Beyond RoboCup: Ad-Hoc Networking for Autonomous Mobile Robots

Alfredo Weitzenfeld, Luis Martínez-Gómez,
Juan Pablo Francois, Alejandro Levin-Pick
Robotics Laboratory, ITAM
Rio Hondo 1, San Angel Tizapan
Mexico City, MEXICO, 01000
alfredo@itam.mx

Katia Obraczka, Jay Boice
Internetworking Research Group (i-NRG), UCSC
Santa Cruz, CA
USA
{katia,boice}@soe.ucsc.edu

***Abstract*—In this paper we describe preliminary results from a collaborative effort between ITAM's Robotics Lab and UCSC's Internetworking Research Group (i-NRG) focusing on extending the Small-Size league RoboCup system architecture to enable multi-robot collaboration beyond communication limited to a soccer field environment. More specifically, we are developing a wireless ad-hoc network architecture that will make it possible for robots to cooperate in carrying out tasks such disaster recovery and emergency response.**

1. Introduction

Moving from single to multiple robot systems is an important challenge in the robot and AI research community. RoboCup competitions [1], initiated 10 years ago, have become a well-known venue where coordination among multiple robots teaming in a soccer game can be evaluated. While there have been significant improvement in the performance of the RoboCup teams over the years, several aspects of the competition simplify the multi-robot coordination problem. One clear example is the fact that the limited size of the soccer arena simplifies communication requirements and that robot control is centralized, in this particular league, in an individual computer sending commands to all robots. Yet, an important challenge in multi-robot applications is managing communication under more stringent and adverse conditions. For example, in disaster recovery and emergency response applications, robot teams collaborating in rescuing or reconnaissance operations are deployed in arbitrarily wide areas with tortuous terrain and subject to communication impairments such as interference, noise, signal fading, etc.

The goal of this paper is to discuss how to extend the current limited sized RoboCup architecture to larger multi-robot environments using ad-hoc networking. The paper is organized in thee major sections, namely: Section 2 describes ITAM's Eagle Knights Small-Size RoboCup team; Section 3 presents an extension to the global visually-guided small-size league architecture into local vision control; Section 4 discusses current algorithms in ad-hoc networking; Section 5 provides a discussion of applications and current collaborative effort between ITAM´s Robotics Laboratory and UCSC's i-NRG on adding ad-hoc networking capabilities to the current multi-robot architecture; Section 6 presents conclusions and future work.

2. RoboCup – Small Size League

The RoboCup Small-Size League (SSL) consists of two teams of five robots, up to 18 cm in diameter each, playing soccer on a 4 by 5.4m carpeted soccer field. ITAM's Eagle Knights [2] have been participating since 2003 in a number of international competitions including the world cup. The system architecture consists of one or two remote computers running remotely from the robots. Computers receive video signals from cameras mounted on top of the field and provide wireless signals to the five robots in the team as shown in Figure 1. The main functional components of the system are: (1) Vision System, (2) AI System, (3) Referee Box, and (4) Robots, as shown in Figure 2. The computation cycle is 60 times per second.

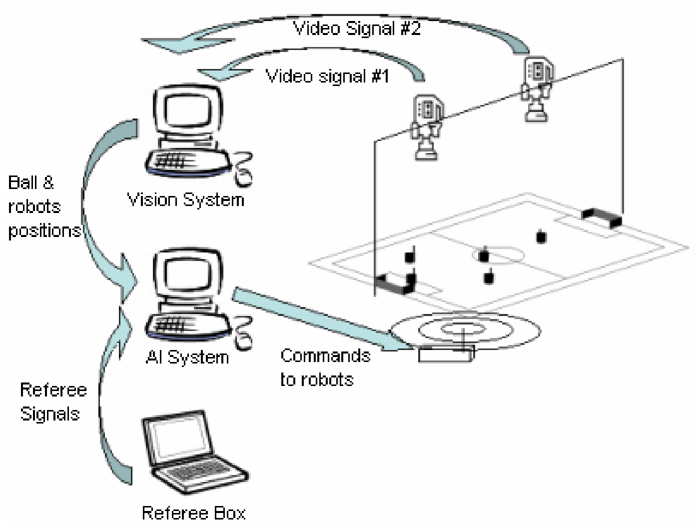

Fig 1. ITAM's RoboCup Small-Size league system architecture

2.1 Vision System

The Vision System is the only source of external input in the application. Its main tasks are to capture video in real time from the two cameras mounted on top of the field. The system needs to recognize the set of colors assigned to the objects of interest in the game, namely robots and ball, according to the SSL rules [3]. Once objects are recognized, the system identifies and computes the position of the ball together with position and orientation of the robots in the team, while computing the position of the robots of the opposite team. Robots in one team must have a blue colored circle patch on top with 50mm diameter while the other team must have a yellow colored patch. Additional patches are used to compute robot orientation as shown in Figure 3.

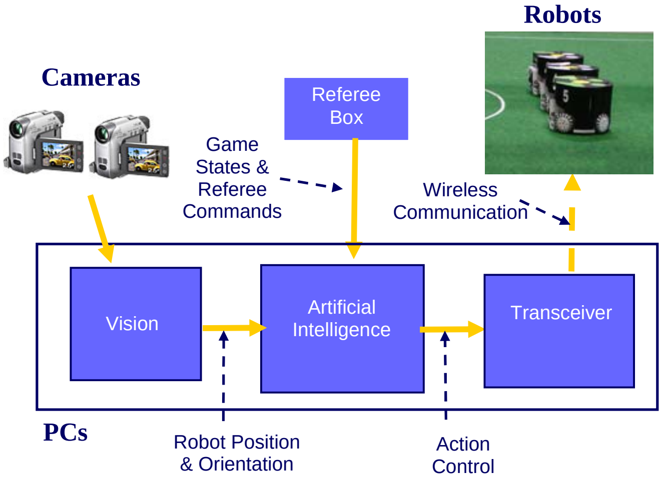

Fig 2. ITAM's RoboCup Small-Size league block diagram

A particularly critical challenge in the Vision System is to adapt to different light conditions by performing a dynamic color calibration scheme. Resulting object positions and orientations, when applicable, are transmitted to the AI System. More details can be found in [4].

Fig 3. Eagle Knights robot showing colored patches

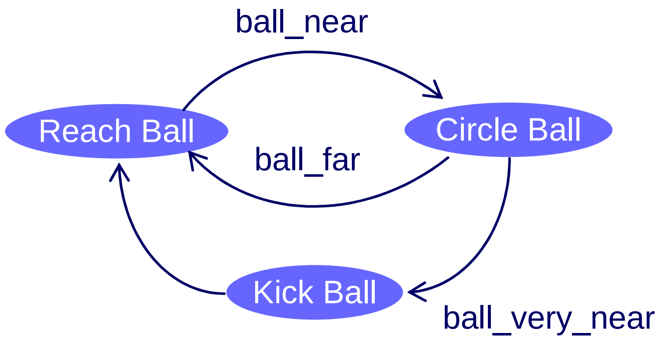

Fig 4. Attacker state machine

2.2 AI System

The AI or High Level Control System receives object positions and orientations, e.g. their localization, from the Vision System in order to produce robot action commands. These actions depend on strategic decisions made a priori depending on robot roles, e.g. goalkeeper, defense, and forward, and on the current state of the game, e.g. attacking or defending. Additional game state information comes from the Referee Box, e.g. regular play, free kick, etc. The AI System is composed of eight modules: (i) Artificial Intelligence, (ii) Simulation System, (iii) Collision Detection, (iv) Transceiver Communication, (v) Omni-directional Drive Control, (vi) User Interface, (vii) Vision System Communication, and (viii) Game Control. The system has a main thread calling each of the modules in sequence, being designed in such a way that modules can be tested individually by linking them to a simulator with dynamics similar to the real robot environment. Robot behaviors can be described in different ways; a simplified state machine diagram used by our robot attackers is shown in Figure 4. Final decisions are converted to commands that are sent to the robots via a wireless link through a transceiver linked to the PC via a parallel port interface. Transmission is fully asynchronous.

2.3 Referee Box

The Referee Box communicates additional decisions (penalties, goal scored, start of the game, etc.) as a set of predefined commands to the AI system through a serial link.

2.4 Robots

The Robots execute commands sent via wireless communication by the AI system in order to produce local robot actions, e.g. move, kick, and dribble. Robots in this league are omnidirectional having either three wheels as shown in Figure 5 (or four wheels as in the robot design shown in Figure 3. We are using in this project the three wheeled design.) A block diagram for the robot architecture is shown in Figure 6. In addition to the three motors controlling the wheels, there is an extra motor controlling the dribbler that "sucks" the ball tight to the robot body as shown in Figure 5. Note the golf ball being handled by the dribbler. Additionally, the robot includes a solenoid to kick the ball. Local robot control is managed by a Texas Instruments TMS320LF2407A fixed-point single chip DSP (Digital Signal Processor) optimized for digital motor and motion control. The DSP receives remote communication from the AI System via a Radiometrix RPC-914/869-64 local transceiver with radio frequency at either 914MHz or 869MHz with 64kbit/sec transmission rate similar to one attached to the PC. Teams alternate in radio frequency. Finally, two rechargeable 9V/1600mA batteries are incorporated in the robot.

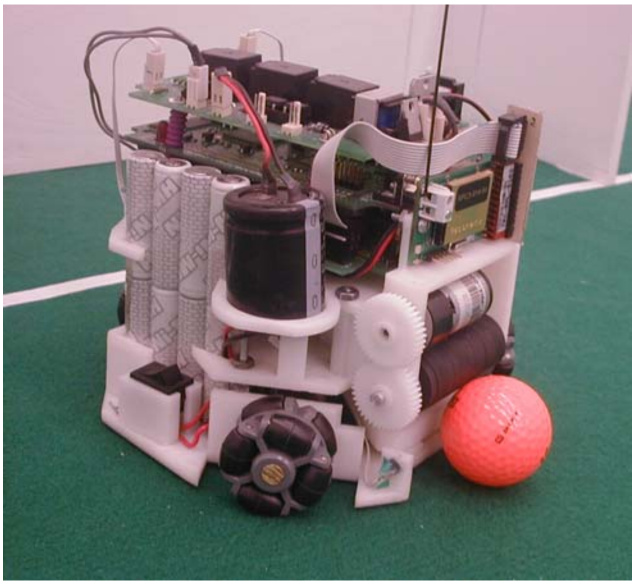

Fig 5. Eagle Knights robot mechanical design

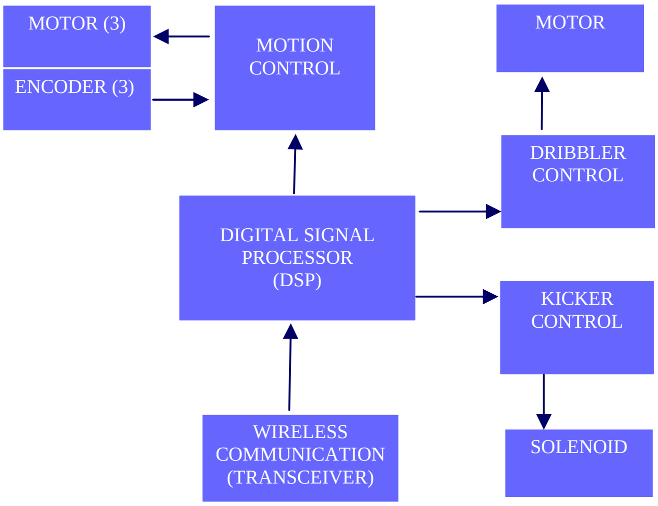

Fig 6. Eagle Knights robot block diagram

3. Local Vision Architecture

A major constraint in the small-size league architecture is the global vision system restricting the mobility of the robots to the soccer field under full camera view. By providing a local vision system as in the case of the Mid-Size and Four-Legged RoboCup leagues [1] it becomes possible to avoid this restriction. For this purpose we have extended our robot design to include a local camera located where the dribbler and kicker used to be while adding a Stargate [5] with WiFi-based [6] communication instead of the local transceiver. The modified robot design is shown in Figure 7. It is based on Eagle Knights second generation robots integrating a Stargate [5], which in turn is outfitted with a webcam and an 802.11 wireless card. Crossbow's Stargate is a relatively powerful, small form factor single-board computer that has found applications in ubiquitous computing and wireless sensor networking. It is based on Intel's 400MHz X-Scale processor and has 32MB flash memory and 64MB SDRAM and provides PCMCIA and Compact Flash connectors on the main board. It also has a daughter board with Ethernet, USB and serial connectors. A Logitech QuickCam Pro 400 webcam is connected through the USB port, and communication carried out by an Ambicom Wave2Net IEEE 802.11b compact flash wireless card. The operating system is Stargate's version 7.2, an embedded Linux system (kernel version 2.4.19).

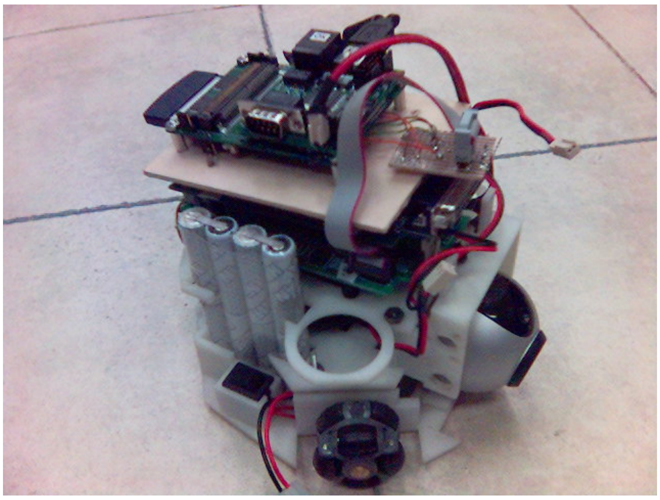

Fig 7. Eagle Knights modified robot having local camera and WiFi communication circuitry

4. Wireless Ad-Hoc Networking

In the RoboCup Small-Size soccer league, robots are very close to each other on the soccer field. This means that all robots are within the transmission range of one another which makes routing of messages between computer and robot, or between robots, trivial; any robot can send a message to any other robot in a single transmission. For other applications, however, as the range of robot mobility is extended, nodes may be too far apart to directly communicate and messages must be routed through intermediate robots to reach their destination. In such situations, known as multi-hop ad hoc networks, nodes must cooperatively establish routes and forward messages in order to maintain communication.

In terms of ad-hoc networking protocols, the Stargate used in our system architecture is shipped with AODV [6], the Ad hoc On-demand Distance Vector routing protocol. We describe AODV in Section 4.1 as a representative routing protocol currently used in ad hoc networks. AODV has been designed under the assumption that end-to-end paths are available at least most of the time. In other words, it is assumed that the network is connected most of the time and that disconnections, when they happen, are short lived. However, in some situations such as disaster recovery or emergency response scenarios, end-to-end connectivity cannot be guaranteed; in fact, it may turn out that the network is disconnected for most of its operational lifetime. For this reason, we have proposed StAR (Steward Assisted Routing), which we describe in Section 4.2. StAR is a routing protocol for networks in which links are often unavailable due to mobility or other interference (see Figure 8).

4.1 AODV

Unlike traditional wired networks, multi-hop ad hoc networks (MANETs) require a routing protocol that can respond quickly to node failures and topology changes. AODV is an example of an on-demand routing protocol. It establishes a route between a source-destination pair only when the source node has data to send to the destination. This notion is in contrast to proactive routing protocols commonly used in the Internet, which can afford the luxury of maintaining all necessary routes since they rarely change. Because routes can change very quickly in a MANET, the signaling overhead required to maintain all routes at all times can be prohibitively high.

AODV's route establishment phase consists of two main control messages, the RREQ (route request) and RREP (route reply). A robot, when desiring to send a message to another robot, must send a route request for the destination. This request is broadcast to all neighbors and relayed by intermediate nodes until it reaches the destination, or a robot with a route to the destination, at which time a route reply message is unicast back to the source robot. This message sequence establishes the (temporary) route so that packets may be forwarded from source to destination. For a much more detailed description of AODV, the reader is referred to the AODV RFC [5].

The major failing point of AODV, and other on-demand routing protocols such as DSR [6], occurs when there is no existing end-to-end path from source to destination, and the route discovery phase fails. In this case, data packets are dropped, and the destination does not receive the intended messages.

4.2 StAR

The objective of StAR is to nominate, for each connected partition in the network, a *steward* for each destination. These stewards are the robots that are next expected to have communication with the destination. For example, if there is a single moving robot who communicates with all other stationary nodes, this robot is likely to be nominated as the steward for all destinations. Messages are sent to the associated steward, who will store them until a route to the destination (or a better steward) is available.

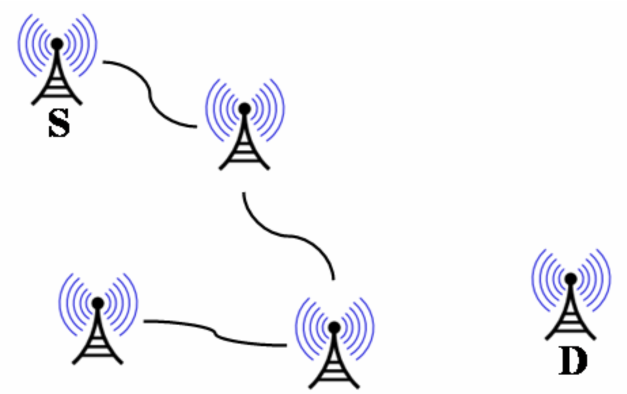

Fig 8. An example network in which there is no route from S to D. Existing on-demand routing protocols fail to deliver messages when a route cannot be established. StAR will buffer data at the node nearest to the destination until a route is available.

StAR routes messages using a combination of global (network-wide) contact information and local (intra-partition) route maintenance. The topological location of active destinations in the network is propagated through periodic broadcasts, or contact exchanges, between neighbors. These broadcasts occur at a fixed interval if there are nearby nodes, and contain only those entries in the routing table that may have changed since the last broadcast to the same set of neighbors. The message includes a unique sequence number indicating the broadcast from which the information came.

Initially, each node nominates itself as the local steward for each destination, and therefore does not route messages to any neighbor. As updates are received from neighbors that advertise better local stewards, routes are formed. The ranking of stewards is based on the most recently heard sequence number for a destination, or route length if two nodes have the same destination sequence number. In a connected network (i.e, a network in which there are connected routes between all robots), each tree will be rooted at the destination itself, and messages routed directly to the destination.

Thus, route maintenance results in one tree per destination of interest in each partition, where each tree is rooted at the locally nominated steward for that destination. Note that it is possible (and quite likely) that a node can be the steward for more than one destination at any given time, and the tree for each destination will contain precisely the same nodes and links.

5. Related Work

In this section, we briefly describe related work in the areas of sensor networking that is being developed in our labs. We also describe some initiatives that employ mobile autonomous robots.

5.1 Sensor Networking

The i-NRG lab at UCSC is currently deploying several related projects in the area of ad hoc sensor networks. Like the Eagle Knights Small-Size RoboCup team, these projects involve the integration of custom-built hardware with ad-hoc network protocols specifically designed for the environments in which they are used, as well as the data that is to be delivered. Experience with each of these projects is being leveraged into the Eagle Knights project.

The CARNIVORE system (Carnivore Adaptive Research Network in Varied Remote Outdoor Environments) was born from the desire to further understand the interplay between coyotes, their predators and their ecosystem in the Santa Cruz mountains. Custom collars have been developed that contain a 3-axis accelerometer, GPS, 74MB of storage space, and communication capabilities. Collared coyotes will continually sense and transmit data to static base stations deployed in the area, and the data will later be aggregated and used in analysis of their behavior. Similar to the Eagle Knights project, the network topology is quite sparse, resulting in a network that is rarely connected. Similar mechanisms will be used to ensure that messages are delivered in a timely fashion to the sink nodes.

Meerkats is a battery-powered wide-area surveillance system incorporating both sophisticated vision algorithms and a power-management scheme to enable long network lifetime. Much of the hardware used in the Meerkats project is similar to the Eagle Knights robot team; the Stargate is used as the base platform, while a Logitech QuickCam Pro 400 webcam is used for image captures. Unlike the Eagle Knights project, the Meerkats network is static, allowing the use of more traditional ad-hoc networking. Detailed analysis of power consumption has enabled the network to be designed such that lifetime is maximized. Power monitoring enables a distributed resource manager to instruct nodes to turn on or off their components such as wireless card and USB camera.

Finally, the SEA-LABS project (Sensor Exploration Apparatus utilizing Low Power Aquatic Broadcasting System) has been designed to monitor remote coral reefs. This project, since it is also battery-powered, must adhere to strict power-consumption guidelines in both sensing and transmission. A successful deployment in the Monterey Bay has provided initial data, and a full deployment in the Midway Atol is planned for the future. The devices, since they used in such extreme environments, must require minimal maintenance and extremely long lifetime. Furthermore, the harsh environment and large distance between nodes (up to 8km) requires that the networking be designed with reliability as a key consideration.

5.2 Surveillance

In addition to outfitting each robot with ad hoc networking capabilities, we have loaded them with a simplified surveillance application. Each robot is defined as either a source (sensor) node or a destination (sink) node. It is the responsibility of source nodes to capture an image of their surroundings through the webcam at 5-second intervals, and transmit them to a designated sink. Because there may be no direct route to the sink at the time the image is taken, StAR ensures that the image is buffered at some intermediate node until a route toward the destination exists. We are currently experimenting with a wide range of network topologies using StAR on the Eagle Knight robots for comparison with standard on-demand routing protocols.

5.3 Search and Rescue

In recent years robots have demonstrated their usefulness in supporting life-threatening human tasks. Among these, Urban Search and Rescue (USAR) [8] has been an area where robotics is starting to have an important impact [9].

In particular, as a result of earthquakes or other collapsed building disasters, one of the most important tasks involves search and rescue of trapped survivors. The main challenges in rescue operations are posed by the unstable nature of the structures, the hard to reach spaces, the lack of oxygen, and the hazards resulting from fire, toxic gases, or other chemicals. In the past, specialized sensory equipment has been used in assisting rescuers, yet most of this technology is used from outside the disaster perimeter. In order to get closer to survivors, scientists are currently experimenting with mobile robots with varying shapes, sizes and capabilities [10]. Until now, most search and rescue robots are remotely operated, resulting in a number of limitations:

(a) The number of robotic devices required to control a large-scale search and rescue operation is significant, requiring a large number of trained human controllers.

(b) Coordination between human controlled teleoperated robotic devices is hard, limiting the possibility of shared decision support systems.

(c) Poor environmental conditions, such as low visibility, make human maneuvering of robotic devices difficult.

(d) Teleoperation relies on continuous availability of robust communication channels and power sources, including the use of wirelines.

In overcoming these restrictions, search and rescue robots will become more autonomous with time, interacting only with human controllers for higher-level decisions making. Thus, robots will help in the overall search and rescue operation, for example, producing maps of how to reach a survivor's location. Eventually, these tasks can be extended in asserting survivor condition and existing hazards. Real robotic environments have been developed to assess and test these capabilities, such as RoboCupRescue arenas [11]. Multiple approaches have been proposed to solve these problems. In particular, our group is currently involved in biologically-inspired robot exploration models [12].

It is important to highlight that there are many strategies for collaborative robot exploration. If we consider that time is probably the most critical factor in human rescue, it is important to achieve this independently from robot cost or number of robots involved. When dealing with topological navigation, there are a number of strategies that can be used including different types of topological map integration and different communication schemes including direct communication between robots and/or base station. Additionally, different numbers of robots and order of navigation can be involved, for example groups of multiples robots sent one group after the other where each group starts its exploration from cognitive maps already produced by previous robots. Obviously the difficulty of the problem is directly proportional to the size and complexity of the search arena.

In order to enable such a collaborative robotic approach, it is also necessary to incorporate adequate ad-hoc network support.

6. Conclusions and Future Work

We present a description of the Eagle Knights Small-Size RoboCup team with modifications for local vision control and WiFi based ad-hoc networking. We motivate the need for ad-hoc networking by overviewing two areas of applications where we are currently working on, sensor networks, surveillance, and search and rescue robotics. In terms of ad-hoc networking, we are currently experimenting delivered with routing protocol designed to handle disruptions from both node mobility and poor link quality in mobile robotic systems. More information can be found in http://robotica.itam.mx/.

Acknowledgements

Supported by the French-Mexican LAFMI, UC-MEXUS CONACYT, CONACYT grant #42440, and “Asociación Mexicana de Cultura” in Mexico.

References

[1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In Proceedings of the IJCAI-95 Workshop on Entertainment and AI/ALife, 1995. http://www.robocup.org.

[2] Martínez-Gómez, L.A, Torres, M., Soto, M., Weitzenfeld, A., 2006, Eagle Knights 2006 - Small Size RoboCup Soccer Team, RoboCup 2006, Bremen, Germany, June 14-18.

[3] Verret, Ball, Kiat Ng. Laws of the F180 League - Release 3.00a. http://www.itee.uq.edu.au/~wyeth/F180%20Rules/index.htm.

[4] Martínez-Gómez, L.A., and Weitzenfeld, A., 2004, Real Time Vision System for a Small Size League Team, Proc. 1st IEEE-RAS Latin American Robotics Symposium, ITAM, Mexico City, October 28-29.

[5] http://platformx.sourceforge.net/

[6] C. Perkins and E. Belding-Royer, 2003, RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing.

[7] David B Johnson and David A Maltz, 1996, Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing Volume 353.

[8] Rescue Robotics, IEEE Robotics & Automation Magazine, 9 (3), September 2002.

[9] Orfinger, B., Robot Responders at WTC Site Fit Into Tight Spaces, Disaster Relief, Oct 2001, (http://www.disasterrelief.org/Disasters/011015robots/).

[10] Osuka, K., Murphy, R., Schultz, USAR Competitions for Physically Situated Robots, IEEE Robotics & Automation Magazine, 9 (3): 26 - 33, September 2002.

[11] RoboCupRescue, Urban Search and Rescue Robot Competitions, 2004 (http://www.isd.mel.nist.gov/projects/USAR/competitions.htm).

[12] Barrera, A., and Weitzenfeld A., 2006, Bio-inspired Model of Robot Adaptive Learning and Mapping, IROS – International Robots and Systems Conference, Beijing, China, Oct 9-13 (accepted for publication).