PATH PLANNING APROACH FOR AUTONOMOUS AND ASSISTED NAVIGATION USING VELOCITY OBSTACLE PRINCIPLES

Marcelo Becker
State University of Campinas, School of Mechanical Engineering, Department of Mechanical Design, PO Box 6051, 13089-970, Campinas – SP, Brazil, Email: becker@fem.unicamp.br
Felix Wullschleger
Swiss Federal Institute of Technology, Institute of Robotics, ETH Center - IfR, CLA H-15.1, CH-8092, Zurich, Switzerland, Email: wullschl@ifr.mavt.ethz.ch

Abstract:

This article relates researches on Path Planning and Obstacle Avoidance techniques for autonomous and assisted navigation in real environments. To test our approach, simulations, using the software Matlab, and experimental tests, using the mobile robot SmartROB-2 equipped with two SICK Laser Scanners and the real-time system XOberon, were done. The results obtained were excellent, a new approach using the velocity obstacle technique was developed, and real tests using the mobile robot, and path planning controller are in progress.

Keywords: Path planning, obstacle avoidance, velocity obstacle, and autonomous and assisted navigation.

1. INTRODUCTION

Path Planning and Obstacle Avoidance approaches consist, respectively, on an algorithm that plans a path that results in achieving a given goal, and another one that allows the controller to detect and track obstacles, generating a new path. Actually, these techniques are being applied in real applications like: robot manipulators avoiding moving obstacles, intelligent vehicles negotiating freeway traffic, etc. This paper focus the problems found in applications similar to intelligent vehicles.

In the case of autonomous navigation of a vehicle, the obstacle avoidance algorithm generates the steering and velocity commands for the vehicle, guiding it to its goal, taking any necessary action to avoid collisions during the path. In the case of assisted navigation, the final decision is always took by the user and the controller should aid the user, providing suggestions of alternative paths to avoid crashes with obstacles. This kind of control is found mainly in applications where the interaction between man-machine is very close and the independence feeling of the user is very important (e.g. intelligent wheelchairs, assisted navigation systems for automobiles, etc.).

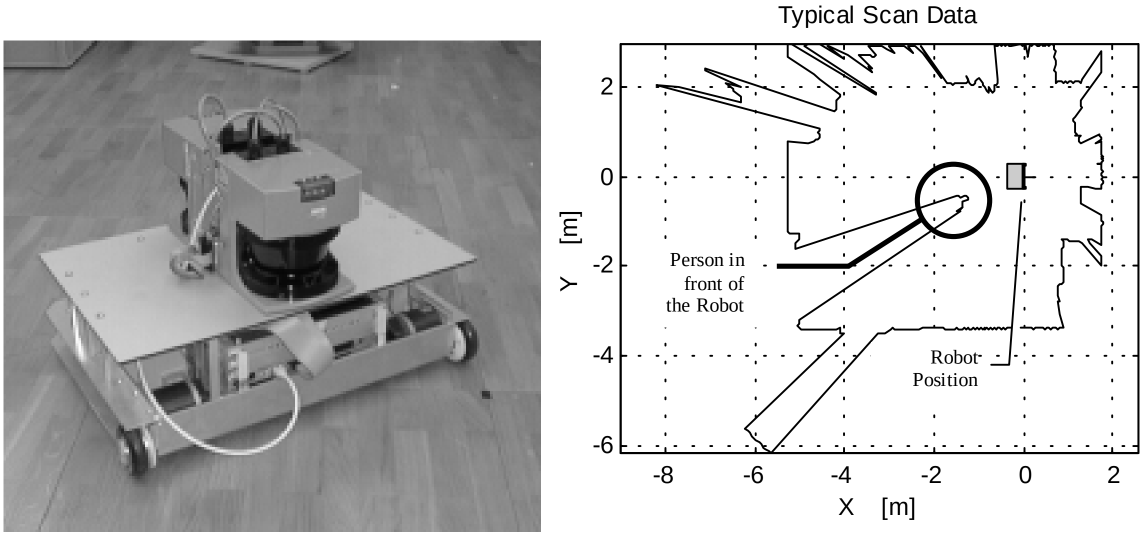
In a general manner, Path Planning can be divided into two main approaches: global path planning, based on a priori complete information about the environment, and local path planning, based on sensory informations in uncertain environment where the size, shape and location of obstacles are unknown (Beom & Cho, 1995). Global path planning methods can solve the path planning problems for completely known environments. But, they can not be used for navigation in complex and dynamically changing environments, where unknown obstacles may be located an *a priori* planned path. To overcome these difficulties, methods considering real-time environment informations from sensors must be used. Local path planning methods uses informations of the sensors to provide environmental information for the vehicle's navigator.

The navigator is a navigation and obstacle avoidance controller that generates the steering and velocity commands for the autonomous vehicle. Based on sensor readings, the vehicle should be able to perform local path planning and to take appropriate control actions. Conflict can appear, e.g.: go to goal position vs. obstacle avoidance. There are many approaches used to solve the Path Planning problem: Visibility Graph (Fu & Liu, 1990); Pattern Recognition (Wang & Tsai, 1991); Feedback Solutions (Feng & Krogh, 1991); Deformable Virtual Zones (Zapata *et al.*, 1994); Fuzzy Sets and Neural Networks (Beom & Cho, 1995 and Baxter & Bumby, 1995); Step-by-Step Planning (Bobyr & Lumelsky, 1999), and many others.

Our approach is based on the Velocity Obstacle approach (Prassler *et al.*, 1999a, b and Fiorini & Shiller, 1998). Nevertheless, we opted for to keep the modulus of the velocity constant during the path, changing its value only if a predetermined maximum steering angle is exceed. This choice generates a new approach that can be used in autonomous or assisted applications.

2. LASER SCANNERS

We decided to use laser range finders (or simply Laser Scanners), although the high costs of this kind of sensor, because actually no other sensor is equally suited for detecting and tracking a large number of moving objects in real time. This function is essential for navigating in real world environments, always subjected to rapid changes. The Laser Scanners provide an accurate 2-D picture of the environment because of theirs high angular and range resolution. Nevertheless, even using a high precision kind of sensor, the presence of differing surface reflectivity's, textures, relative orientations, etc can produce erroneous signals. These problems should be analyzed during the design of the sensor (Adams, 1999) or later, by the user, when processing the data. For the data acquisition procedure we used a real platform, a differential drive mobile robot, SmartROB-2, (see Figure 1-a and Badreddin, 1992) equipped with two SICK LMS200 Laser Scanners covering 360° with a resolution of 2° and radial error measurement less than ±20mm.



(a) (b)

Figure 1 – (a) Photo of the mobile robot SmartROB-2 equipped with two SICK LMS200 Laser Scanners and (b) plot of a typical laser scan data obtained by the robot during the experimental tests.

3. NAVIGATION CONTROLLER

The basic navigation controller used was previously described in Becker (1999). Nevertheless, a new Obstacle Avoidance procedure was adopted. It essentially consists of three components: an algorithm for Obstacle Detection, one for Motion Detection and Tracking (or simply Motion Estimation), and another one for computing Evasive Maneuvers, which is based on the *Velocity Obstacle* approach, Prassler *et al.*, 1999a, b (Figure 2).

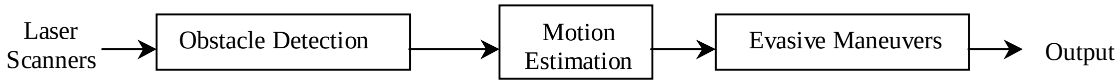


Figure 2 - Block diagram of the algorithms used for Obstacle Avoidance Controller.

3.1 Obstacle Detection and Motion Estimation

Based on the scan data, the robot can build a model of the environment and use this model to detect obstacles, motion, and to track the motion. However, how to represent and use these data in a real-time application? An efficient scheme for mapping scan data is the *Occupancy Grid Representation* (Elfes, 1989). In this approach, all the scan data are represented on a 2-D rectangular grid, where each grid element (cell) describes a small region of the real world. But, due to the necessity to completely initialize the grid and set each cell to some default value at step time *t*, this representation is too expensive in computer time. To avoid this problem and save computer time, a variation of this representation, called *Time Stamp Map* (Fiorini & Shiller, 1998), is used. In this case, only the occupied cells observed at each time step *t* are mapped, all the other cells in this grid remain untouched. This procedure allows the implementation in real-time, avoiding spend time mapping free space.

The easiest way to identify changes in the environment is to consider a sequence of the *Time Stamp Map*'s and to investigate where the steps of this sequence differ. A discrepancy between two subsequent steps is a strong indication of a potential modification in the environment. An important parameter to be considered is the distance from the bottom to the Laser Scanners position (i.e. the height of the sensor). Changing this position, the characteristics of the same objects is also changed. In the Figure 4 it is possible to observe the differences between two different positions of the Scanners (a- sensors positioned close to the ground and b- one meter of height). The method proposed to treat this problem is shown in the block diagram bellow (the double line blocks consider more than one *Time Stamp Map*):

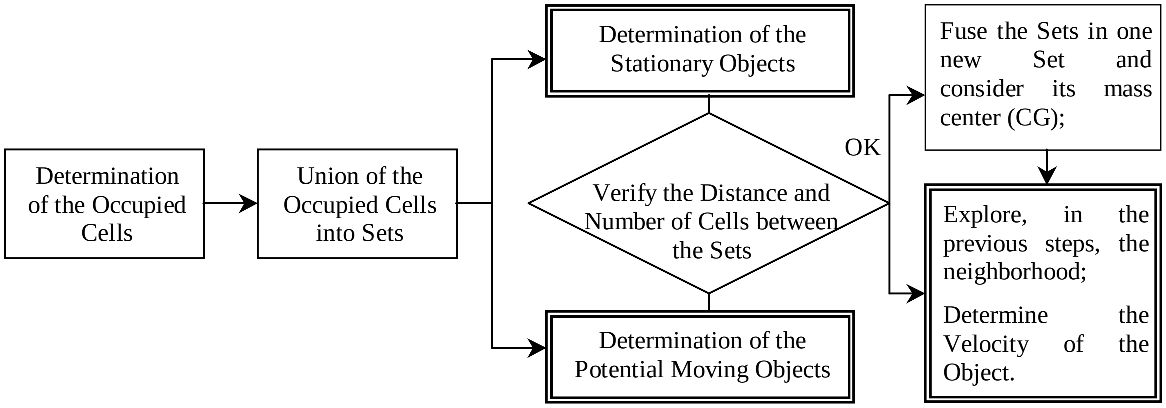


Figure 3 – Simplified Block Diagram of the method used to detect and track moving obstacles.

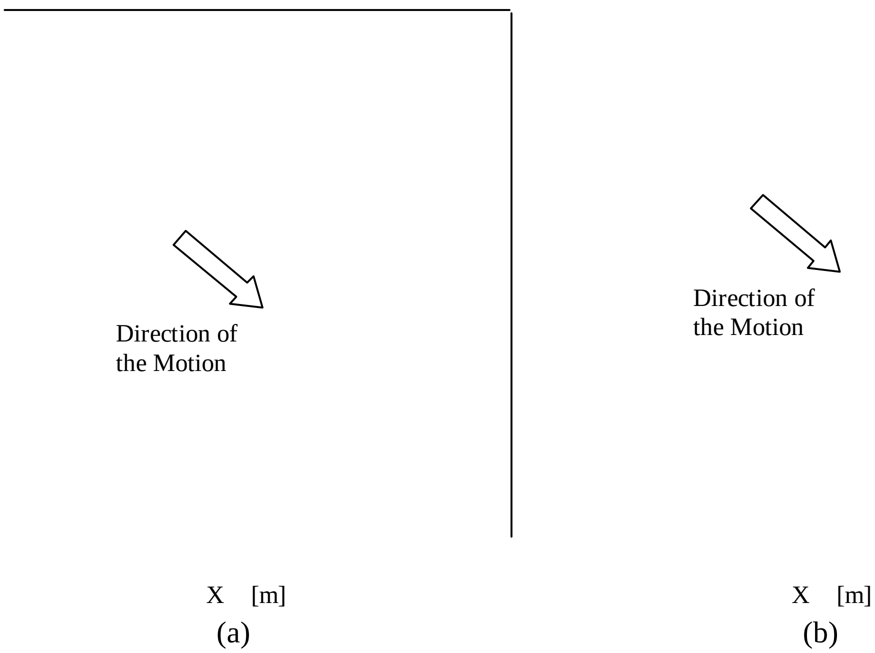


Figure 4 - Sequence of Scans of one person walking in a diagonal path: scanners located (a) one meter of height and (b) close to the ground. In both cases the robot was kept stationary.

Using the ideas proposed above, we obtained the following results:

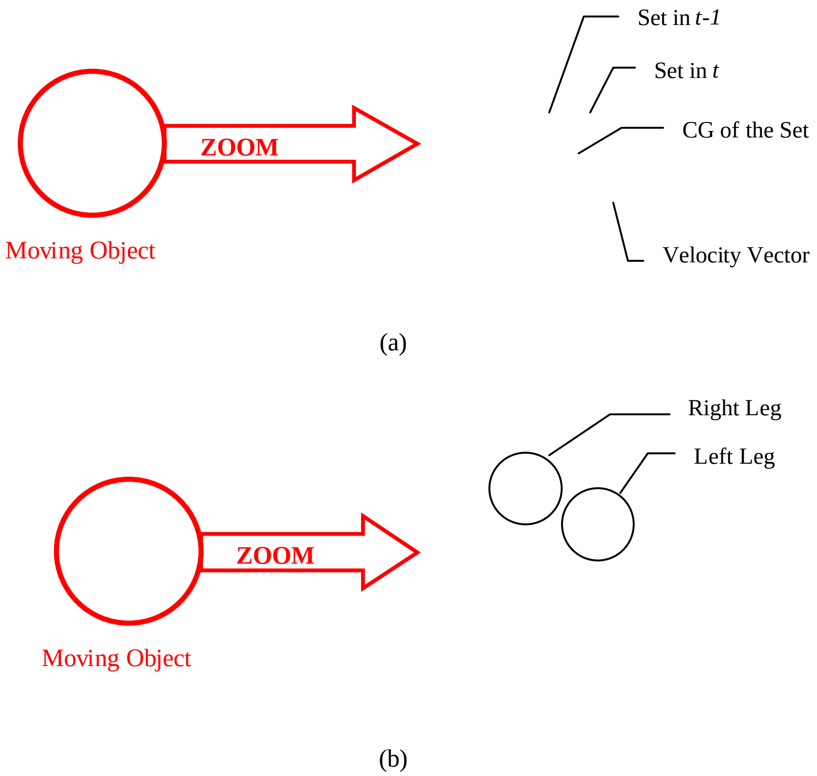
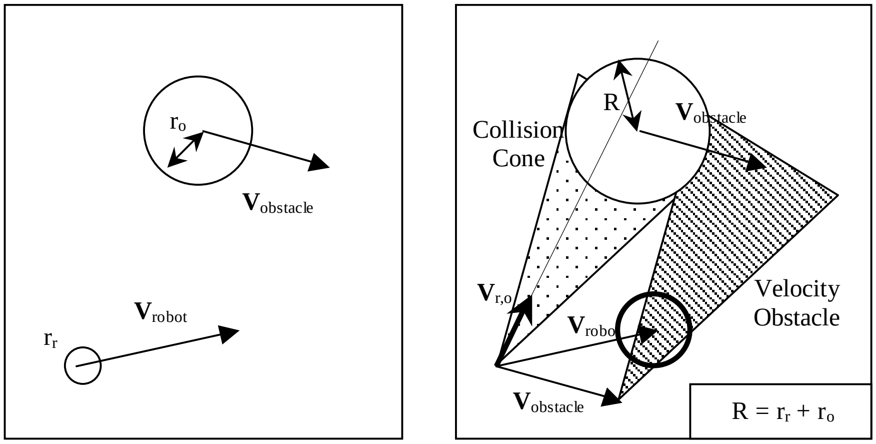


Figure 5 – Detection and tracking of one person walking in a diagonal path: scanners located (a) one meter of height and (b) close to the ground. In both cases the robot was kept stationary.

As shown in Figures 4 and 5, it is not easy to detect and track people walking (Kleistsch, 1998). Many times, in real environments, the informations provided by the Laser Scanners are not enough to avoid collisions with a lower height objects or objects that do not reflect the laser beam properly (e.g. glass doors).

3.2 Evasive Maneuvers

The basic idea of this algorithm is to obtain a "*collision cone*" and an equivalent condition on the absolute velocity of the robot ("*velocity obstacle*"), specific to a particular pair of robot / obstacle. The velocities of the robot inside of the *velocity obstacle* would cause collision between the robot and the obstacle, velocities outside of velocity obstacle would avoid collisions, and velocities on the boundaries of the velocity obstacle would result in the robot grazing the obstacle. Combining the results for each obstacle and the output of a navigation controller, it is possible to choose the better control output, which results in the robot going to the goal position and avoiding the obstacles. It is possible to see in Figure 6-b that for the actual velocity of the robot (*vector velocity* inside of the velocity obstacle area, see circle) the collision is imminent. Nevertheless, any velocity orientation that provides a position outside of the filled area would avoid the collision.



(a) (b)

Figure 6 - The relative velocity ($\mathbf{V}_{r,o}$), the Collision Cone and the Velocity Obstacle.

There are two options to avoid the collision: the first one is to change the orientation of the *velocity vector* (V_{robot}), steering the vehicle and, the second one, to change the modulus of the *velocity vector,* accelerating the vehicle (positive or negative acceleration). As one of the principles adopted for the controller behavior is to use a cruiser velocity during the path, the modulus of the *velocity vector* is changed only if the maximum steering angle is exceeded.

Based on the above premises, we used a circle with radius equal to the desired cruiser velocity to represent the set of possible orientation angles of the robot. Then, for each obstacle observed in the neighborhood, the *velocity obstacle* and the intersections between it and the circle are calculated. The results obtained for all obstacles are combined and result in the *Sets of Dangerous Orientation Angles* for the robot (Figure 7). In other words, if the orientation angle of the robot is inside of one of these sets, a collision between the robot and the obstacle will occur. Nevertheless, if the maximum steering angle is exceeded, the algorithm searches a new value for the modulus of the *velocity vector* that avoids the crash and does not exceed the maximum steering angle.

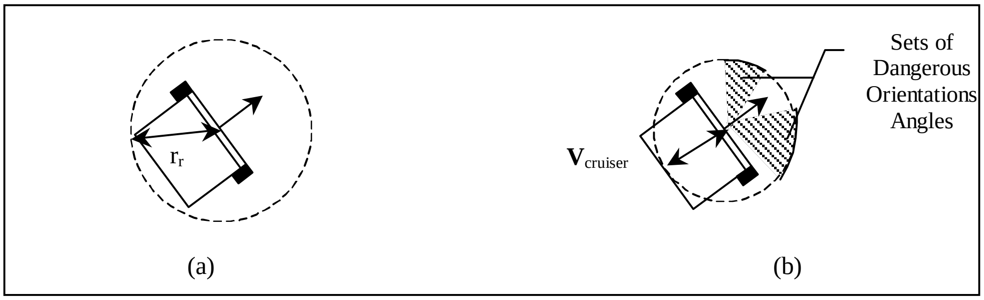


Figure 7 – (a) Detail of the robot radius and, (b) Sets of Dangerous Orientation Angles for the robot on the circle with radius equal to $\mathbf{V}_{\text{cruiser}}$.

To simulate the vehicle behavior in complex environments (with moving and fixed obstacles), the software Matlab was used. Simulations are made in an environment with 10 obstacles using the technique described above: motion planning using velocity obstacle. A cruise velocity is used during the path. The Figure 8 shows the environment used: the moving obstacles (O_1, O_2, O_4, O_5 and O_6), the fixed obstacles (O_3, O_7, O_8, O_9 and O_{10}), the initial and final positions, and orientations.

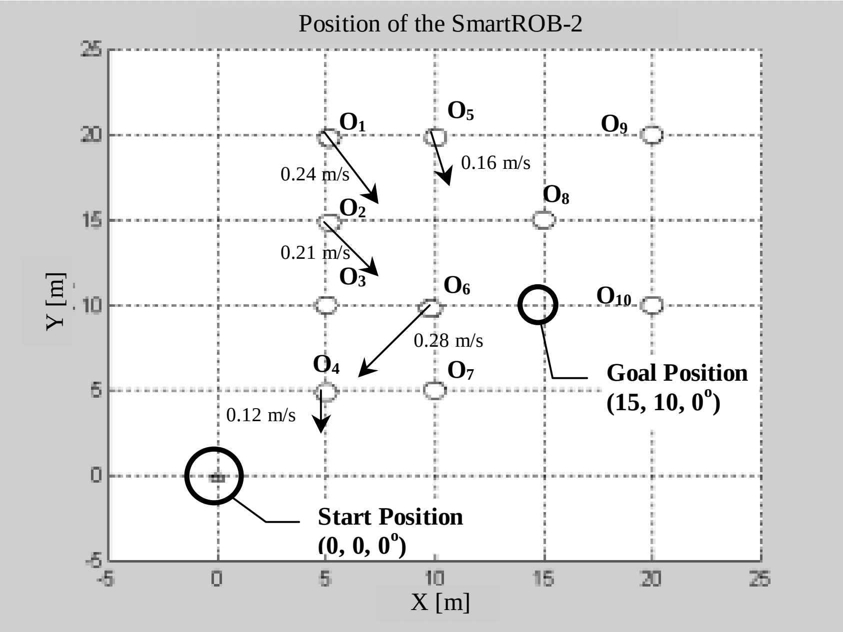
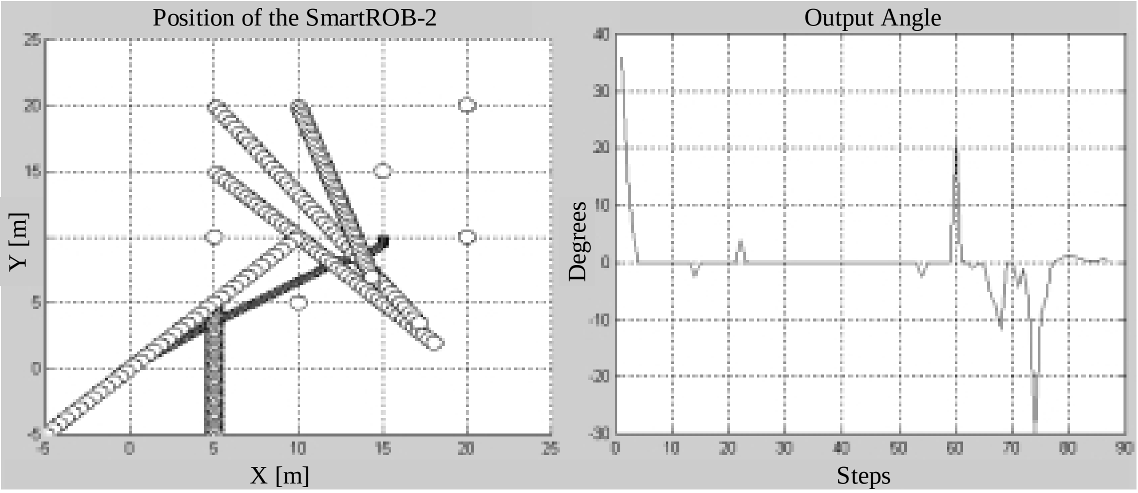


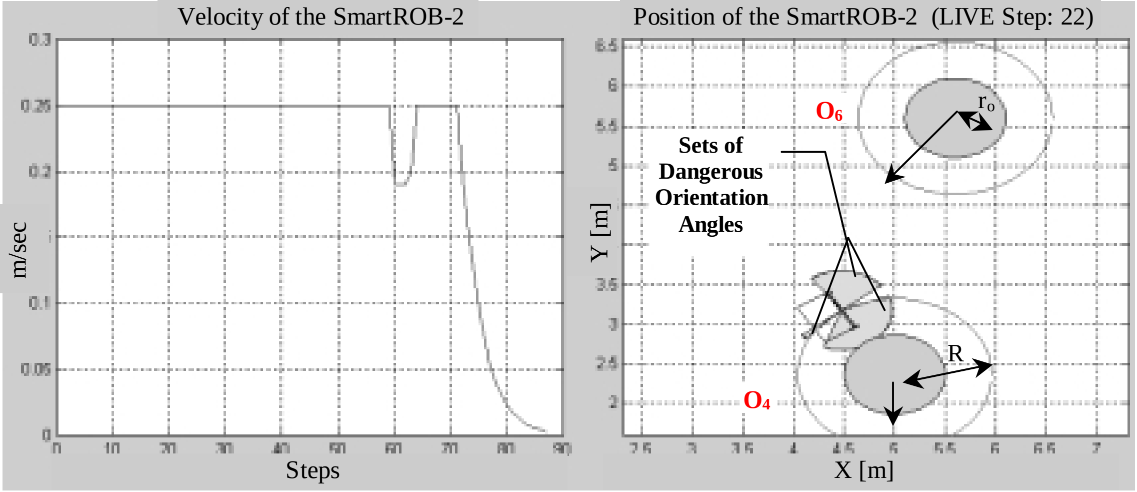
Figure 8 - Beginning of the simulation – environment with ten obstacles (five moving obstacles, their velocities are represented by vectors).

The results obtained using the obstacle avoidance controller were excellent. The capacity of the controller to predict and determine collision courses with several moving obstacles and compute an avoidance maneuver, which is close to its original heading as possible, allows the controller to avoid undesired and unnecessary accelerations and steering angles. It is possible to observe in Figures 9 and 10 that the vehicle moves close to the obstacles, grazing some of them. Only when the maximum steering angle is exceeded, the controller changes the cruise velocity to compute a new output angle. The behavior of the vehicle grazing the obstacles reduces the total displacement and the consumption of energy of the vehicle and increases its range.



(a) (b)

Figure 9 - (a) Paths simulated for the robot and the obstacles and (b) controller output angle.



(a) (b)

Figure 10 - (a) Velocity simulated for the robot and (b) detail of the sets of dangerous orientation angles (if the orientation angle of the robot is inside of these sets, the collision is imminent).

4. CONCLUSIONS AND OUTLOOK

We developed researches on Obstacle Avoidance for autonomous and assisted vehicles using a real sensor data and a mobile robot (SmartROB-2). The researches included simulations using the software Matlab, experimental tests using the mobile robot, and the real-time system XOberon.

The results obtained using the variation of the *Velocity Obstacle* approach were excellent, nevertheless it is necessary to do experimental tests to evaluate the behavior of the controller in real world environments. Tests about the position of the sensors on the robot, i.e., the height of the sensors, and its influence in the performance of the motion-tracking algorithm shown the efficiency of the algorithm developed. Due to the situations found in real world environments, we recommend the use of the Laser Scanners mainly to track moving obstacles and add Sonar or Vision sensors to detect objects not recognized by the Laser Scanners (e.g. lower height objects, glass doors, etc). The use sensor fusion would improve the performance of the algorithm.

ACKNOWLEDGMENTS

The authors wish to express their thanks to Prof. Gerhard Schweitzer and all the staff of the Institute of Robotics, ETH Zurich (Switzerland) for their contribution to this paper. The CNPq - National Research Council (Brazil) and the ICSC - World Laboratory (Switzerland) funded this study.

REFERENCES

Adams, M. D., 1999, “Sensor Modeling, Design and Data Processing for Autonomous Navigation”, World Scientific Series in Robotics and Intelligent Systems, Vol. 13, p.232.

Badreddin, E; 1992, “Mobile Roboter – Einführung in den Aufbau und die Funktionsweise Mobiler Roboter”, Kurs, June 15 – 16, 1992, IfR – ETH Zurich, Switzerland.

Baxter, J. W. & Bumby, J. R., 1995, “Fuzzy control of a robotic vehicle”, Proc. of Instn. Mech. Engrs. - Part I: Journal of Systems and Control Engineering, Vol. 209, pp. 79 - 91.

Becker, M., 1999, “Simulation of a fuzzy controller for autonomous vehicles in complex and dynamic environments”, Proc. of COBEM’99 – XV Brazilian Congress on Mechanical Engineering, Águas de Lindóia – SP, Brazil, Nov 22 to 26, 1999, CD-ROM.

Beom, H. R. & Cho, H. S., 1995, “A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning”, IEEE Trans. on Syst., Man, and Cyber., Vol. 25, n. 3, pp. 464 - 477.

Bobyr, S. & Lumelsky, V., 1999, “Control of dynamics and sensor based motion planning for a differential drive robot”, Proc. of the 1999 IEEE International Conference on Robotics and Automation, May 1999, Detroit – MI - USA, pp. 157 – 162.

Elfes, A., 1989, “Occupancy grids: a probabilistic framework for robot perception and navigation”, Ph.D. Thesis, Electrical and Computer Engineering Department / Robotics Institute, Carnegie-Mellon University, USA.

Feng, D. & Krogh, B. H., 1991, “Dynamic steering control of conventionally steered mobile robots”, Journal of Robotic Systems, Vol. 8, n. 5, pp. 699 - 721.

Fiorini, P. & Shiller, Z., 1998, “Motion planning in dynamic environments using velocity obstacles”, International Journal of Robotics Research, Vol. 17, pp. 760 - 772.

Fu, L.-C. & Liu, D.-Y., 1990, “An efficient algorithm for finding a collision-free path among polyhedral obstacles”, Journal of Robotic Systems, Vol. 7, n. 1, pp. 129 - 137.

Kleistsch, C., 1998, “Objekte / Menschen extrahieren aus Distanzdaten”, Semesterarbeit SS’98, IfR – ETH Zurich, Switzerland.

Prassler, E., Scholz, J. & Fiorini, P., 1999a, “Navigating a robotic wheelchair in a railway station during rush hour”, International Journal of Robotics Research, Vol. 18, pp. 711 - 727.

Prassler, E., Scholz, J. & Elfes, A., 1999b, “Tracking people in a railway station during rush hour”, Proc. of 1st Int. Conf. on Computer Vision Systems (ICVS’99), Gran Canaria, Spain, pp. 162 – 179.

Wang, L.-L. & Tsai, W.-H., 1991, “Collision avoidance by a modified least-mean-square error classification scheme for indoor autonomous land vehicle navigation”, Journal of Robotic Systems, Vol. 8, n. 5, pp. 677 - 698.

Zapata, R., Lépinay, P. & Thompson, P., 1994, “Reactive behaviors of fast mobile robots”, J. of Robotic Syst., vol. 11, n. 1, pp. 13 - 20.