Control and Obstacle Avoidance of a Mobile Platform Used as Robotic Assistant for Elderly and Disabled

Alina Voda, Adrian Filipescu, Ion Susnea, Eugénia Minca

Control and Obstacle Avoidance of a Mobile Platform Used as Robotic Assistant for Elderly and Disabled

A. Voda[1], A. Filipescu[2], I. Susnea[2], E. Minca[3]

1. Department of Control Systems, GIPSA-lab, Grenoble University Joseph Fourier, 38000 Grenoble, France
alina.voda@gipsa-lab.grenoble-inp.fr

2. Department of Automation and Industrial Informatics, University "Dunarea de Jos" of Galati, 800008, Romania
Adrian.Filipescu@ugal.ro, Isusnea@yahoo.com,

3. Department of Automation,Computer Science and Electrical Engineering, University "Valahia" of Targoviste, Romania, Eugenia.Minca@gmail.com

Abstract: In this paper a fuzzy control and an obstacle avoidance system, together with a distributed system of embedded microcontrollers, are presented. In the real-time control, a wheeled mobile robot (WMR), Pioneer 3-DX, from Mobile Robots, has been used. The solution adopted can be easily ported for the implementation of an intelligent wheelchair, capable either to carry an elderly or disabled person, or to move independently in a smart environment, as a sensorial extension of the assisted individual. A number of control modules are located on the mobile robot, while others are deployed in an intelligent environment. This solution can significantly reduce the cost of developing a robotic assistant for the elderly and disabled. The structure of the real-time setup is described in detail, as well as the main algorithms used for each individual task: path following, obstacle avoidance, data acquisition. Also, an obstacle avoidance system, based on ricochet method, named "the bubble rebound method", is presented. This method is tested only by simulation,

1 WHY ROBOTIC ASSISTANT?

Since 1985 when Borenstein, J. and Koren, Y in [1] proposed the concept of "nursing robot", seen as a mobile platform, equipped with a robotic arm, a considerable number of researches have added new facets to the idea of creating a robotic aid for the elderly or disabled. References [2] and [3] described the concept of "personal robotic assistant", while other researchers, [4], focused on the development of intelligent wheelchairs, provided with a certain level of autonomy in motion. Advances in telecommunications and Internet technology allowed the introduction of the concepts of tele-embodiment, and virtual visit, in connection with the idea of a robotic assistant ([5], [6]). This progress seems to justify the optimism of Bill Gates who speaks about "a robot in every home", and estimates that the impact of the personal robots in everyday life will be comparable with that of the personal computers. In spite of this optimism, and of the great number of individual solutions described in the scientific literature, the number of commercial applications of home robots is, so far, limited to the cleaning robots and autonomous mowing machines. Significant results have been also reported in the field of intelligent wheelchairs (see, for example [10]).

Meanwhile, the problem of aging of the population aggravates, and the number of people in need of a minimal permanent medical assistance is constantly increasing, thus bringing a considerable pressure on the social security budgets (see [7]). Under these circumstances, it seems that there is an urgent social need for affordable robotic assistant for the elderly or disabled persons.

This paper describes a possible solution to this problem – the use of a distributed network of low cost, embedded microcontrollers to handle the tasks associated with the navigation of a mobile robot, as well as with the interaction between the robot and the environment. The experiment is not only limited to a software simulation using ARIA (Advanced Robot Interface for Applications), [14], and real-time implementation has been made. In simulation and real-time, the kinematic model of the real robot, WMR Pioneer 3-DX, from Mobile Robots Inc, [14], has been used. The whole application is written in ANSI C, so that it is easily portable to an embedded controller carried by the robot. Pioneer 3-DX figure 1, provides a base for service or performance robots. Pioneer 3-DX has the ability to: communicate with other robots, connect to PC's via the Internet or LAN, run autonomously. Pioneer 3-DX is designed for development use by seasoned programmers with ARIA. Pioneer 3-DX runs indoors on flat floors. It can traverse low sills and household power cords. With upper and lower sensing, the Pioneer 3-DX will turn away from nearly all obstacles. Pioneer 3-DX can navigate autonomously, avoiding obstacles, leading tours or providing remote surveillance. Display the robot's sonar or laser map on a remote monitor or plasma screen overhead to demonstrate how robots sense the world. Using ARIA development tools included with Pioneer 3DX, multitudes of behaviors may be created. Custom accessories plugged into Pioneer's user I/O bus are already integrated into ARIA through ARCOS (Advanced Robot Controller Operating System) packets, [14]. It is simulated above mobile robot

This work was supported in part by the Romanian High Education Scientific Research National Council, under project IDEAS, ID_506.

platforms with MobileSim, built on the Stage simulator platform. MobileSim provides access to simulated robots and their environment to ARIA programs.

Fig.1: Mobile platform Pioneer 3-DX WMR.

The paper is structured as follows: The first section is a quick overview of the existing art in the field of robotic assistants. Section 2 contains the general assumptions and requirements deriving from the previous work, considered in the current experiment. Section 3 describes fuzzy control, the structure of the real-time setup. Section 4 presents bubble rebound obstacle avoidance. In Section 5 some conclusions are presented.

2 FUNCTIONS OF A ROBOTIC ASSISTANT

The complexity of the tasks associated with the control of autonomous mobile robots require high computational power for the control system, which results in high cost of the equipment, and even higher costs for developing the software for the control applications. Besides that, most of the existing control solutions tend to achieve the maximum generality, by assuming that the robot is operating in a totally unknown/unstructured environment. In fact, the environment where a robotic assistant is destined to operate, such as a home, a nursing facility, or a community center for the elderly is fully known, and easy to structure in order to reduce the on-board equipment. For example, it is much more convenient to provide the robot with the capability to communicate with the alarm system of the building, rather than programming the robot to patrol, and try to use its on-board sensors to emulate the functions of the motion, or smoke detectors.

This leads to the idea of a robotic agent that operates in a "smart environment". This idea is not entirely new. Reference [8] describes a "smart house" for people with physical disabilities, wherein an intelligent wheelchair and a smart bed are capable to support the disabled for movement. Moreover, in [9] a robotic system aimed to support blind people for shopping.

Based on the above mentioned literature, and on the analysis of many other projects not listed here for lack of space, we have extracted a list of distinctive features required for a low cost robotic assistant.

A simple and robust navigation system is an absolute requirement for such a robot. Although LASER and LIDAR sensors have better performances, sonar based navigation should be also considered, for economic reasons. A self calibration system to compensate odometric cumulative errors is recommended, to improve the overall robustness.

It is highly desirable that the robotic assistant can perform the function of walking aid for persons with limited locomotion impairment. An intelligent wheelchair or even a walker could be the ideal solution.

A robotic arm may seem desirable, but it is likely to dramatically increase the cost of the whole project. Therefore we did not consider this option.

It is desirable that the robot have the capability to recognize a limited number of vocal commands. The attempts to design robots able to sustain a conversation in natural language with human operators have produced questionable results at much higher costs.

It is desirable that the robot can monitor some signals from the environment (e.g. the alarm system of the building, or some medical equipment) and control various appliances, lights, heating, air conditioner, etc. With this approach, the robot is seen as part of a "smart environment".

It is desirable that the robot can interact with the communication system of the environment and select between various levels of emergency calls.

The function of cognitive prosthesis (e.g. reminding the assisted person to eat, take medication etc. at certain moments of time) is optional, since it can be easily implemented on a stationary computer.

And, last but not least, it is highly desirable that the robot can be easily programmed by a user with little or no knowledge of robotics.

3 FUZZY CONTROL OF THE MOBILE PLATFORM PIONEER 3-DX

Considering the above stated requirements, we propose a minimal structure for the robotic assistant as shown in figures 2 and 3. Figure 2 depicts the equipment located on the robot. Figure 3 shows the equipment deployed in the environment.

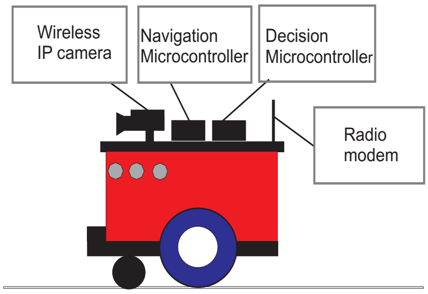

Fig.2: The on-board equipment

Besides the equipment presented in figures 2 and 3, the robot have additional built-in electronics including power drivers for the wheels, sonar sensors, odometric system, and communication interfaces, figure 4.

Note the presence of four embedded microcontroller units, having the following functions:

The *navigation module* receives as input the odometric information and sonar data reported by the built-in

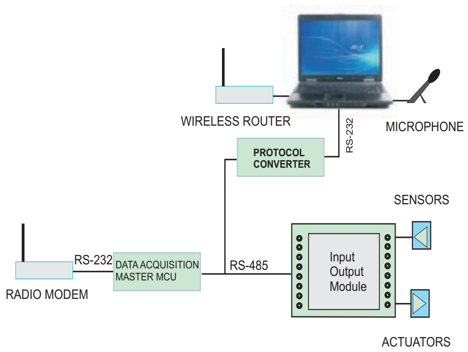

Fig.3: The equipment located on the ground

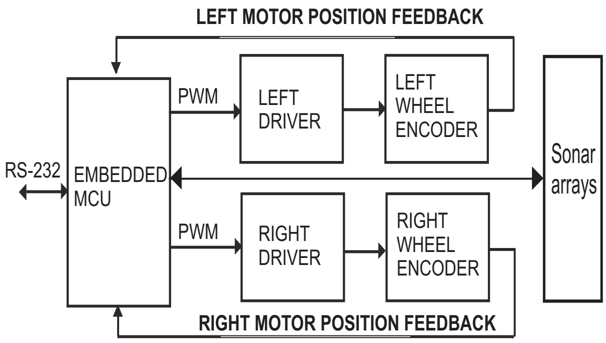

Fig. 4: The built-in electronics of the robot

electronics of the robot, and the coordinates of the current goal point transmitted by the decision module. Generates as output elementary motion commands, by transmitting reference values for the speed v_R, v_L of each drive wheel. This module is also responsible with the obstacle avoidance;

The decision module receives as input general robot status information (such as: battery status, bumper status, stall, etc.), as well as the state of the sensors connected to the inputs of the data acquisitions modules and the codes associated with a set of voice commands. It generates motion commands for the robot, and activation commands for the actuators connected to the I/O modules (e.g. to control the appliances, lights etc., or to transmit emergency calls);

The *data acquisition master* module is responsible with the time consuming task of scanning the data acquisition modules. It generates synthetic reports for the decision module;

The *protocol converter* module was introduced to convert the output of the speech recognition software to a format compatible with the protocol required by the data acquisition modules, so that all the information from the environment can be handled by the same data acquisition master module.

In addition to the four microcontrollers described above, a regular personal computer and a wireless router connected to the Internet were included in the system. The tasks associated with this equipment are:

Run dedicated software for speech recognition. When a predefined voice command is recognized, the computer sends an identification string through a serial port to the protocol converter module;

Provide the physical interfaces for transmitting the images and sound collected by the wireless IP camera carried by the robot through the internet.

Besides the hardware modules described above, the experimental setup includes the following software applications:

A robot simulator (MobileSim) offered by MobileRobots. This is a modified version of the Player/Stage Simulator, a very useful resource to speed up control software development;

Since the robot simulator communicates via TCP, and the actual robot is controlled through a RS232 communication line, a special software application, called “Serial to Ethernet Connector” is required to assure direct communication compatibility between the microcontroller navigation module and the simulator software.

With this structure of the control system, each of the four embedded microcontroller module can be reduced to a minimal configuration, as shown in figure 5A. Figure 5B shows the actual size of the embedded module. Two navigation controls have been addressed in this study: the path following problem, and the obstacle avoidance.

Since the environment is fully known, the user can define a number of routes (a route is an oriented path) as a succession of line segments. Each segment is described by (1).

$$Ax + By + C = 0 \qquad (1)$$

where:

$$\begin{aligned} A &= y_{i+1} - y_i \\ B &= -x_{i+1} - x_i \\ C &= x_{i+1} y_i - x_i y_{i+1} \end{aligned} \qquad (2)$$

(x_i, y_i), (x_{i+1}, y_{i+1}) are the coordinates of the points that determine the segment.

With these notations, the distance from the point (x_0, y_0) to the current segment of the segment, defined by two successive points (x_i, y_i), (x_{i+1}, y_{i+1}) is determined by (3).

$$d = \frac{| Ax_0 + By_0 + C |}{\sqrt{A^2 + B^2}} \qquad (3)$$

The position error for a vehicle attempting to follow the curve from (1) is the signed distance between the current point (x_0, y_0) and the line (1):

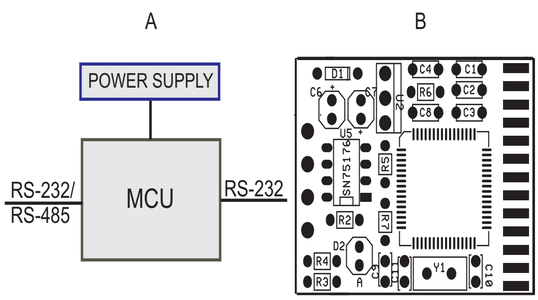

Fig. 5: Structure and layout of the MCU modules

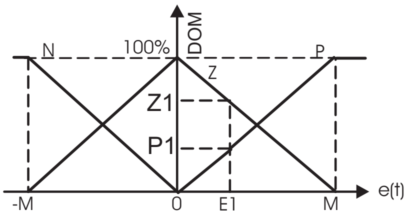

Fig. 6: Shape of the membership functions for e(t), e’(t)

Tab. 1: Rule base for the Fuzzy controller

error derivative e’(t)	error e(t)		
	N	Z	P
N	HL	LH	LM
Z	HL	MM	MH
P	ML	MH	LH

$$e = \frac{Ax_0 + By_0 + C}{\sqrt{A^2 + B^2}} \tag{4}$$

and the derivative of the position error is:

$$e' = \frac{\Delta e}{\Delta t} \tag{5}$$

By defining three fuzzy domains N (Negative), Z (Zero) and P (Positive) for each of the variables e and e', with membership functions as shown in figure 6, it is easy to define a set of linguistic rules to describe the desired behavior of the fuzzy controller.

The general structure of the logic sentences for the fuzzy controller is:

“If the error is positive and the error dot is positive, then v_R must be HIGH and v_L must be LOW.” The entire “rule base” describing the fuzzy controller is presented in Table 1. Each cell of table 1 contains a logic sentence and should be read as: “If $e(t)$ is Negative AND $e'(t)$ is negative, THEN v_L must be HIGH and v_R must be LOW”, where H, M, L designate the singleton values for HIGH, MEDIUM and LOW fuzzy domains of the outputs v_L, v_R respectively.

The truth value of the antecedent of the above sentence is $min(N(e), N(e'))$, where $N(e)$ *and* $N(e')$ are the degrees of membership of $e(t)$ and $e'(t)$ to the domain N.

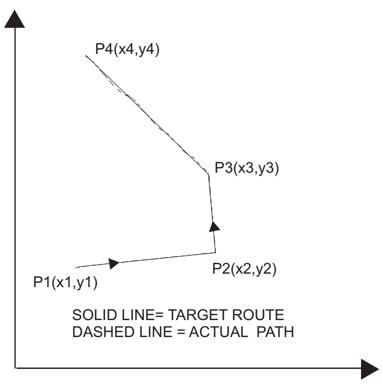

Fig. 7: Results recorded with the fuzzy controller

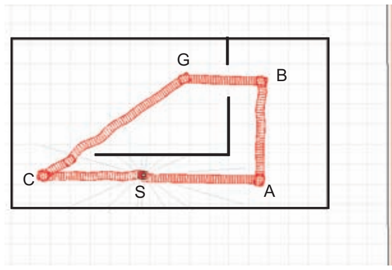

Fig.8: Trajectory of the simulated robot

The crisp output of the fuzzy controller is a combination of all the rules in the rule base as follows:

$$v_{OUT} = \frac{\sum_{i=1}^{K} z_i S_i}{\sum_{i=1}^{K} z_i} \tag{6}$$

where:

$$z_i = min(E_i, E'_i) \tag{7}$$

S_i is the corresponding singleton value of the fuzzy output, and K is the total number of rules in the rule base. E_i E'_i are the degrees of membership of $e(t)$ and $e'(t)$ to the domain corresponding to the cell i. The actual output variables of the fuzzy controller are the reference values for the speed v_R, v_L of each drive wheel. Thus, both translation speed (8) and steering (9) are controlled by the same fuzzy controller.

$$v = \frac{v_R + v_L}{2} \tag{8}$$

$$\omega = \frac{v_R - v_L}{b} \tag{9}$$

where b is the kinematic “bias” of the robot (distance between the planes of the drive wheels).

See [11] and [12] for more details on the implementation of the fuzzy controller. Figure 7 shows a plot of the successive positions reported by the real robot, following a path in the absence of obstacles, and figure 8 is a snapshot of the trail of the simulated robot moving from a start point S to a goal point G, on two different routes, as recorded by MobileSim.

4 BUBBLE REBOUND OBSTACLE AVOIDANCE

A new obstacle avoidance algorithm, based on ricochet method is presented here. The robot used in this experiment have an array of 8 sonar cells located on a semi-circular band on the front panel of the robot, as shown in figure 9. The readings of the sonar are unsigned integers in the range $[100mm, \;\; 5000mm]$. Therefore, the “visible” universe for the robot is a semi-circular area, delimited approximately by the curve A, in figure 9. Under these conditions, it would be difficult, if not impossible to attempt a rigorous path, a priori, computed based on environment mapping.

We have defined an obstacle detection area as a petal-shaped region, more elongated along the motion direction (the gray area delimited by the curve B in figure 9), which is permanently scanned by the navigation unit. If any of the sonar cells reports an obstacle within this area, the control unit computes the least resistance angle:

$$\alpha_0 = \frac{\sum_{i=1}^{8} \alpha_i D_i}{\sum_{i=1}^{8} D_i} \qquad (10)$$

where α_i is the position angle of the sonar cell i, relative to the current heading, and D_i is the reading of the sonar cell i. Note that the entire sonar visibility range is considered in (10).

The rotation angle computed with (10) is also relative to the current heading.

See figure 10 for an example of a possible distribution of obstacles.

Having the least resistance angle computed, the robot abandons its current path, and adjusts its heading to the new path, then moves forward a distance $d' \, min(D_i)$, or until it detects a new obstacle. After this detour, it checks whether the original goal point became visible, and if so, it defines a new path to the goal point, as a line defined by the current point and the goal point.

A schematic of the process of moving between a start point S and a Goal point G, in the presence of an obstacle is presented in figure 11.

The actual recording of the motion of the robot while detecting and avoiding obstacles is presented in figure 12. The ricochet algorithm, by robot simulator MobileSim, has been tested and in real-time on PeopleBot has been implemented. No any significant differences between simulation and real-time has been obtained. Test results for several obstacle configurations, in figures 13, 14 and 15 have been shown.

The obstacle avoidance algorithm described above produces reasonably good results with static obstacles. It can even handle dynamic obstacles, provided that the moving objects have speeds comparable to the speed of the robot.

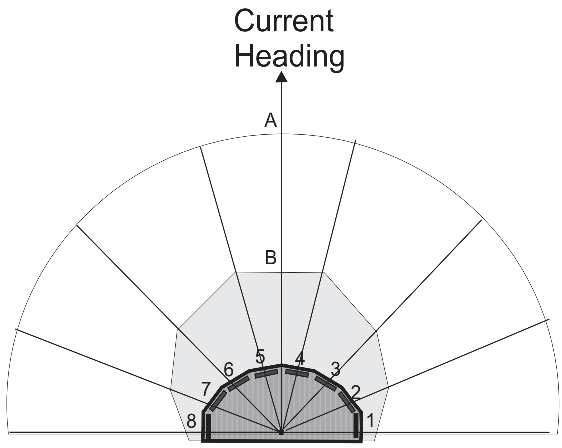

Fig. 9: Angular disposition of the sonar cells

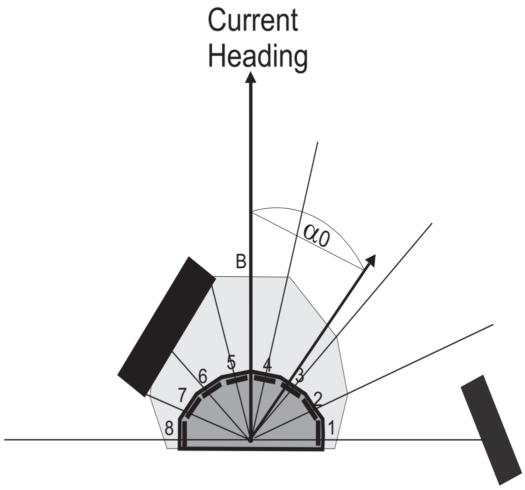

Fig.10: Illustration of the concept of least resistance angle

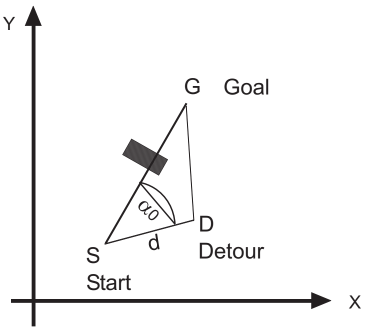

Fig.11: Modified path upon detection of an obstacle

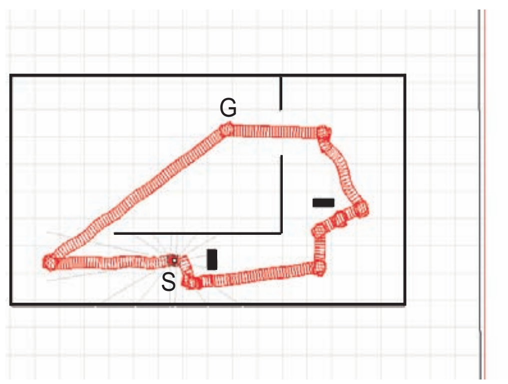

Fig.12: Trajectory of the simulated robot with obstacles

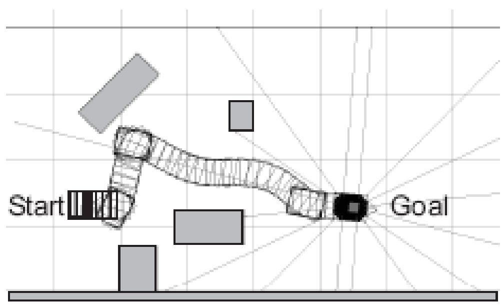

Fig.13: Robot trajectory for an obstacle configuration

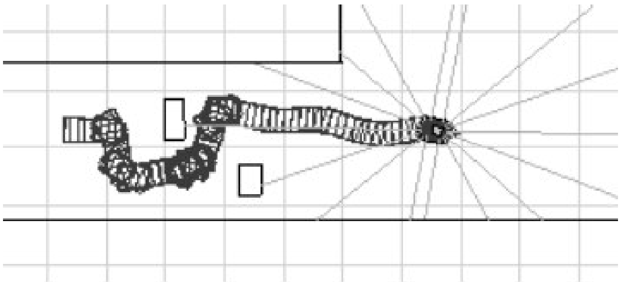

Fig.14: Robot trajectory for tight path

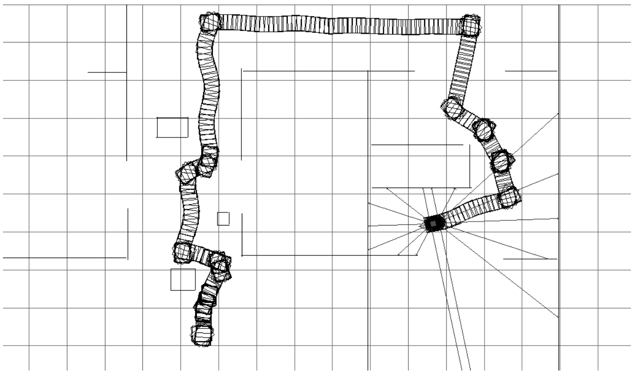

Fig.15: Robot navigation by labyrinth and intermediary obstacles

The selection of the solution for the robot decision problem, was mainly determined by the requirement that the robot must be easily programmable by means of translating simple linguistic sentences like: “If the input I_i is activated, AND there is no voice command, then move on route R_n”.

This leads to a PLC-like programming language, with the difference that, voice commands are treated like any other inputs, and predefined routes are treated as regular digital outputs.

Another difference is that the actual inputs and outputs of the system are integrated in the environment.

5 CONCLUSIONS

This experiment demonstrates that the idea of controlling a robotic assistant with a distributed structure of embedded microcontrollers is entirely feasible and can lead to significant cost reduction.

Besides the low cost, there are many other advantages in the proposed solution: low power consumption, less effort to develop the software, use of regular, low cost data acquisition modules, etc. Obviously, any of the sub-systems described here is almost endlessly perfectible and further research for improving the navigation algorithms, and the user programming techniques are definitely worth the effort.

REFERENCES

[1] J. Borenstein, Y. Koren, “A Mobile Platform For Nursing Robots”, *IEEE Transactions on Industrial Electronics*, May 1985, pp.158-165.

[2] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, “Towards robotic assistants in nursing homes: Challenges and results”, *Robotics and Autonomous Systems,* Volume 42, Issues 3-4, 31 March 2003, pp. 271-281.

[3] J. Matthews, S. Engberg, M. Montemerlo, J. Pineau, N. Roy, J. Rogers, S. Thrun , S. Handler, T. Starrett, D. Ting and R. Travis, “The Nursebot project: Results of preliminary field studies during development of a personal robotic assistant for older adults”, *Proceedings of the Greater Pittsburgh 14th Annual Nursing Research Conference*, Pittsburgh, PA, 2002.

[4] Lu Tao, Yuan Kui, Zhu Haibing and Hu Huosheng, “An Embedded Control System for Intelligent Wheelchair”, *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference*, Issue , 2005 pp: 5036 – 5039

[5] E. Paulos, J. Canny, Social “Tele-Embodiment: Understanding Presence, Autonomous Robots 11”, pp: 87–95, Kluver Academic Press, 2001

[6] N. A. Sachs, P. L. Nulud P and G. E. Loeb, “Virtual Visit: improving communication for those who need it most”, *Studies in Health Technology of Information,* (Vol. 94, pp: 302-310), 2003

[7] George Magnus, “The Age of Aging: How Demographics are Changing the Global Economy and Our World”, John Wiley & Sons, 2008

[8] Hyun Park Kwang-, B. Zeungnam , Ju-Jang Lee, B. Kook Kim, Jong-Tae Lim, Jin-Oh Kim, Heyoung Lee, D. H. Stefanov, , Dae-Jin Kim, Jin-Woo Jung, Jun-Hyeong Do, Kap-Ho Seo, Chong Hui Kim, Won-Gyu Song and Woo-Jun Lee, “Robotic smart house to assist people with movement disabilities”, *Journal of Autonomous Robots ,* Volume 22, Number 2, pp: 183-198, February, 2007

[9] C. Gharpure, V. Kulyukin, “Robot-Assisted Shopping for the Blind”: Issues in Spatial Cognition and Product Selection, International Journal of Service Robotics, March 2008

[10] A. Filipescu, A. Stancu, S. Filipescu and G. Stamatescu, “Real-time Sliding-mode Adaptive Control of a Mobile Platform Wheeled Mobile Robot”, Proceedings of the 17th World Congress, The International Federation of Automatic Control, Seoul, Korea, July 6-11, 2008, pp 4393-4399, ISBN: 978-3-902661-00-5, ID: 978-1-1234-7890-2/08/$20.00 © 2008 IFAC, 10.3182/20080706-5-KR-1001.0881, Elsevier IFAC Publication

[11] I Susnea, G. Vasiliu and A. Filipescu**,** “Real-Time, Embedded Fuzzy Control of the Pioneer3-DX Robot for Path Following”, Proceedings of 12th WSEAS International Conference on SYSTEMS, Heraklion, Crete, Greece, July 22-24, 2008, pp.334-338, ISBN: 978-960-6766-83-1, ISSN: 1790-2769,

[12] Adrian Filipescu, Susnea I, Adriana Filipescu, Stamatescu G.,Control of Mobile platforms as Robotic Assistants for Elderly, Proceedings of the 7th Asian Control Conference, Hong Kong, China, August 27-29, 2009, IEEE Catalog Number CFP09832, ISBN:978-89-956056-9-1, pp.1456-1461.

[13] Adrian Filipescu, Susnea I, Adriana Filipescu, Stamatescu G., Distributed System of Mobile Platform Obstacle Avoidance and Control as Robotic Assistant for Disabled and Elderly, Proceedings of 2009 IEEE International Conference on Control and Automation Christchurch, New Zealand, December 9-11, 2009, IEEE Catalog Number CFP09537, ISBN: 978-1-4244-4707-7, pp: 1886-1891, Library of Congress:2009904841.

[14] http://mobilerobots.com/ResearchRobots/ResearchRobots/ResearchPatrolBot.aspx