Proceedings of the 10th Mediterranean Conference
on Control and Automation - MED2002
Lisbon, Portugal, July 9-12, 2002.

FUZZY LOGIC NAVIGATION AND CONTROL OF A NON-HOLONOMIC VACUUM CLEANER

Ali Okatan[1], **Georgi M. Dimirovski**[1†]

[1] Doğuş University, Faculty of Engineering, Dept. of Computer Engineering,
Acibadem, Zeamet Sk. 21, Kadıköy, TR-81010 Istanbul, Turkey
Fax#: + 90- (0)216- 327-9631
e-mail: aokatan@dogus.edu.tr

[†] SS Cyril and Methodius University, Faculty of Electrical Engineering,
Inst. of Automation & Systems Engineering, MK-1000 Skopje, Macedonia
e-mail: gdimirovski@dogus.edu.tr

Keywords: Embedded control, fuzzy-logic control; mobile robots, non-holonomic systems, vacuum cleaner.

Abstract

In this work, a simple fuzzy logic navigation and control design for a non-holonomic robotic vacuum cleaner is presented. The embedded control has been designed by using the Motorola flash micro-controller. A circular cleaning algorithm is used to clean the space, the execution of which is driven according to the underlying philosophy of binary random sequences and obstacle occurrence. Any kind of obstacle is sensed by infra-red detectors before collision may take place. In turn, navigation and control algorithm is redirects the robot cleaner toward some other area in the vicinity while avoiding a possible collision.

1 Introduction

The field of science and engineering of robotic systems is well matured since a decade ago [6], [12], [23], [25]. Even the application of soft-computing methods has advanced considerably [5], [9], [10], [17]. The underlying essence behind its successful development is the synergism of control, computing, industrial electronics and systems engineering [3], [5], [7], [14], [15], [21], [24], [25]. Several branches of applied robotics emerged, nonetheless, manipulation and mobile robotics remain the two main areas [10], [25]. In the area of mobile robotics [25], the branch of service robotics is also well established within which the topic of mobile manipulation robots is gaining momentum [10], [11], [25]. In addition to the main service function, it involves some issues of motion planning [1], [13], [14], [22] navigation with obstacle avoidance [4], [6], [8], and coping with non-holonomic constraints [1], [2], [6], [12].

The idea of self-navigated vacuum cleaner has been around for quite some time. Different groups have been working on the subject, e.g. see [14], [20]. Of course, the underlying control problem is the one of a non-holonomic mobile robot motion control [8], [9], [12]. On the grounds of this existing theoretical knowledge, however, we have handled this problem in a more pragmatic way by using the aforementioned synergism, and also employing the typical features of a cylindrical-body, four-wheel vacuum cleaner [19], [20].

A cleaning robot must sense all floor and clean whole surface. For accurate coverage and optimal operation, the navigation system must be capable of accurate positioning the robotic cleaner while simultaneously avoiding collisions with obstacles such as furniture and people. In floor cleaning task, the desired aim is to reach the obstacles as close as possible, but not collide with them [20].

2 On Kinematics Features of Car-like Mobile Robots and the Non-Holonomic Motion

Despite the fact that cylindrical geometry of vacuum cleaner facilitates its navigation and obstacle avoidance, still it is a “Car-like mobile robot” with a front-wheel-drive four-wheel vehicle typically (see Fig.1). The fact that the two rear wheels are not steerable and roll without slipping on the ground introduces a constraint on the vehicle motion, which expresses the fact that the velocity of the centre of the rear axle is co-linear with the orientation axis of the vehicle base. This type of constraint is a ‘*non-holonomic’* one and does not allow a closed form analytical solution for the vehicle trajectory [1], [13], [14]. It is this kind of physical constraints that make navigation and obstacle avoidance of a vacuum cleaner considerably involved if it is to operate as an autonomous unmanned vehicle in real-world environment of a limited 2D space. This is clearly demonstrated in the sub-section below.

2.1 On Kinematics Features of Car-like Mobile Robots

Let observe Figure 1 [6], and let use the following notations: by L the distance between the two axes of the wheels is denoted; by θ the angle between the major axis of the robot and x-axis of the absolute reference frame; by ϕ the steering angle (i.e. the orientation of the front wheels with respect to the major axis of the robot) is denoted; and by x_M, y_M, and V_M the co-ordinates and the velocity of car-like mobile robot, respectively, at the middle point of the axle of the rear wheels are denoted. Then, assuming that there is no slipping of the rear wheels, it is well known [5], [6], that the major velocity vector of M is always co-linear with the axis of the mobile robot:

$$(dx_M/\ dt)\sin\theta - (dy_M/\ dt)\cos\theta = 0. \qquad (2.1)$$

Apparently, this equation is not integrable. It is a constraint on the velocity of the robot but does not affect the dimension of the space of configurations. On the other hand, for a given configuration, the space of achievable velocities has a dimension of only two. These are therefore a non-holonomic constraint.

Another characteristic of the car-like robots is that the steering angle is constrained from above:

$$|\phi| \le \phi_{max} < \pi/2 \qquad (2.2)$$

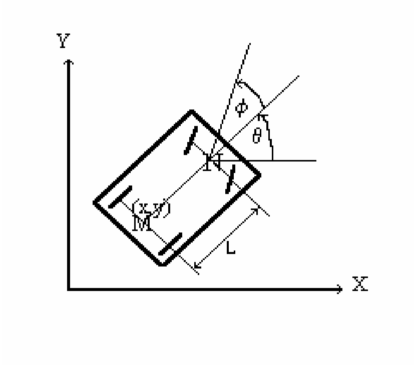

Fig. 1. The schematic of a car-like mobile robot, a non-holonomic moving object in a 2D real-world space (Dimirovski et al, 2001)

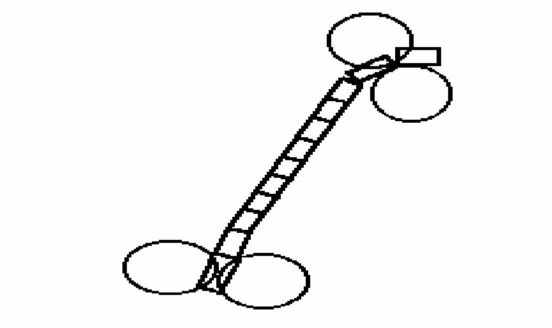

Fig. 2. A trajectory joining two positioning configurations of a non-holonomic car-like robot in motion (Dimirovski et al, 2001)

And, yet another characteristic is that the turning curvature radius of steered mobile robot is constrained from below. The radius of curvature of the trajectory, R, is computed according to the steering angle. Given the constraint on the steering angle ϕ_{max}, we can express the minimum turning radius of the vehicle as follows:

$$|R| \ge R_{min} = \mathrm{L}/\sin\phi_{max}\,. \qquad (2.3)$$

For the purpose of a certain physical verification of the above presented discussion examine the simulation of the motion in Figure 2 in conjunction with the schematic of car-like mechanics in Figure 1. Figure 2 depicts a cruise between start and target end positions, turning manoeuvre, and parking alignment [12], [25].

2.2 On Motion Trajectories of Non-holonomic Car-like Mobile Robots

In traditional robot motion planning equality or inequality constraints over the configuration variables are considered. These types of constraints are the '*holonomic'* ones, i.e. topological properties of the configuration space can be altered, or the robot is constrained to a lower dimensional manifold of the configuration space. Yet, any car-like mobile robot as well as our vacuum-cleaner robot is in fact a non-holonomic mechanical object. Non-holonomic constraints [1], [12], on the other hand, are imposed on the velocities instead of on the position variables of a system. A non-holonomic constraint is expressed as a non-integrable equation involving derivatives of the configuration parameters and it cannot be reduced to equality constraint on the position parameters. Such constraints are expressed in the tangent space at each configuration defining allowable velocities of the system. For example the equation $\mathbf{w(x)x'=0}$ expresses a linear velocity constraint, where $\mathbf{x(t)\in R^n}$ represents configuration of the system and $\mathbf{x'}$ belongs to the tangent space of the configuration space at $\mathbf{x}$.

In here, we consider the problem of motion of rigid body system from one position to another with certain configuration, and this motion to be repeated as many times as needed. There are many systems that are not capable to move between two arbitrary configurations due to kinematics constraints. These limit the set of reachable configurations to a space of lower dimension. Unlike such systems, we are interested in the cases in which the kinematics constraints do not restrict the reachable configurations. An example of such a system is any front-wheel drive car-like mobile robot and/or front-wheel drive vacuum-cleaner (as in our experimental research). Its kinematics is constrained because the front wheels can only roll and spin, but not slide sideways. Despite this, we can park a car at any position and orientation. It is convenient to convert the problem of finding a path between two given configurations subject to k constraints of form $\mathbf{w(x)x'=0}$ into the problem in control theory. Namely, we are interested in describing the directions in which we can move car-like mobile robot (our smart vacuum cleaner) instead of those in which we cannot by means of appropriate navigation control.

There are two different kinds of techniques to the problem of generating trajectories for car-like mobile robots with non-holonomic constraints. Following the first kind of methodological techniques, a geometric reasoning is utilized to construct feasible trajectories by assembling arcs of simple curves [22]. In contrast, the philosophy within the second type relies on generating paths by searching the configuration space of the robot via applying the non-holonomic constraint as an additional heuristic at every step of the search. Barraquand and Latombe [1] have proposed an elegant method along these lines based on hierarchical bit-map discretization and potential-field functions. Their method may be generalised to robots with large number of degrees of freedom, however, its significant drawback is in large memory and computation time required, and is limited to bounded domains because search in configuration space is necessary. Our analytic-simulation technique belongs to the first type of techniques, and it is extended with fuzzy-logic control for the case of environments with fixed obstacles.

In our respective studies, we have found that the simplest way to comply with and resolve the constraint issues within a limited 2D environment, which were discussed insofar, is to employ geometric trajectory generation. Note that a car-like mobile robot moving in a 2D-world has 3 degrees of freedom: the co-ordinates of a given point of the robot, and the orientation of the robotic system itself. A path is a curve joining 2 points and providing a change in the robot orientation.

The orientation change of the vehicle of robotic system can be accomplished only when the vehicle is moving. To actually change the orientation θ of the vehicle with a minimum length of trajectory, in fact, requires the maximisation of the curvature of the trajectory within the limits of non-holonomic constraints. This may be achieved when the vehicle is moving along a circle the radius of which is the minimum radius of curvature. The problem is that circles take a heavy toll on the length of the trajectory: for a long trajectory, a straight-line segment is really needed. Hence the idea of combining arcs of circles and straight-line segments resulted. Therefore the trajectories are typically designed the following way [25]: the robot moves along one of the circles related to the initial configuration in order to be oriented towards the goal target, no matter what the final orientation must be. Then, it moves along a straight-line segment in the direction of the target.

The last part of the trajectory is meant to provide the desired final orientation: the robot moves along one of the circles related to the final configuration. In order this be physically feasible for implementation, the trajectory must provide a continuous orientation to the mobile robot. Hence, the straight line segment must be tangential to the circles (see Fig. 2). The geometric method provides a continuous orientation to the robot because the circles and the segment are tangent. However, the curvature of the obtained curve is discontinuous: a non-zero constant on the circles, and a zero constant on the straight-line segments.

3 Simple Fuzzy Navigation Control for Obstacle Avoidance

In the realistic case when the environment of motion contains obstacles, e.g. such as in an industrial environment, the mobile robot is treated as an autonomous control vehicle which can operate in camera mode and trajectory generation be based on fuzzy logic control [4], [6]. The predefined fuzzy subsets can be modelled by means of triangular-shaped (e.g., see Fig. 3) or bell-shaped fuzzy sets. These are employed to define three information-files covering the distance from an object, the orientation and the speed. Clearly, the first two files are directly linked with the obstacle detection whereas the last one is to determine the strategy for steering. The location variable of mobile robot relative to some obstacle may be specified by means of fuzzy variable given in linguistic terms set:

$$O_{\text{distan}} = \{\textit{near distance, middle distance, far distance, no obstacles}\}. \quad (3.1)$$

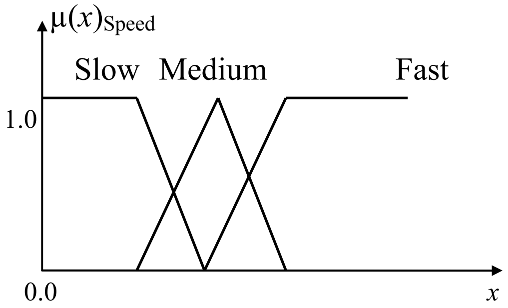

Fig. 3. Fuzzy subsets of speed variable for car-like mobile robot, which can be replaced by bell-shaped ones; actual crisp values depend on engineering design parameters of the robot

To obtain the required orientation, this information is concatenated respectively with orientation fuzzy variable given by the following set:

$$C_{\text{orient}} = \{\textit{more left, slightly left, straight, slightly right, more right}\}. \quad (3.2)$$

And also, a third fuzzy variable C is defined to cover the ranges of motion speed, which in linguistic terms is represented by:

$$M_{\text{speed}} = \{\textit{slow, middle, fast}\}. \quad (3.3)$$

Then, in general terms, the fundamental rule of vacuum-cleaner motion can be specified by means of the following linguistic expression:

$$\text{M} = \text{“}\textit{Depending on } O_{\text{distan}} \textit{ move } C_{\text{orient}} \textit{ at speed } M_{\text{speed}}\text{”}. \quad (3.4)$$

Now, it should be noted, for the fuzzy navigation strategy which emanates from the above analysis and modelling in terms of such a pre-defined fuzzy system the following result (e.g., see [4], [6]) applies:

$$N = (\Sigma_{0,\ldots,p} C_i^{\,p})^k . \quad (3.5)$$

Here, N represents the number of different possible situations, p is number of fuzzy subset terms for obstacle angle fuzzy regions, and k is number of fuzzy subset terms for obstacle distance fuzzy-regions. Following model representation Eqs. (3.1)-

(3.5), one can notice that N = 32768 different situations have to be observed. It has been shown [6], [7], however, that in the case of car-like robot motion in limited 2D space (e.g., as in an industrial job shop floor) these different situations in navigation control may be accommodated by means of the simple fuzzy knowledge base presented below.

Therefore a simple, but practically efficient, control strategy has been elaborated, which is easy to implement and employ. It can be represented by the following linguistic model in terms of the respective pre-structured fuzzy-rule base [6] as follows:

- if no obstacles, go to the end position fast;
- if there is an obstacle at far distance more to the left, then go slightly right with medium speed;
- if there is an obstacle at middle distance more to the left, then go slightly right with slow speed;
- if there is an obstacle at far distance slightly left, go more to the right with medium speed;
- if there is an obstacle at middle distance slightly left, go more to the right with slow speed;
- if there is an obstacle at far distance more to the right, go slightly left with medium speed;
- if there is an obstacle at middle distance more to the right, go slightly left with slow speed;
- if there is an obstacle at middle slightly right, go more to the left with slow speed.

In the course of this research, we have demonstrated by simulation experiments that this set of rules does represent an almost complete imitation of a human operator using conventional vacuum-cleaner. This leads to an development of a tree-structured fuzzy-rule base with only 32 or less rules of the class 'if-than-else' [6]. Apparently, this is a fuzzy-rule production system based control that can be real-time executed and efficiently used in a number of application cases of car-like mobile robots for sample operations in limited 2D space.

4 An Outline of the Experimental Engineering Implementation

In here we present a brief summary only of the experimental implementation of our robotic vacuum-cleaner, based on [20]. For the area of 20 square-meter and 10 minutes total operating time the robot covered 95 % of the total area on the average. As the operating time has been increased the area covered has increased. Physical experiments and field tests have shown this method enables 95% of the floor to be cleaned up provided a sufficient time is available. A special analysis must be made, however, in order to get a good prediction on how large floor space can be covered in a given interval of time. Insofar this analysis has not been done yet in this work. For the time being, we rely solely on obtained empirical results. This vacuum cleaner robot can be used to clean the big areas otherwise many workers are needed to do the same cleaning.

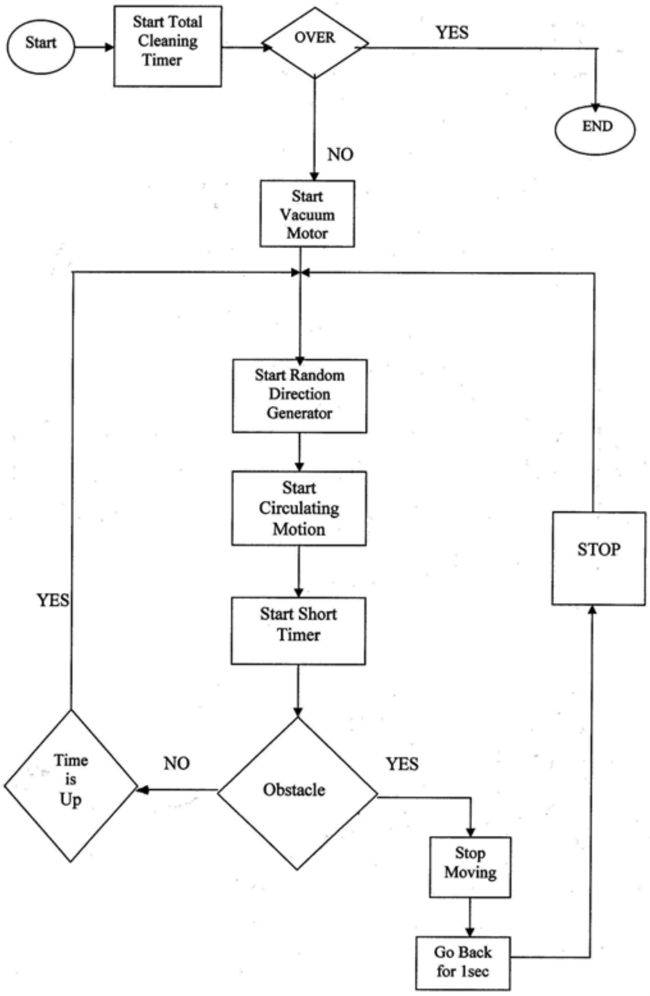

Fig. 4. Simplified algorithmic flow diagram of application program implementing control and navigation of vacuum-cleaner robot

The hardware for the control sub-system system of vacuum-cleaner robot, which has been designed and implemented [19], [20], is consisted of: and embedded 8-bit µcontroller made by Motorola, infrared collision detector, and electrical motor drivers [21], [24]. The basic engineering design structure of the vacuum-cleaner control sub-system may be inferred from the schematic in Figure 5. The Motorola microcontroller MC68 HC908JL3 has one external interrupt input, two channel 16-bit timer interface module (TIM) [16], [18]. This micro-controller has 4096-bytes of flash memory. This makes the programming the microcontroller reasonably easy. A simplified block diagram depicting the overall structure of the control system is shown in Figure 5. External interrupt input is used for obstacle detector. Therefore, whenever the robot approaches an obstacle the IR detectors senses it and an interrupt is generated. The 16-bit timer interface unit is used for different timing events required in the software.

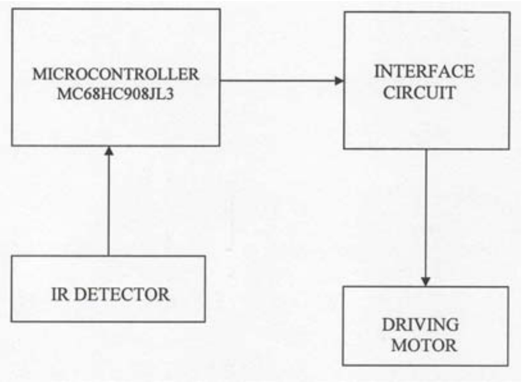

Fig. 5. Block diagram of the physical arrangement of vacuum-cleaner control sub-system

The applications software [3], [19], [20] was written to accomplish the following tasks presented below. All software was written in assembly language. Instead of mapping the place, we followed different approach. Initially the robot is directed randomly. After travelling a pre-defined distance it stops and stars rotating around itself to the right or left for pre-defined amount of time. A random number generator helps whether it will rotate right or left. During this procedure, should the robot senses an obstacle, it stops, goes back for some pre-specified distance, defined by the user, and then stops. In the next step of the navigation, it chooses a new direction and repeats the same procedures already executed, and so on. Overall algorithmic flow diagram is depicted in Figure 4.

5 Conclusion

In this paper we have reported on a design and implementation project for a robotic vacuum cleaner capable of efficient work while retaining its capability of good self- navigation and obstacle avoidance in limited floor space such as a hall, room, etc. Both early experimental trials as well as proper field tests in cleaning various surfaces, our robotic vacuum cleaner has demonstrated quite satisfactory performance. Design and implementation were done on the grounds of existing theoretical knowledge and by using standard hardware, in particular, the Motorola microcontroller and associated electronics needed.

The purpose of application of fuzzy control in navigation system of a car-like mobile robot, e.g. an autonomous vacuum-cleaner, within an operating environment having obstacles is to realise an optimum navigation control performance by implementing the experience of a human operator. Fuzzy-rule knowledge base (control rules) is implemented by means of linguistic synthesis using ‘if-then-else‘ rules. The application software structure should be brought to a stage “ready to operate” by simple specification of the desired final frontier coordinates only. This point along with the fact that only a few milliseconds of run time on a standard PC platform is necessary to generate the trajectories makes this methodology quite attractive for real-time navigation in an a-priori unknown environment with fixed obstacles such as in a factory floor or living/ working premises.

Although vacuum-cleaner robots, by the very conceptualization, are not conceived to apply path planning, because trajectories are not to and cannot be generated beforehand, this has to be observed in order to take into account the problem of non-holonomic constraints. For, the steering a vacuum-cleaner robot has to be implemented as steering a non-holonomic mobile robot by means of randomly generated trajectories as close as possible along a physically admissible trajectory. This can be done by observing the controllability aspects of actually non-holonomic non-linear systems. A kind of envisaged geometries of possible manoeuvring are helpful in the design of navigation algorithms.

Acknowledgement

Authors express their gratitude to the authorities of Dogus University for supporting in part their research in mobile robotics.

References

[1] J. Barraquand, J.C. Latombe, "Non-holonomic multi-body mobile robots: Controllability and motion planning in presence of obstacles", in *Proceedings of IEEE Int. Conference on Robotics and Automation,* Sacaramento CA, The IEEE New York pp. 2328-2335 (1991).

[2] A.M. Bloch, M. Reyhanoglu, N.H. McClamroch, "Control and stabilization of non-holonomic dynamics systems", *IEEE Trans. On Automatic Control,* **AC-37**, pp. 1746-1757 (1992).

[3] C. Britton, and J. Doake, *Software System Development: A Gentle Introduction* (2nd edition), McGraw-Hill International Ltd., Maidenhead UK (1996).

[4] G.M. Dimirovski, O.L. Iliev, N.E. Gough, A.T. Dinibütün, O. Kaynak, "A contribution to fuzzy-logic analytical simulation methods for AGV motion navigation", in *Proceedings of ICRAM95 on Recent Advances in Mechatronics,* O. Kaynak, M. Ozkan, N. Bekiroglu, I. Tunay, Eds., UNESCO Chair on Mechatronics – Bogazici University, Istanbul, Vol. II, pp. 678-683 (1995).

[5] G.M. Dimirovski, O. Kaynak, "Soft computing, complex systems and control integration in mechatronics systems", in *Proceedings of the 2nd Asian Control Confrence,* Myoung-Sam Ko (General Chiar) and Zuengnam Bien (IPC Chair), The Institute of Control, Automation and Systems Engineers, Seul, Vol. III, pp. 443-446 (1997).

[6] G.M. Dimirovski, Z.M. Gacovski, K. Schlacher, and O. Kaynak, "Mobile non-holonomic robots: On trajectory generation and obstacle avoidance", in *Robot Control (SYROCO 2000),* P. Kopacek, Ed., Pergamon Elesevier Science, Ofxord, pp. 549-554 (2001).

[7] G.M. Dimirovski, "Research in robotics and flexible automation: What may be feasible in small developing countries?" (Invited Survey Paper), in *Robot Control (SYROCO 2000),* P. Kopacek, Ed., Pergamon Elesevier Science, Ofxord, pp. S3.(1-10) (2001).

[8] W.E. Dixon, D.M. Davson, E. Zergeroglu, F. Zhang, "Robust tracking and regulation control for mobile robots", *Int. J. Robust & Nonlinear Control,* **10**, pp. 199-216 (2000).

[9] R. Fierro, F.L. Lewis, "Control of a nonholonomic mobile robot using neural networks", *IEEE Trans. On Robotics & Automation,* **RA-9**, pp. 589-600 (1998).

[10] S.S. Ge, T.H. Lee, C.J. Harris, *Adaptive Neural Network Control of Robotic Manipulators,* World Scientific, Singapore (1998).

[11] U.D. Hanebeck, C. Fischer, G. Schmidt, "ROMAN: A mobile robot assistant for indoor service applications", in *Proceedings of the 1997 IEEE/RSJ/GI Int. Conference on Intelligent Robots and Systems*, Grenoble (FR), pp. 518-525 (1997).

[12] I. Kolmanovsky, N. McClamroch, "Developments in nonholonomic control problems", *IEEE Control Systems,* **15**, pp. 20-36 (1995).

[13] G. Laffeirere, H.J. Sussman, *Motion Planning for Controllable Systems without Drift.* Technical Research Report of Rudgers Center for Systems and Control, Rudgers University (USA), 1990.

[14] J.C. Latombe, *Robot Motion Planning*, Kluwer, Boston MA (USA), 1991.

[15] M. Mano and C.R. Kime, *Logic and Computer Design Fundamentals.* Prentice Hall, Upper Saddle River NJ (USA), 2000.

[16] Motorola, MC68HC908JL3/H, Technical Data (2000).

[17] R. Murphy, *Introduction to AI Robotics*. The MIT Press, Cambridge MA (1998).

[18] NOMADIC TECHNOLOGIES Inc., XR4000 Mobile Robot, http://www.robots.com/xr4000.

[19] A. Okatan, "Control of a robot car through Web", in *Automatic Systems for Building the Infrastructure in Developing Countries*, G.M. Dimirovski, Ed., Pergamon Elsevier Science, Oxford, pp. 121-124 (2001).

[20] A. Okatan, Smart Robotic Vacuum Cleaner, Technical Documentation on Development and Testing. Technical Research Notes, Dogus University, Istanbul (2001).

[21] E.J. Pasahow, *Microprocessor Technology and Microcoputers.* McGraw-Hill Inc., New York (USA), 1988.

[22] F.G. Pin, H.A. Vaseur, "Autonomous trajectory generation for mobile robots with non-holonomic and steering angle constraints", in *Proceedings of IEEE Int. Workshop on Intelligent Motion Control,* O. Kaynak, General Chair, Istanbul (TR), The IEEE, New York, Vol. I, pp. 295-300 (1990).

[23] Vukobratovic, M.K. and G.M. Dimirovski, "Modelling, simulation and control of robots and robotized FMS" (Invited Plenary Lecture), in *Proceedings of the IFAC WS on Automatic Control for Quality and Productivity,* A. Kuzucu, I. Eksin, A.T. Dinibütün, Eds., The IFAC, the TOK and Istanbul Technical University, Istanbul, Late Paper IPL3.(1-32) (1992).

[24] J.L. Whitten, and L.D. Bently, *System Analysis and Design Principles* (4th edition). Irwin McGraw-Hill, Boston (USA), 1998.

[25] Y.F. Zhang, *Recent Trends in Mobile Robots,* World Scientific, Singapore (1993).