ANNALS of the ORADEA UNIVERSITY.
Fascicle of Management and Technological Engineering, Volume XI (XXI), 2012, NR3

NAVIGATION SIMULATION FOR AN AUTONOMOUS ROBOT

ŢARCĂ Radu, CORNEA Mihai, VESSELENYI Tiberiu, ŢARCĂ Ioan
University of Oradea, rtarca@uoradea.ro

Keywords: autonomous robot, navigation algorithm, simulation.

Abstract: The work presented in this paper pretends to provide a navigation algorithm for an autonomous robot which is moving in an environment, in this case, is that the experiment uses grid lines painted on the floor, for robot navigation purpose.

1. INTRODUCTION

Form computational point of view navigation and control algorithms are currently using either probabilistic or soft computing approaches.

A good example of probabilistic approach is given in [4]. This study proposes a new hierarchical formulation of Partially Observable Markov Decision Processes (POMDPs) for autonomous robot navigation that can be solved in real-time, and is memory efficient.

Soft computing methods are usually employed because their capacity to solve nonlinear dynamic problems. Among them fuzzy logic method is one of the most widely used methods, for example, a novel technique is used to autonomously select different motor schemas using fuzzy context dependant blending of robot behaviours for navigation [7] and also in [10] a switching fuzzy logic controller for mobile robots with a bounded curvature constraint is presented.

In addition to fuzzy logic methods, a large variety of neural structures, such as neuro-fuzzy approaches [6], nonlinear neural lattices [2], Cellular Neural Networks (CNN) processing techniques [5] and behaviour-based artificial structures [8] have been studied.

In addition, evolutionary robotics (ER) is a field of research that applies artificial evolution to the automatic design and synthesis of intelligent robot controllers. Biologically inspired distributed intelligence algorithms gained more popularity recently [3] competing with neural and evolutionary methods.

The goal of our research team is to develop a navigation algorithm for an autonomous robot system to be able to explore a dynamic environment. The particularity of the environment, in this case, is that the experiment uses grid lines painted on the floor, for robot navigation purpose. For this a model for the robot and its environment was developed to realise the robot navigation simulation.

2. THE SYSTEM DESCRIPTION

The system developed by us consists in an autonomous robot, its environment (a gridded flour) containing obstacles in which the robot system works and a CCD camera which observes the scene.

An image of the environment (including obstacles) in which the robot system works is presented in figure 1.

In the central points of some squares, obstacles are placed on the gridded table, as it is shown in figure 1. The scene is observed by a web camera.

The first sequence of tasks of the system is to acquire the scene image with the objects, to calibrate the image and to indentificate the robot and also extract the obstacles locations.

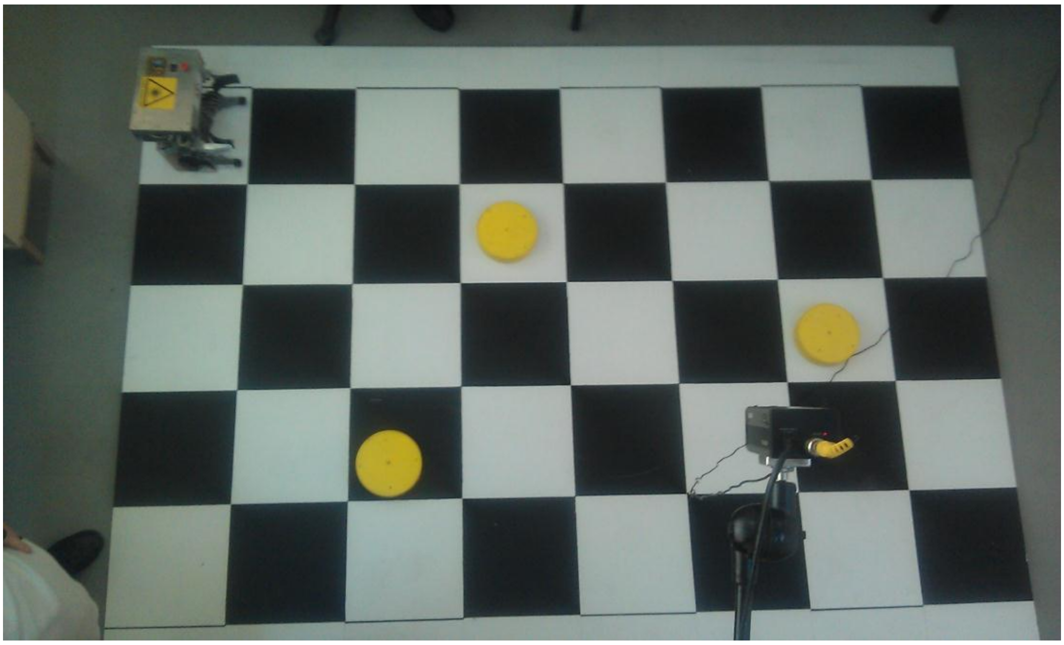

Fig.1 The system components.

If an obstacle is placed on the table after the process started, the robot is able to locate that obstacle and to take it into account while applying the strategy to reach the target object.

The autonomous robot is presented in figure 2.

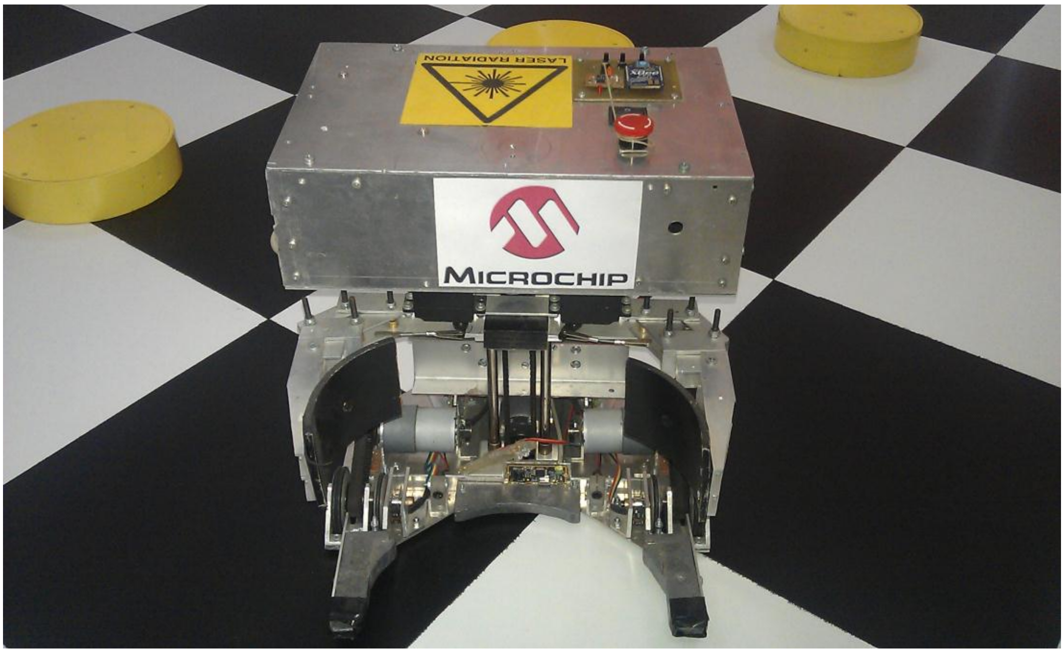

Fig 2. The autonomous robot

The robot is equipped with three distance sensors used for obstacle detection, placed one in front and the other two on the left and right side of the robot.

The sensors are based on laser triangulation realized with webcams. Because of the fact that the front lasers are close enough, a single webcam can be used to measure the distance to the laser spots.

Fig 3. The front distance sensors

The front distance sensor is presented in figure 9 and it consists in a CCD camera and two lasers which are placed at a distance equal with the obstacles diameters.

The right side distance sensor is presented in figure 10 and it consists in a CCD camera and one laser distance sensor. The left side distance sensor is similar with the right side distance sensor.

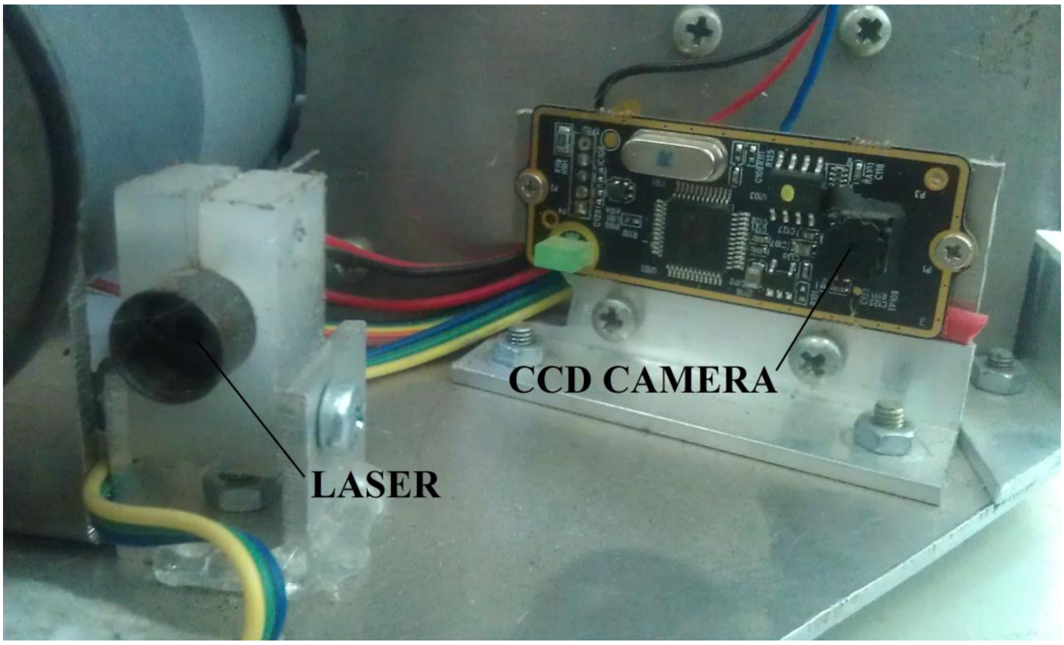

Fig 4. The right side distance sensors

The distances measured by sensors have been used to determine the position of the obstacles (in this case cylinders of 200 X 50 mm) on the table. This is why we needed a precise measurement range (in respect to the measurement area, which has to be as small as possible).

3. ENVIRONMENT MODELS FOR ROBOT NAVIGATION SIMULATION

Environment modeling is inherent for robot navigation algorithm design, since the design is a continuous testing and adjustment cycle. Even after the algorithm is ready to use, some kind of environment model is still required for the navigation map.

A good example for software simulation of navigation problems is given in [1], where both classical and fuzzy based algorithms of a mobile robot avoiding obstacles in a static environment can be tested. In [11] the researchers focused on distributed coverage of environments with unknown extension using a team of networked miniature robots. Recurrent neural networks are used in [12] in order to make a topological description of the environment.

For autonomous robot control, many researches are using simultaneous localization and mapping (SLAM) systems. In [9], the control system is hybrid in nature and tightly coupled with the SLAM system, using a combination of high and low level deliberative and reactive control processes to perform obstacle avoidance, exploration and global navigation.

In our case we developed a software for the robot and its environment to simulate the robot navigation algorithm in its environment and to visualize it was realized a graphical interface as is presented in figure 5.

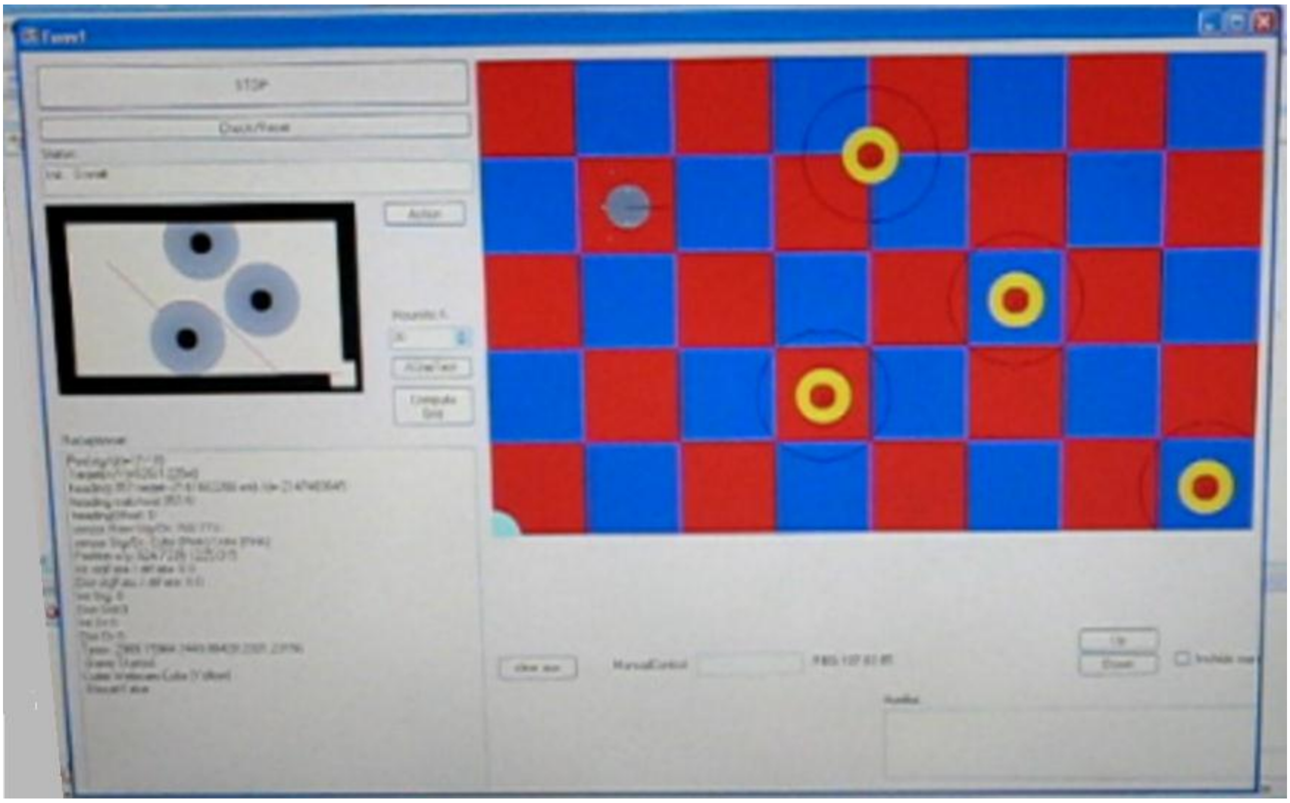

Fig 5. The graphical interface of the software developed by our team

Each obstacle is posed on the grid table model according with its real location determined by CCD camera (in the real time interface figure 6, the obstacles are designed with yellow with an red mark). The initial robot position is identified also by the CCD camera and its model (represented by a grey circle) is placed in that location on the table model.

The operator has the possibility to select from the object the target one. (let suppose that in this case the target was selected the object from right-down corner).

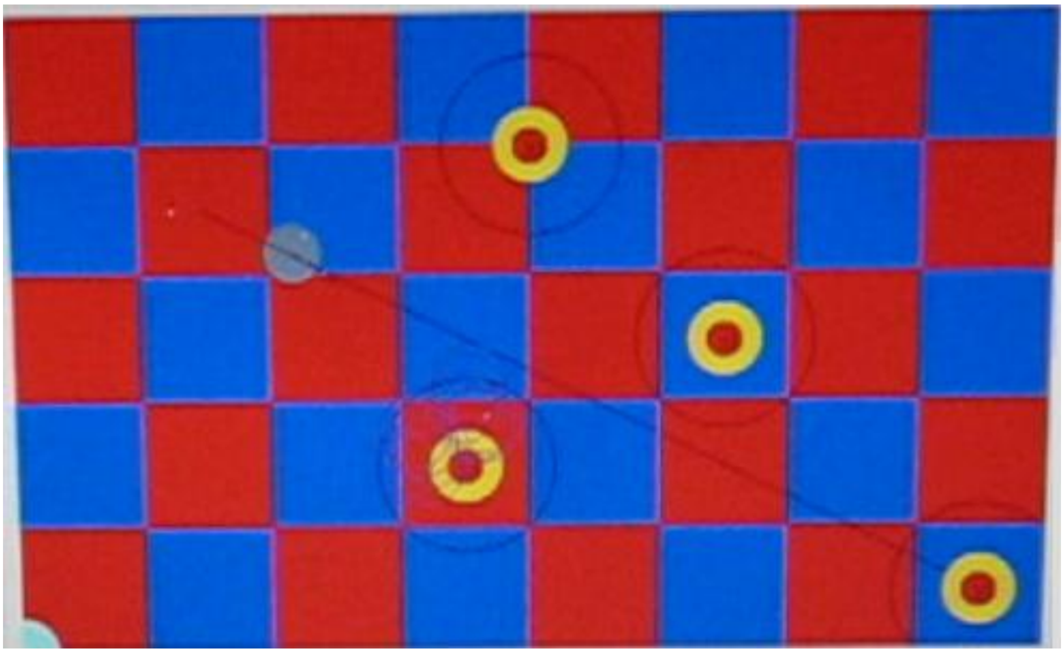

Fig 6. The real-time graphical interface for the robot and its environmental model

The path calculation for the autonomous robot form its initial location to reach the target object, to prehense it and to return in another location is done with the A* algorithm, using the Manhattan heuristic formula.

The A * algorithm uses a heuristic based on the type of distance + cost (usually denoted f (x)) to determine the order of visiting the nodes. Heuristic is the sum of two functions g (x) - cost function calculated from the initial node to the current node and h (x) - based on a heuristic that estimates the distance from the current node to target node. Value given by the function h (x) at a time must be less than or at most equal to the value given by f (x). If heuristic is well chosen, the A * algorithm guaranteed to find the optimal solution. Time complexity of the algorithm A * is dependent heuristics used. When calculating the optimal path in road applications function h (x) is an optimal heuristic because it is a Euclidean distance (length of a straight line connecting the current node to target node) and always has a value less than f (x). Thus, the complexity of the problem is polynomial rather than exponential as with other search algorithms.

Implementation is also done with “river crossing” so that for the case in which no path can be calculated without hitting an obstacle, the robot can force its way pushing the obstacles, so that he will follow the least resistance path.

After the robot reaches the target, it will collect the object, and then, using the same navigation algorithm, it will transfer the collected object to the delivery location – defined at the beginning by the human operator.

4. CONCLUSION

In this paper we have presented a robot navigation algorithm and simulation system capable to execute tasks in a dynamic environment. The robot navigates in the environment using intelligent algorithms allowing it to set up paths, recognize and avoid obstacles.

The navigation algorithm was realized using the A* algorithm, based on Manhattan heuristic formula.

Bibliography:

1. **Abiyev et al,** Navigation of mobile robots in the presence of obstacles, Advances in Engineering Software, vol. 41, 2010, pp. 1179–1186.
2. **Arena et al** Cellular Nonlinear Networks for the emergence of perceptual states: Application to robot navigation control, Neural Networks 22, 2009, pp. 801-811.
3. **Castillo et al,** Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot, Information Sciences, Vol. 192, 2012, pp. 19-38.
4. **Foka et al** Real-time hierarchical POMDPs for autonomous robot navigation, Robotics and Autonomous Systems, vol. 55, 2007, pp. 561–571.
5. **Gavriluţ, I, et all,** Motion planning for two mobile robots in an environment with obstacles by using cellular neural networks, Proceedings of the IEEE International Symposium on Signals, Circuits and Systems (ISSCS 20005), ISBN 0-7803-9029-6, Iaşi, pp. 801-804, 2005.
6. **Hui et al,** Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches, Fuzzy Sets and Systems 157 (2006) pp. 2171 – 2204.
7. **Huq et al,** Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation, Applied Soft Computing, vol. 8, Iss.1, 2008, pp. 422–436.
8. **Manzotti et al,** From behaviour-based robots to motivation-based robots, Robotics and Autonomous Systems 51, 2005, pp. 175–190.
9. **Milford et al,** Hybrid robot control and SLAM for persistent navigation and mapping, Robotics and Autonomous Systems, vol. 58, 2010, pp. 1096-1104
10. **Moustris et al,** Switching fuzzy tracking control for mobile robots under curvature constraints, Control Engineering Practice, 19, 2011, pp. 45-53.
11. **Rutishauser et al** Collaborative coverage using a swarm of networked miniature robots, Robotics and Autonomous Systems, vol. 57, 2009, pp. 517-525.
12. **Ster et al,** An integrated learning approach to environment modelling in mobile robot navigation, Neurocomputing, vol. 57, 2004, pp. 215 – 238.
13. **Tarca R et all**, Designing A Networked Telerobotic System Using Internet, Proceedings of the ASME 2012 11th Biennial Conference On Engineering Systems Design And Analysis, ESDA2012, pp. 353-361.