Efficient Path Planning Algorithm for Mobile Robot Navigation with a Local Minima Problem Solving

Jasmin Velagic, Bakir Lacevic and Nedim Osmic
Faculty of Electrical Engineering, University of Sarajevo, Bosnia and Herzegovina
E-mails: jasmin.velagic@etf.unsa.ba, bakir.lacevic@etf.unsa.ba, nedim.osmic@etf.unsa.ba

Abstract – **This paper proposes a new reactive planning algorithm for mobile robot navigation in unknown environments. The overall navigation system consists of three navigation subsystems. The lower level subsystem deals with the control of the linear and angular volocities using a multivariable PI controller described with a full matrix. The position control of the mobile robot is in the medium level, and it is a nonlinear. The nonlinear control design is implemented by a backstepping algorithm whose parameters are adjusted by a genetic algorithm. The high level subsystem uses the Fuzzy logic and Dempster-Shafer evidence theory to design the fusion of sensor data, map building and path planning tasks. The path planning algorithm is based on a modified potential field method. In this algorithm, the fuzzy rules for selecting the relevant obstacles for robot motion are introduced. Also, suitable steps are taken to pull the robot out of the local minima. A particular attention is paid to detection of the robot's trapped state and its avoidance. One of the main issues in this paper is to reduce the complexity of planning algorithms. Simulation results show a good quality of position tracking capabilities and obstacle avoidance behavior of the mobile robot.**

I. INTRODUCTION

The basic feature of an autonomous mobile robot is its capability to operate independently in unknown or partially known environments. The autonomy implies that the robot is capable to react to static obstacles and unpredictable dynamic events that may impede the successful execution of a task [1]. To achieve this level of robustness, methods need to be developed to provide solutions to localization, map-building, planning and control. The development of such techniques for autonomous robot navigation is one of the major trends in the current robotics research [2].

The robot has to find a collision-free trajectory between the starting and the goal configurations in a static or a dynamic environment containing some obstacles. To this end, the robot needs the capability to build a map of the environment, which is essentially a repetitive process of moving to a new position, sensing the environment, updating the map and planning subsequent motion.

Path planning is one of the key issues in mobile robot navigation. Local path planning methods use ultrasonic sensors, laser range finders, and on-board vision systems to perceive the environment to perform an on-line planning [3]. In our paper, the workspace for the navigation of the mobile robot is assumed to be unknown and it has stationary obstacles only.

In local path planning methods a particular attention is paid to local minima problem. This problem occurs when a robot navigates towards a desired target with no priori knowledge of the environment and gets trapped in a loop [4], [5]. This happens usually if the environment consists of concave obstacles, mazes, and the like objects. To get out of the loop the robot must comprehend its repeated traversal through the same environment, which involves memorizing the environment already seen [5].

The main contribution of this paper is design of a robust autonomous mobile robot control system suitable for on-line applications by using soft computing methodologies. This system provides the mobile robot that may navigate in an a priori unknown indoor environment using sonar sensors information. To achieve these requirements the proposed system is hierarchically organized into three distinct separated subsystems with arbitrary responsibility. At each level of this system one or more soft computing methodologies are adopted to solve its specific problems.

A low level velocity controller is developed using the standard PI multivariable control law. The medium level position control law has to be nonlinear in order to ensure the stability of the error, that is its convergence to zero [6], [7]. Some of control parameters are continuous time functions, and usually the backstepping method [6], [8] was used for their adjustment. In order to achieve the optimal parameters values we used a genetic algorithm.

A high level subsystem contains map building and path planning algorithms. In this paper the occupancy based map [9] using Dempster-Shafer evidence theory based on sonar measurements is demonstrated. Also, we propose a new path planning approach based on the repulsive and attractive forces, which sets the fuzzy rules for determining which obstacles should have influence on the robot motion. This approach provides both obstacle avoidance and target following behaviors and uses only the local information for decision making of the next action. At the end, a new algorithm for identification and solving of the local minima situation during the robot's traversal using the set of fuzzy rules is proposed.

II. NAVIGATION SYSTEM

The proposed three level navigation system is shown in Fig. 1. The low level velocity control system is composed of a multivariable PI controller and a dynamic model of mobile robot and actuators. At the medium level the position control system generates a non-linear control law whose parameters are obtained using a genetic algorithm. The high level system performs map building and path planning tasks. This system is in charge of sensor interpretation, sensor data fusion, map building and path planning. All the modules are designed using fuzzy logic and Dempster-Shafer theory of evidence.

In the following sections the design of the navigation system blocks from Fig. 1 is described.

1-4244-0726-5/06/$20.00 '2006 IEEE

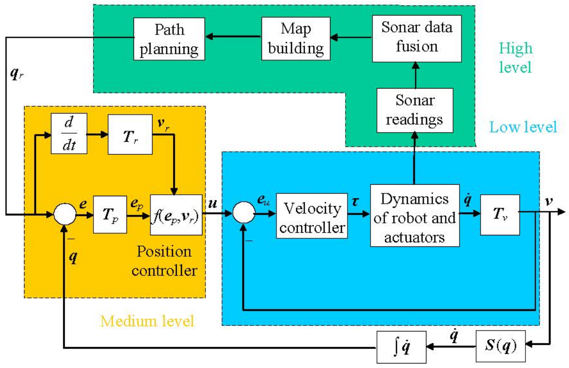

Fig. 1. Mobile robot navigation system for real-time requirements.

A. Dynamics of Mobile Robot

The dynamic model of our mobile robot was derived in recent papers [10], [11]. The robot has two driving wheels which are independently driven by two actuators to achieve both the transition and orientation. Dynamic equations of the motion can be written as in [10]:

$$A\ddot{\theta}_R + B\ddot{\theta}_L = \tau_R - K\dot{\theta}_R\,, \tag{1}$$

$$B\ddot{\theta}_R + A\ddot{\theta}_L = \tau_L - K\dot{\theta}_L\,, \tag{2}$$

where

$$A = \left(\frac{Mr^2}{4} + \frac{(I_A + Md^2)r^2}{4R^2} + I_0\right), B = \left(\frac{Mr^2}{4} - \frac{(I_A + Md^2)r^2}{4R^2}\right)$$

and M is the mass of the entire vehicle, r is the radius of wheels, I_A is the moment of inertia of the center of mobile robot gear, I_0 is the moment of inertia of the rotor/wheel complex system, τ_R and τ_L are the actuation torques (right and left) and $Kd\theta_R/dt$ and $Kd\theta_L/dt$ are the viscous friction torques of right and left wheel-motor systems, respectively. The angular velocities of wheels ($d\theta_R/dt$ and $d\theta_L/dt$) are main variables.

B. Low-level Subsystem

The dynamics of the velocity controller is given by the following equations in Laplace domain [11]:

$$\boldsymbol{\tau}(s) = \begin{bmatrix} \tau_R(s) \\ \tau_L(s) \end{bmatrix} = \frac{1}{r}\begin{bmatrix} g_1(s) & g_2(s) \\ g_1(s) & -g_2(s) \end{bmatrix}\begin{bmatrix} e_v(s) \\ e_\omega(s) \end{bmatrix}, \tag{3}$$

where $e_v(s)$ is the linear velocity error, and $e_\omega(s)$ is the angular velocity error. Transfer functions $g_j(s)$ are chosen as follows:

$$g_1(s) = K_1(1+\frac{1}{T_{i1}s})\cdot R, \quad g_2(s) = K_2(1+\frac{1}{T_{i2}s})\cdot R\,. \tag{4}$$

The choice of the multivariable PI controller described by (3) and (4) is justified with our theorem [11].

The controller parameters used for the simulation are K_1=129.7749, K_2=41.0233, T_{i1}=11.4018, T_{i2}=24.1873, which are tuned using standard GA.

From the simulation results obtained (Fig. 2), it can be seen that the proposed PI controller successfully tracks the given linear and angular velocity profiles.

C. Medium-level Subsystem

The trajectory tracking problem for a mobile robot is based on a virtual reference robot that has to be tracked. The tracking position error between the reference robot and the actual robot can be expressed in the robot frame as:

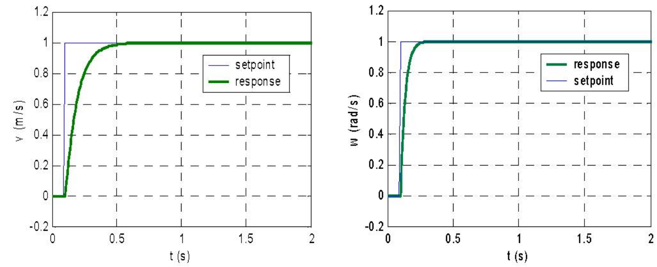

Fig. 2. Linear and angular velocity step responses.

$$\boldsymbol{e}_p = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \boldsymbol{T}_p\boldsymbol{e}_q = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}. \tag{5}$$

where $\boldsymbol{e}_q=[e_x\ e_y\ e_\theta]^T$.

The position error dynamics can be obtained from the time derivative of the (5) as:

$$\dot{e}_1 = \omega e_2 + u_1, \dot{e}_2 = -\omega e_1 + v_r \sin e_3, \dot{e}_3 = u_2\,, \tag{6}$$

where inputs u_1 and u_2 are new control inputs.

In this paper we propose the following nonlinear control inputs as the servos for velocity control loop:

$$\begin{aligned} u_1 &= v_r \cos e_3 + \frac{k_1 e_1}{\sqrt{k_4 + e_1^2 + e_2^2}} \\ u_2 &= \omega_r + \frac{k_2 v_r e_2}{\sqrt{k_5 + e_1^2 + e_2^2}} + \frac{k_3 v_r \sin e_3}{\sqrt{k_6 + e_3^2}} \end{aligned}, \tag{7}$$

where k_1, k_2, k_3, k_4, k_5 and k_6 are positive parameters.

Equations (7) are a modified backstepping control law given first in [6]. The modification consists of the introduction of denominators. In [6], Lyapunov's stability theory was used to prove that the considered control law provides uniformly bounded norm of error $||e_p(t)||$. The issue of rigorous proof of stability for introduced control law (7) remains however open.

In this paper a simple genetic algorithm is used for parameter evolution. Coefficients k_1 to k_6 are encoded into a binary chromosome. If $e_x(t)$, $e_y(t)$, and $e_\theta(t)$ $(0 < t < t_s)$ are error functions, the objective function is calculated as

$$\begin{aligned} F = a_x \sum_{i=0}^{N-1} \ln(1 + \left|e_x(i\frac{t_s}{N})\right|) + a_y \sum_{i=0}^{N-1} \ln(1 + \left|e_y(i\frac{t_s}{N})\right|) \\ + a_\theta \sum_{i=0}^{N-1} \ln(1 + \left|e_\theta(i\frac{t_s}{N})\right|) \end{aligned}. \tag{8}$$

Parameters a_x, a_y and a_θ are some positive real numbers. N is the number of error samples. In order to evaluate the fitness of the individual it is necessary to run the simulation.

The coefficient evolution is performed using a lemniscate as suitable complex trajectory [10]. The evolution process (population size is 40) is depicted in Fig. 3. Values of the objective function parameters are: $a_x = a_\theta = 1$, $a_y = 2$, N=1500 and t_s=15 sec. Evolution yielded the following coefficient values: k_1=18.262, k_2=299.732, k_3= 9.8229, k_4=26.537,

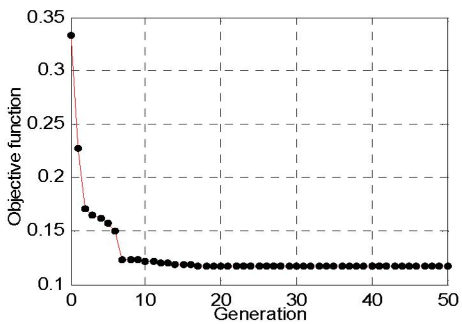

Fig. 3. Best fitness through generations.

k_5=1.0164, k_6=2.0028, K_1=129.7749, K_2=41.0233, T_{i1}=11.4018, T_{i2}=24.1873.

The results of the lamniscate trajectory tracking task are shown in Fig. 4. From this figure, it can be concluded that satisfactory tracking results are obtained using the proposed control system. More results of this medium-level control system are given in [10].

D. High-level Subsystem

The High-level subsystem contains map building and path planning processes. Here, the proposed map building process utilizes Dempster-Shafer evidence theory [10]. The sonar sensor readings are interpreted by this theory and used to modify the map using Dempster's inference rule.

More specific, to build an occupancy map of the environment, we first construct a grid representing the whole space. Every discrete region of the map may be in two states, *Empty* and *Full*. A series of range readings $\{r_1,...,r_n\}$ collected at known sensor locations are available. In principle, the task of the Map Building System is to process the readings in order to assess which cells are occupied or empty and thus suitable for robot navigation. The complete description of our map-building system can be found in [10]. The path planning process will be described in the following section.

III. PATH PLANNING

The avoidance of obstacles problem in this paper is based on a modificated of the potential field method.

A. Calculation of Attractive and Repulsive Forces

In the potential field approach obstacles exert a repulsive force on the robot and target position exerts an attractive force.

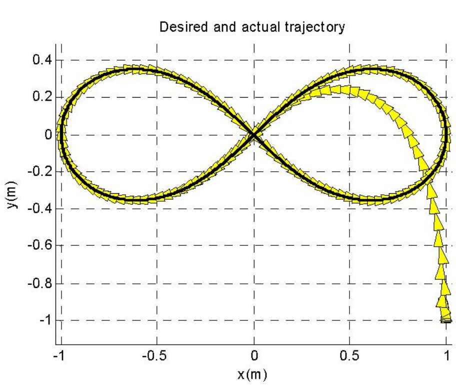

Fig. 4. Tracking a lemniscate path.

The amplitude of the repulsive force is defined as

$$F = \frac{1}{R_r^2}, \tag{9}$$

where R_r is distance between the current robot location and obstacle location. Its value is calculated as follows (Fig. 5):

$$R_r = \sqrt{\Delta x_o{}^2 + \Delta y_o{}^2} = \sqrt{(x_o - x_m)^2 + (y_o - y_m)^2}, \tag{10}$$

where Δx_o and Δy_o are coordinate distances between mobile robot position and obstacle position. The attractive force is defined using R_a as

$$R_a = \sqrt{\Delta x_t^2 + \Delta y_t^2} = \sqrt{(x_t - x_m)^2 + (y_t - y_m)^2}. \tag{11}$$

In this paper we proposed the following law for attractive forces:

$$F_a = \begin{cases} |R_a|, & \text{for } |R_a| > 1\,\text{m} \\ \kappa |R_a|, & \text{for } |R_a| \le 1\,\text{m} \end{cases}, \tag{12}$$

where κ=5. This parameter is involved to reach the target when obstacles (relevant obstacles which satisfy pruning rules in the next subsection) are located into the circle with radius 1 m and then they produce strong repulsive forces. The discontinuity at point of 1 m causes the orientation of the robot to be slightly modified since the new intermediate target point is moved to another location. This parameter is related with radius R of sonar model. In the following consideration the R is 4 m.

The resulting vector is obtained as a sum of the vectors of repulsive and attractive forces (Fig. 5). The new target coordinates of the robot are:

$$x_n = x_r + \Delta x_t,\ y_n = y_r + \Delta y_t. \tag{13}$$

To avoid a possible standstill which may arise if the robot, the obstacle a target aligned, we introduce a stochastic force orthogonal to planned trajectory when sum of forces is zero, and robot is not yet at the target. This situation is explained by fuzzy rules:

If $|F_a| = |F|$ and *angle between* F_a and F *is less then* 0.01 *rad*
then F_s *is big and acts orthogonal on robot*
In other cases F_s *is small.* (14)

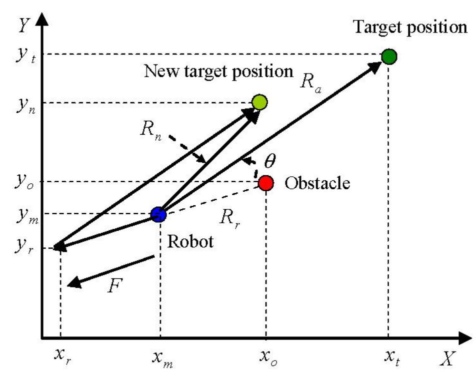

Fig. 5. The concept of path planning based on repulsive and attractive forces.

In the following subsection the proposed concept of pruning of relevant obstacles is demonstrated.

B. Pruning of Relevant Obstacles

The number of obstacles in the environment may be large. If all the obstacles exert repulsive force, the calculation of the field is extensive and yet it may produce erratic trajectories. Therefore, we introduce a pruning of obstacles with an aim to preserve only these relevant for the trajectory planning. This pruning is done in a set of steps. In these steps we use the concept of the *visibility field* of the mobile robot [10].

The practical visibility field of a mobile robot here is a circle with the radius r_v= 4m. However, in our approach we use only a square inscribed in this circle. The reason is that the local map has $m\times m$ points. This local map is now represented as a $m\times m$ matrix, and each cell in the map is one matrix element.

Due to this geometry, some obstacles will be in visibility field given by

$$R_r < r_v. \tag{15}$$

but they will not be taken into account. For example, in Fig. 6, the obstacles 3, 4 and 5 satisfy (15). However, only obstacles 4 and 5 remain as relevant.

Next, all the obstacles in the square are not equally relevant to the future robot motion toward target. To pinpoint the most relevant we first define a circular area around the target named *target circle* with a radius R_a [10]. The value R_a is slightly larger than the distance between the robot and target, as represented in Fig. 7 (R_a+0.1). The extension of 0.1 ensures a safe region around robot. We consider that only the obstacles in the section where the target circle and the visibility square overlap are really relevant to the motion planning. Then, among these obstacles we take the one nearest to the robot. The distance between the target and the obstacle is R_p:

$$R_p = \sqrt{(x_t - x_o)^2 + (y_t - y_o)^2}\ . \tag{16}$$

In our approach the rule for visibility of obstacle inside square is given as:

If ((R_a +0.1)>R_p) and (R_r<r_v) then
obstacle is visible and it should influence the motion. (17)

In Fig. 7 only obstacle 3 influences the new target point determination. The further reduction of obstacles is still possible. An obstacle located in the above defined area (17) but behind the current robot position, or behind the target, and not on the direction of motion should not be taken into account. The force produced by such an obstacle can be strong but obviously it should be neglected. To solve this problem we introduce a new circle around the robot with a radius equal to the distance from the robot to the target(Fig. 8). In this way the obstacles located behind the target and outside of this circle are not considered and robot will not be pushed aside to go back.

So, we introduce a new restriction in the final visibility rule:

If ((R_a +0.1)>R_p) and (R_r<r_v) and (R_r < R_a) then
obstacle is visible and it should influence the motion. (18)

Finally in our example (Fig. 9) we have only one relevant obstacle 3.

In the case that no obstacles are located on the robot motion toward the target, its velocity is governed by position controller. When a distance between the robot and relevant obstacles, which satisfy above conditions, is less then 1 m we introduced the parameter k which multiplies velocities of the left and the right wheels. Using this parameter the possible collision between the robot and obstacle and the robot turning over are avoided. The graph of robot velocity in dependence of a distance between the robot and relevant obstacle is shown in the Fig. 10.

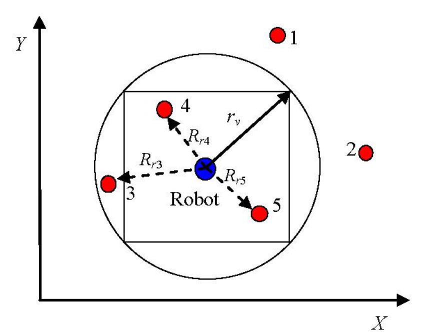

Fig. 6. The visibility region of mobile robot using sonar sensors.

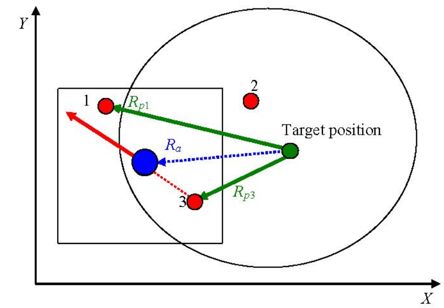

Fig. 7. The forces act on the robot motion.

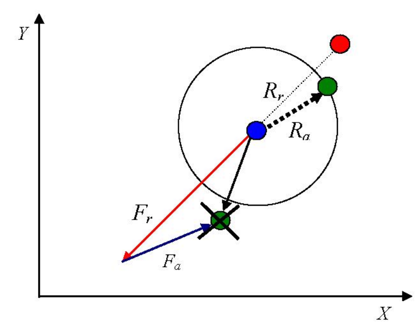

Fig. 8. Introduction of a new circle around the target to avoid the influence of the strong repulsive force.

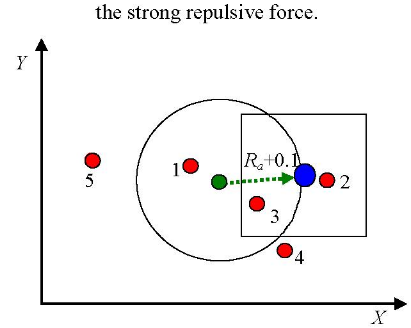

Fig. 9. General concept of determination of new target position.

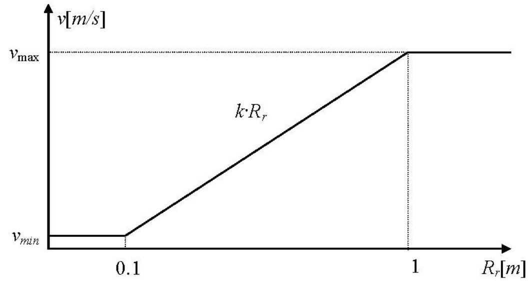

Fig. 10. Dependence of robot velocity and obstacle distance.

In the rest of section the steps of a local minima problem solving are introduced.

C. Local Minima Problem Solving

For a local minima problem avoidance (Fig. 11) we introduce an algorithm which does not require memorizing the already seen environment. This algorithm is capable to identify the local minima situation during the robot's motion, by recognizing its trapped state (infinite loop). In this state the robot oscilates between two points.

This algorithm is described as follows:

Step 1: *Calculate attractive and reflexive forces and their sum (resultant force)*

Step 2: **If** *resultant force tends to push robot to go back* **then** *do not take into consideration the attractive force* **and** *keep the reflexive force to be constant and equal to the closest obstacle*

else *apply the principle of attractive and reflexive forces*

end

Step 3: If *robot does not reached target point* **then**

go to Step 1

end

The above steps are executed until the robot reaches the target. The algorithm turns the robot right in **switching point a** (point in which robot tends to go back) and the robot continues to see the wall on its left until arrives in **switching point b** after it go toward to target point using principle of attractive and reflexive forces (Fig. 12). The simulation results of local minima avoidance using proposed algorithm are given in Fig. 13. In this case the proposed algorithm executes a sequence of steps that pulls it out of the trap.

In addition, more simulations are done in the next section to present the effectiveness of the proposed approach in cluttered, environment and environments with obstacle loops and mazes.

IV. SIMULATION RESULTS

To show the usefulness of the proposed approach, a series of simulations have been conducted using an arbitrarily

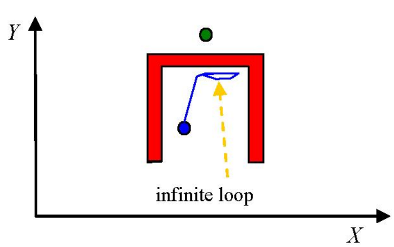

Fig. 11. Illustration of a local minima problem.

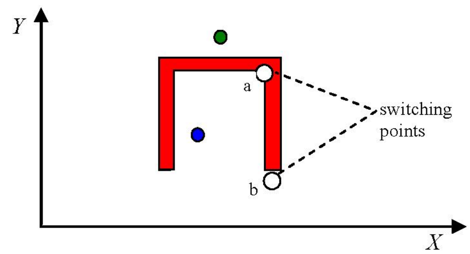

Fig. 12. Identification of point a (switching point) in which robot trapped.

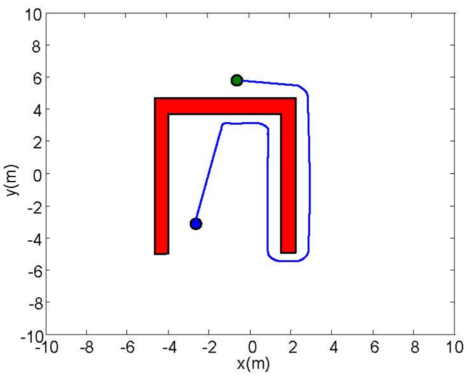

Fig. 13. Simulation results of local minima problem solving using proposed algorithm.

constructed environment including obstacles. The robot has been modeled as a small circle and imaginary sensors, eight in number, are placed in the form of an arc along the circumference of the robot. The minimal distance obtained within the cone of each sensor is considered as the distance of the obstacle from that sensor, and it is available as an input to the algorithm. All of simulations are done in the MATLAB/SIMULINK environment. The used mobile robot parameters are M=10kg, I_A=1kgm^2, r=0.035 m, R=0.175 m, d=0.05 m, m_0=0.2 kg, J_0=0.0001 kgm^2, K/A=0.1.

In these simulations the position of all the obstacles in the workspace is unknown to the robot. The robot is aware of its start and final positions only. Simulation results obtained in seven different working scenarios are presented in Figs. 14-17.

The passing through a narrowdoor processes using our algorithm is shown in Figs. 14. From results obtained can be concluded that robot successfully passes through the narrowdoor.

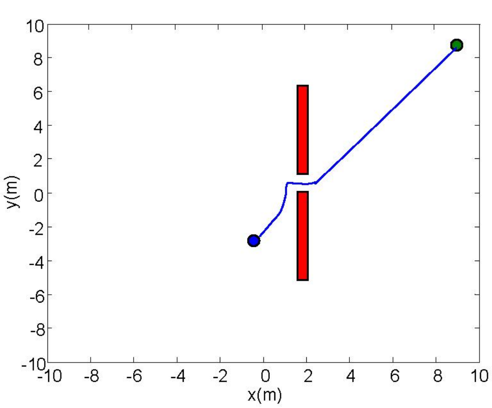

Fig. 14. Passing through a narrowdoor.

In all of the cases in which target is surrounded by cluttered obstacles, which form the convex loop, proposed algorithm described by (9)-(18) helps the robot to find the target (Fig. 15).

A new problem arrises if the environment consists of concave obstacles and robot gets into an infinite loop or into a local minima. Even more complex problem happens when the obstacles are long and have many bends and kinks the target attracting and goal repulsing behavior conflict and the robot also gets itself into an infinite loop (Fig. 16).

The effectiveness of proposed algorithm for solving this problem is illustrated in the same figure. In this way the robot does not get trapped. Finally, behaviour of mobile robot in more cluttered environment is illustrated in Fig. 17.

These results clearly showed that this system is able to navigate in complex environments, such as mazes.

V. CONCLUSIONS

In this paper we present a new mobile robot navigation strategy in an a priori unknown environment suitable for on-line applications with real-time requirements.

A dynamic model of a mobile robot with nonholonomic constraints is derived first. The special feature of this model is that main variables are angular velocities of wheels.

The low level control of robot velocities is designed using PI controllers. The efficiency of this controller is demonstrated by step trajectory tracking.

The coefficients of the medium level nonlinear backstepping position controller are adjusted by a genetic algorithm. This controller reaches a good position tracking performance.

The high level control system consists of map building and path planning. The sonar data interpretation and fusion were realized using Dempster-Shafer theory of evidence. The local map is represented by an occupancy grid with the updates obtained by Dempster rules of combination.

An efficient local path planning algorithm is required to achieve real-time operation. The proposed local planning is based on repulsive and attractive forces and pruning of relevant obstacles by fuzzy rules. It shows a robust and stable performance. For the local minima avoidance problem an algorithm that does not need to memorize previous movement is proposed.

A possible extension of this work be navigation of the mobile robot in a dynamic environment.

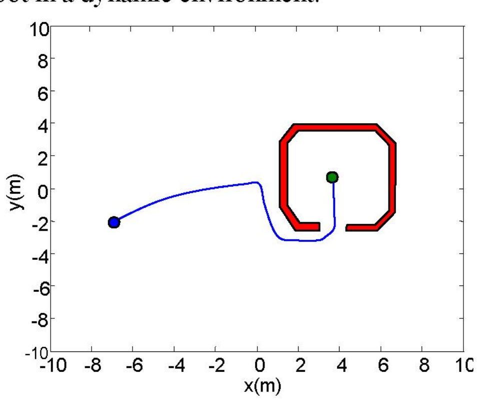

Fig. 15. Finding a target surrounded by obstacles.

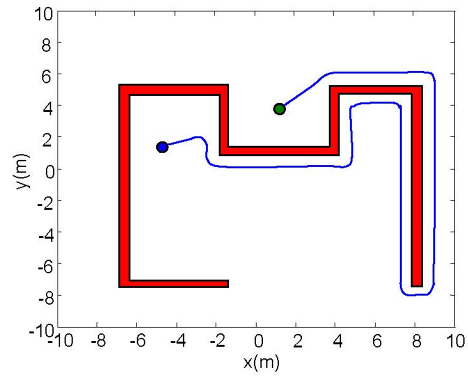

Fig. 16. Finding a target in the maze.

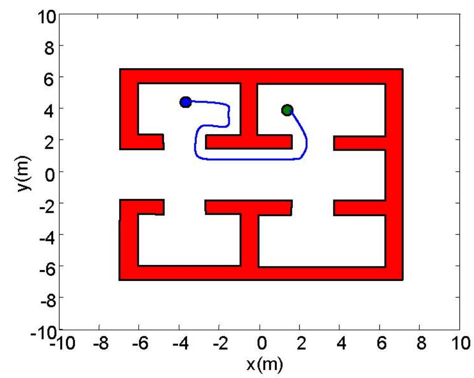

Fig. 17. Finding a target in a more cluttered maze.

REFERENCES

[1] Y.C. Kim, S.B. Cho and S.R. Oh, "Map-Building of a Real Mobile Robot with GA-Fuzzy Controller," *International Journal of Fuzzy Systems*, vol. 4, pp. 696-703, April 2002.

[2] S. G. Tzafestas (ed.), *Advances in Intelligent Autonomous Systems*, Kluwer, Dordrecht/Boston, 1999.

[3] J. Borenstein and Y. Koren, "he Vector field histogram - Fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, 1991.

[4] H.P. Huang and P.C. Lee, "A real-time algorithm for obstacle avoidance of autonomous mobile robots," *Robotica*, vol. 10, pp. 217-227, 1992.

[5] K.M. Krishna and P.K. Karla, "Solving the Local Minima Problem for a Mobile Robot by Classification of Spatio-Temporal Sensory Sequences," *Journal of Robotic Systems*, vol. 17, no. 10, pp. 549-564, 2000.

[6] Hu, and S.X. Yang, "Real-time Motion Control of a Nonholonomic Mobile Robot with Unknown Dynamics," in Proceedings of the Computational Kinematics Conference, Seoul, Korea, 2001.

[7] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation and experimental validation," *IEEE Transactions on Control System Technology*, vol. 10, , pp. 835-852, November 2002.

[8] H. G. Tanner, and K.J. Kyriakopoulos, "Backstepping for nonsmooth systems," *Automatica*, vol. 39, pp. 1259-1265, 2003.

[9] G. Oriolo, G. Ulivi and M. Vendittelli, "Real-time Map Building and Navigation for Autonomous Robots in Unknown Environments, " *IEEE Trans. Syst. Man Cyber*., part B, vol. 28, pp. 316-333, June 1998.

[10] J. Velagic, B. Lacevic and B. Perunicic, "New Concept of the Fast Reactive Mobile Robot Navigation Using a Pruning of Relevant Obstacles," in Proceedings of the IEEE International Symposium on Industrial Electronics - ISIE 2005, 20-23 June, Dubrovnik, Croatia, pp. 161-166, 2005.

[11] J. Velagic, B. Lacevic and B. Perunicic, " Reduction of Control Torques of Mobile Robot Using Hybrid Nonlinear Position Controller," in Proceedings of the IEEE International Conference "Computer as a tool", 22-24 November, Belgrade, Serbia and Montenegro, pp. 161-166, 2005.