International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 1, January-2014, pp: (215-221), Impact Factor: 1.252, Available online at: www.erpublications.com

Visual Fuzzy Logic Path Planning Controller for Mobile Robots

Rami Al-Jarrah[1], Mohammad A. Al-Jarrah[2], Hubert Roth[3]
[13]Department of Automatic Control Engineering, Siegen University, Siegen, Germany
[2]Department of Electrical Engineering, Salman Bin Abdulaziz University, Al-Ryadh, Saudi Arabia

Abstract: The fuzzy logic controller for mobile robots that tuned by genetic algorithm is designed for path planning in unknown environments. The path planning is very important for moving a robot to a specific position even there is a changing in the environment. The robot should recognize the environment around himself by using navigation plan which make the robot to accomplish the mission. This design has two levels: the planner level and motion controller level. The planner level based on visual data by Speeded Up Robust Features algorithms to generate the path to the destination and avoid obstacles. The motion controller responsible about the wheels velocity of the robot. The simulations results show a good performance for the approach as well as the real experiments show the applicability of the system.

Keywords: Ground robot, fuzzy control, path planning, genetic algorithm.

Introduction

Making a robot to move from start point to a specific position in a constantly changing environment is the greatest goal in a navigation system. To do so, the robot should recognize the environment around itself by using the navigation plan which can make the robot to accomplish the mission using a set of sensors in the varying environment [1]. Different methods and approaches have been developed to solve the path planning problem, such as the cell decomposition, road map and potential field [2, 3]. Most of these methods were based on the concept of space configuration [4]. These techniques show lack of adaptation and a non robust behavior. To overcome the weakness of these approaches researchers explored variety of solutions [5]. Therefore, in order to have a suitable motion planning scheme in an unknown environment, the control system should be able to adapt in nature. The soft computing or intelligent systems include such as fuzzy logic, genetic algorithm, neural network can solve such complex real-world problems within a reasonable accuracy [6, 7]. The non-holonomic constraints of the wheeled mobile robots make it difficult to derive stable trajectory control laws [8]. Moreover, there are few limitations because of the geometric relation among the positions of the robot with respect to the target.

The main issues of the mobile robot navigation in unknown dynamic environment can be divided into three parts. First, find the exact position for robot, target and obstacles by using computer vision system. Second, navigate to the target by using the coordinate information. Finally, follow the planned path and detect any new obstacles in the path. This system is very important for some tasks and missions such as soccer robots and rescue jobs because there are some dynamics obstacles which are moving in the space areas. In most of the approaches, the desired paths were generated by a higher level planner. In this paper, we apply fuzzy logic and Speeded Up Robust Features algorithms (SURF) vision algorithm to this planner level as well as fuzzy logic to the motion control level . The SURF has been proposed for real-time processing because the processing time is faster than other features techniques [9]. In [10], they proposed a distance propagating dynamic system and gets a real time collision-free motion planning. Yong Han combined the influence of obstacles velocity with the potential field function and gets a good result [11]. In this paper, we consider that the robot has no information about the environment and should avoid obstacles to approach its target position. The SURF will detect the target, robot, obstacles . Then, genetic algorithm will find the initial short path for the robot in order to reach its final destination. The fuzzy logic controller works on the map and has ability to detect any new obstacles in the path and build new path by using the sensors information. We simulate the algorithm on various complex and simple maps environments as well as tested on real environments. The robot was easily able to detect and escape from obstacles and reach the target in an optimal time.

International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 1, January-2014, pp: (215-221), Impact Factor: 1.252, Available online at: www.erpublications.com

Modeling of a mobile robot

The posture vector $P = [x, y, \theta]^T$ and the translational and rotational velocity of the mobile robot $V=[v, w]^T$ define with respect to the center of the robot. The kinematics given as :

$$\dot{P} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{1}$$

$$[v \quad w]^T = \begin{bmatrix} \frac{V_R+V_L}{2} & \frac{V_R-V_L}{2} \end{bmatrix} \tag{2}$$

where V_R, V_L are the right and left wheels velocities. θ is the robot angle w.r.t original coordinates. Fig. 1 shows the schematic diagram for the system components (robot, obstacles, target). Where d_{ob}, α are the distance and orientation of the obstacle from the robot respectively. d_t, β are the distance and angle between the robot and the target location, respectively.

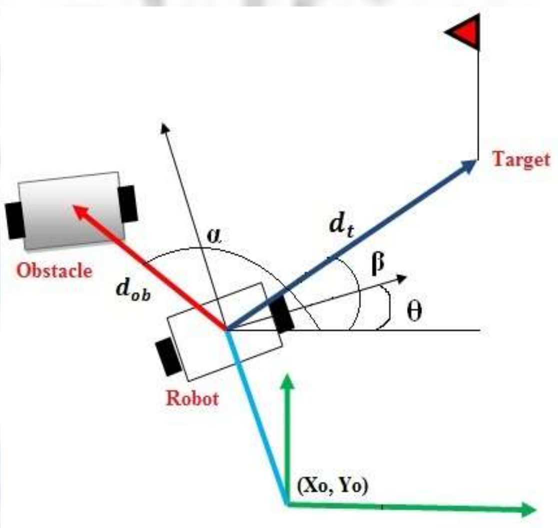


Figure 1. Schematic diagram of the system components

Computer Vision Algorithm

In computer vision the image processing is used to find the image characteristics to recognize interest points. These features are ranging from points or edges to more complex structures such as ground robots. Such features can be used as references for several control systems and visual servoing tasks. The problem of tracking features can be solved with different approaches. The Harris corner detector is one of the most reliable interest point detectors. It is robustness to any rotation, image noise and illumination variation. However, it is very sensitive to changes in image scale and not suitable different sizes images [12]. Even though the Scale-Invariant Feature Transform (SIFT) [13] is widely used in several applications, but it can't satisfy the real time requirements because it needs large amount of calculation time. However, SURF algorithm has an improvement in terms of speed. The SURF allows having good detection with scaled invariant, rotation invariant, and robust against noise. The first step of the algorithm is determining the keypoints based on the determinant value of the Hessian matrix and using the Haar-wavelet.

Then, construct a distinctive features vector for each key-point by using a sliding orientation window around the keypoints and determine the dominant orientation. The last step is matching the features vectors of the image against those of the reference image. It uses a metric to compute the distance between the feature vectors depends on Euclidean distance. However, the SURF algorithm has outliers due to the Gaussian distribution. To filter the total set of matched points and eliminate erroneous matches the Random Sample Consensus RANSAC algorithm is used to discard the outliers from the set of matched points [14]. Fig.2 shows the SURF algorithm detect the ground robot and the obstacle.

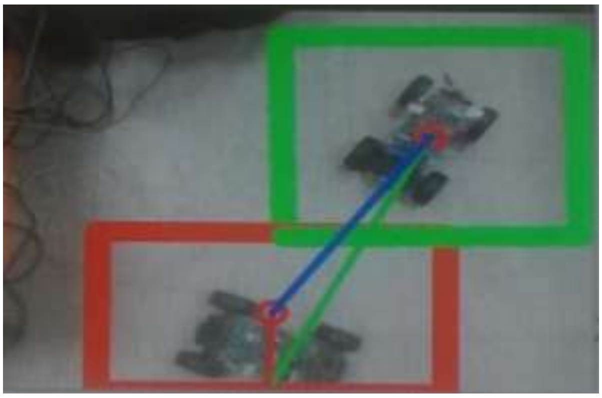


Figure 2. The SURF algorithm test

Let us define I (x, y) is the image frame. The coordinates of the most left-bottom and the most right-top points are (0, 0) and (263,167), respectively. The pose (location and orientation) of the ground robot is Pr = (x_r, y_r, θ_r) and the obstacle pose is Pob = (x_{ob}, y_{ob}, α), and the target pose is Pt= (x_t, y_t, β) in the image plane. These locations and orientations are related to camera projection point in the image plane and it is not global localization.

$$\theta_r = \tan^{-1}\left(\frac{x_r}{y_r}\right) \tag{3}$$

$$\alpha = \tan^{-1}\left(\frac{x_{ob}}{y_{ob}}\right) \tag{4}$$

$$\beta = \tan^{-1}\left(\frac{x_{ob}}{y_{ob}}\right) \tag{5}$$

$$d_{ob} = \sqrt{(x_{ob} - x_r)^2 + (y_{ob} - y_r)^2} \tag{6}$$

$$d_t = \sqrt{(x_t - x_r)^2 + (y_t - y_r)^2} \tag{7}$$

$$\theta_r^{ob} = \tan^{-1}\left(\frac{x_{ob} - x_r}{y_{ob} - y_r}\right) \tag{8}$$

$$\theta_r^t = \tan^{-1}\left(\frac{x_t - x_r}{y_t - y_r}\right) \tag{9}$$

Where, θ_r is the oruentation of the ground robot to the original axes. α is the projection angle between the obstacle and robot in the image plane, β is the angle between the robot and the target in the projection image plane. Where d_{ob} is the distance between the robot and the obstacle in image plane. dt is the distance to the target. θ_r^{ob} and θ_r^t are the angle of the robot with respect to the obstacle and target in image frame, respectively. We should not that all distances here are in pixels and in order to convert them to the metric distances we use pinhole projection point, so the distances in Eq. 6 and Eq. 7 are rewritten as the following.

$$D_{ob} = -\frac{H_b \times d_{ob}}{f} \tag{10}$$

$$D_t = -\frac{H_b \times d_t}{f} \tag{11}$$

Where, Hb is the camera height in the environment.

Fuzzy Logic

The overall system is shown in Fig. 3and it has three main steps. First, The SURF vision system that locates objects on the environment by the camera information which is mounted above the area. Second, the visual information will feed the fuzzy controller with data. The last step is the outputs of the fuzzy controller that provide the robot with the information of the velocities to act and reach the target. Since the action is represented as a number of sequential motions, the host computer sends each robot's left and right wheel velocities rather than higher linguistic commands. The fuzzy logic itself contains three blocks or three controller. The fuzzy destination controller, fuzzy avoid obstacles controller and fuzzy motion. Three of them work parallel in order to control the main behaviors of the robot. We should note that the avoid obstacles controller also has the ability to work with robot sensors. Therefore, any delay in vision system information the robot can detect the obstacles and change its orientation to avoid them. Fig. 4 illustrates the inputs and output membership functions. These membership functions were designed after simulations studied then they are optimized by using genetic algorithm in order to get better results.

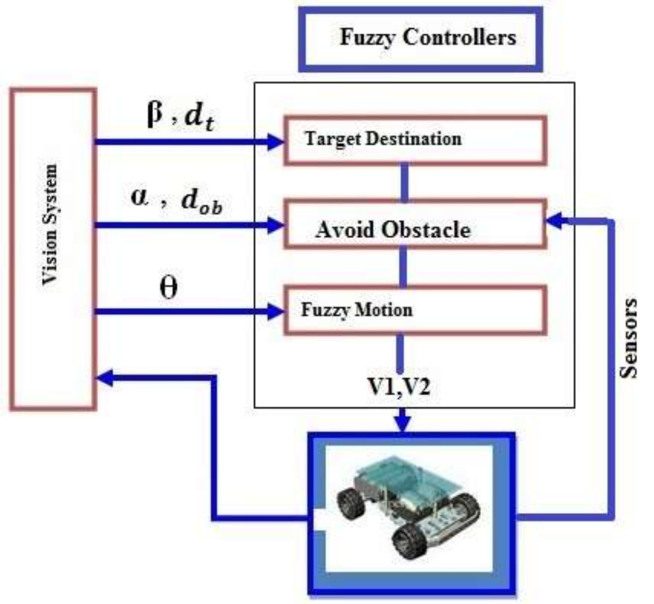


Figure 3. The overall system

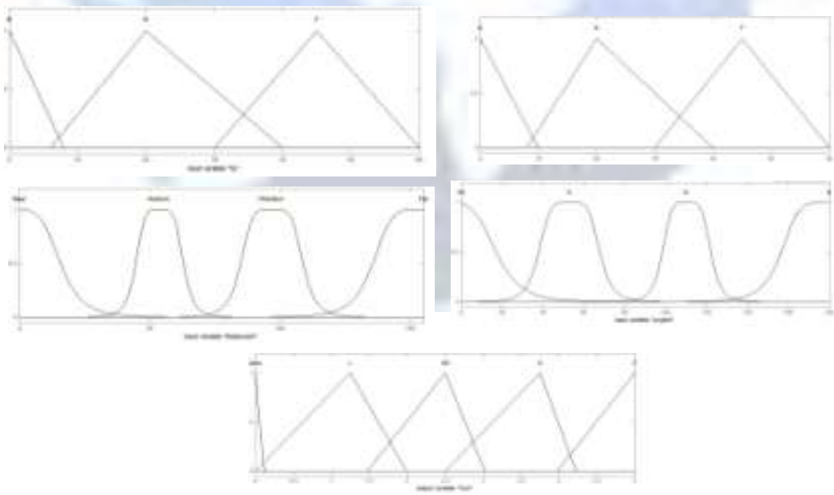
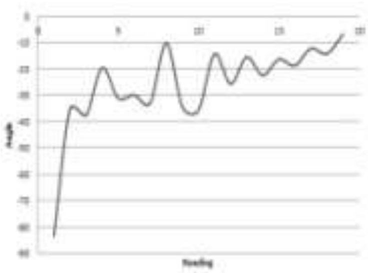


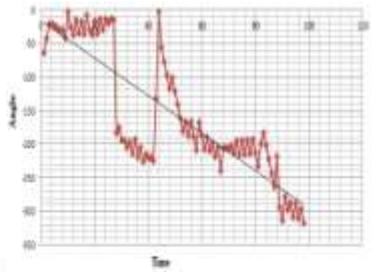
Figure 4. Membership Function: Distance to target, Angle to Target, Distance to Obstacle, Angle to Obstacle, and the output is Turn the robot

Simulation and Experiment

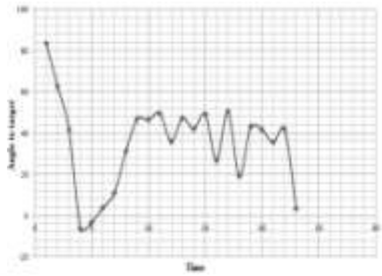
In order to test the algorithm, we had made a simulation by MATLAB. Every attempt was made to ensure that the simulation behaves in a way similar to actual robot. This would ensure that the algorithm can be easily deployed on a real robot in the next step of the project. All simulations were done on a 3.3GHz dual core system with 8 GB RAM. The image Map information in the simulation based on fuzzy edge detection in order to find the robot, goal and the obstacles. The simulation results based on static environment to test the fuzzy control. In all the cases the robot was initially facing at the direction of the x-axis and not towards the goal. the physical quantities of the robot are: wheelbase (D) = 8 cm, wheel radius (r) = 4.0 cm, and the sampling interval (Ts) = 0.5 ms. Fig. 4 shows the simulation results. The robot was supposed to reach the goal from the source. The top left corner is the source and bottom left corner is the goal. The first test case was the obstacle avoidance test. Here we placed numerous small and big obstacles which were kept on the path of the robot. We observed that the robot was easily able to guide itself by avoiding obstacles towards the target. It could smoothly manage to overcome the numerous obstacles of all sizes and shapes that were on its way. The path traversed by the robot was quite optimal.



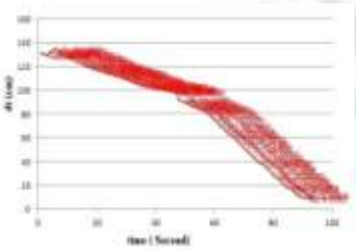
A) Angle between robot and target



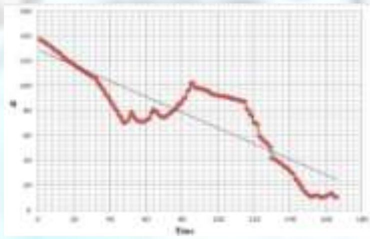
B) angle between robot and target



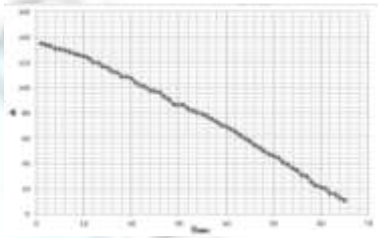
C) angle between robot and target



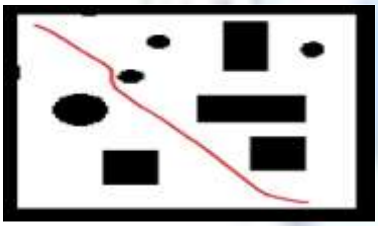
D) Distance to reach the target



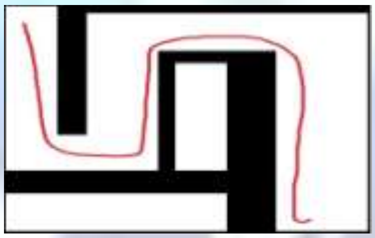
E) Distance to reach the target



F) Distance to reach the target

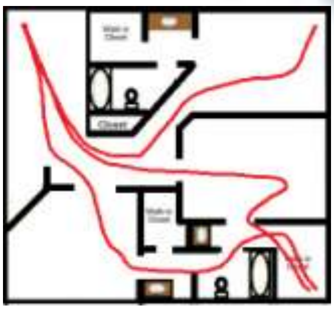


G) Map1



H) Map2

L) Map3



M) Map4

Figure 5. The simulation results

In order to verify the complete proposed system, some experiments were conducted. During these experiments the camera was at a certain height (1 meter). The experiment was done in lab, the background is very clear and there is not any disturbance except for the illumination variance. We assume that the ground robot is already in the view of the on-board camera. The ground robot and the obstacle would be identified and detected in the video sequence by the vision system. One obstacle was taken into account during these experiments. The related distances and orientations between the ground robot and the obstacle and the target are shown in in Fig.6 and Fig. 7. These initial tests were implemented to check the feasibility of the system and the results are quite good and show how the ground robot can navigate by using the visual information. Fig. 8 and Fig. 9 are the sequences images of the experimental results for two environments. Fig. 8 illustrates how the ground robot behave when it detect another robot in the environment and drive away to avoid collision with this obstacle. Also, as it is clear in Fig. 9 the robot has ability also to turn away when it detects a static obstacle in its path.

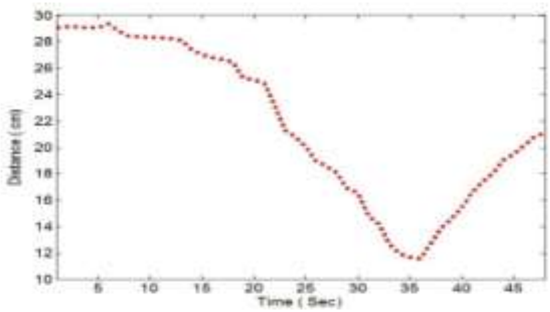


Figure 6. **The related distance d_{ro}.**

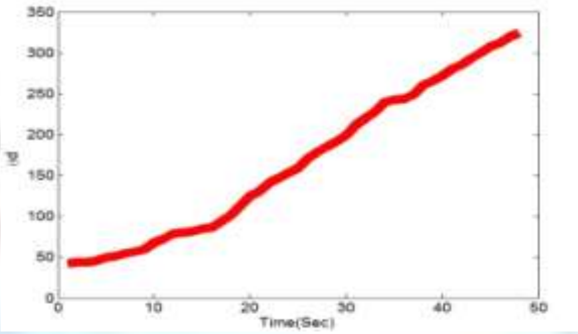


Figure 7. **The related oriented angle.**

Figure 8. **Dynamic Environment : robot and obstacle (robot)**

Figure 9. **Static Environment: robot and static obstacle.**

Conclusion

In this paper we have proposed a method to solve the problem of path planning using a combination of the genetic algorithm and Fuzzy Planning. In addition, the transformations from different coordinate frames have been formulated. The experiments results validate that the algorithm is not only able to help the ground robot navigates effectively, but also it improves the robustness and accuracy of the system. We tested the algorithm for various test cases. In all the test cases we observed that the algorithm was able to find the correct solution and navigate from start point to end point with avoiding obstacles. The solutions generated by the algorithm were often close to optimal. The Fuzzy Planner can run in a very dynamic environment and help the ground robot to be safe from colliding obstacles. Any change in the environment at any time can be easily adapted in real times. Simulation and experimental results showed the applicability of the proposed control scheme.

References

[1]. Ching-Chang Wong, Chi-Tai Cheng, Kai-Hsiang Huang, and Yu-Ting Yang, "Fuzzy Control of Humanoid Robot for Obstacle Avoidance," International Journal of Fuzzy Systems, vol. 10, no. 1, March 2008.

[2]. B. Graf, J. M. Hostalet Wandosell, "Flexible path planning for nonholonomic mobile robots,". In Proc. 4th European workshop on advanced Mobile Robots (EUROBOT'01)., Fraunhofer Inst. Manufact. Eng. Automat. (IPS). Lund, Sweden, Sept. 19-21, 2001, pp. 199-206.

[3]. J. C. Latombe, Robot Motion Planning. 1st ed, Boston, UK, MA: Kluwer Academic Publishers (Published by Springer), 1991, ISBN: 0-7923- 9129-2.

[4]. T. Lozano-Perez and M. Wesely, (1979, October). An algorithm for planning collision-Free paths among polyhedral obstacles. Commun. of the ACM [Online], 22(10), pp. 560-570.

[5]. A. Elshamli, H. A. Abdullah and S. Areibi, "Genetic algorithm for dynamic path planning". In: Proc. Canadian Conf. Elect. and Comput. Eng., Niagara Falls, Vol.2, May 2-5, 2004, pp: 677-680, 2004 IEEE.

[6]. NirmalBaran Hui,V.Mahendar, Dilip Kumar Pratihar, "Timeoptimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches," Fuzzy sets and Systems 157 (2006) 2171-2204.

[7]. Jelena Godjevac, Nigel Steele, "Neuro-fuzzy control of a mobile robot," Neurocomputing 28 (1999) 127-143.

[8]. J.-M. Yang and J.-H. Kim, "Sliding Mode Control for Trajectory Tracking of Nonholonomic Wheeled Mobile Robots," IEEE Transactions on Robotics and Automation, vol. 15, no. 3, Jun. 1999.

[9]. Bay, H., Ess, A., Tuyelaars, T., and Gool, V. (2008). Speeded up robust features. Computer vision and image understanding, 346-359.

[10]. Allan R. Willms and Simon X. Yang. An efficient dynamic system for real-time robot-path planning[J], IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, v36, n4, August, 2006, p755-766.

[11]. Yon Han, Guo-dong LIU, Mobile Robot Motion Planning Based on Potential Field in Dynamic Environment[J]. Robot. 2006, 8(1):45-49.

[12]. C. Harris, M. Stephens, A combined corner and edge detector, Proceedings of the Alvey Vision Conference, 147-151, 1988.

[13]. D.G. Lowe, Distinctive image features from scale-invariant key point, cascade filtering appraoch, International Journal of Computer Vision, Vol.60, No.2, pp. 91-110, 2004.

[14]. M.A. Fischer and R.C. Bolles, Random Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communication of the ACM, 1981, vol. 24(6), pp. 381-395.