting applications, is the assistance to human beings in hostile environments. Thanks to initiatives like RoboCup Rescue there are interesting works in this area [9], [10] aimed on the assistance to natural disasters like earthquakes. Other works, focus their efforts in the exploration or inspection of known but not safe artificial environments like the Makro project [11] or the MOIRA project[12]. Other authors [13] have dealt with these environments, marking the lines to follow and emphasizing the importance of inspection tasks in these places. Also, there are works [14], [15] focused in the design of appropriate robots for these environments.

One of the handicaps on this type of underground environments are the difficulties found to keep continuous communication among the members that conform the team. This requires (1) high degree of autonomy for each member of the group and (2) to share as much information as possible when the environment allows the communication between the members of the group. Despite the emergence of new techniques [16], [17], empirical evaluations [18] show that the presence of oil and the soil composition are decisive aspects. Thanks to the work made in decentralized communications [19], new techniques could be developed to establish a reliable communication between nodes of a sensor or robot mesh.

In this paper, inspired by the selfish behavior of herd members that achieve the most benefit for the group, we present a Selective Inspection method of Maps Based in Auctions (SIMBA), that follows the line started by a previous work [20]. We show the robustness and fault-tolerant capacity of this method, demonstrating their competence to be used in environments where the communication is not a guaranteed characteristic.

The paper is structured as follows. First, we describe how the proposed method works, explaining each of the components parts and its role. Second, we expose the different experiments in a brief comparative with the line started in our previous work. Finally, we comment the conclusions and suggest some proposals for future works.

II. SELECTIVE INSPECTION OF MAPS BASED ON AUCTIONS

A. Preliminaries

The objective of the presented method is the inspection of a difficult communication environment by means of the collaboration of an agent-team. The environment is modeled as a graph: nodes are points of interest like joints or special rooms, edges are corridors or areas that connect two different nodes. We consider an inspected graph as a graph where all edges have been visited by an agent at least once. For a difficult communication environment we are referring to a place whose intrinsic characteristics prevents to the agent-team establishing communication at any moment.

An special characteristic of SIMBA, is that the number of members of the agent-team may vary over time. Either the members fail and can not work anymore, either new members join to the agent-team.

B. Formal definitions

In the sequel we enumerate the structures and elements used in our method:

- Graph: The graph is a representation of the real environment where nodes represent intersections or areas of special interest and edges represent corridors, tunnels or areas that join nodes. Isolated nodes are not present in the graph neither small groups of disassociates nodes or islands, i.e our graph is a connected graph.
- Route: It is a *path* of the graph, where a *path* is *"a sequence of arcs where all the arcs are directed in the same way, i.e. the end of an arc coincides with the origin of the following one"* [21]. The first node of a path always is the agent's current node. The routes represent a set of nodes (and therefore a set of edges) that the agent decides to visit.
- Agent: An agent is a member of the team that tries to inspect the graph. The agent calculates routes composed by nodes and edges to be visited. Subgroups of agents compete in auctions, betting routes that maximize its own interest. These auctions are carried out on the shared memory.
- Shared Memory: The shared memory has two functions: First, as the entity where the global knowledge about inspection task is stored; second, as the place where auction process is started. The agent-team needs a place where each member can share its knowledge about the state of the inspection task and get this information from other members. When a subgroup of agents want to start an auction process, is the shared memory the computational space where it is carried out.

C. Global description of the method

Viewed from a global way, each agent updates the global knowledge about the environment, calculates its best option of all possible routes, competes in an auction with other agents and finally inspects its own winner route. This global algorithm is showed in Fig II.1. An agent only participates in auctions and updates the global knowledge if the communication conditions allow it, else, the agent takes the best action, with the limited knowledge about the global situation, for the global interest.

```
while not task_achieved:
  if communication_available:
    update_global_knowledge()
    calculate_best_route()
    join_to_auction_process()
    inspect_my_winner_route()
  else:
    calculate_best_route()
    inspect_my_provisional_route()
```

Figure II.1. Global algorithm pseudo-code

Thanks to the auction process, the system tries to achieve that all agents have a good route for the global objective (global benefit) and at the same time the best route for its own purpose (personal benefit). This method of bids tries to balance both interests, attempting to overcome problems of total agent failure, communication and possible new agents arrivals.

The following sections describe in more detail the role played by each element.

1) Shared Memory: As we said previously, this component has two main tasks: it develops the role of shared memory for all agents and it is also a place where the agents perform the auctions.

As shared memory provides to the agents with the necessary information to keep their own maps updated, storing information about the global state of the inspection. When an agent can communicate with this element, the shared memory sends to the agent all updates in the global task for the last communication of the agent. Also, the agent communicates to the shared memory the places already inspected.

As place to hold an auction, the shared memory starts a process to lead the auction process. To start this process, any agent can indicate to the shared memory its intention to celebrate an auction. The shared memory then starts a process to lead the auction. This process waits a determined time for more auction players. Each player makes a bid, this bid is its best route. To simplify this process, we want to remark that the route is treated as a set of edges, using set theory at following.

In the auction process, the bids are evaluated to determine if there are winners. A bid is marked as winner if, in terms of set theory, it does not intersect with any other bid. If a bid is not marked as winner, this is communicated to the player, that can know the winner bids of other players and use this knowledge to calculate, again, its best route. Now, the player has two options: (1) to propose a new bid or (2) to retire from the auction. At the end of the auction process, all bids presented are winners, that means there are not intersections between the routes. This process is showed as pseudo-code in Fig II.2.

```
wait_for_new_players()
while not all_bids_are_winners:
  calculate_winners()
  communicate_results_to_losers()
  receive_new_bids_from_losers()
```

Figure II.2. Auction algorithm pseudo-code

Thus, if Σ is the set of proposed routes by agents (in auction terms: the bids presented by the players), the route a is declared winner if and only if the following equation is satisfied:

$$\exists a\epsilon\Sigma,\ \forall b\epsilon\Sigma,\ a\neq b\wedge(a\cap b)=\phi \qquad \text{(II.1)}$$

2) Agent: The goal of agents is to achieve all the graph's edges visited in the fewest possible steps. To accomplish this objective, the agents keep its map updated and trade with other agents in auctions if environment allow it.

An agent try to keep update its map when possible, for this task, the agent uses the shared memory to store its own changes and recover changes from other agents. If an agent arrives to a node across an edge that does not have marked as visited, then the agent marks this edge. When the communication allows it, the agent stores this information in the shared memory. Other agents will recover the information to complete their own knowledge about the environment.

To inspect the map, the agent traces routes of a determined depth, that includes so many non visited edges as possible. If the environment allows the communication, the agent tries to join to an existing auction or create a new one.

If the agent can participate in an auction, it sends its calculated route. If its route is declared as looser, it updates the map and calculates a new one, knowing now what edges will be visited by other team mates. If all edges in possible routes are already visited, or will be visited by other team mates, the agent retires from the auction and calculates the shortest path to the closest non visited edge, we call these routes *long routes*. When the agent has a route, either a long route or a winner route, covers all the edges of it.

If communication is not possible, and therefore there is no possibility to join to an auction or update the map, the agent calculates anyway the best route and covers the first edge, trying to establish communication at the next opportunity. This action ensures that the team's member always try to do the best for the group interest.

This algorithm is showed in Fig II.3 for better compression.

```
while not task_achieved:
  calculate_best_route()

  if communication_possible:
    join_to_auction_process()

    while i_have_possible_routes:
      propose_route()
      if not winner:
        update_knowledge()
        calculate_best_route()

    if not i_have_possible_routes()
        calculate_long_route()

    cover_the_route()

  else:
    go_first_stage()
```

Figure II.3. Agent algorithm pseudo-code

III. EXPERIMENTAL RESULTS

We have conducted several experiments on a software simulator to check the worth of the proposed method. We present these results in two sections. First, we have focused our results in the number of steps needed to inspect all the map. We define a step as the action of going from one node to another. Secondly, we contrast the results of SIMBA along different communication probabilities.

We use agent-teams of different size, from teams with only one agent to teams with thirty members. We have conducted experiments where the probability of communication failure ranges between 0.9 and 0.1 in steps of 0.1, in addition to a very small probability of communication failure of 10^{-5}.

We have used two different maps depicted in Fig. III.1 and III.2. The former has 180 edges and the latter has 351 edges. We want also to check out the impact of the branching factor in our algorithm. Each experiment configuration has been repeated one hundred times. We are giving the average results obtained in all the runs.

A. Inspection of the map

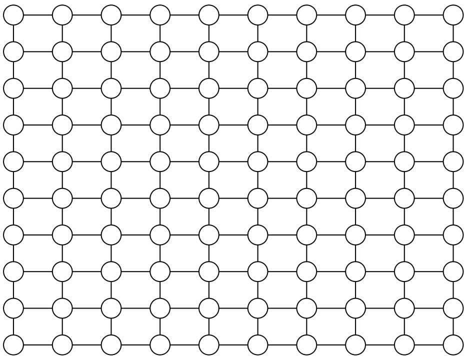


Figure III.1. Graph composed by 100 nodes and 180 edges used in the simulations refered in the text as *Map 1*

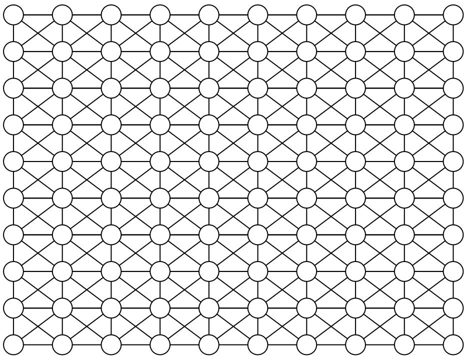


Figure III.2. Graph composed by 100 nodes and 180 edges used in the simulations refered in the text as *Map 2*

Fig. III.3 shows the average steps needed to accomplish the map inspection by teams of different sizes with a probability of communication failiure equals to 0.9. The black bars represent the average steps of the SIMBA algorithm when all participants in an auction compite until all of them have a winner route, while the gray bars are the average steps when the auction participants retire when it have a winner route (they do not wait to other participants to make new bids). These are denomited respectively as *Bloking auctions* and *Non bloking auctions* on figures.

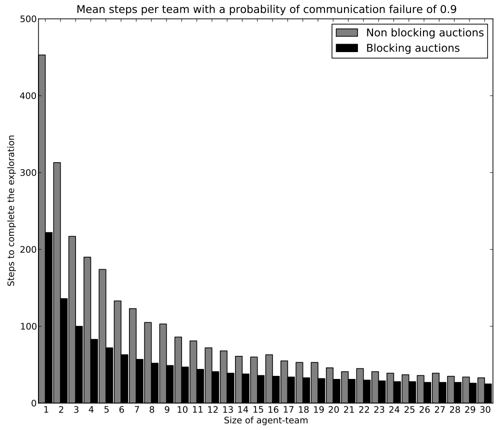


Figure III.3. Mean step over Map 1

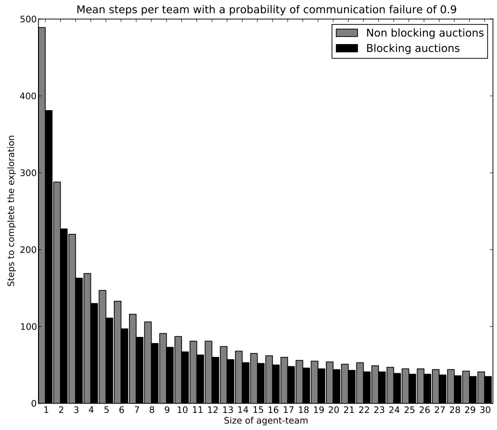


Figure III.4. Mean step over Map 2

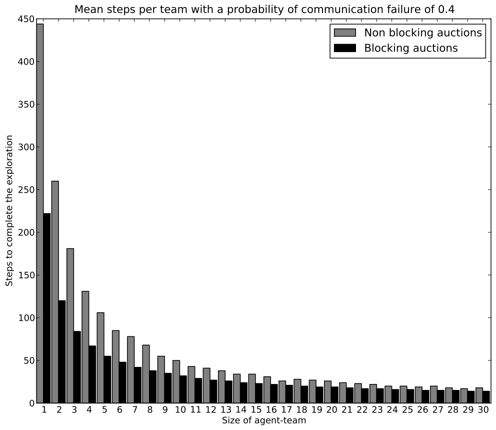


Figure III.5. Mean step over Map 1

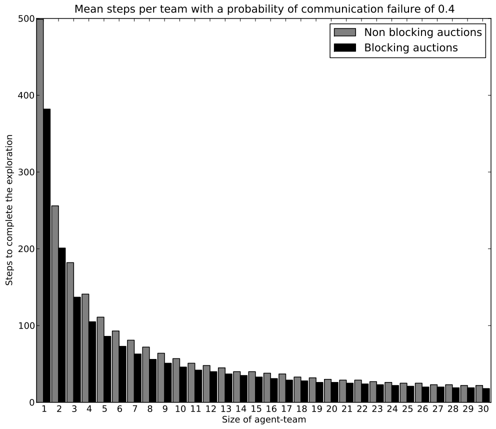


Figure III.6. Mean step over Map 2

Fig.III.4 shows the results over the *Map 2* with the same communication failure probability, i.e. 0.9. In this case the results are less impresive. Nevertheless, despite the number of edges to be inspected in *Map 2* are almost double that edges in *Map 1*, the number of steps needed to accomplish the inspection is not double. Later we reintroduce this point.

Fig. III.5 and III.6 show the number of steps needed for inspecting the maps by the different robot teams in environments with more reliable communication (communication failure probability of 0.4).

The relation between both maps is showed in Fig. III.7. Each point represesents the difference, in steps, between both maps, where *Y* axis represents the number of steps and the other axis represents the number of agents. We can see how (1) for more complex maps there is not a significant improvement if we use bigger teams, and (2) the difference in steps to acomplish the task, between high and low probabilities is small (in relation to the number of edges) and practically constant.

Another important aspect that could be appreciated in these results is the number of robots needed to carry out the most optimum inspection. This size is around seven or nine robots, not a very big team, even with four or six robots the obtained results are quite satisfactory and the difference with teams of ten or eleven robots is not significant.

B. Influence of the size of the route

Another important aspect of SIMBA is the size (number of nodes) of the routes calculated by the agents. We have carried out experiments with different size for these routes to check the influence of the number of nodes in the final number of steps. Results are showed in Fig III.8. We can observe that a route with more nodes does not improve the results achieved. In fact, it makes even worst the results obtained with smaller routes, because is more difficult to find a route that does not intersect with others.

C. Comparison between different failure probabilities

Finally, we have used the results showed in section III-A to contrast it with different probabilities of communication failure. As we have said before, we have made experiments where the probability of communication failure ranges between 0.9

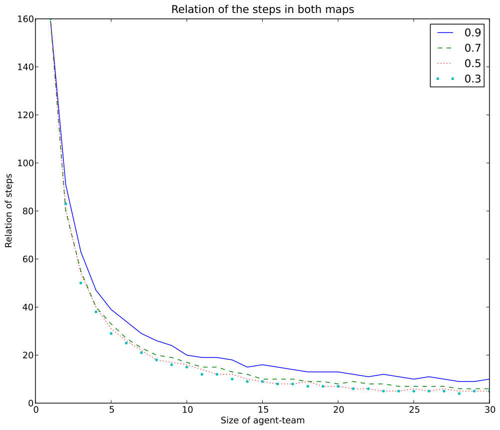


Figure III.7. Relation of steps on both maps per probability

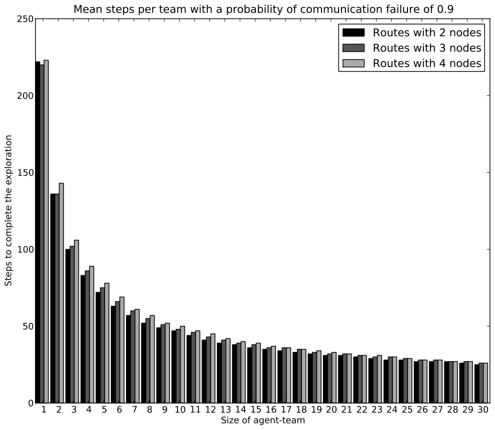


Figure III.8. Influence of the route size

and 0.1. In Fig III.9 are showed the steps needed by teams of different size to achieve their task over *Map 1* along different probabilities of communication failure. Its remarkable how the steps needed to achieve the task are almost the same in environments where the communication is almost perfect and in places where an attempt of two fault. This is the same for *Map 2* as is showed in Fig III.10

In previous figures it could be appreciated how the improvement in the number of steps is not significant for teams greater than approximately 15 members. i.e. the number of steps converges along the size of the team. This illustrate how we can adjust the size of the team, to a relatively small team, achieving good results in the inspection task.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a selective inspection of maps based on auctions (SIMBA). This method inspired in the nature of herds and communities try to achieve good results for the group when communication is difficult and therefore the coordination. The experiments conducted shown how our method is able to overcome the communication difficulties, achieving very good results over the simulator, for the task it has been designed.

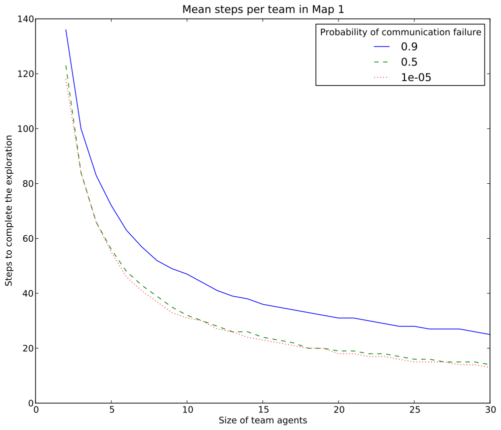


Figure III.9. Relation between communication fail and steps on Map 1

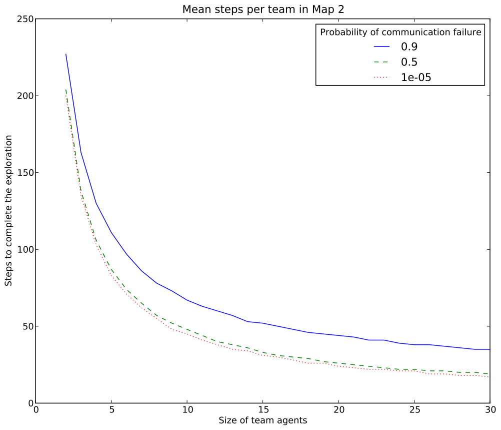


Figure III.10. Relation between communication fail and steps on Map 2

Our next step is to apply SIMBA in a controlled real environment with robots in our laboratory. In order to achieve a fully robust communication between members of the team, a decentralized and modular communication system is needed. Thus, in the future we will replace the shared memory by a fully decentralized shared memory.

To provide our robots with exploration skills and to construct a map of their environment are another important aspects in our research. These new skills are accompanied by the need of sharing and merging the knowledge that each team member has about the environment.

REFERENCES

[1] R. Wood, “The first takeoff of a biologically inspired at-scale robotic insect,” *Robotics, IEEE Transactions on*, vol. 24, no. 2, pp. 341–347, 2008.

[2] C. Siagian and L. Itti, “Biologically inspired mobile robot vision localization,” *Robotics, IEEE Transactions on*, vol. 25, no. 4, pp. 861–873, 2009.

[3] J. Cuadra Troncoso, J. Álvarez-Sánchez, F. de la Paz López, and A. Fernández-Caballero, “Improving area center robot navigation using a novel range scan segmentation method,” *Foundations on Natural and Artificial Computation*, pp. 233–245, 2011.

[4] J. Martín H., J. de Lope, and D. Maravall, “Adaptation, anticipation and rationality in natural and artificial systems: computational paradigms mimicking nature,” *Natural Computing*, vol. 8, pp. 757–775, 2009.

[5] R. Duro, M. Graña, and J. de Lope, “On the potential contributions of hybrid intelligent approaches to multicomponent robotic system development,” *Information Sciences*, vol. 180, no. 14, pp. 2635–2648, 2010.

[6] R. Pfeifer, M. Lungarella, and F. Iida, “Self-organization, embodiment, and biologically inspired robotics,” *Science*, vol. 318, no. 5853, p. 1088, 2007.

[7] M. Mataric, “Behaviour-based control: Examples from navigation, learning, and group behaviour,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2, pp. 323–336, 1997.

[8] T. Anderson and M. Donath, “Animal behavior as a paradigm for developing robot autonomy,” *Robotics and Autonomous Systems*, vol. 6, no. 1-2, pp. 145–168, 1990.

[9] S. Ramchurn, A. Farinelli, K. Macarthur, and N. Jennings, “Decentralized coordination in robocup rescue,” *The Computer Journal*, vol. 53, no. 9, p. 1447, 2010.

[10] D. Calisi, A. Farinelli, L. Iocchi, and D. Nardi, “Multi-objective exploration and search for autonomous rescue robots,” *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 763–777, 2007.

[11] E. Rome, J. Hertzberg, F. Kirchner, U. Licht, and T. Christaller, “Towards autonomous sewer robots: the MAKRO project,” *Urban Water*, vol. 1, no. 1, pp. 57–70, 1999.

[12] K. Osuka and H. Kitajima, “Development of mobile inspection robot for rescue activities: MOIRA,” in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4. IEEE, 2003, pp. 3373–3377.

[13] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hahnel, D. Montemerlo, A. Morris, Z. Omohundro *et al.*, “Autonomous exploration and mapping of abandoned mines,” *Robotics & Automation Magazine, IEEE*, vol. 11, no. 4, pp. 79–91, 2005.

[14] J. Lee, S. Roh, D. Kim, H. Moon, and H. Choi, “In-pipe robot navigation based on the landmark recognition system using shadow images,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, 2009, pp. 1857–1862.

[15] Y. Zhang and G. Yan, “In-pipe inspection robot with active pipe-diameter adaptability and automatic tractive force adjusting,” *Mechanism and Machine Theory*, vol. 42, no. 12, pp. 1618–1631, 2007.

[16] I. Akyildiz, Z. Sun, and M. Vuran, “Signal propagation techniques for wireless underground communication networks,” *Physical Communication*, vol. 2, no. 3, pp. 167–183, 2009.

[17] A. Chehri, P. Fortier, and P. Tardif, “Application of Ad-hoc sensor networks for localization in underground mines,” in *Wireless and Microwave Technology Conference, 2006. WAMICON’06. IEEE Annual*. IEEE, 2007, pp. 1–4.

[18] A. Silva and M. Vuran, “Empirical evaluation of wireless underground-to-underground communication in wireless underground sensor networks,” *Distributed Computing in Sensor Systems*, pp. 231–244, 2009.

[19] A. Ortiz, F. Royo, T. Olivares, L. Orozco-Barbosa, J. Castillo, and A. Fernández-Caballero, “Protocol integration for intelligent monitoring applications in wireless sensor networks,” *Foundations on Natural and Artificial Computation*, pp. 511–520, 2011.

[20] M. Martín-Ortiz, J. Pereda, J. de Lope, and F. de la Paz, “Selective method based on auctions for map inspection by robotic teams,” *Foundations on Natural and Artificial Computation*, vol. LNCS 6686, pp. 175–184, 2011.

[21] P. Mathis, *Graphs and Networks: Multilevel Modeling (Geographical Information Systems Series)*, 2007.

Apéndice C

Artículo Robotics and Autonomous Systems

Auction based method for graphic-like maps inspection by multi-robot system in simulated and real environments

Manuel Martín-Ortiz*, Javier de Lope**, Félix de la Paz

Abstract

Nature teaches us how the collaboration between the members of a herd is an important aspect to ensure their survival as a group. Collaboration requires communication, but it may happen that it could not be established for some reason. In this context, members of the herd have to make decisions on its own, trying to do the best for the group. Inspired by this principle, we have designed a method to overcome the communication difficulties of the environment. The team members compete using their own made decisions, this competition affects on the best possible way to the team benefit.

1. Introduction

When groups of animals are observed, it could seen how they collaborate to achieve tasks that are important for the group. However, when individuals are observed, they show selfish behaviours, like if the group interest would not be as important as the group behaviour proves. Probably, the most representative example of these facts are the Caribou migrations. When the herd is attacked, each member shows a selfish behaviour, trying to save itself instead of organizing a collaborative defence. However, this selfish behaviour is the best option for the group at the end, ensuring the group survival.

Obviously, nature is a very good model for inspiration, as evidenced works on robot construction [1], emulation in biological sensors [2], human navigation systems [3] or human cognitive processes [4, 5]. But, as some studies shown [6], remains much to learn from nature.

Following this nature-inspired line, the robotic community has developed very important works, like the *Mataric's Nerd Herd* [7] or the experiments conducted by Anderson and Denath [8]. Focusing our attention on the collaboration between robots, one of the most interesting applications is the assistance to human beings in hostile environments. Thanks to initiatives like *RoboCup Rescue* there are interesting works in this area [9, 10] aimed on the assistance to natural disasters like earthquakes. Other works, focus their efforts in the exploration or inspection of known but not safe artificial environments like the Makro project [11] or the MOIRA project[12]. Other authors [13] have dealt with these environments, marking the lines to follow and emphasizing the importance of inspection tasks in these places. Also, there are works [14, 15] focused in the design of appropriate robots for these environments.

One of the handicaps on this type of underground environments are the difficulties found to keep continuous communication among the members that conform the team. This requires (1) high degree of autonomy for each member of the group and (2) to share as much information as possible when the environment allows the communication between the members of the group. Despite the emergence of new techniques [16, 17], empirical evaluations [18] demonstrate that the presence of oil and the soil composition are decisive aspects. Thanks to the work made in decentralized communications [19], new techniques could be developed to establish a reliable communication between nodes of a sensor or robot mesh.

---

*Robotics Laboratory. ITRB Labs

**Computational Cognitive Robotics, Universidad Politécnica de Madrid
Dept. Artificial Intelligence, UNED

Email addresses: `manuel.martin@itrblabs.eu` (Manuel Martín-Ortiz), `javier.delope@upm.es` (Javier de Lope), `delapaz@dia.uned.es` (Félix de la Paz)

Preprint submitted to Elsevier *April 4, 2012*

This paper expands the work presented in [20] and [21], presenting a brief theoretical analysis and the results obtained after experiments in a simulated environment and a real one. The theoretical proposition of Selective Inspection Method Based on Auctions (SIMBA) is presented in section 2, later a case of use is described in section 3 as well as various pseudo-code fragments to illustrate the role of each actor. To validate SIMBA, two kind of experiments have been carried out in the previous scenario: first, in a simulator developed to validate the main idea of SIMBA and explained in section 4; second, in a real environment to test if the results obtained in the simulations are valid in the real world that is described in section 5. Finally, in section 6 the experimental results are analysed and future work presented.

2. Theoretical Analysis

The method to select how the team is coordinated is based on game theory, more concretely, in auction processes. So, the method has the following elements:

- *Players*: The players are the members of the team.
- *Options/Bids*: The bids are selected by the players. The bid can be decomposed in elements common to all players, i.e. a bid is composed by elements that any player can have. The solution to achieve the task can be decomposed in these elements. So, a bid is a subgroup of the elements needed to achieve the task.

Thus, if Σ is the set of proposed options by players, the bid a is declared winner if and only if the following equation is satisfied:

$$\exists a \epsilon \Sigma,\ \forall b \epsilon \Sigma,\ a \neq b \wedge (a \cap b) = \phi \tag{1}$$

If a player is declared as loser, it can ask for winner bids, update its knowledge and make a new bid. This process is repeated until all players are declared as winners. This guarantees that at the end of the auction all players have a bid that does not interfere with the others for the global solution. Fig. 1 (A) shows a non finished auction, and Fig. 1 (B) shows the relation between the different options when the auction is finished.

3. Scenario for validation

To validate SIMBA a real problem has been adapted, such some experiments have been carried out over this scenario. The scenario selected has been the inspection of a known area by a team of robots, this scenario can be divided in the following elements:

- *Task*: The main task to be achieved in the scenario is the inspection of all the area. This task finalise when all parts of the map have been visited almost by one member of the team. To illustrate how robots try to complete the task the pseudo-code of Fig. 2 is presented.
- *Map*: The environment map is modelled as a graph. Nodes represent joints of the real scenario and edges represent tunnels or corridors between nodes.
- *Robots*: The team that have to inspect the map is formed by robots, these robots are independent and know of the existence of other robots, collaborating with them to achieve the task as efficient as possible. A pseudo-code of the robot behaviour is showed in Fig. 3.
- *Central Server/Shared Memory*: This element has a dual function: allowing to the robots sharing knowledge of the global status and executing the auctions. A pseudo-code of the auction labour is showed in Fig. 4.

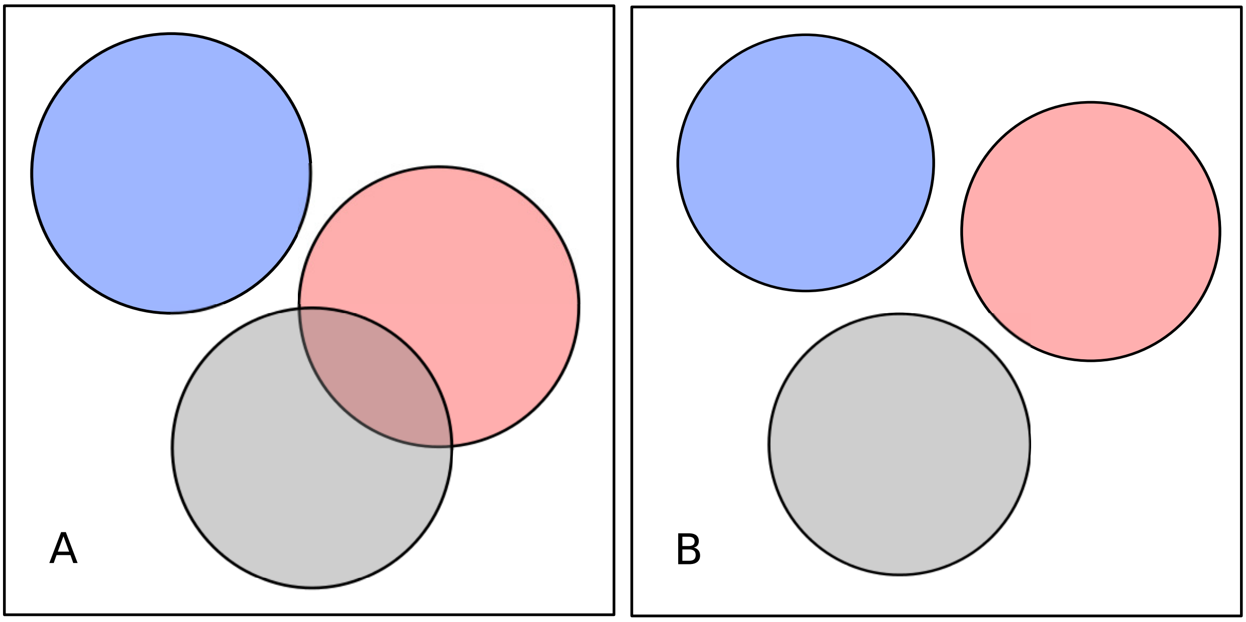


Figure 1: Representation of the options relation as set theory

- *Route*: The route is a set of graph edges, or a path of the graph. According to [22] a path is *“a sequence of arcs where all the arcs are directed in the same way, i.e. the end of an arc coincides with the origin of the following one“*

According to the nomenclature used in section 2 the robots are the *players* and the *options* bided are the routes.

```
while not task_achieved:
  if communication_available:
    update_global_knowledge()
    calculate_best_route()
    join_to_auction_process()
    inspect_my_winner_route()
  else:
    calculate_best_route()
    inspect_my_provisional_route()
```

Figure 2: Global algorithm pseudo-code

4. Experiments in a simulator

For the validation of our proposition, a simulator has been designed and developed to provide a controlled environment where software robots could interact. Thanks to this software simulator, a great number of experiments could be executed, saving its result for a posterior analysis. In this section the simulator is presented and the results of the experiments analysed.

The simulator pretends to provide to the robots a virtual environment where some real characteristics could be emulated. Thus, the simulator provides two uncertainties for the experiments:

```
while not task_achieved:
  calculate_best_route()

  if communication_possible:
    join_to_auction_process()

    while i_have_possible_routes:
      propose_route()
      if not winner:
        update_knowledge()
        calculate_best_route()

    if not i_have_possible_routes()
        calculate_long_route()

    cover_the_route()

  else:
    go_first_stage()
```

Figure 3: Robot algorithm pseudo-code

```
wait_for_new_players()
while not all_bids_are_winners:
  calculate_winners()
  communicate_results_to_losers()
  receive_new_bids_from_losers()
```

Figure 4: Auction algorithm pseudo-code

- *Communication*: This aspect answer a simple question in the robot algorithm. *Is the communication with my team-mates possible?*. On the simulations, the experiments are executed with a probability that ranges between 0.9 and 0.1 in steps of 0.1. Additionally, a very small communication failure probability of 10^{-5} is used to simulate a very reliable communication infrastructure.
- *Hardware failure*: In real environments, some issues can affect to real robots: low battery, wheels blocked or unexpected general hardware failure. So, this uncertainty try to represent this real aspect when the robots are used. In our simulated experiments is fixed to 0.05

These two variables were used in experiments with the following configuration:

- *World*: Each robot has a local copy of the map and is able to know in which node is located because the nodes are identified by an unique tag. Two different worlds where used for the experiments, they are noted as *Map 1* and *Map 2* and are showed in Fig. 5 and 6 respectively.
- *Central Server/Shared memory*: Under this symbolic name, the auction process is implemented. So, in this independent process, the bids and therefore the maps updates are executed and stored for the benefits of all the robots.
- *Window size*: This parameter represents how many time the central server wait for players before the auction starts. For these experiments this value has been adjusted to avoid all robots to participate in all the auctions.

- *Route depth*: Indicates how many edges have the routes calculated by the robots. In the results showed bellow this value is set to 2.

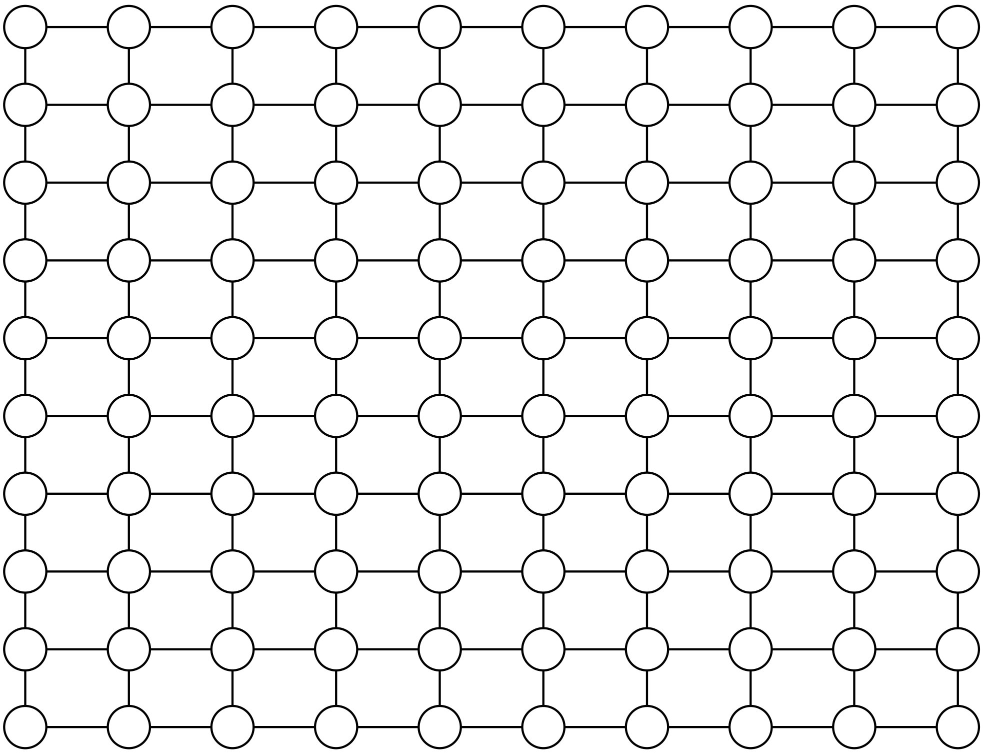


Figure 5: Graph denoted as *Map 1*

With these parameters the simulation was executed for both maps, with a different size of teams formed by robots and different communication probabilities. We have paid special attention to the number of steps needed to go over all edges of the graph almost once. As each configuration has been executed 100 times, the results are showed as a mean value. The results of these different configurations are presented in Fig. 7 for *Map 1* and Fig. 8 for *Map 2*.

These figures show how close are the results for an environment with a communication probability of 0.5 and 10^{-5}. Meaning that the proposed method is robust to communication failure.

5. Experiments in a real scenario

For the real experiments there are some element with a high importance: the robots, the environment, how the robots acquire the information from the world and the robot navigation system.

- *The robots*: The robots selected for our environment are the educational ePuck robot [1]. This robot is small and easily controlled through bluetooth.

---

[1] `http://www.e-puck.org`

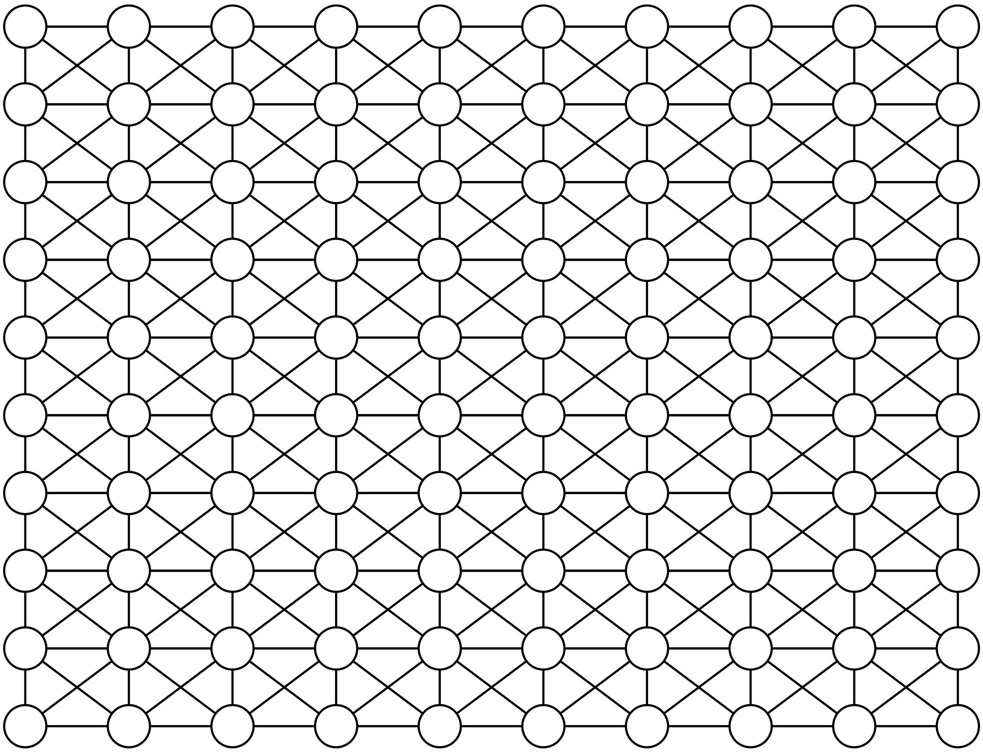


Figure 6: Graph denoted as *Map 2*

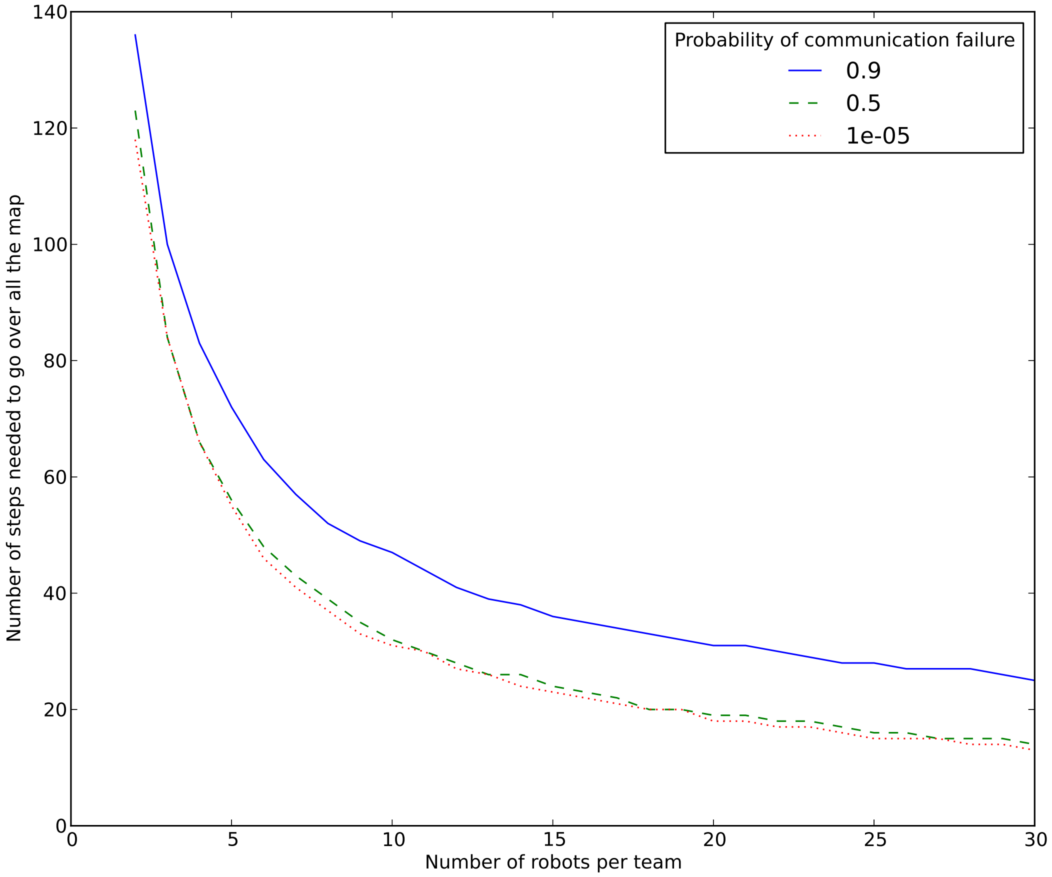


Figure 7: Steps needed to go over *Map 1* with different communication probabilities by teams of robots

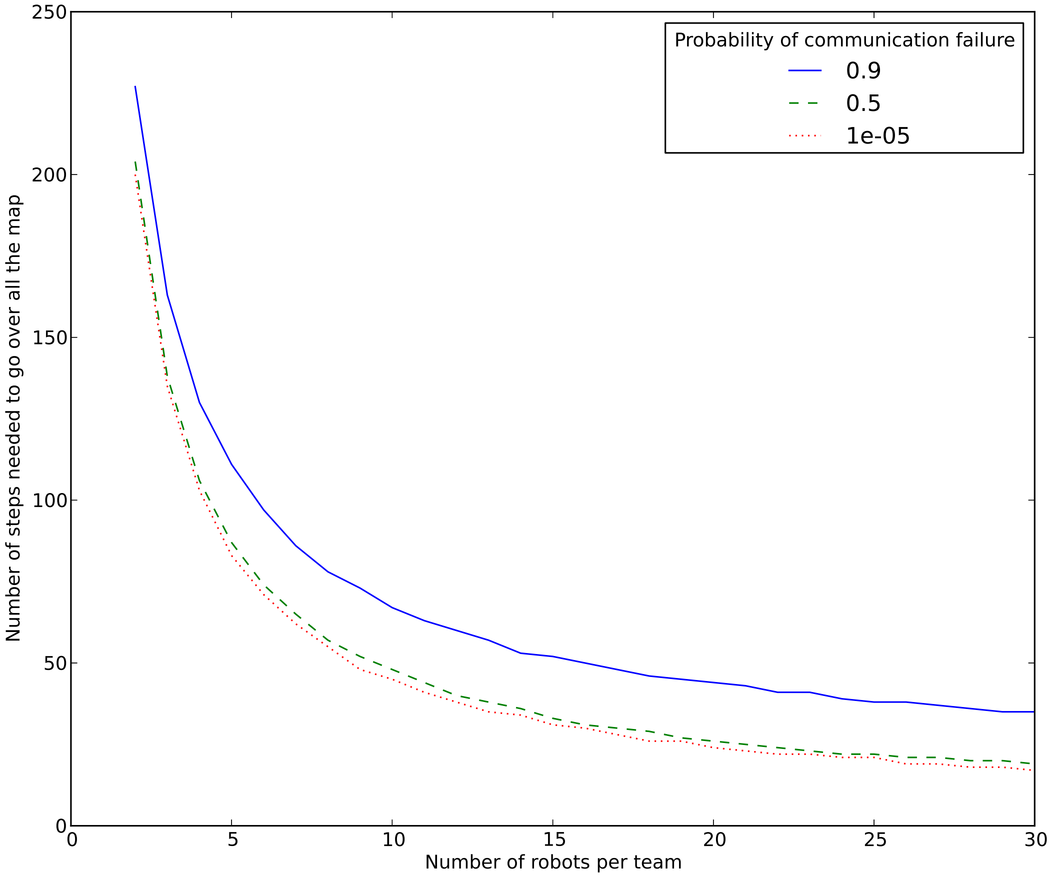


Figure 8: Steps needed to go over *Map 1* with different communication probabilities by teams of robots

- *The environment*: Thanks to the small size of the robots a big white table has been enough to build all the environment, using black paper to represent walls.
- *Information acquisition*: A RGB camera is set in zenith position over the environment for save videos of the experiments and provide information to the robots.
- *Robot navigation system*: The navigation system used is the Area Center method [3]. Unfortunately, this method needs the information provided by a laser, so has been necessary to implement a mechanism to provide to the ePuck robot a laser sensor.

For the information acquisition both the modelled of the environment and the robot navigation system, has been employed an augmented reality library. This library, called ArUcO [2], uses special marks that can be uniquely identified in a RGB image. These marks are used on robots to find them inside the image and as landmarks to identify the nodes of the graph. Fig. 9 shows an ePuck robot with the mark that identifies the robot uniquely.

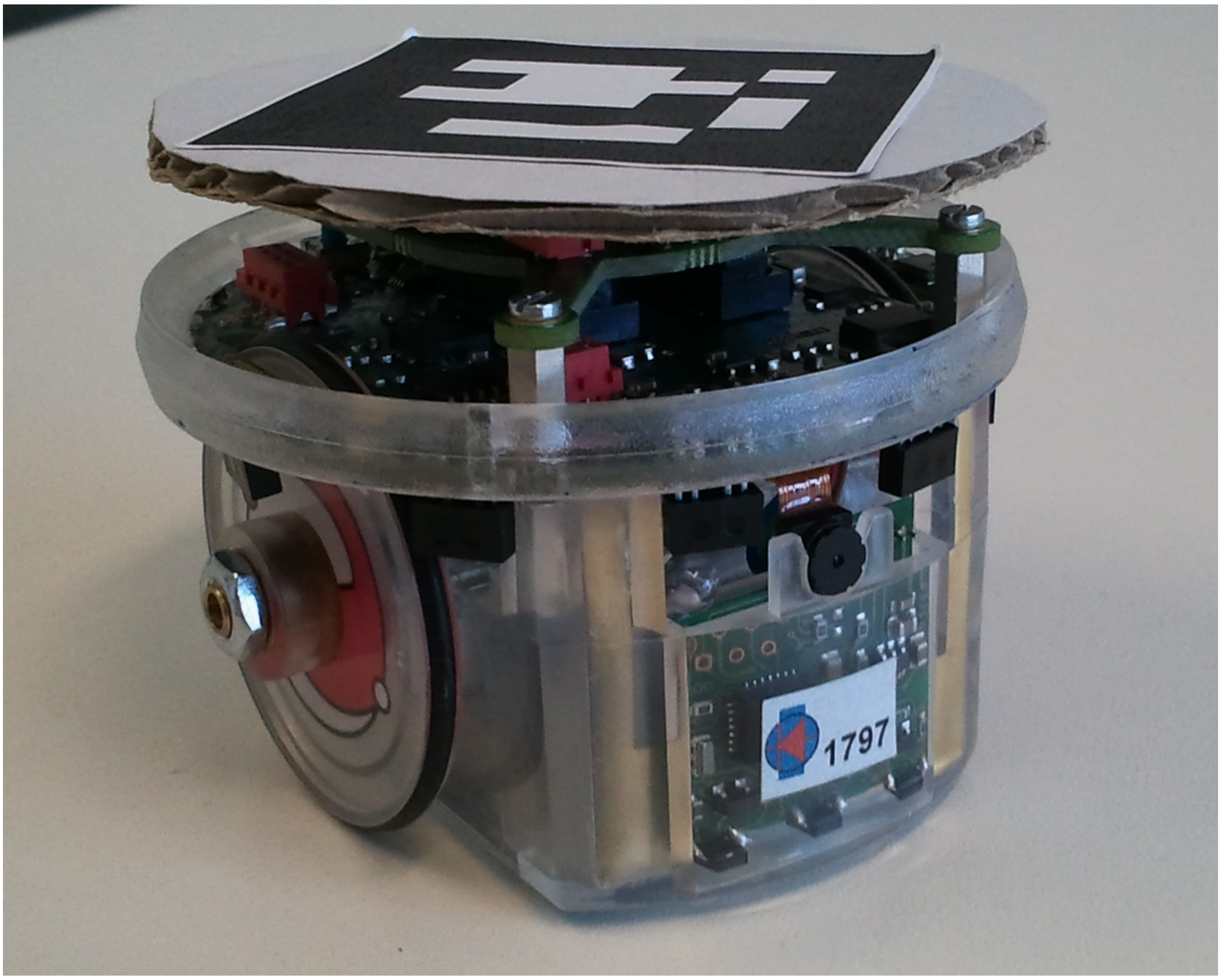


Figure 9: ArUcO mark over an ePuck robot

---

[2] `http://www.uco.es/investiga/grupos/ava/node/26`

Once the marks are identified inside the image, it easy to know if it belongs to a robot or to a landmark. If it belongs to a robot, its possible to know the robot central point coordinate and calculate the laser beam for this robot. If it belongs to a landmark, then the distance in millimetres from all robots to this landmark is calculated. All this information is used by the robots to know its relative distance to a target, and by the Area Center Method to control the robot, driving it to its goal. Fig. 10 illustrates the representation from the real environment with walls and marks, and the modelled map, with nodes and edges in red colour. Videos and more information about all the software developed for this experiments can be found in the website dedicated to it[3] or in the ITRB Labs video section[4].

This world is simpler than the world used in simulations due to the limitation of space and the number of real robots that can be used in such space. Fig. 11 illustrates the aspect of this environment.

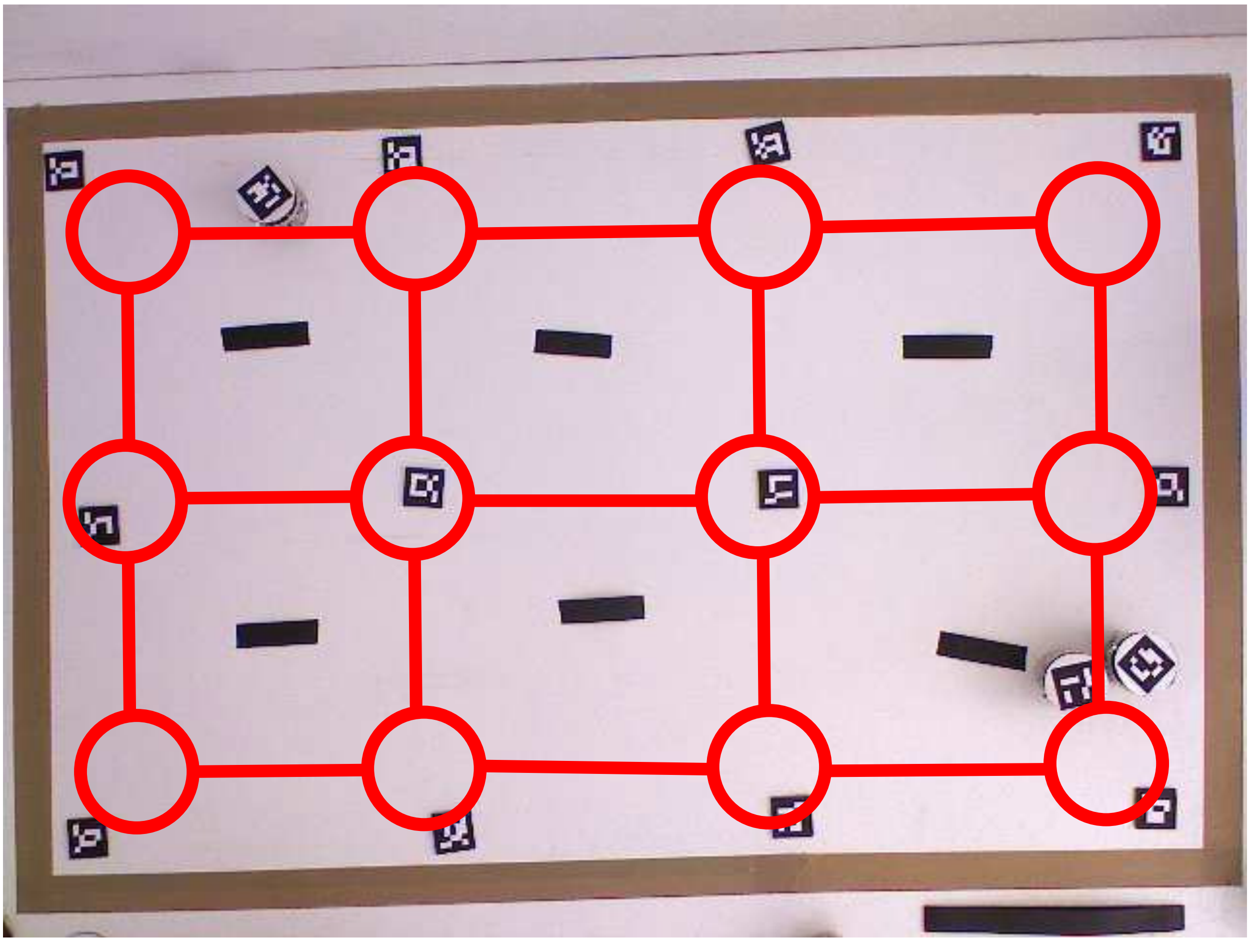


Figure 10: Graph modelled of the real world environment

The set up for real experiments are the following:

- *World*: Twelve nodes and twenty-one edges, each node is identified by an unique mark (Fig. 11).
- *Central Server/Shared memory*: A process running independently to the robots. The communication is through TCP connections.

---

[3] `http://www.ia.uned.es/~delapaz/investigacion.htm`

[4] `http://www.itrblabs.eu/videos`

- *Window size*: In this case and due to our world dimensions are 9 seconds.
- *Route depth*: For these experiments the selected depth is 2.

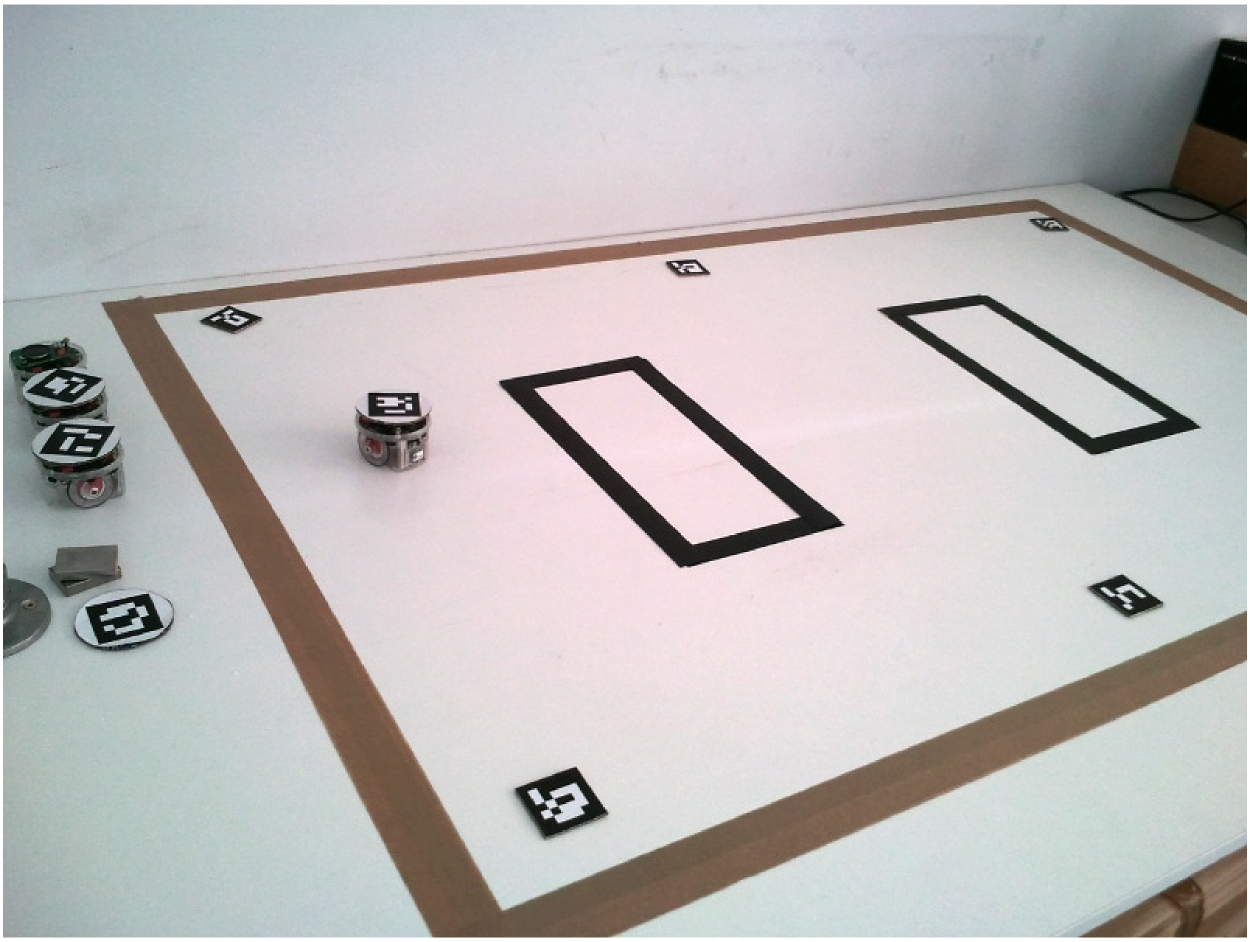


Figure 11: Real environment built for the experiments

The algorithms executed in the robots are the same that in the simulation, as well as the central server/shared memory process. However, as it was expected, the robot navigation system produces more time lapses and some errors that must be overcome to make the robot reach its goal. Fig. 12 shows the steps needed by different teams to achieve the task. During the experiments, it could be observed that the small space for the world construction provide a very reduced space for the robot movements, this causes a lot of time inverted by two or more robots trying to avoid other team mates. Therefore, the task was executed only for few robots of the team, not for all.

Finally, a comparative of the obtained results in the real scenario and a simulation with the same set up as the first is showed in Fig. 13. It exists a big difference for the same set up, probably because the issues with the robots movements.

6. Conclusions and future work

A method based in auctions and set theory for robot coordination has been presented in this paper. Our main idea, inspired by the mixed cooperative-competitive spirit of animal and insect communities, has been

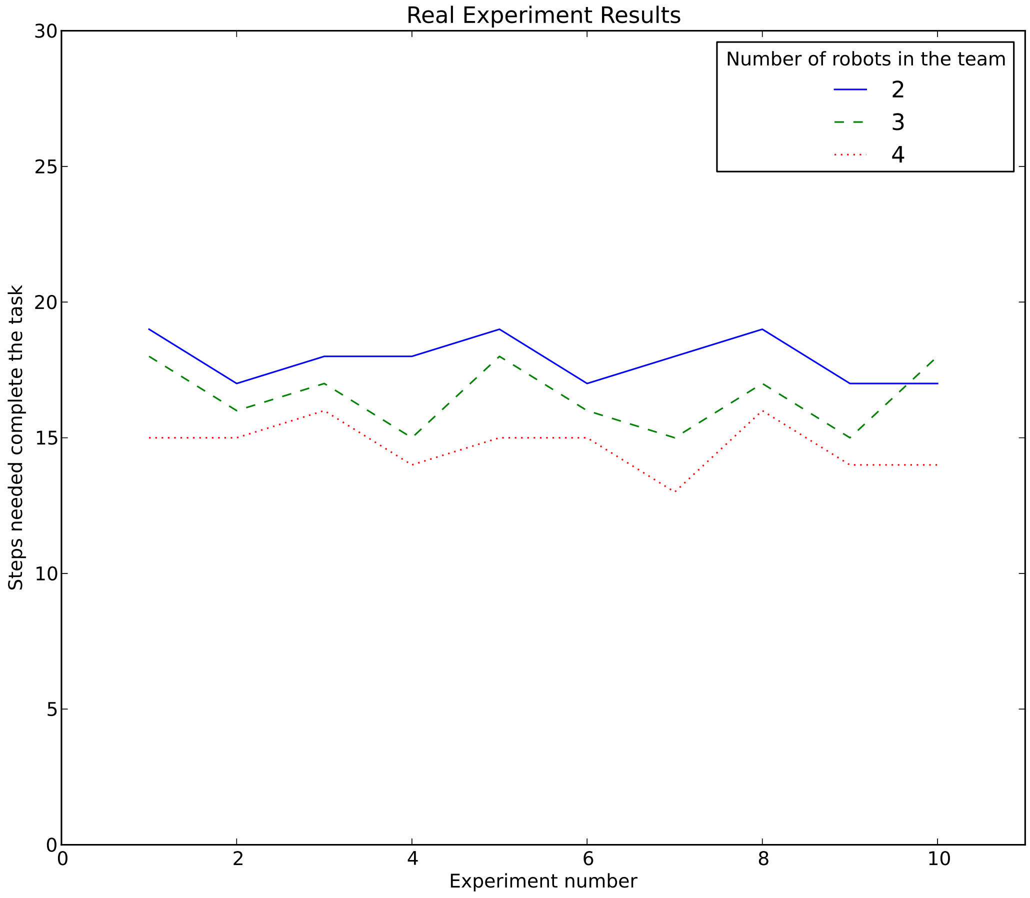


Figure 12: Results obtained in real experiments

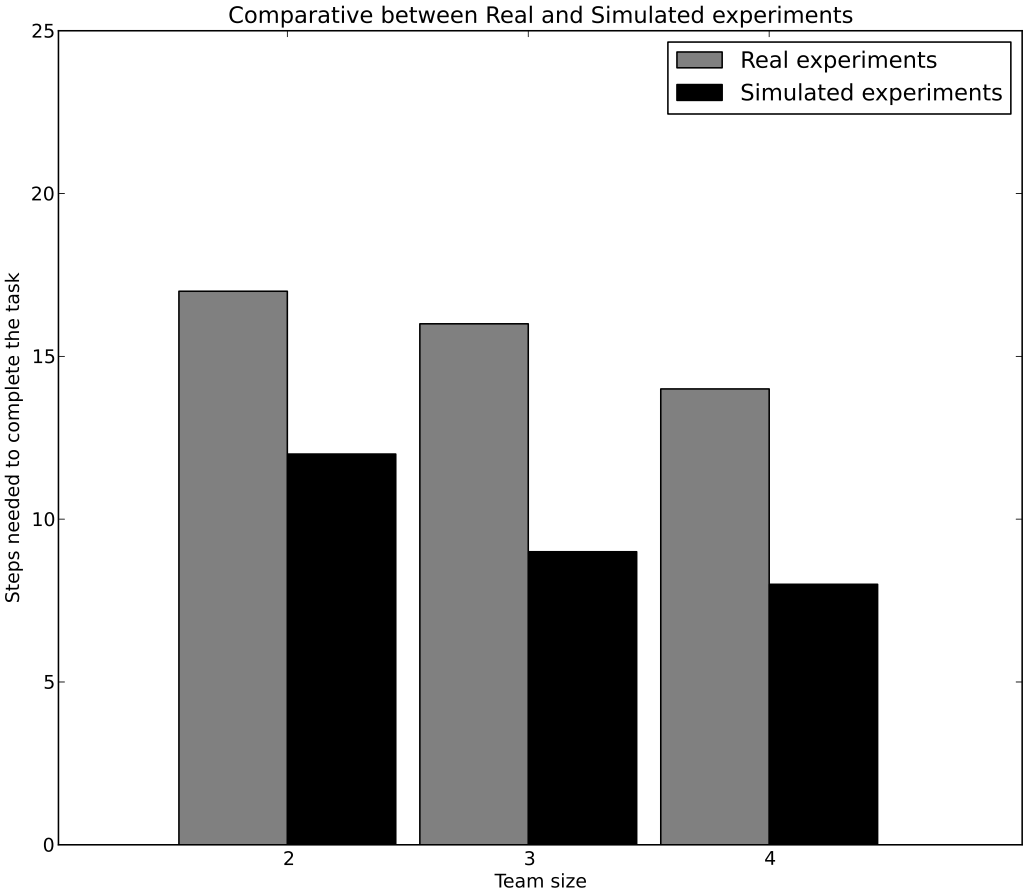


Figure 13: Comparative between Real and simulated experiments

analysed, implemented and validated with both real and simulated experiments. For real experiments, has been necessary to design and construct an ad-hoc platform with the same set-up as simulated experiments.

The results of the experiments carried out in the real environment are equivalent to the obtained in the simulated environment. Obviously, some differences are marked by the intrinsic difficulties of the real world. Despite we have tried to implement some of the innumerable variables of the real world, it is impossible simulate all of them, and this factor is reflected in the results obtained. However, the algorithm works perfectly and the robots always choose the best option for himself and for its team, independently of the communication probability. Some improvements over the real scenario are planned: a bigger world and a reduction of the space needed to identify interesting zones.

We are currently exploring the possibility to apply the SIMBA principle to others issues in robotics, a field where the individuality of the elements provide a great advantage to apply game and set theory.

References

[1] R. Wood, The first takeoff of a biologically inspired at-scale robotic insect, Robotics, IEEE Transactions on 24 (2) (2008) 341–347.

[2] C. Siagian, L. Itti, Biologically inspired mobile robot vision localization, Robotics, IEEE Transactions on 25 (4) (2009) 861–873.

[3] J. Cuadra Troncoso, J. Álvarez-Sánchez, F. de la Paz López, A. Fernández-Caballero, Improving Area Center Robot Navigation Using a Novel Range Scan Segmentation Method, Foundations on Natural and Artificial Computation (2011) 233–245.

[4] J. Martín H., J. de Lope, D. Maravall, Adaptation, anticipation and rationality in natural and artificial systems: computational paradigms mimicking nature, Natural Computing 8 (2009) 757–775.

[5] R. Duro, M. Graña, J. de Lope, On the potential contributions of hybrid intelligent approaches to multicomponent robotic system development, Information Sciences 180 (14) (2010) 2635–2648.

[6] R. Pfeifer, M. Lungarella, F. Iida, Self-organization, embodiment, and biologically inspired robotics, Science 318 (5853) (2007) 1088.

[7] M. Mataric, Behaviour-based control: Examples from navigation, learning, and group behaviour, Journal of Experimental & Theoretical Artificial Intelligence 9 (2) (1997) 323–336.

[8] T. Anderson, M. Donath, Animal behavior as a paradigm for developing robot autonomy, Robotics and Autonomous Systems 6 (1-2) (1990) 145–168.

[9] S. Ramchurn, A. Farinelli, K. Macarthur, N. Jennings, Decentralized Coordination in RoboCup Rescue, The Computer Journal 53 (9) (2010) 1447.

[10] D. Calisi, A. Farinelli, L. Iocchi, D. Nardi, Multi-objective exploration and search for autonomous rescue robots, Journal of Field Robotics 24 (8-9) (2007) 763–777.

[11] E. Rome, J. Hertzberg, F. Kirchner, U. Licht, T. Christaller, Towards autonomous sewer robots: the MAKRO project, Urban Water 1 (1) (1999) 57–70.

[12] K. Osuka, H. Kitajima, Development of mobile inspection robot for rescue activities: MOIRA, in: Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, Vol. 4, IEEE, 2003, pp. 3373–3377.

[13] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hahnel, D. Montemerlo, A. Morris, Z. Omohundro, et al., Autonomous exploration and mapping of abandoned mines, Robotics & Automation Magazine, IEEE 11 (4) (2005) 79–91.

[14] J. Lee, S. Roh, D. Kim, H. Moon, H. Choi, In-pipe robot navigation based on the landmark recognition system using shadow images, in: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, 2009, pp. 1857–1862.

[15] Y. Zhang, G. Yan, In-pipe inspection robot with active pipe-diameter adaptability and automatic tractive force adjusting, Mechanism and Machine Theory 42 (12) (2007) 1618–1631.

[16] I. Akyildiz, Z. Sun, M. Vuran, Signal propagation techniques for wireless underground communication networks, Physical Communication 2 (3) (2009) 167–183.

[17] A. Chehri, P. Fortier, P. Tardif, Application of Ad-hoc sensor networks for localization in underground mines, in: Wireless and Microwave Technology Conference, 2006. WAMICON'06. IEEE Annual, IEEE, 2007, pp. 1–4.

[18] A. Silva, M. Vuran, Empirical evaluation of wireless underground-to-underground communication in wireless underground sensor networks, Distributed Computing in Sensor Systems (2009) 231–244.

[19] A. Ortiz, F. Royo, T. Olivares, L. Orozco-Barbosa, J. Castillo, A. Fernández-Caballero, Protocol Integration for Intelligent Monitoring Applications in Wireless Sensor Networks, Foundations on Natural and Artificial Computation (2011) 511–520.

[20] M. Martín-Ortiz, J. Pereda, J. de Lope, F. de la Paz, Selective method based on auctions for map inspection by robotic teams, Foundations on Natural and Artificial Computation LNCS 6686 (2011) 175–184.

[21] M. Martin-Ortiz, J. Pereda, J. de Lope, F. de la Paz, Inspection method based on multi-agent auction for graph-like maps, in: Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on, IEEE, 2011, pp. 317–322.

[22] P. Mathis, Graphs and Networks: Multilevel Modeling (Geographical Information Systems Series), 2007.

Apéndice D

Artículo WAF 2012

Hardware And Software Infrastructure To Provide To Small-Medium Sized Robots With Advanced Sensors

Manuel Martín-Ortiz and Javier de Lope and Félix de la Paz

***Abstract*—Experiments with small robots have numerous advantages: a great number of robots in reduced space, easy world construction, experiment replication, etc. But usually, small robots have not some useful sensors like laser or accurate proximity sensor. These sensors are usually present in bigger robots, but make use of them in groups is complicated due to its size. In this paper, we present an infrastructure to provide accurate sensors for small robots, taking the big robot sensors advantages in small ones.**

***Index Terms*—Robotics, Autonomous Robots, Virtual Sensors, Augmented Reality, Artificial Sensors, Experimental Framework**

I. INTRODUCTION

THE big robots advantages are undeniable: accurate sensors, computing capacity or adaptation to real world scale are some of the advantages that these *great boys* have in opposite to other alternatives. But usually its computer capacity and sensor accurate is directly related with its cost, and it make difficult to make swarm experiments with big robots. It is here, in swarm robotic, where the small robots became a great option: they can be controlled by a powerful computer if you need it, they have a long battery life compared with the *big boys* and you can get a lot of them for the same money that you acquire a big one.

Despite simulators are one of the best options to make experiments [1] [2] [3], there is a time when our work must be validated on real environments [4] [5]. At this moment, the best options in swarm robotics is make use of small robots. In this field, the ePuck [6] or Khepera [7] robots are two of the most used options.

However, for some experiments these small robots can present some difficulties due to its limited number of sensors and its accuracy. Becoming to a nightmare some simple experiments. Here is when a couple of accurate or advanced sensors can help to researchers to overcome some uncomfortable situations, and it is here where our work is focused.

In this work we present a simple infrastructure composed by hardware platform and a software library to provide to small robots with virtual advanced sensors, such as laser, proximity sensor or compass. Making use of modern augmented reality techniques, we present a cheap solution to improve seriously a swarm robotic team.

Fig. 1. Some popular small robots

This paper is structured as follow: first, we present the potential end users of this system. Later, we explain how the system works, detailing the foundation of each implemented sensor, we present the visual interface of the library and some works where this infrastructure has been used. Finally, we expose our conclusions and the future deployment for this work.

II. SCOPE

The main propose of this system is to provide to small robots with a set of sensors that allow them to execute advanced algorithms. Our main objective have been the navigation algorithms, but these sensors can be used in a great variety of applications, like object detection or coordination.

To provide these sensors, we have divided the system in two parts: (1) a hardware platform and (2) a software library. The objective of the hardware platform is to get a controlled environment where execute the experiments. In this place, we can localize the robots and build the sensors through the software library. Later, these sensors are provided as requested by the robots. Fig. 2 shows the aspect of our controlled environment.

Manuel Martín is with ITRB Labs.
E-mail: manuel.martin@itrblabs.eu

Javier de Lope is with Universidad Politécnica de Madrid.
E-mail: javier.delope@upm.es

Félix de la Paz is with UNED.
E-mail: delapaz@dia.uned.es

This system allow to research centres, universities and amateurs to provide sensors like laser, compass or proximity-sensor to any kind of robots, with a simple VGA camera and our library. So, making experiments with an advanced navigation system and small robots is now possible.

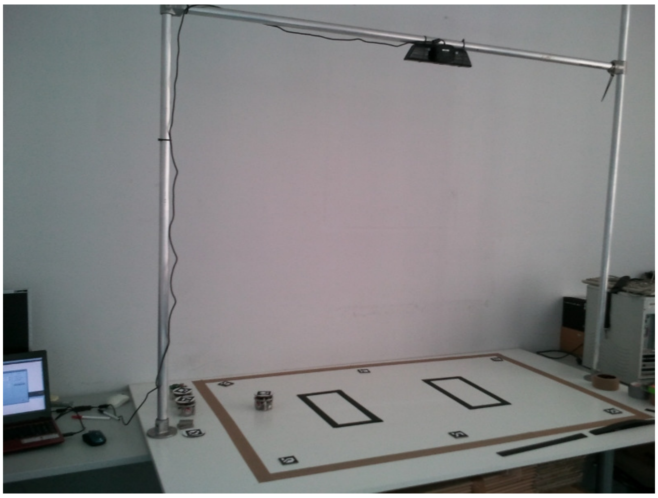


Fig. 2. An example scenario built for experiments with swarm robots

III. How does it work

The behaviour of the software library is as follow:

```
forever:
    fetch_image()
    identify_robots()
    detect_orientation_and_direction()
    process_laser_sensor()
    process_compass_sensor()
    process_proximity_sensor()
    communicate_new_data_exists()
```

a VGA camera is set in zenith position, so it captures video images of the environment, identifying what a robot is and detecting its position and its orientation on the image. Finally, all sensors are processed. For robot detection we make use of augmented reality, in our case, we have used Augmented Reality Univerdidad de Córdoba (ArUcO)[1], a library developed in Córdoba University by the *Applications of Artificial Vision* group. The library is able to detect with high precision determined *marks*, so, we have printed one of these marks that the robot *wears* as a hat. For a more detailed description of how the library works, we recommend to visits its web site, where it is explained and some fabulous use cases showed.

The robot used in our case is the ePuck robot, but any other robot can be used, the only requisite is that the robot *wears* the mark identified by ArUcO, as is showed in Fig. 3. Due to the limited ePuck computation capacity, it is controlled by Bluetooth and all the hard work computed in a more powerful machine.

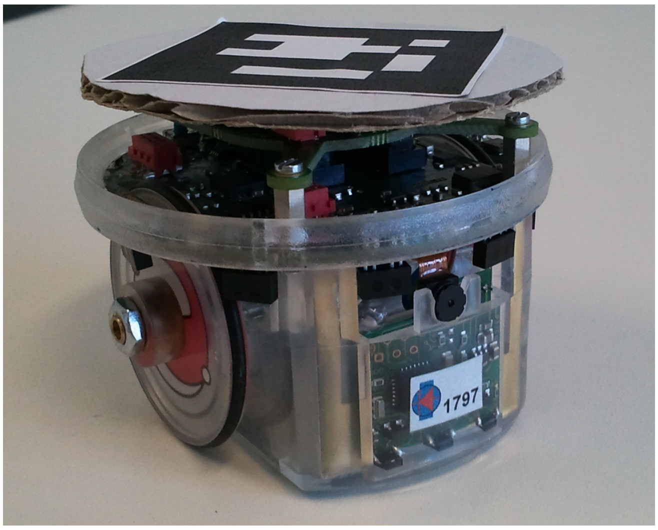


Fig. 3. ePuck robot *wearing* a mark used for the robot detection

Once the mark is detected and identified as a robot, the ArUcO library provides its central point. This point is used as origin for the Robot Coordinate System (RCS) (showed in Fig. 4), used to calculate each robot sensor.

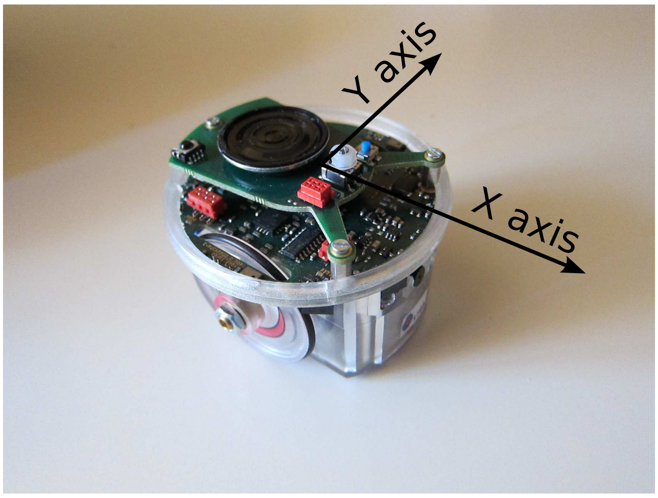


Fig. 4. Local robot coordinate system

A. *Virtual Laser Sensor*

For the laser sensor it is needed to use a binary image, where the obstacles and the free space are clearly identified. All laser beams are calculated from the RCS origin and the nearby objects. Usually, a laser with a range of 180° is enough for navigation, but a more wide laser is easy to configure allowing us to have a maximum laser of 360°.

The first laser beam corresponds with the Y axis and the last with the -Y axis, the beams are separated 1 degree (0.039 radians, approximately), but this parameter is easily configurable. The laser is provided to the robot as a list of vectors, with its corresponding magnitude r (in millimetres)

---

[1] http://www.uco.es/investiga/grupos/ava/node/26

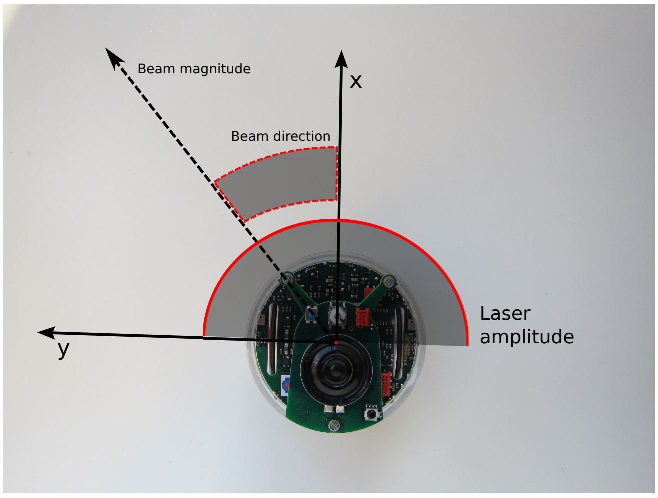


Fig. 5. Laser distribution

and direction α (in radians). Fig. 6 shows how the laser is represented over the RGB image acquired by the camera.

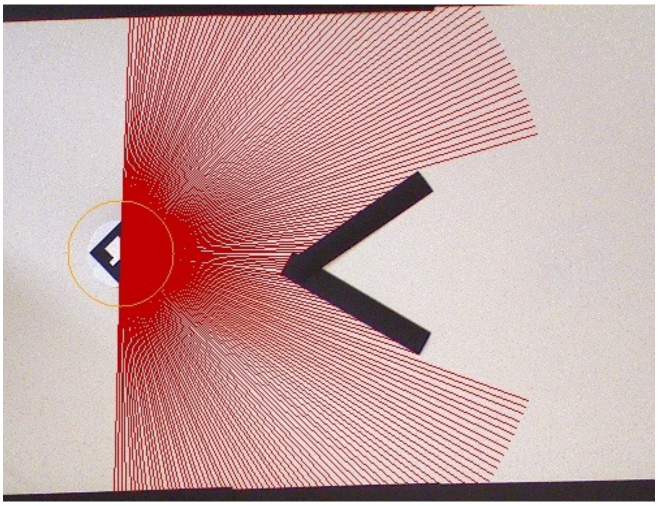


Fig. 6. Laser Sensor

B. *Virtual Proximity Sensor*

The Virtual Proximity Sensor returns a list of vectors that represent the distance and direction (in millimetres) to other robots, specific ArUcO marks or Zone Of Interest (ZOI). The ZOI are equivalents to ArUcO marks, but these are zones that only can be established manually on the image. The sensor maximum range is a configurable parameter that indicates the maximum working distance, i.e. if the object that must be detected is too far, the virtual sensor will ignore it and, therefore, not detected.

In Fig. 7 can be appreciated how the proximity sensor is represented. Four of the ZOI in the environment are detected because it is inside the proximity range, the other two are not detected because are too far. In this image, the measure is showed in millimetres.

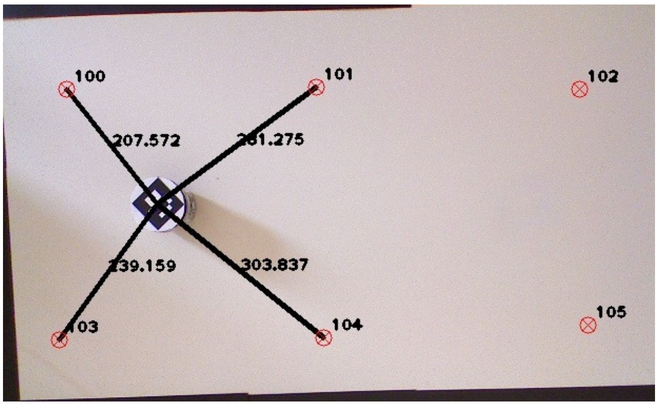


Fig. 7. Proximity Sensor representation

C. *Virtual Compass Sensor*

A specific ArUcO mark can be configured to work as compass. The direction marked by its X axis is considered North. The sensor return the offset (in radians) of the RCS respects the North.

IV. INTERFACE

A graphical interface based on the cross-platform Qt Libraries [2] has been developed to interact with the Virtual Sensor Library. The aim of this interface, is to provide a way to control all configurable parameters as well as show the current status of the different sensors.

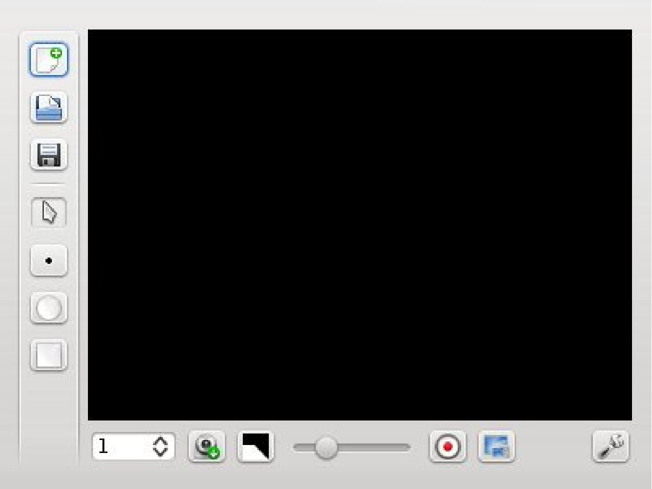


Fig. 8. Main Qt Widget for the Virtual Sensors

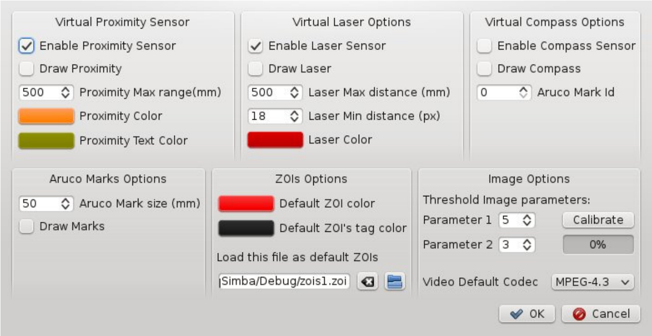
Thereby, and using the Qt Widget showed in Fig. 8, the user can develops easily an application to see, record, and debug its experiments. This widget provides some very useful tools to interact with he main library:

- Zone Of Interest tools: These tools allow to the user to clear, load and save to a file the current ZOI established. Also, it provides tools for the creation of three kind of ZOI to be used by the proximity sensor.

---

[2] http://qt.nokia.com/

- Device section: A simple input that indicates the device to be used to capture the image.
- Image section: Various buttons to switch between colour and threshold image, set the threshold for the binary image used for the Virtual Laser and buttons to record a video of the experiments.


Virtual Proximity Sensor
☑ Enable Proximity Sensor
☐ Draw Proximity
500 Proximity Max range(mm)
🟧 Proximity Color
🟩 Proximity Text Color

Virtual Laser Options
☑ Enable Laser Sensor
☐ Draw Laser
500 Laser Max distance (mm)
18 Laser Min distance (px)
🟥 Laser Color

Virtual Compass Options
☐ Enable Compass Sensor
☐ Draw Compass
0 Aruco Mark Id

Aruco Marks Options
50 Aruco Mark size (mm)
☐ Draw Marks

ZOIs Options
🟥 Default ZOI color
⬛ Default ZOI's tag color
Load this file as default ZOIs
|Simba/Debug/zois1.zoi

Image Options
Threshold Image parameters:
Parameter 1 5
Parameter 2 3
0%
Video Default Codec MPEG-4.3

Fig. 9. Properties dialogue

The properties dialogue, showed in Fig. 9, presents a simple interface to configure various parameters of the Virtual Sensor.

V. EXAMPLES

The sensors provided by this system have been used in the experiments designed to test the work presented in [8] and [9]. All the system presented in this paper worked perfectly, providing a invaluable information in form of sensors to apply complex behaviour proper of expensive robots with powerful sensors.

On the website dedicated to these experiments[3] are hosted videos where the use of the different sensors can be appreciated. For these test, both the laser as the proximity sensor have been used intensively in the robot navigation system.

As main navigation system, the robot uses the Area Center Method (ACM) [10]. This navigation algorithm is able to avoid efficiently both static as dynamic obstacles and simply require a laser to determine the wheels robot velocities. That is the first time that a robot so small is able to navigate using the ACM.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented an economic platform to aid small robots to execute complex algorithms that need of some sensors that usually they does not implement. This system, built by less than 400€, will allow to research centres, university laboratories, schools or amateurs, to reproduce experiments in controlled environments using all kind of small robots. Also, thanks to the library license, researches or enthusiastic around the world can develop new sensors or improve the library, allowing others to make use of new features.

The system presented has been deployed in two different places, proving in both cases its reliability and easy set up.

Thanks to this system, can be done classical robotics experiments reproduction for educational purposes, or the designing and testing of new ones.

ACKNOWLEDGMENT

The authors would like to thank to Jose Manuel Cuadra Troncoso all the hours that has dedicated to support the deployment of the Area Center Method in this system, and to the INT3 project (TIN2010-20845-C03-02)[4] by its support.

REFERENCES

[1] M. Polycarpou, Y. Yang, and K. Passino, "A cooperative search framework for distributed agents," in *Intelligent Control, 2001.(ISIC'01). Proceedings of the 2001 IEEE International Symposium on*. IEEE, 2001, pp. 1–6.

[2] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle *et al.*, "ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 5027–5034.

[3] O. Michel, "WebotsTM: Professional mobile robot simulation," *Arxiv preprint cs/0412052*, 2004.

[4] R. Brooks, "Artifical life and real robots," in *Toward a practice of autonomous systems: Proceedings of the First European Conference on Artificial Life*, 1994, pp. 3–10.

[5] H. Lund and O. Miglino, "From simulated to real robots," in *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. IEEE, 1996, pp. 362–365.

[6] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th conference on autonomous robot systems and competitions*, vol. 1, no. 1, 2009, pp. 59–65.

[7] F. Mondada, E. Franzi, and A. Guignard, "The development of khepera," in *Proceedings of the 1st international Khepera workshop*, vol. 64, 1999, pp. 7–13.

[8] M. Martín-Ortiz, J. Pereda, J. de Lope, and F. de la Paz, "Selective method based on auctions for map inspection by robotic teams," *Foundations on Natural and Artificial Computation*, pp. 175–184, 2011.

[9] M. Martin-Ortiz, J. Pereda, J. de Lope, and F. de la Paz, "Inspection method based on multi-agent auction for graph-like maps," in *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*. IEEE, 2011, pp. 317–322.

[10] J. C. Troncoso, J. Álvarez-Sánchez, F. de la Paz López, and A. Fernández-Caballero, "Improving area center robot navigation using a novel range scan segmentation method," *Foundations on Natural and Artificial Computation*, pp. 233–245, 2011.

---

[3] http://www.ia.uned.es/~delapaz/investigacion.htm

[4] http://www.dsi.uclm.es/personal/AntonioFdez/nais/proyecto_INT3.html