Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Procedia Social and Behavioral Sciences 2 (2010) 565–571

Procedia
Social and Behavioral Sciences

WCES-2010

Teaching robot navigation in the presence of obstacles using a computer simulation program

B. Erin[a], R. Abiyev[a], D. Ibrahim[a] *

[a]Near East University, Faculty of Engineering, Departtment of Computer Engineering, Lefkosa 98010, TRNC

Received October 6, 2009; revised December 15, 2009; accepted January 4, 2010

Abstract

In this paper, an educational software tool called *EDURobot* has been developed to enhance the understanding of robotics for undergraduate and graduate students of computer and electrical and electronic engineering departments. The software tool mainly teaches students the navigation problems of a mobile robot avoiding obstacles in a static environment using different algorithms. The simulation environment is of a menu-driven one where students can draw obstacles of standard shapes and sizes and assign the starting point of the mobile robot. The robot will then navigate among these obstacles without hitting them and reach the goal point given by the user. Parameters associated with the different algorithms may also be changed to observe their effects which will further enable comprehension of characteristics of different path planning algorithms.

Keywords: Robotics education; robot navigation; obstacle avoidance; path planning; mobile robot algorithm.

1. Introduction

In engineering education one of the basic problems is understanding of theoretical knowledge gained in engineering courses and its use in practice. For this purpose, various experimental exercises or real physical training systems are being developed for education of students. Because of the use of special hardware resources and their expensive nature the design of physical systems have become difficult. In this paper the development of an educational tool for robotics courses is considered.

The computer program which we are introducing in this paper serves the purpose of increasing the understanding of students as to how robots can move in a predetermined environment avoiding obstacles of standard shapes and sizes utilizing different navigation and path planning algorithms. The objective of the software is to let the students understand the main idea behind some of the well known robot navigation algorithms and the differences between them. Different types of commercially available computer simulators are available in all branches of science and

* D. Ibrahim. Tel.: +0-90392-2236464; fax: +0-903922236627
E-mail address: dogan@neu.edu.tr.

1877-0428
doi:10.1016/j.sbspro.2010.03.064

engineering education. Some examples are, electronic simulation package *B^2 Spice* (B2Spice), microcontroller electronic simulator *Proteus* (Proteus), a chemical engineering simulator *MDDS* (MDDS), a graphical simulation software for complex dynamic systems *VisSim* (Vissim), a powerful mathematical tool for simulating all types of complex engineering systems *MATLAB* (Gilat; Hahn & Valentine; Chapman), and many more.

The use of mobile robots in robotics education has gained popularity over the last two decades. Different applications can be discussed for mobile robot navigation. Examples of such applications are given by Das & Kar (2006) and Dongbing & Housheng (2006) and include trajectory tracking, target tracking (Luo, Chen, and Su), obstacle avoidance (Ye, Yung, and Wang; Fairchard and Garnier; Hurley, Xu, and Bright), landmark recognition systems and soccer robot navigation for mobile robot navigation (Vadakkepat, Miin, Peng, and Lee)].

Many educational institutions are now offering courses in robotics. For example, Carnegie Mellon University (CMU) offers undergraduate, as well as MS and PhD level courses in robotics (Carnegie, 2009). A summer camp program, *RoboCamp* (NREC, 2009), is offered by the *National Robotics Engineering Consortium* with the aim of teaching the practical aspects of building and programming mobile robots during the summer vacation. Another summer vacation educational robotics program is the *Andrew's Leap* (Andrew, 2009), run by the CMU with the aim of teaching the programming of mobile robots to students. In addition to teaching, many colleges and universities organize robot competitions, such as robot navigation and speed competitions (MIT, 2009), maze navigation (Verner and Ahlgren) competitions and so on.

Robotics simulation is not new and several educational establishments with varying capabilities have been developed by various researchers in the past in order to inspire students' interest in mobile robotics and motivate them to participate actively in the learning process. The educational robotic system presented by Khamis et al. (2006) has a multi-layered architecture. The educational mobile robot platform *MBR-01* (Gunes and Baba) was designed as an educational tool for the course 'Control and Robotics'. Its aim is to teach the fundamental skills of control systems, autonomous robots, FLC, image processing, edge detection, RF communication techniques and internet based remote control. This simulator gives students the ability to control the speed and position of an autonomous mobile robot, thus providing experience of using a real environment situation. The next system *RoboLab* (Torres et al., 2006) allows students to work with a simulation of an industrial robot and carry out operations with a robot. Students perform exercises on the simulated virtual environment and then, after checking that the results are correct, they can execute them in the real system. Students are able to practice and carry out correct movement sequences. A computer program named *RoboKol* is introduced by Conkur (2006), which performs path planning for redundant manipulators and mobile robots using potential field based algorithms. Through user interface, the user can interact with the program and can perform a variety of operations such as drawing obstacles and robots on the screen, obtaining two- and three-dimensional images of the potential field and performing robot simulations.

In this paper three different algorithms, potential field, vector field histogram, and local navigation, have been implemented for robot navigation problems. The development of such a system will allow students to better understand the problems of obstacle avoidance and their solution mechanisms.

2. Mobile Robot Navigation Algorithms

Navigation is one of the basic problems in robotics. The research paper by Fujimori et al., (1997) classifies the robot navigation algorithms as global and local, depending on surrounding environment. In global navigation, the environment surrounding the robot is known and the path which avoids the obstacles is selected. Here graphical maps which contain information about the obstacles are used to determine a desirable path (Khatip; Iwan and Borenstein; Brooks; and Brooks and Connell). The global navigation problem is solved using different path planning algorithms. In local navigation, the environment surrounding the robot is unknown, or partially known, and sensors are used to detect the obstacles and collision avoidance system is incorporated into the robot to avoid these obstacles.

In this paper global navigation using path planning algorithms and local navigation algorithms are modelled for mobile robot to avoid obstacles. Path planning methods for mobile robots is based on the idea of finding a optimal and smooth path consisting of many points close enough to each other avoiding obstacles. The software which is the subject for this paper utilizes three methods as described below:

2.1. Potential field method

Robot dynamic characteristics and navigation law are important in path planning, where information about location of obstacles is used to determine the desirable path. One of the path determination methods used in the simulator software which is the main topic of this paper is the ***potential field*** method (PFM) described by Khatip (1985). The philosophy of the potential field approach is that the mobile robot moves in a field of forces. The goal position to be reached is an attractive potential while each obstacle generates a repulsive potential. A potential field can be viewed as an energy field and so its gradient at each position is a force. The idea of obstacles exerting virtual repelling forces towards a robot, while the target generates a virtual attractive force uses a similar concept that takes into consideration the robot's velocity in the vicinity of obstacles. In one example called the Brooks implementation (Brooks, 1986), if the magnitude of the sum of repulsive forces exceeds a certain threshold, the robot stops, turns into the direction of the resultant force vector, and moves on. The theory of the potential field method is given below briefly.

A potential, $\phi(r)$, is defined by the Laplace equation $\nabla^2\phi = 0$ in a closed region, Ω, of continuous, equal connectivity. The boundary of Ω, $\partial\Omega$, does not have to be connected. It includes the surfaces of all obstacles and the goal point. $\phi(r) = \phi_1$ at the surfaces of obstacles and $\phi(r) = \phi_0$ at the goal point. There are no local minima on $\phi(r)$. Nevertheless, the exponential decay of the field from any point leads to areas where the magnitude of the gradient on ϕ, $|\nabla\phi|$, is very small while the range of $|\nabla\phi|$ may be very large. The field decays rapidly near the goal, and far from the goal there is only a slight change in the field. The Laplace equation in two dimensions is represented on equally spaced and connected grid by the following partial differential equation:

$$\phi_{(i,j)} = \frac{\phi_{(i+1,j)} + \phi_{(i-1,j)} + \phi_{(i,j+1)} + \phi_{(i,j+1)}}{4} \qquad (1)$$

where i is position on the grid in the x direction, j is position on the grid in the y direction. Field values are calculated for any point in the workspace by using linear interpolation. For detailed description the reader for PFM method can refer to Conkur (2006), Atsushi et al., (1997), and Khatip (1985).

2.2. Vector Field Histogram Plus Method

The Vector Field Histogram Plus (VFH+) method described by Ulrich & Borenstein (1998) includes four stages for computing direction of robot motion. In first three stages the two-dimensional map grid is transformed into one-dimensional polar histograms. These are implemented using primary polar histogram, binary polar histogram and masked polar histogram. In last stage, using masked histogram and cost function the algorithm selects the suitable direction for the robot. The brief description of these stages is given below.

In VFH+ method, there exists a circular window with diameter w_s where the robot scans it environment. This forms our histogram grid which has dimension w_s x w_s. In this histogram grid, each cell has a certainity value $c_{i,j}$ which has value 1 where we are confident there exists part of an obstacle and has value 0 where there is no obstacle. In developed algorithm where obstacles are represented as rectangles and ellipses in a static environment, the circular window mentioned above is obtained from a two-dimensional array called savepoint which holds the information whether each cell is part of an obstacle or not. Next steps are the building of primary polar, binary and masked histograms. Using masked histogram in last stage the selection of the new steering direction is carried out.

2.3. Local Navigation Method

A local navigation method (LNM) with obstacle avoidance is considered for mobile robots in which the dynamics of the robot are taken into consideration (Atsushi et al., 1997). The goal is known but the geometry and the location of the obstacles are unknown. The mobile robot position is represented by the Cartesian coordinates and can move in three directions, forward, left or right. The starting point and goal points of robot are given. Using these points the directional angle of robot $\theta(t)$ ($0 \le \theta(t) \le 2\pi$) is determined. There may be obstacles in the plane of motion and the objective is to navigate the robot to the goal avoiding the obstacles. To determine optimal path the following navigation law is used.

$$\dot{\theta}(t) = -\eta[\theta(t) - \theta^*(t)] \qquad (2)$$

here, $\theta(t)$ is current directional angle of robot, $\theta^*(t)$ is desirable path, η is positive constant.

The problem is to navigate the robot to its goal by avoiding obstacles by witching $\theta^*(t)$ based on the information available from three- left, centre and right distance sensors. The detailed description of the local navigation is given by Ulrich & Borenstein (1998).

3. Simulation Results

The program menu includes File, Edit, Draw, Simulate, Settings, Parameters, and Help pull-down menus. During a typical run of the program different obstacles can be created, and starting and goal positions of the robot can be selected. Then, an algorithm is selected and path planning or navigation of the mobile robot is performed. In all of the software runs below, the gridSize has been taken as 1.0 and linelength has been taken as 0.1. gridSize is the length and width of each cell on the grid.

Fig. 1 and Fig. 2 show typical runs where the parameters are changed from their default values. The first figure shows the results of the three algorithms before the parameters were changed and second figure shows the situation after the parameters were changed. What has happened is that in potential field method the robot was out of line as the gridSize was increased to 5.00 from 1.00 and the time increased twofold as the number of iterations was increased from 1000 to 1500. For Vector Field Histogram plus the workspace size(w_s) was increased from 12 to 22 which dramatically increased the ***time taken to reach goal*** from 11325 msecs to 24070 msecs. The fact that we increased minDistance from 1 to 10 has taken the path away from the obstacles. The other parameters for VFH+ remained constant (namely b and alpha). Finally there is the Local Navigation parameters where the parameter d_{max} has been increased from 4 units to 8.5 alpha for local navigation had to be increased anyways from 30 to 85 degrees to fix the path. The change was again slight movement of the path away from the obstacles.

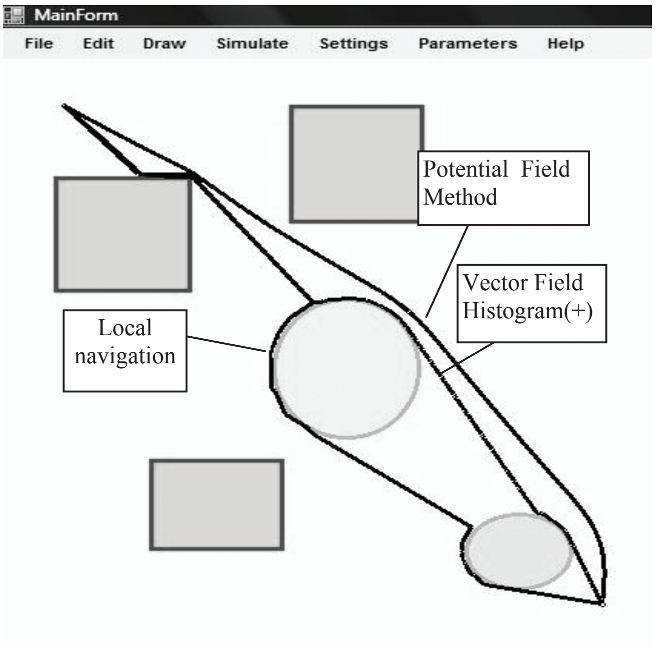

Figure. 1: Graph before parameters were changed

(The parameters are: **PFM:** Number of Iterations: 1000, gridSize = 1.00, **VFM(+):** w_s = 12, minDistance = 1, b = 1.5, alpha = 2.0, **LNM:** d_{max} = 4 units, alpha = 85 degrees - it had to be fixed from 30 degrees to get the correct path, linelength=0.1)

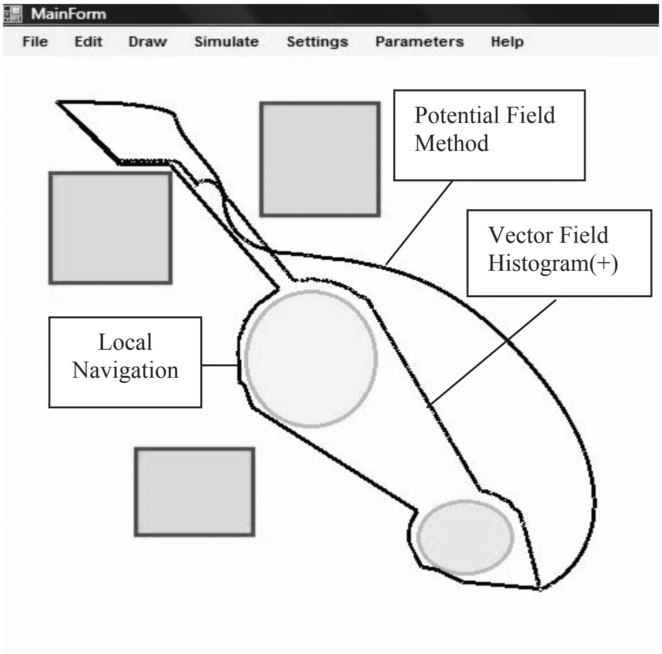

Figure. 2. Graph after parameters are changed

(The parameters are: **PFM:** Number of Iterations: 1500, gridSize = 5.00, **VFH(+):** w_s = 22, minDistance = 10, b = 1.5, alpha = 2.0, **LNM:** d_{max} = 8.5 units, alpha = 85 degrees - it had to be fixed from 30 degrees to get the correct path, linelength=0.1)

Table 1 Statistics for Fig. 1

Algorithm	Time taken to reach goal (ms)	Number of points generated
PFM	2682	5431
VFH+	1325	5538
LNM	764	6304

Table 2 Statistics for Fig. 2

Algorithm	Time taken to reach goal (ms)	Number of points generated
PFM	45396	6436
VFH+	24070	5779
LNM	1544	6465

4. Case study – opinion of students

A pilot study was carried out at the Near East University to find out the opinions of students to using the EDURobot simulation program. The study consisted of carefully prepared questionnaire, completed by undergraduate students who used the EDURobot simulation software as part of their normal lecture sessions during one month. The aim of the questionnaire was to learn the opinions of students about the usefulness of the system.

Fourty-four questions from eight subjects were prepared using the Likert-5 scale. An extract from the questionnaire is shown in Table 4. The participants were given time to complete the questionnaires at the end of their training. Training includes the presentation of theoretical and practical sections. In theoretical section the navigation problem of mobile robots, the methodologies used for navigation were explained. After the theoretical section students attended their laboratory sessions. In these laboratories the EDURobot simulation software was used with different parameters. After completion of practical section the evaluation of the simulator software was done by students by completing the questionnaire. The results of the questionnaire were analysed using the SPSS statistical package and below is a summary of the important outcomes:

- A majority of students (85%) strongly agree that the EDURobot is easy to use and is user friendly.
- Over 90% of students agreed that the effects of parameter changes can easily be observed.
- A majority of students (89%) agree that the simulator has helped them to understand the functional differences between the three different algorithms.
- Over 94% of students agree that they were able to thoroughly understand the PFM, VFH+ and LNM algorithms.
- Finally, all of the students agree that a computer simulation is an effective way for them to learn.

5. Conclusions and suggestions for future work

The development of a computer simulation program called EDURobot for navigation of mobile robots in the presence of obstacles has been described. The objective of the software was to improve students understanding of robotics navigation. EDURobot is an interactive program and is used successfully in laborotary sessions at the Near East University. Various parameters of the algorithms can be adjusted by students and this helps students to compare the advantages and disadvantages of the algorithms. Result show that the software tool has increased students knowledge and understanding of robotics and gave them a better insight into the various robotic path planning and navigation algorithms. Considering the interest and enthusiasm of students it is planned to include the developed tool as a permanent experiment of the undergraduate robotic laboratory work.

References

Andrew's Leap, http://www.cs.cmu.edu/~leap [Accessed on August, 2009]

Brooks, R.A., (1986). A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, 14-23

Brooks, R.A., and Connell, J.H. (1987). Asynchronous Distributed Control System for a Mobile Robot. *Proceedings of the SPIE*, Vol.727, 77-84.

B2Spice electronic Simulation Software, Beige Bag Software, http://www.beigebag.com [Accessed on: June, 2009]

Chapman, S.J. (2007). *MATLAB Programming for Engineers*, CL-Engineering, (4h Ed).

Conkur, E.S. (2006). RoboKol: A computer program for Path Planning for Redundant and Mobile Robots. *Computer Applications in Engineering Education*, 14(3): 198-210.

Das, T and I.N. Kar, I.N. (2006). Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots, *IEEE Transactions on Control Systems Technology*, 14, 501-510.

Dongbing, G and Housheng, H. (2006). Receding horizon tracking control of wheeled mobile robots, *IEEE Transactions Control Systems Technology*, 14, 743-749.

Fairchard, T., and Garnier, P. (2001) Fuzzy control to drive car-like vehicles, *Robotics and Autonomous Systems*, 34, 1-22.

Fujimori, A. Nikiforuk, P.N., and Gupta, M.M. (19977). Adaptive Navigation of Mobile Robots with Obstacle Avoidance. *IEEE Transactions on Robotics and Automation*, Vol.13, No.4.

Gilat, A. (2008). *MATLAB: An Introduction with Applications*, Wiley.

Gunes, M., and Baba, A.F. (2006). An Approach for Building Innovative Educational Environments for Mobile Robotics, *International Journal of Engineering Education*, Vol. 22, No. 4, 732-742.

Hahn, B. and D. Valentine, D. (2007). *Essential MATLAB for Engineers and Scientsts,* Newnes.

Hurley, M.D., Xu, W.L., and Bright, G. (2005) Implementing fuzzy logic for machine intelligence: a case study, *International Journal of Engineering Education*, 21(5), 178-186.

Khamis, A., Rodriguez, F., Barber, and Salichs, M.A. (2006). An Approach for Building Innovative Educational Environments for Mobile Robotics, *International Journal of Engineering Education*, Vol. 22, No. 4, 732-742

Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. *Proceedings of the IEEE Conference on Robotics and Automation*, 500-505.

Luo, R.C., Chen, T.M., and Su, K.L. (2001). Target tracking using hierarchical grey-fuzzy motion decision-making method, *IEEE Transactions on Systems, Man. and Cybernetics-Part A: Cybernetic*, 13, 179-186.
MDDS Simulator, http://dambiren.tripod.com/ mddssimSetup.zip [Accessed on September,2009-08-31
National Robotics Engineering Center, http://www.rec.ri.cmu.edu [Accessed on: July, 2009]
MIT Autonomous Robot Design Competition, http://web.mit.edu/6.270/www [Accessed on August, 2009]
Proteus Simulation Software, Labcenter Electronics, http://www.labcnter.co.uk [Accessed on: June, 2009]
Robotics Institute of Carnegie Mellon University, http://www.ri.cmu.edu [Accessed on: July, 2009]
Sokolowski, J.A., and Catherine, M.B. (2009). *Principles of Modeling and Simulation: A Multidisciplinary Approach*, Wiley.
Torres, F., Candelas, F.A., Puente, S.T., Pablo Gil, J.P.P., and Ortiz, F.G. (2006). Experiences with Virtual Environment and Remote Laboratory for learning Robotics at the University of Alicante, *International Journal of Engineering Education*, Vol.22, No.4, 732-742.
Ulrich., I and Borenstein, J. (1998). VFH+: Reliable obstacle avoidance for fast mobile robots, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, 1572-1577.
Vadakkepat, P., Miin, O.C., Peng, X., and Lee, T.H. (2004). Fuzzy behaviour-based control of mobile robots, *IEEE Transactions on Fuzzy Systems*, 12, 559-564.
Verner, I.M., and Ahlgren, D.J. (2004). Robot Contest as a Laboratory for Experiential Engineering Education, *Journal on Edu. Resources in Computing*, Vol. 4, Issue 2..
Ye, C. Yung, N.H.C., and Wang, D. (2003). A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance, *IEEE Transactions on Systems, Man. and Cybernetics-Part B: Cybern*33, 17-27..

Table 5. An extract from the questionnaire

QUESTIONNAIRE

Criteria	Sub-Criteria	S. Agree	Agree	Neutral	Disagree	S. Disagree
User Friendliness	EduRobot is easy to use and user-friendly	17	4			
	My interaction with the EDURobot tool is clear and understandable.	16	5			
	EDURobot is not difficult to use	12	9			
	The tool has visual capability to facilitate the understanding of the various path planning algorithms	14	7			
	Changing of parameters is easy and has significant impact	15	6			
Application-specific self-efficiency	I have the ability to set the start and and goal positions of the path plan	14	6	1		
	I have the ability to place obstacles anywhere in the grid	14	7			
	I have the ability to run any of the three different algorithms for any configuration	13	6	2		
	I can observe the effects of changes in the parameters easily	12	8	1		
	I can thoroughly understand the differences in functioning between the three different algorithms	10	8	3		