Creating wireless coverage maps for mobile robot applications

Joaquín López, Manuel Álvarez, Miguel Cacho, Enrique Paz and Diego Pérez

Abstract— **This paper describes a method to create wireless coverage maps for mobile robots. Most indoor mobile robot applications need some kind of wireless communication in order to receive commands and transmit information to users. However, robots that work in wide areas might not have coverage in all the working area. For example, surveillance robots that move in a building where wifi antennas do not cover all the area leaving some blind spots.**

Even in the case of autonomous robots that they do not need to be connected all the time because they can do tasks autonomously, the problem arise when one robot mission ends in one of these blind spots. If we do not provide mechanisms to avoid it, the robot will keep waiting for a new command that will never arrive since we can not connect to it. For these cases it comes handy a coverage map in order to avoid the blind spots if possible or to move to a highly coverage point in case the mission ends in a blind spot. In this paper we present different ways to create these wireless coverage maps according to the information that we have about the wireless system.

Index Terms— **mobile robot, WiFi, coverage map.**

I. INTRODUCTION

DURING the last years, the mobile robot industry has grown in investment, research and applications. Different degrees of autonomy are required in different applications but in most of them it is necessary to provide some kind of connection between the user and the robot. The information to be transferred from the user to the robot depends on the application and degree of autonomy. It can go from basic motion commands in robots with remote control to more general and abstract tasks in autonomous robots. On the other direction, most applications also need to transfer information from the robot to the user such as camera images and other on-board sensor data, reports while executing different tasks, etc. Therefore a communication system between user and robot is necessary in most of the applications. Furthermore, a wireless system is required in most of the case while the robots move on their working areas.

Wireless Local Area Networks (WLANs) provide network services where it is difficult or too expensive to deploy a fixed infrastructure. WLANs can coexist with fixed infrastructure to provide mobility and flexibility to users.

The primary WLAN standards are IEEE 802.11 [1]. The standard defines both physical and MAC layer protocols [2] and specifies the Received Signal Strength Indicator (RSSI) that is the measure of the RF energy received by the radio. The 802.11 protocol set, popularly known as Wi-Fi, includes wireless network standards that allow data transmission up to a theoretical 54 Mbps.

Wireless communication systems cover areas that depend on parameters of the system selected and the environment where it is installed. For example, the Wi-Fi coverage in a building is going to depend on the devices used but also on the building construction materials, wall distribution, columns, etc. There are two different ways to approach this problem:

- Make sure that we have full coverage on the area where the robots can move. This is, remove the coverage blind spots.
- Provide the robots with mechanisms to deal with the blind spots.

Most of the applications use the first approach providing coverage over the complete area where the robots are going to work. However, we address here the second problem for the following reasons:

- We do not want to modify the building facilities. Our goal is to bring the robots and use the communication system already installed.
- In some cases developing a system to provide full coverage can be expensive.
- It is very difficult to obtain an indoor RF propagation model wave travels to make sure there are not dead spots where no signal can be received.
- Even in the case we have full communication coverage, the malfunctioning of an access point might jeopardize the mobile robot application.

Several researchers have already addressed the task to create Wireless coverage maps for outdoors [3][4] and also indoors [5][6]. However, since their purpose was not a mobile robot application, they focus their research in using

Joaquín López, Manuel Álvarez, Miguel Cacho, Enrique Paz and Diego Pérez are with the Dep. Ingeniería de Sistemas y automática, University of Vigo, 36200 Vigo, Spain (phone: +34 986 812222; fax: +34 986814014; e-mail: joaquin@uvigo.es, su_manu2003@yahoo.es, (mcacho, epaz, dplosada)@uvigo.es

radiofrequency propagation models to estimate the RSSI. In most of the mobile robot applications, it is quite easy to measure the RSSI or some related value in different points of the environment while navigating the robots.

Most of the research in the robotics field regarding indoor WiFi is devoted to positioning systems [7]. There are different ways to focus the positioning problem utilizing many different properties. Some use physical properties of the signal, while others use the time taken for the signal to reach the destination node. For example, Angel Of Arrival (AOA) methods use the direction of incoming signals while Time of Arrival and Difference of Arrival use time or difference of arrival times.

Results obtained from those methods have little to do with our problem. However, there are a set of power based wireless position methods that can provide ideas to solve our problem. We can divide these solutions in two different approaches. One common approach employs surveying of signal strength information in a particular area. This information is stored in a database that is later used to determine the location of a mobile device by a particular pattern matching algorithm [1].

The other approach uses different propagation models to estimate the relationship between the signal strength and distance from transmitters [7][8]. Even that our problem is different, we can use the solutions provided by these methods to estimate the signal strength in the different points of the environment to construct a coverage map.

The rest of this paper is organized as follows. Next section introduces the architecture used to control one mobile robot we are using for applications such as surveillance. Robots will connect via WiFi to a central server to share information with the user interfaces. The method that we use to create the coverage map is described in section III. Finally, section VI includes some comments about the results and concludes the paper.

II. CONTROL ARCHITECTURE

The kind of applications we work with involve several autonomous robots that can be monitored from the web. Figure 1 shows one of these scenarios where the robots are connected to the central server using some kind of wireless communication. On the other side, users will also connect to the central server using an application GUI to command different tasks to the robots and to monitor them. They might do it from different kind of terminals such as a computer or a mobile phone.

The kind of information to be transferred from the robots to the central server and from this station to the users will depend on the application. For example, in a security system we will include images from cameras and information about security sensors on-board of the robots. It can also be useful to report the robot position and planned tasks.

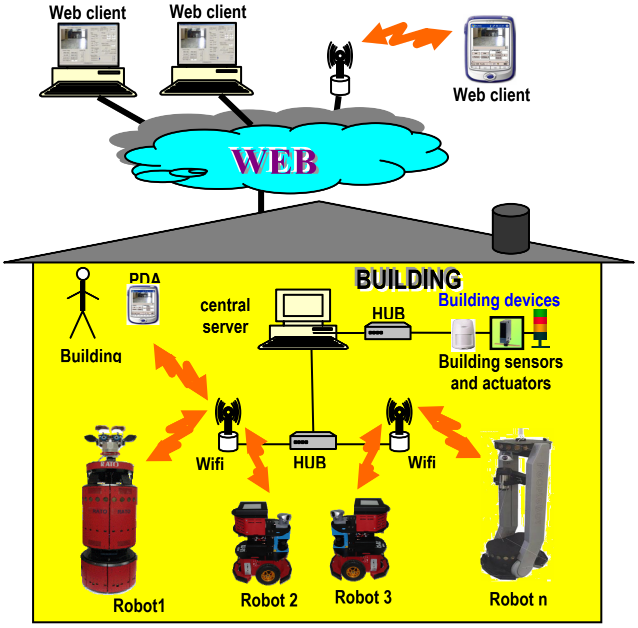

Fig. 1. Multirobot application scenario where different robots are connected to the central server .

The robots are able to execute autonomously different tasks and they do not need to be always connected to the central server during their execution. However, a problem will arise if one robot finishes a task in a point were it has no wireless coverage and a mechanism to move the robot to a point where it can connect the server is necessary.

In this section we describe first the control architecture on-board of the robots to carry out autonomously the commanded task describing in more detail the modules that deal with the wireless communications. In next section we will describe the general architecture that connects users, robots and other devices on the building.

The on-board control architecture is organized as shown in figure 2. Even though the different modules are organized in four sets, they can be mapped in the three layer architecture popularized by Bonasso de al. [9]. The hardware servers and control set implement the functional layer while RoboGraph Dispatch implements the executive and planning layer. Finally, the architecture includes a set of processes to interact with the users and connect to the central server using a wireless communication system.

The navigation platform is based on CARMEN [10] and some modules such as localize, navigator and base hardware servers remain basically the same. Unlike CARMEN, motion control is divided into high-level (strategic) planning [11] and lower-level (tactical) collision avoidance using the Beam method [12]. CARMEN integrates obstacles in the map and plans a new trajectory in order to avoid obstacles. Integrating all but the lowest-level motor control into a single module can produce optimal plans. However, due to the lack of precision in the localization system, the obstacle integration process can narrow some openings in the map. When the opening is only a little bit wider than the robot diameter, this difference can lead the path planning to discard a possible path through that opening. We have observed this behavior in several points of our office environment using CARMEN since the robot has to go through very narrow doors and corridors.

A. Hardware server modules

The hardware server modules govern hardware interaction providing an abstract set of actuator and sensor interfaces and isolating the control methods from the hardware details. Most of the hardware devices are connected to a CAN bus using RoboCAN [13]. Some of these devices are used in navigation such as the laser and sonar while others are specific for the application such as the robot head, sound and speech system, etc. The hardware servers also provide low-level control loops for rotation and translation velocities. Thanks to this layer changes in hardware components can be made without changes on higher layers modules while keeping the same interface. Special attention must be paid to the RF_Server module that is in charge of obtaining information from the wireless communication device about the signal strength from the different APs. During the application installation it might be used to create a log file while the robot moves through the different areas of the building. This log file can be used to create a coverage map as we will describe in section III. Even though it is not included in figure 2, there is a module named RF_Simulator that simulates the RF_Server behavior when we use the robot simulator. In simulation mode, RF_Simulator uses the log-normal shadowing propagation model to calculate the expected signal strength values.

B. Control modules

The control modules integrates sensor and motion information to provide improved sensor odometry, provide basic navigation capabilities (localization, path planning, follow path, etc) and basic application specific functions (say text, make expression, etc). These modules should be programmed by experts (platform developers) since they have to deal with different implementation details.

C. Executive modules

All the modules in this layer belong to the RoboGraph application that includes two modules: The Gui that is used only for application development and the dispatch without graphical interface that should be working when the robot is doing application operations.

RoboGraph Gui is the programming IDE that should be used by the application programmers. They must build the tasks that coordinate the work of the rest of the modules including some actions carried out by the users on the different interfaces. The tasks are programmed using a simple graphical tool and stored in an xml file.

Robograph Dispatch loads from an xml file and executes tasks requested mainly by user interfaces.

D. Interface modules

There are several interface modules for the programmer to debug and trace the control and hardware server modules. However, there is only one interface module on board that allows the user to interact with the robot. There are two modules (Robot_Web and RF_GUI) on this level directly related to this research that we outline in figure 2. Robot_Web is the module that exchanges information between the robot and the building central server station. RF_GUI is the module used for this research that creates and shows coverage maps, creates a new APs configuration and stores it in a xml file and shows information published by RF_Server or the RF_Simulator in real time

Each module in figure 2 is a Linux process that exchanges information with other modules using IPC Inter Process Communication [14]. Developed at Carnegie Mellon's Robotics Institute, IPC provides a publication-subscription model for processes to pass messages to each other via a central server process. Each application registers with the central server, and specifies what types of messages it publishes and what types it listens for. Any message that is passed to the central server is immediately copied to all other processes subscribed.

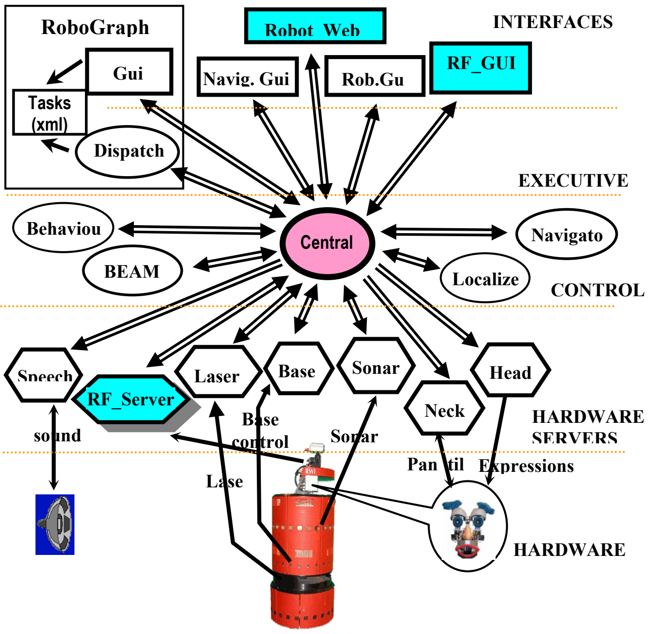

Fig. 2. ISANAV Control architecture. Different modules are divided in several sets. The hardware servers set reads sensor data and controls actuators. The control set provides several basic functions that can be used by other modules of this set and the executive set.

Our current research addresses architectures that extend the architecture described in figure 1 to control several mobile robots and connects them to a central server(figure

1). Figure 3 shows the proposed architecture for two robots connected to Ethernet but it is extensible to multiple robot systems. Regarding the communication topology, the framework shown in figure 3 is very similar to the one using inside each robot. The main reasons why we did not use IPC are:

- Native library: The Web User Interface is a Java applet in a web page that we want to start from any web browser. IPC provides a nice Java package to communicate any Java coded module with central. However, this package includes a C native library that needs to be installed in any computer where we want to run the module (in this case the Web User Interface applet).
- Address selection: Using the publish/subscribe (we want to avoid queries) model does not seem possible to publish a message direct to only a module. Modules subscribe to a kind of message and they will receive all the messages of that kind that any module will publish. This is a nice feature when thinking of frameworks like in figure 2. However due to the symmetry of the framework in figure 3, a problem arises when dispatch wants to send a command (such a go to point A) to only a robot since all the robots will get the message. Even though it can be easily done adding a new field to the messages, we don´t want to touch any module in figure 1.
- Public-private connection: As in RFC 1918, it is not possible to establish a connection with a private address from a public network. For this problem, JIPC clients use a pulling mechanism to get information from the server instead of the event mechanism used by IPC.

- Access control: According to figure 5 any user can try to monitor and send commands to any robot however in a surveillance application this can not be allowed. When the Web user interface starts, the user needs to be identified. Afterwards, all the commands he sends are checked by Jcentral referee for privileges. In a similar way, subscription to different robot resources (view camera, robot position, etc) must be approved to Jcentral referee.
- Referee: According to figure 5 several users can try to monitor and send commands to the same robot. Jcentral includes a referee that defines wich user can monitor some or all the robot resources and command some tasks.

The building might also have a set of devices such as sensors and actuators that in some way must be connected to the central server.

The building sensor and actuator system should be connected to the home server as in figure 1. A new module named building control unit (figure 2) manage data from and to the acquisition card. Besides security sensors and alarms, any sensor or actuator present in the building such as open/close doors and windows can be connected in a similar way to any domotic system.

There are different ways to connect all the sensor and actuators to the central unit. The first option is to use an acquisition card even though field buses or any sensor network can also be used. In order to avoid wires, a wireless sensor network can also be configured such as in [15].

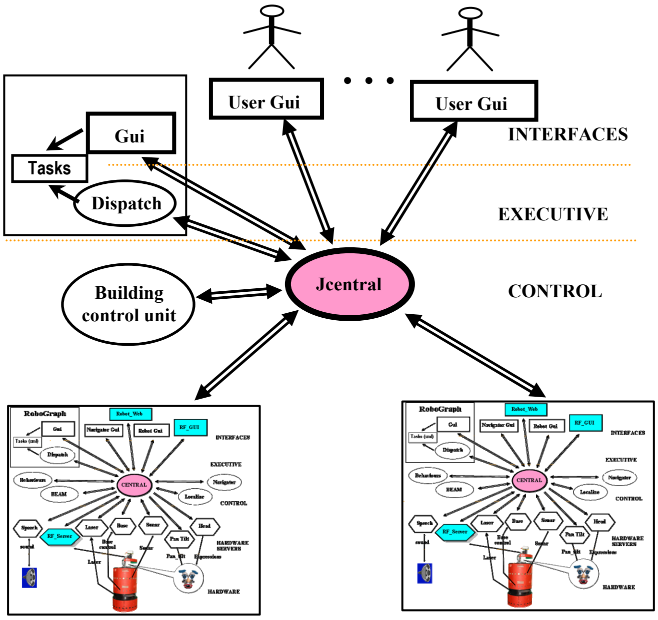

Fig. 3. Multirobot control framework. The framework used to control a robot using IPC is extended to control and connect the robots, the building control unit that manages all building sensors and actuators, RoboGraph that is in charge of executing the tasks (Petri nets) and the user interface that is an applet running on the users web browser.

III. LOG-NORMAL SHADOWING PROPAGATION MODEL

In some projects we can easily obtain information about the APs (Access Points) such as its position, attenuation, etc. For these cases we might want to obtain the expected RSII measurements in the different points of the environment. However these power signals are complex radio waves and there are many properties underlying electromagnetic wave propagation attributed to reflection, diffraction and scattering that should be considered.

The free space propagation model is used to predict received signal strength when the transmitter and the receiver have an unobstructed line-of-sight (LOS) path between them. For this case, the received signal power is well known and decreases with the square of the distance. However in many practical indoor situations there is no LOS path between transmitter and receiver.

In real world situations, the channel changes according to the movement of the communicating entities and many objects, still and moving ones, produce reflections, diffraction and scattering. Attenuation of the signal is not only a factor of distance, but also of the obstacles between the transmitter and receiver. Even people moving in a room will affect signal strength. Multipath propagation may result in dead spots where no signal can be received. Therefore, it is difficult to predict the propagation of a RF wave in an indoor environment [16]. Thus RSSI measurements are sensitive to multi-path, fading, non line of sight measurements, noise, refraction and diffraction.

In the literature, there are numerous experimental and theoretical studies of indoor propagation models [17][18][19][16] that tend to focus on a particular characteristic like temporal fading or inter-floor losses. However, we are more interested in empirical models that help in reducing computational complexity as well as increasing the accuracy of the predictions [20]. We do not use models such as Two-Ray model [17] that rely on detail knowledge about the indoor environment because it will be difficult to obtain such information for most of the mobile robot applications.

The kind of models we are interested are such as COST 231, Ericsson model 9999, etc that can be seen in [20]. For our case, we have decided to use the Log-normal Path Loss Model that defines the path loss for a receiver separated a distance d from the transmitter as:

$$PL(d)_{dB} = PL(d_0)_{dB} + 10n\log\left(\frac{d}{d_0}\right) + X_\sigma(dB) \qquad [1]$$

Where PL(d) is the path loss, PL(d_0) is the path loss at the reference distance d_0, $X_\sigma = 10\log(X)$ is zero mean Gaussian[1] distributed random variable (in dB) with standard deviation σ (also in dB) and the value of n depends on the specific propagation environment. Finally, n is the path loss exponent which indicates the rate at which path loss increases with distance d. This value of n depends on the specific propagation environment, type of construction material, architecture and location within the building. For example, for buildings in areas where the transmitter is in the line-of sight of the receiver, n ranges from 1.6 to 1.8 while in the obstructed areas ranges from 4 to 6. However, in obstructed areas in factories the path loss exponent ranges from 2 to 3.

However, we can obtain PL(d_0), n and the standard deviation σ from data collected by the robot. In this case, First we should measure the path loss (PL) at the reference distance (d_0). Next we "walk" the robot through our environment while the RFServer (figure 2) logs RSII samples in order to obtain the $PL_{Measured}(d_i)$ for different distances (d_i) and different kind of areas (line-of-sight and obstructed). Then we obtain n as the value that minimizes the function:

$$F(n) = \sum_{i=1}^{k}\left[PL_M(d_i) - PL_{LN}(d_i)\right]^2 \qquad [2]$$

Where $PL_M(d_i)$ is the Path Loss measured by the receiver on the robot, $PL_{LNl}(d_i)$ is the Path Loss obtained according to eq.1 and k is the number of samples that we have. Using eq.1 in eq.2 we have:

$$F(n) = \sum_{i=1}^{k}\left[PL_M(d_i) - PL(d_0)_{dB} + 10n\log\left(\frac{d}{d_0}\right) + X_\sigma\right]^2 \quad [3]$$

F(n) has a minimum[2] for n:

$$n = \frac{\sum_{i=1}^{k}\left[\log\left(\frac{d_i}{d_0}\right)\left(PL_M(d_i) - PL(d_0)\right)\right]}{10 \times \sum_{i=1}^{k}\left[\log\left(\frac{d_i}{d_0}\right)\right]^2} \qquad [4]$$

We can see in figure 4 a portion of the samples taken by the robot while travelling in a corridor. White areas are places where the robots can move. In the area depicted in the figure we can see three APs. In order to get a better idea and a simple map, we will consider the signal coming from the AP labelled as "AP1" in figure 4. The red line on the big corridor show the trajectory followed by the robot and the samples taken by the robot. Darker points represent better coverage than the lighter points.

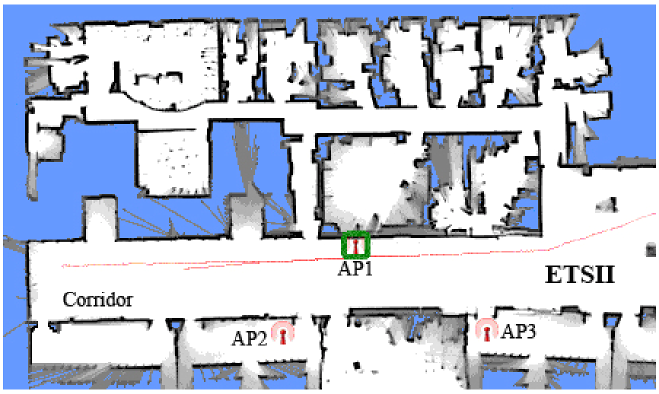

Fig. 4. Occupation grid map of our department and path traveled by the robot while taking signal strength samples.

Figure 5 shows the signal level from the robot while travelling the path described in figure 4 and the logarithmic approach as in equation 1 without the random variable with n obtained using a similar dataset from the same robot. We

---

[1] X itself is not Gaussian, its log is Gaussian, so it is called log-normal. This is, normal in the log scale.

[2] The minimum is obtained from d(F(n))/dn =0

can see that the logarithmic approach is good in average even though the data set has a quite high standard deviation. This deviation is the reason to add the random variable in model defined by equation 1.

After obtaining *n*, the deviation σ can be calculated using equation 5:

$$\sigma^2 = \frac{1}{k}\sum_{i=1}^{k}\left[PL_M(d_i) - PL_{LN}(d_i)\right]^2 \qquad [5]$$

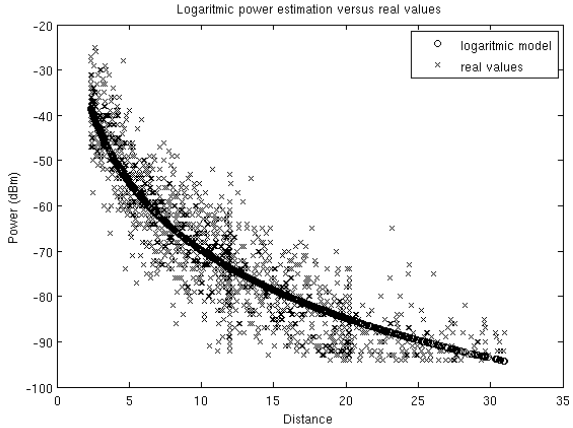

Fig. 5. Samples obtained with the robot for different distances versus the values obtained with the logarithmic model (Eq. 1 without the random variable).

Finally, we have all the Log-normal Path Loss Model parameters (eq.1) and we can use eq.1 to obtain the Path Loss in the different cells of the environment. We can see one example of the coverage map created using this method in figure 6.

Fig. 6. Coverage map created using log-normal shadowing propagation model.

Two different values of n have been used to create the map on figure 6. One is for the AP line-of-sight areas and a different one for the rest of the areas.

IV. RESULTS AND CONCLUSIONS

Different tests have been carried out in the "Escuela Técnica Superior de Ingenieros Industriales" (ETSII). We have built a map for every AP so that if we do not get coverage in an area that should be covered by an AP, the robot moves to an area with good signal from a different AP. When the robot navigates in the FOV (field of view) of some AP, we obtain similar coverage values as in the maps obtained with the method described in the paper. However, in some areas not in the FOV of any AP we can appreciate important differences. We believe it is mainly due to the construction materials such as big concrete and iron columns.

It is necessary to point out that we have found some spots even in the AP's FOV with no coverage such as in front of some vending machines located in one of the corridors.

As a future work, we are working on a different approach to obtain the coverage maps. This new approach uses data gathered by the robots while moving through the environment. In order to obtain the coverage on the different cells of the map, we extend the real values on the closest cells obtained by the robot. Using this new approach we expect to identify some patterns such as blind spots in front of the vending machines, etc

We have presented a method to create wifi coverage maps that can be used in applications based in mobile robots that need to be connected using this wireless communication system. These maps are used as a reference to maintain the robot connected especially when a new task needs to be commanded.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish Comisión Interministerial de Ciencia y Tecnología, CICYT, project DPI2008-06738-C02-02 and Xunta de Galicia (PGIDIT06PXIC 303194PN).

We would like to thank all the people that have influenced this work.

REFERENCES

[1] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," IEEE Infocom, 2004.

[2] ISO/IEC 8802-11, Wireless LAN Medium Access Control (MAC) andPhysical Layer (PHY) specifications, 1999.

[3] T. K. Sarkar, et al., "A survey of various propagation models for mobile communication," IEEE Antennas and Propagation Magazine, Volume 45, Issue 3, June 2003, pp. 51-82.

[4] M. F. Iskander and Z. Yun, "Propagation Prediction Models for Wireless Communication Systems," (Invited paper)IEEE Trans. on Microwave Theory and Techniques, Vol. 50, pp. 662-673 (March 2002)

[5] A. Hills, et al., "Estimating Signal Strengths in the Design of an Indoor Wireless Network, " IEEE Trans. Wireless Comm., vol. 3, pp. 17-19 (January 2004).

[6] C. Cheon, G. Liang, and H. L. Bertoni, "Simulating Radio Channel Statistics for Different Building Environments," IEEE J. on Selected Areas in Communications, vol. 19, pp. 2191-2200 (November 2001).

[7] Bose, A. and Chuan Heng Foh, practical path loss model for indoor WiFi positioning enhancement, Proceeding of the 6th International Conference on Information, Communications & Signal Processing, December 2007. pp: 1-5

[8] A. Kotanen, M. Hännikäinen, H. Leppäkoski, T. Hämäläinen, “Positioning with IEEE 802.11b wireless LAN”, IEEE Personal, Indoor and Mobile Radio Communications, vol.3, pages 2218- 2222, 7-10 Sept. 2003.

[9] R. Bonasso, D. Kortenkamp, D. Miller and M. Slack, “Experiences with an architecture for intelligent, reactive agents”. Journal of Artificial Intelligence Research Vol 9(1), pp: 237-256, 1997

[10] M. Montemerlo , N. Roy, S. Thrun, “Perspectives on Standardization in Mobile Robot Programming : The Carnegie Mellon Navigation (CARMEN) Toolkit”, Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2003). 27-31 Oct. 2003. Vol. 3, pp: 2436- 2441

[11] A.R. Diéguez, J.L. Fernández, R. Sanz, “A global motion planner that learns from experience for autonomous mobile robots”, Robotics & CIM Vol. 23 pp: 544–552 (2007), Ed. Elsevier.

[12] J.L. Fernández, R. Sanz, J.A. Benayas and A.R. Diéguez, “Improving collision avoidance for mobile robots in partially known environments: the beam curvature method.” Robotics and Autonomous Systems. vol. 46, pp. 205-219, April 2004.

[13] Fernández, J.L., Souto, M.J., Losada, D.P., Sanz, R., Paz, E., “Communication framework or sensor-actuator data in mobile robots”, Proceedings of ISIE 2007. IEEE International Symposium on Industrial Electronics, 2007. pp: 1505-1507

[14] R. Simmons, “The interprocess communications system (IPC). http://www.cs.cmu.edu/afs/cs/project/TCA/www/ipc/ipc.html. Accessed: 2007-08-29.

[15] Y. Kim, H. Kim, S. Yoon, S. Lee, K. Lee, “Home security robot based on Sensor Network”, SICE-ICASE International Joint Conference 2006, pp: 5977-5982.

[16] H.J. Zepernick and T.A. Wysocki. Multipath channel parameters for the indoor radio at 2.4 GHz ISM band. IEEE Veh. Technol. Conf., 1:190–193, May 1999.

[17] A.F. Agelet, A. Formella, J.M.H. Rabanos, I.F. de Vicente, and P.F. Fontan. Efficient ray-tracing acceleration techniques for radio propagation modeling. IEEE Veh. Technol. Conf., 49:2089–2104, November 2000.

[18] M. Iskander, Z. Yun, and Z. Zhang. Outdoor/indoor propagation modeling for wireless communications systems. IEEE Antennas and Propagation Society International Symposium, 2:150–153, July 2001.

[19] J. Tarng and T. Liu. Effective models in evaluating radio coverage on single floors of multifloor building. IEEE Veh. Technol. Conf., 48:782–789, 1999.

[20] A. Neskovic, N. Neskovic, and G. Paunovic. Modern approaches in modeling of mobile radio systems propagation environment. IEEE Communications Surveys, http://www.comsoc.org/pubs/surveys, 2000.