ABCM Symposium Series in Mechatronics - Vol. 1 - pp.18-24

A NAVIGATION AND PATH PLANNING SYSTEM FOR THE NOMAD XR4000 MOBILE ROBOT WITH REMOTE WEB MONITORING

Alberto J. Álvares
Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle (GRACO), CEP 70910-900, Brasília, DF
alvares@AlvaresTech.com

Gabriel F. Andriolli
Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica, GRIMA-GRUCON, Caixa Postal 476, CEP 88040-900, Florianópolis, SC
gabriel@das.ufsc.br

Paulo R. C. Dutra
Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle (GRACO), CEP 70910-900, Brasília, DF
prcdutra@hotmail.com

Márcio M. de Sousa
Universidade de Brasília, Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle (GRACO), CEP 70910-900, Brasília, DF
marcio_ms79@hotmail.com

João C. E. Ferreira
Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica, GRIMA-GRUCON, Caixa Postal 476, CEP 88040-900, Florianópolis, SC
jcf@grucon.ufsc.br

Abstract. *The development of teleoperated autonomous robots occurred due the existence of environments unsuitable for human action. These robots usually need a representation of the environment (a map) and the capacity to interpret that representation to be able to plan a path towards some target location and to move safely in an environment where there may be variations in the position of the objects. This work describes a navigation system with remote monitoring through the Internet for the Nomad XR4000 mobile robot. The system builds bidimensional maps of the environment that surrounds the robot, through a combination of sonar and infra-red sensors, and also the estimated position of the robot. The navigation is carried out through stored environment maps. The user, through a graphical interface in a Web browser, can define a target to be reached by the robot. A path planner based on artificial intelligence defines a trajectory and controls the robot motion without collisions.*

Keywords*. Navigation system, mobile robot, teleoperation*

1. Introduction

Nowadays, mobile robots are utilized in many applications such as performing tasks that are hazardous for humans or even simply guiding people in museums (Thrun et al. 1999). The great variety of tasks that can be accomplished by autonomous mobile robots has stimulated a growing interest in this field of knowledge in recent years.

For an autonomous mobile robot, the control problem is translated into a navigation concept. According to Nehmzow (2000), navigation can be defined by three fundamental competences:

- Self-localization;
- Path planning;
- Map-building and map-interpretation.

Self-localization denotes the robot’s capacity to establish its own position in relation to a reference system. Path planning requires the determination of the robot’s current position and the position of a target location in relation to the same reference system. Those two competences usually require the use of a representation of the environment (a map) and the ability to interpret that representation.

In general, approaches to representing the environment are divided into topologic and geometric maps. Topologic maps consider the environment as a series of places with connections between them, these being naturally represented by a graph. One of the main advantages of this representation is the ease of path planing.

Geometric maps, such as occupancy grids (or evidence grids, Martin, 1996 and Moravec, 1985), represent the environment in discrete cells forming a grid. Here, each cell represents a square area of the environment and it stores the value that indicates the occupation state for this area. This is usually done by labeling the cells with “unknown”, “free” or “occupied” values or with a value that represents the probability of the cell being occupied or not.

To fill the grid through a probabilistic treatment may be computationally expensive, and becomes more expensive as the environment map increases. In this way, some applications in real time become nonviable (Dissanayake, 2001).

The facility to represent the environment and the possibility for integration of data from different kinds of sensors constitutes the main advantages of the geometric maps.

The fundamental problem of map-building and map-interpretation is knowing the robot's location in the environment. A simple method for mobile robot localization is called dead reckoning, in which the position and the orientation are estimated by summing each movement in relation to an initial point (path integration). Because of the odometer drift problems that are continually accumulate it is more common to find systems that also add sensors information, since this error cannot be eliminated without an external perception.

Another problem that has been getting attention in the field of robotics is teleoperation. The existence of environmentally unsuitable sites, where human action is dangerous or not viable, brought about the need for the development of remotely operated systems. In Byrd (1996), a mobile robot remotely operated is used to visualize and monitor dangerous places. Nehmzow et al. (1996) and Simmons (1998) use the Internet as a communication vehicle to remotely control mobile robots.

This work presents aspects of the methodology and the implementation of a navigation system for a mobile robot. The system developed allows teleoperation through the Internet.

In section 2 we describe the mobile robot used in this work. In section 3 we detail the navigation system implemented and some simulated results. The teleoperation architecture is described in section 4. In the section 5 the experimental results are presented. The conclusions are given in section 6.

2. The Nomad XR4000 mobile robot

The Nomad XR4000 mobile robot (fig 1) is an integrated system designed for industrial applications or research. Its features include (Nomadic Technologies, 1999):

- Onboard dead reckoning system (odometer);
- Infrared, sonar and tactile sensors;
- Two PC Pentium computers running the Linux operational system (programmed in C language) interconnected by TCP/IP protocol;
- Camera, pan-tilt unit and frame grabber for vision task;
- Wireless network connection.

The robot's motor system is holonomic, providing three degrees of freedom (X, Y and q): two of translation and one of rotation. The wheels have independent translation and rotation axes, summarizing eight motors for the four system wheels. Three DSP (Digital Signal Processing) units and a dedicated microcontroller control the eight axes and calculate the estimated position (dead reckoning) based on the robot's kinematic model and input data obtained from encoders located at each actuator (motors and wheels).

The sonar sensor system, Sensus 250, consists of two rings containing 24 sensors each. A sonar can provide information on objects at between 150 and 7000 mm distances. The Sensus 350 system consists of two rings of 24 infrared light sensors capable of providing information on close objects (typically between 200 and 500 mm distance).

The Nomad XR4000 also has tactile sensors that return information about physical contact with objects in the environment. In this work, these sensors were only used for emergencies, since robot collisions should be avoided.

The Nomad XR4000 is programmed using the XRDev architecture provided by the manufacturer. This multi-process architecture is composed mainly of three processes:

- Nrobot, the server process that communicates with the robot’s hardware;
- Ngui, a graphical interface for robot control;
- User process, which communicates with Nrobot in order to allow the performance of several tasks defined by the user.

The network connection is made through a Proxin RangeLan2 wireless Ethernet card adapter (http://www.proxim.com). This system communicates with a bridge server that connects the robot with the local network using the TCP/IP protocol.

Figure 1. The Nomad XR4000 mobile robot

3. Navigation System

Figure 2 shows the navigation system architecture based on static and dynamic maps developed for the Nomad XR4000 mobile robot. In the following subsections aspects of its architecture and implementation will be detailed.

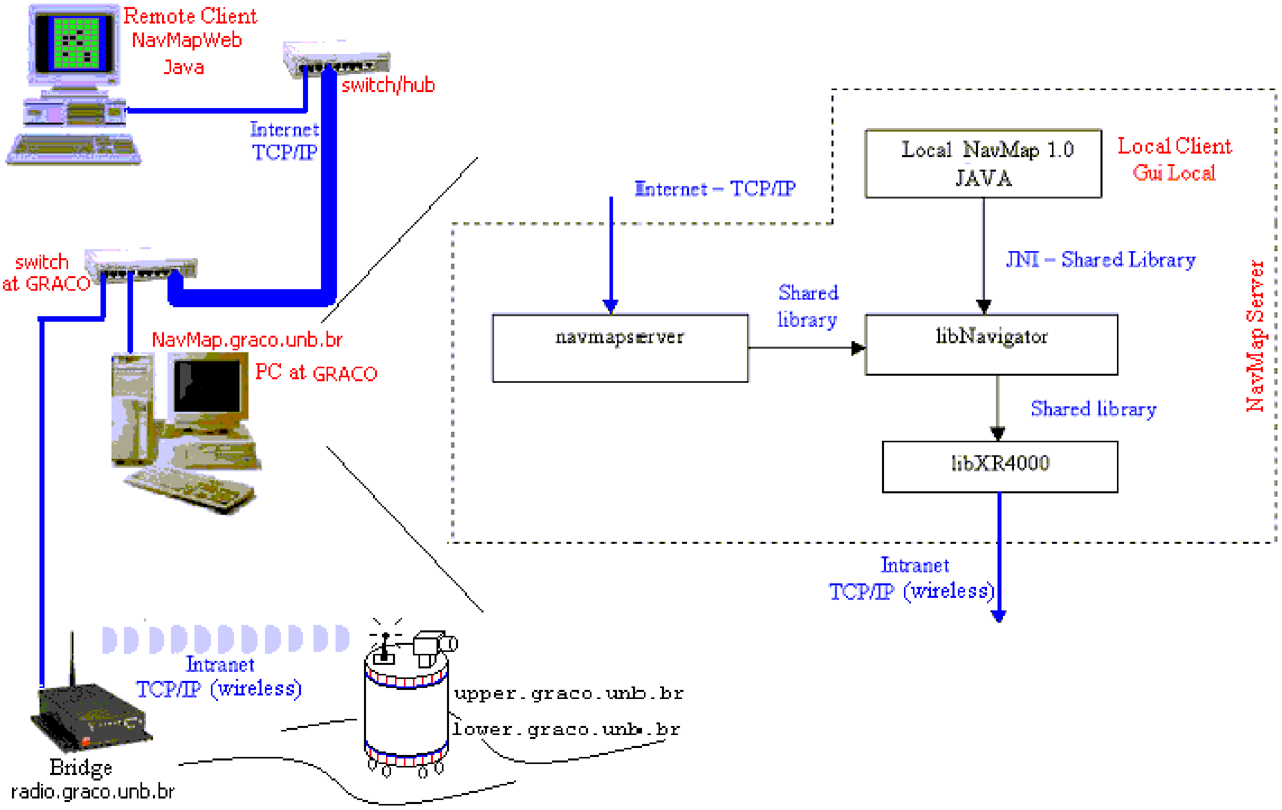


Figure 2. Navigation system architecture remotely operating through the Internet.

3.1. Mapping and Self-localization Module

In our system we use occupation grids to represent the environment that surrounds the robot. This approach was chosen mainly because it supplies a simplified way to integrate different sensorial perceptions.

In this approach, the environment is divided into cells, and each one represents a square area of the real world. In this work the state of occupation of each cell is indicated by an integer value. These values and their respective meanings are:

(-1) Non-explored area of space. Initially, the state of occupation is unknown in all cells. Then, all cells are initialized with -1;
(0) Free space;
(8) Tolerance: This information is used to avoid collision with obstacles due to difference between the robot's diameter and the cell dimension.
(9) Occupied space: This identifies the obstacles in the environment.

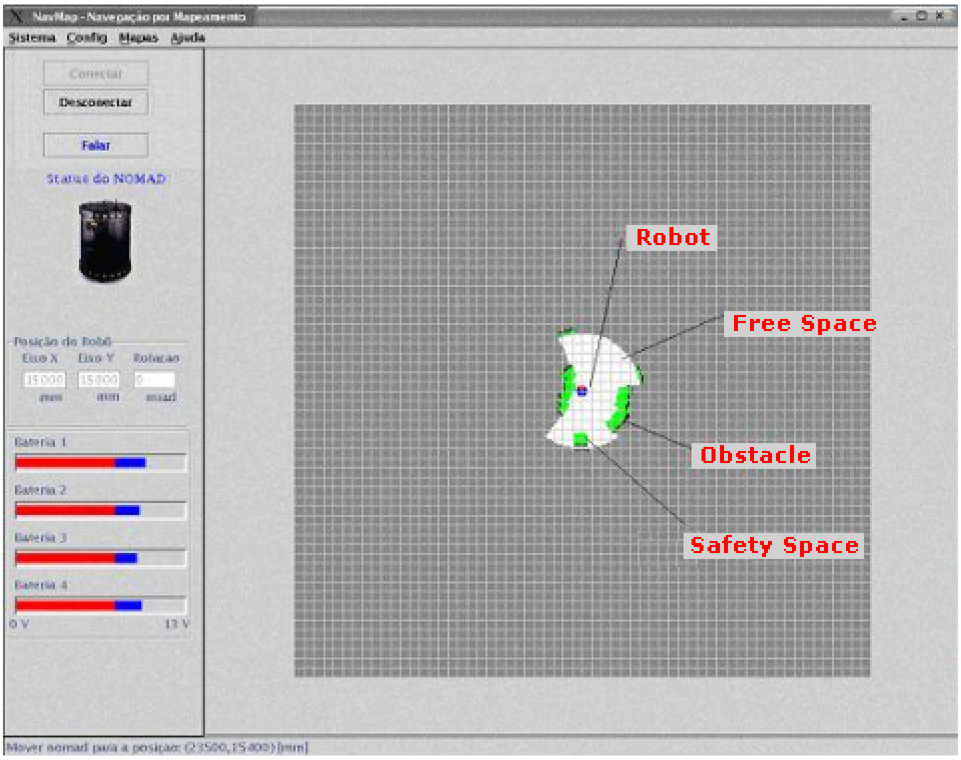
The cell size in the occupation grid must be chosen considering aspects such as computational cost, precision and real environment dimensions. In the developed system, the cell size was defined as 100 x 100 mm. The structure to store the maps is a one-dimensional vector with 90000 elements, equivalent to a three-hundred-order square matrix. Thus, the map can represent an area with maximum dimensions of 30 x 30 m or 900 m^2.

The cells are filled out based on the sensorial processing that integrates data from ultrasonic and infrared sensors with the estimated robot position in the environment. A safety margin is added around the obstacles to prevent collisions.

The position and orientation of the robot are obtained from the dead reckoning system. Dead reckoning, however, suffers from several sources of inaccuracy, like noise and wheels slippage, and the cumulative effects of these errors can result in a large error in the robot position estimate. Position errors can affect not only the building of the map but also its interpretation.

Figure 3 shows details of the graphical user interface (GUI), called NavMapWeb, developed in Java programming language (Sun Microsystems, 2003a). Through this GUI, the user can constantly visualize the environment map and the robot position, both being constantly updated. In the map, free spaces are represented by white, obstacles by black, safety areas are colored green and unknown spaces are in gray. The robot is represented by a blue circle in its respective position. This interface also allows a map to be saved or loaded, the grid resolution to be adjusted and the state of the batteries to be visualized.

It can be observed in figure 2 that the NavMapWeb interface is remotely executed and communication with other parts of the system is carried out through the Internet. This interface can also execute locally and, in this case, it is referenced by Local NavMap.



Eixo X: 15000 mm
Eixo Y: 15000 mm
Rotacao: 0 mrad

Mover nomad para a posição: (23500,15400) [mm]

Figure 3. GUI (NavMapWeb) running the application.

3.2. Path Planning Module

The path planner's problem is to find a free path leading from the robot's current position to a target position. Besides the current and target position, to plan a path for a mobile robot a map of the environment is necessary.

Since in most cases there are several possible ways to lead a robot from an initial point to a target point, some desirable characteristics for the route can be defined. For example, some routes may be shorter while others may be more secure. These characteristics directly affect the choice of the path planning method.

Most of the approaches to path planning visualize the search space in a graph of possible roads, for example, the approaches by Voronoi diagram and visibility graph (Latombe, 1991). This approximation can naturally lead to the learning of the environment, where the robot can take several paths to map the environment.

To select the best path from a graph of several possible paths we utilize, also, many very well known artificial intelligence techniques (AI), such as the branch and bound search variations (Winston, 1984).

In the developed system the A* search algorithm was used (Winston, 1984), which is a refinement of the branch and bound search. In this algorithm, the search space is reduced deleting multiple paths to a subnode, and leaving only the lowest cost path. The choice of the lowest cost path for expansion at each stage is improved by adding an estimated cost for the remaining path to the actual cost of the path so far. If the estimated cost is always lower than the actual cost, the search A* produces an optimal solution. A typical lower bound used in path planning is the straight-line distance between the position represented by the current node and the target position.

Figure 4 displays a simulated result using an imaginary map. Free cells are white. The black cells represent the obstacles. Letters *R* and *M* indicate, respectively, the robot's initial and target position. The path was found by the A* search and it is represented by the gray cells. In this simulation, a safety margin to avoid collisions was incorporated.

In the implemented system, the path planning module is activated every time a movement is requested. The user can specify the target position, i.e. the target coordinates, simply by clicking the mouse on the GUI map (figure 3). When this event occurs, the path planning module immediately verifies the existence of a path without collisions between the origin and destination point and, when it exists, the robot is commanded by the navigation system to the target position. The control of the whole movement is automatic.

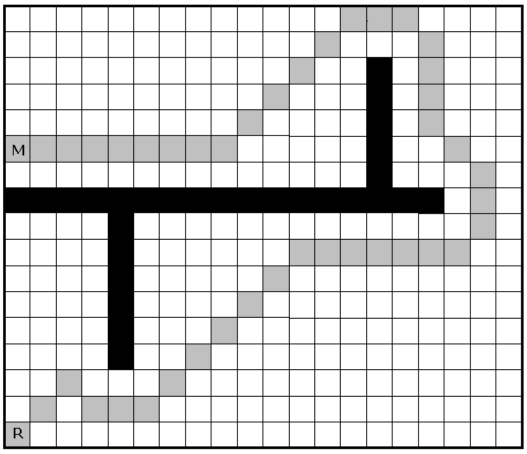


Figure 4. Simulated A* search result.

While the navigation system drives the robot to the target position, it executes sensorial routines to detect obstacles not foreseen in the map. When an obstacle is detected too close to the robot (distances equal or inferior to 300 mm), a new path is calculated to avoid such obstacle, and thereby any potential collisions.

4. Internet teleoperation

The implemented system can be teleoperated through the Internet based on the client-server architecture defined by Álvares (1998, 1999a, 1999b), which is composed of the following parts:

- The server, represented by programs that run in the robot;
- The client, represented by the Java applets located in the user browser.

A similar architecture was used in another work in the laboratory, in which Tourino (2002) directs the Nomad XR4000 mobile robot through the WEB.

In the implemented system, the developed server, called NavMap, runs on a Pentium II, 600Mhz microcomputer, with a Linux Suse 8.1 operational system, located in the laboratory. The NavMap server communicates with the robot's server, Nrobot, through the Ethernet local network using the TCP/IP protocol, as shown in figure 2.

The NavMap server was implemented using two libraries developed in the C++ programming language: libXR4000 and libNavigator. The libXR4000 library makes a map of the functions, variables and structures available at the manufacturer's library (Nclient_host) for a simple object orientated model, to be used within other applications. The libNavigator contains the implementation of the mapping and path planning modules. The communication between the Java interface and the C++ written system uses the JNI (Java Native Interface, Sun Mycrosystems, 2003b) technology, that allows Java code to operate with applications and libraries written in other languages, such as C, C++, and Assembly.

The client executes the application, whose graphic interface is shown in figure 3, in the browser (Netscape, MS IExplorer, etc.) of the user or in a GUI for the Xfree Linux environment. It can be located anywhere in the world. The images of some laboratory’s cameras and of the robot's own camera are also available to the user. The application can be accessed at the address http://NavMap.graco.unb.br. The code source, binaries and documentation are also available in this URL.

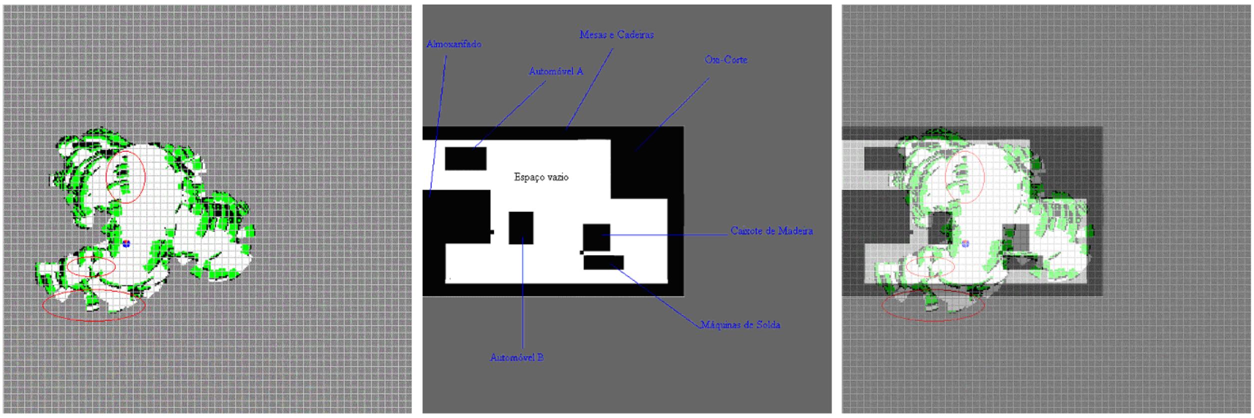
Communication between client and server is accomplished through sockets using the TCP protocol. It requires the IP address and the port of the server machine.

5. References

To observe and to test the system, the Automation and Control Group (GRACO) laboratory was mapped using the developed application. The robot was initially positioned in a fixed point whose coordinates (in relation to the global reference) were known. The system needs these coordinates to locate the robot in the environment. The robot was then guided through the environment as described in the last section. A map was simultaneously generated.

The result can be observed in figure 5 (a). Figures 5 (b) and (c) show, respectively, the Graco map obtained manually and the superimposition of the maps for comparison.

It was observed that some cells indicated nonexistent obstacles (circled in red in figure 5) and that these mistakes happen with more frequency in the cells most distant from the robot, indicating that a revaluation of the sensor model adopted and/or a better data processing and sensorial fusion is necessary.



(a) (b) (c)

Figure 5 – (a)Graco’s map obtained through the developed system, (b)Graco’s manual map and (c)Maps superimposed.

In Lora (1998) a Kalman filter is used to integrate the information from a digital compass with that from the dead reckoning system. Several other applications using Kalman filtering to integrate sensorial information are mentioned in Lora (1998).

In spite of the observed problems, the acquired map can be used as a first view of the environment. The motion control was carried out automatically by the developed system and there was no collision while the environment exploration was accomplished, showing the robot's capacity to avoid obstacles in an autonomous and efficient way.

It was observed that the dead reckoning error did not significantly affect the obtained results due to the low robot displacement speed (limited to 300mm/s) and to the relatively small dimensions of the explored environment. However, future improvements to the system should include some calibration module to correct the robot's coordinates based on the global reference system of the map.

In Yamauchi (1996), for example, the system returns the robot to a fixed calibration position every 10 minutes to calibrate the odometer. In Thrun (1999) the system corrects the robot’s position with an error smaller then 100 mm.

6. Conclusion

In this work the implementation of a navigation system for the mobile robot Nomad XR4000 is presented. For the environment representation the occupancy grid approach was used due to the possibility of easily integrating different sensorial perceptions to compose the space representation. The robot location was determined by the dead reckoning system.

The developed system has a graphical interface where the user can visualize the map being built simultaneously with the robot’s motion. It allows, through high level user commands, the sending of target coordinates to the navigation system. The interface also gives the state of the batteries and the images picked up by the camera carried by the robot.

The path planning is carried out automatically by the system every time a movement is requested. This uses an AI technique to draw the route that the robot should proceed along. The module of path planning was able to guide the robot around the obstacles of the environment, thereby avoiding collisions.

The system also offers the client-server architecture characteristic that allows the robot's remote monitoring through the Internet.

The mapping related problems could be lessened through better data processing, such as by using Kalman filtering or artificial neural nets. To be able to accomplish more efficient navigation and to increase the robot's autonomy, a dead reckoning calibration system is also necessary.

7. References

Álvares, A. J. Romariz, L. S. 1998. “Desenvolvimento de um manipulador com dois graus de liberdade controlado remotamente via Internet”. V CEM-NNE98. Fortaleza.

Álvares, A. J. 1999a. "Telerobótica: Metodologia Para o Desenvolvimento De Sistemas Robóticos Teleoperados Via Internet", Águas de Lindóia, SP, XV COBEM.

Álvares, A. J. 1999b. "Telerobotics: A Methodology for the Development of Through-the-Internet Teleoperated Robotic Systems" Telemanipulator and Telepresence Technologies V – SPIE Boston, USA.

Bilgiç, T. e Turksen, I.B., “Model-based Localization for an Autonomous Mobile Robot Equipped with SonarSensors”, IEEE Conference on Systems, Man and Cybernetics, Oct. 22-25, Vancouver, B.C. Canada, 1995.

Byrd, J. S. “An Intelligent Inspection and Survey Robot”, University of South Carolina. Columbia. USA. 1996.

Dissanayake, M. W. M. G. et al. “A solution to the simultaneous localization and map building problem”, IEEE Transaction on Robotics and Automation, v.17, n.3, p.229-241, 2001.

Latombe, J.C., “Robot Motion Planning”. Kluwer Academic Publishers, 1991

Lora, F.A.S., Hemerly, E.M. e Lages,W.F., "Sistema para Navegação e Guiagem de Robôs Móveis Autônomos", Controle & Automação, Vol. 9, No. 3, pp. 107-118, 1998.

Martin, M. C. Moravec, H. “Robot Evidence Grids”, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 1996.

Moravec, H. P. Elfes, A. “High resolution maps from wide angle sonar”, IN: IEEE International Conference on Robotics & Automation (ICRA), 1985.

Nehmzow, U. Bühlmeier, A. Dürer, H. “Remote Control of Mobile Robot via Internet”, Manchester, 1996.

Nehmzow, U. , “Mobile Robotics: A Practical Introduction”, Springer, London, 2000.

Nomadic Technologies, “Nomad XR4000 Hardware Manual Release 1.0”, California, USA, 1999

Simmons, R. “Xavier: An Autonomous Mobile Robot on the Web”. Carnegie Mellon University. USA, 1998.

Sun Microsystems, 2003a, “The Java Tutorial”, Available on the Internet at: http://java.sun.com/docs/books/tutorial/index.html.

Sun Microsystems, 2003b, “Java Native Interface: Programmer’s Guide and Specification”.

Thrun, S., et alli, “MINERVA: A second generation mobile tour-guide robot”. In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1999.

Tourino, S. R. G. Álvares, A. J. “Teleoperation Architeture for the Navigation through WEB of the Nomad XR4000 Mobile Robot”, João Pessoa, PB, II CONEM, 2002.

Winston, P. H. “Artificial Intelligence”, Addison-Wesley, 1984.

Yamauchi, B. “Mobile Robot Localization in Dynamic Environments Using Dead Reckoning and Evidence Grids”. Journal of Robotics Systems, Special Issue on Mobile Robots, Vol. 14, No. 2, pp. 107-120, 1996.