Autonomous Mobile Robot Navigation to Exit Path Planning for Floor Tile Room

J. SANDOVAL, J.J. MEDEL
Applied Science & Advanced Technology
Research Center at National Polytechnic Institute IPN
Calzada Legaría 694 Col. Irrigación MÉXICO D.F. 11500
jacobosandoval@hotmail.com, jjmedelj@yahoo.com.mx.

Abstract: This paper shows an algorithm for path planning to reach an exit load inside to unknown environment but dimensional conditions bounded and with obstacles avoidance. The algorithm implemented consider to implant into a Pioneer 3-DX with the Arnl tools, so that, the grid-map navigation use two dimensional space; limiting only by floor tiles adjust the algorithm respect to the real environment. It could be seen as a whole where the path plan algorithm achieve a specific path for anywhere position or orientation into two dimensional space defined into the knowledge base; having it as objective, to exit with autonomous decision. It has a set answers and landmark motion with time restrictions. In this case, the first simulation requires knowing the environment, but don't know the random position obstacles. The position reached by sonar sensor with Doppler information, requires the odometer orientation and position into a grid-base with motion on cardinal and intermediate direction as long as the environment permits to follows the desired exit path; in other words, with a plane defined, the autonomous mobile robot move into valid position with specific obstacles set. The strategy represents random paths, step by step, evaluating in each interval the path achieved for the desired goal.

Keywords: Autonomous Navigation, Indoor environment, Local Path Planning, Mobile Robot, Real Time Avoidance, Uncertainly.

1 Introduction

The most basic capability of an autonomous mobile robot in agreement to [1] and [2], consist in to recognize environments, avoid obstacles and decide the actions based in its upon knowledge. Therefore, high performance world demands precise and scientific solutions, perform a varied number of tasks, all these independently of human control strategies.

Into robotic system, intelligence meant for [15] to learn tasks and performing these at the same way respect to the human actions or other decision making as autonomous system. Technically speaking, these tasks target to observability trough sensors and controllability respect to actuators, allowing the robot algorithms blend, with specific goal, conditions bounded with calculability and complexity.

A superior difficult into robot equilibrium point, is to correlate the desired goal in the worse situations ending at the time interval required. These combination facts obligate to use an inference set into algorithm robot decision. In addition, supposing that the robot navigation was near to risk area but the next step of it need to be ejected after the deadline robot operation; so that, the shortest period was goal fault, while compute algorithm error, was the largest path; then, the calculability didn't solve adequately. A mobile robot in any environment with proper abilities operates as real-time system by nature environment interaction.

On Autonomous Guided Vehicle (AGV's) was a system designed to complete some useful task in environment trough the model [16]. Therefore, was necessary build a strategy in order to use the inference tools combined with the knowledge superficies base, requiring to prove its before to implement into mobile robot system.

2. Environment

In mobile robotic, the basic operations allow to recognize two kinds of environments, considering in the general form as indoor and outdoor. Indoor environment consider the most important area aspects respect to sensor data information actualized for each step, remembering through the knowledge base and inference sets the new relative coordinates; algorithms inferences actions response by actuator inside at short step; in this case, the paths action was composed by simple straight lines joined by curves. The outdoor environments considered the longer distances where the possibility good enough answer in each step was low for finding the same reference by the mobile robot and the paths tending to be sequences of points that represent undulating converge curves; satisfying different inference level tasks, conditions and solutions in each step evolution trajectory.

The Arnl software tool allowed and generated to the relative distance measures between mobile robot steps. The sonar data information with inference mechanism algorithms, generating a new politic ejected by the actuator set by each condition environment, as is showed in Fig. 1.

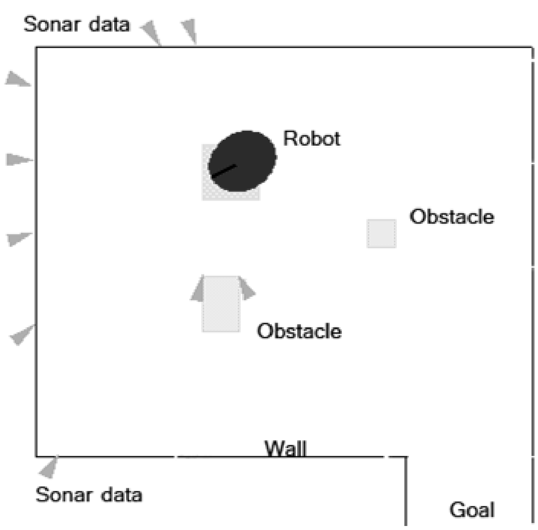

Fig. 1. Simulation block scheme.

3. Obstacles Avoidance

In the previous decade, the obstacles considered into trajectory respect to mobile robot position commonly were statics; but recently changing it as random walk with distribution bounded.
Inside of each step into the mobile robot position sensor the obstacles were considered statics although in the real life this situation never was observed, nothing stayed constant in the physical world, and nothing looks the same condition twice times. But in the discrete sampled computing, the obstacles in each step were freeze position defining the neighbor region into inference sense; where the mobile robot was trunked to reach a valid path. Identifying by radar Doppler sensors, the obstacles place where didn´t mobile robot found the valid paths, the negative inference situation was valid. Some algorithm identification places tasks into prediction strategies allowed building the adjust knowledge base into the mobile robot, modifying its inference loads dynamically in middle-scale respect to the mobile evolution obstacles [1], [2].

Collision-avoidance algorithm [5], generated a collision free motion trajectory considered the neighbor interval respect to the obstacle place sensor data information modified the control law loads in each step.

In this paper, navigation inference algorithms combined with inference prediction tools with inference neighbor collision-avoidance implemented in Arnl software, respecting great advantages for to seek the allowed trajectory for each interval time.

4. Navigation

The navigation problem [12], as decision-making of an optimal local work satisfying the inference algorithm task; the concept of this approach could be similar to choosing the best place for next foot step in a mountain climbing in human sense. Actually has two almost completely separate processes that take place at the same time; the first process assumes that the position estimates were correct and execute a plan and the second process monitors the progress with sensor systems, predicting the obstacle condition and with corrects odometer estimates approximates to the first of its. The planning into mobile robots basically could be divided in positioning and path planning. Approaches may, may be divided in two categories according to decision-making: a) *Navigation based on global path planning* and, b) *reaction-based navigation*.

4. 1. Path Planning

A complement for autonomous navigation suggest a decision-making process called path planning, where the process must be based from the current location to the desired location given a priori knowledge of world like environment map; but, the most difficult environment were without the aid of pre-installed maps or models [14]. Furthermore, path planning process must be capable of having its preconditions changed on a continual basis without becoming disoriented. This meant that the process could be dynamic, keeping little static information about its previous plans.

Some approach refers the performance of local path planning, which deals only with part of the room around obstacle by planning genetic algorithm [8], [9]. Other technical, with known model where the mobile robot has information about the Braille block arrangement as an environmental map so that the robot can plan the path using the map [11].

Within path planning process, other situation to complete was a robot motion, if we had known the path, by defect should be haven a strategy for getting onto and tracking that path. But if the path was success, many changes were obtains at different times and circumstances, only known truth until were it moved onto desired position.

4.2. Reaction based navigation

The position and orientation of the mobile robots depends of the capability from their sensor and actuators, among these, responsive time according to the environment. Reaction based navigation had been considered more suitable for navigation in complex and dynamically changing environments, because it controls the mobile robot in on-line manner utilizing instantaneous sensor measurements; however the most powerful mobile robot was an hybrid design from reactive and path planning system.

Strategy for grid-based navigation

The rules implemented could be defined such decision-learning, when the robots starts to navigate its state could be a valid position when it moves at the next state, remember that the position was valid and created a discrete path, but when the obstacle was presented in a path, the rule decides to change and wrong position was remembered too.

Let a set of discrete points called state: $(x_{11}, y_{11})(x_{12}, y_{12}), ..., (x_{mn}, y_{mn})$ creating a virtual map from the environment and adjusted to the best path.

The robot's motion had been design for a minimum step from one degree at time, for this reason we designed a discrete direction such as north, northeast, east . . . and successively, as showed in Table 1. Furthermore each direction corresponds to binary set from three bits.

Table 1. Robot motion input set in cardinal and intermediate directions with theirs binaries values.

N	NE	E	SE	S	SW	W	NW
	*		*		*		*
		*	*			*	*
				*	*	*	*

Each output, was considered a motion in x and y axis, blend in accord with the global map, although the algorithm was building in a local map, as showed in Table 2. The "+" symbol represent increment and "-" decrement into position.

Table 2. Output robot motions set by x_n, y_n.

Motion	N	NE	E	SE	S	SW	W	NW
x_n		+	+	+		-	-	-
y_n	+	+		-	-	-		+

The algorithm routine meanly was designed with two evaluations: first process, asks, the valid path position, after it, in the second process evaluate the reach path as a global goal; when the path was invalid only change to other new function, as shown in Fig. 2.

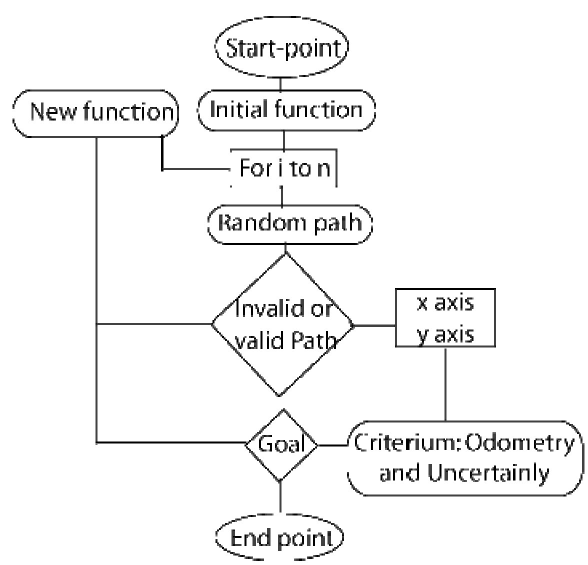

Fig. 2 General Algorithm Process

Stochastic Navigation functions tested:

$$\begin{aligned} x &= x + u(1+\omega_u) \\ y &= y + v(1+\omega_v); \end{aligned} \qquad (1)$$

$$\begin{aligned} x &= x - u(1+\omega_u) \\ y &= y - v(1+\omega_v); \end{aligned} \qquad (2)$$

$$\begin{aligned} x &= x + u(1+\omega_u) \\ y &= y - v(1+\omega_v); \end{aligned} \qquad (3)$$

$$\begin{aligned} x &= x - u(1+\omega_u) \\ y &= y + v(1+\omega_v); \end{aligned} \qquad (4)$$

With Normal Distribution function:

$$\{\omega_v,\ \ \omega_u\} \subseteq N(|\mu| < 1, \sigma^2 < \infty).$$

4.3 Autonomous Navigation

4.3.1. Controllability

In mobile robot, an important fact for decision-making is a good response from mechanicals and electronics devices, these kinds of limited behavior put the truth rules considering finite valid motion set into the whole rules, but to solve locomotion problems, the mobile robot algorithms need the inferences so that adjust the parameter load energy to mechanism changing the inertial condition; through the fuzzy control strategies.

4.3.2 Observability

For each mobile robot process requires autonomous behavior, necessarily must be use instrumentation and measurement equipment to obtain information for specific environments [3] through a set of sensor. While, more information had the system, decision-making process carries out the precise path, respect to inference rules.

4.3.3. Complexity

The complexity of the navigation processes depends of the general context, in which the tasks set were executed [10], but in a compute represents a relationship between inputs and output system.

4.3.4. Calculability

The capacity from taking an action, were stimulated by an environment (inputs signal sensors), such as position, localization, state, status, and others, describing a parameters set from path possibilities (outputs) through the algorithms schemes with inference decision, as well exist two reference conditions: global reference considered a parameter concerned to robot in a Cartesian measure and; local reference used the integrate data into knowledge base (KB) from sensors into the intelligent robot sense. Therefore, the natural coordinate in this kind of systems use a polar description.

4.3.5. Odometry

In relative information the odometry considered how it had been moved, but not where did it move. If it knows where its starts, then it could add the relative movement to the starting position and obtain an estimation of its current position. However, if the mobile robot accumulated all errors considered in its motions, then it could be subject to drift. So that, the odometer technical into inference rules considering the position estimates in accord to floor tile position.

4.3.6. Uncertainly

A mobile robot never was exacted in position estimate in deterministic sense. Each time the odometry advances the mobile robot position estimates respect to measure sensors containing always the little errors. Establishing into the KB a minimum uncertainly levels. Maintaining the stability and integrity, such as, the distance from obstacles along the path planning is very significative. In the real world, often is necessary to set up the constraints that the algorithm satisfies the inference mechanism in order to be used. The concept of constraints is different from the performance metrics described above: while the latter can be used to optimize the performance or to measure the differences of two algorithms, the former have to be considered strict needed of application at hand: if the robot motion did not satisfy one of these constrains, the mission is declared failed.[7].

For mobile robot tested was used a sonar data and wheel information. Sonar data has an accuracy from $\pm$ 10 mm, which is important to know; furthermore the accuracy in a motion of the wheel was from $\pm$ 1 mm, as shown in Fig, 3; in this way, we tested the error ω, with different values founding that five to fifty percent were the best and was concatenated to random functions.

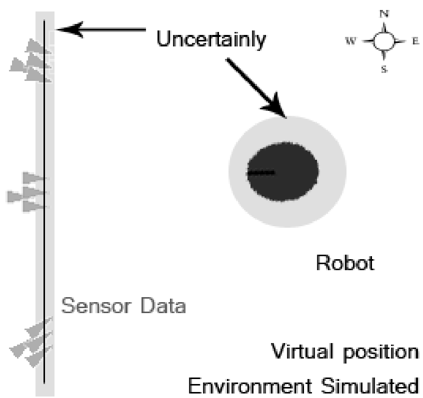

Fig. 3. Uncertainly in motion and sonar data with respect cardinal position.

5. Navigations Methods

The most convenient autonomous navigation methods, as a function time, space, complexity, calculability, contrability, observability, and others, could be converges to desired goal. Hence some simulator makes possible for robotics researchers to debug their code and experiments conveniently for achieving the goal. Likewise useful simulators, works with ideal conditions without noise and other adverse conditions.

Some methods uses functions such as environment model-building (walls, obstacles), driving system and noise level, observation model setting, Real Time display of the running processing and observations [6], others, as a grid robot is a virtual fuzzy base situating into environment made of square grids [4], until a concept like the concept of fuzzy navigation with natural interpretation.

6. Desired goal

When somebody in an earthquake or terrorist attack occurs in a building or place and it collapse, the human being danger, because loses the orientation and position with respect to environment known. This hypothetic condition unfavorable to each person had the capability from orientation respect to local information. However some could be valid and other invalid path with respect the best trajectory.

Now, in this unfavorable context, change the person by autonomous mobile robot, which try to find the best conditions and had successful to exit from the dangerous place, without tool from global position, only with prior knowledge floor tile. The indoor environment known, but the obstacle and the robot start-position appears in random form, with finite conditions and absolute compute.

The random path found with $0.5 \leq \omega \leq 0.15$, simultaneously with the function (1), depicted in Fig. 4, with respect to obstacles avoidance was complete, but the path failure near to desired goal.

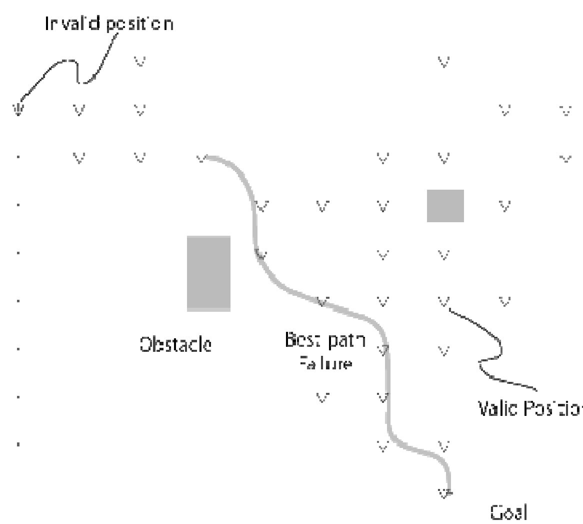

Fig. 4. Path planning with $0.5 \leq \omega \leq 0.15$, and $x = x + u(1 + \omega_u)$, $y = y + v(1 + \omega_v)$;

The function number (1) found the free-collision path, but for this case, the best path failure until one thousand interactions. Other path tested with equal threshold ω, using (2) obtaining a collision and the worst path, as showed in Fig, 5

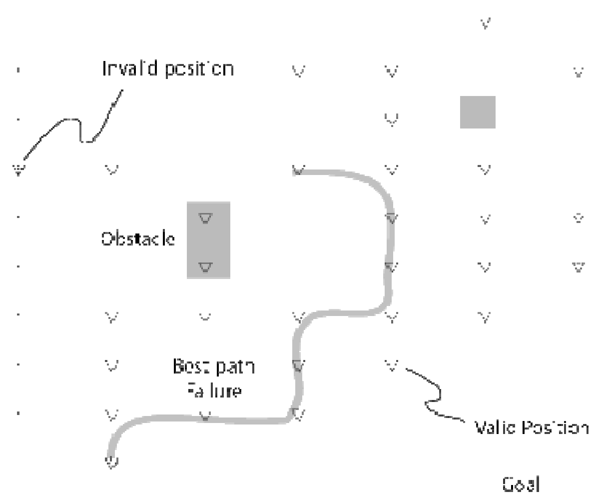

Fig. 4. Path planning with $0.5 \leq \omega \leq 0.15$, and $x = x - u(1 + \omega_u)$, $y = y - v(1 + \omega_v)$;

The best path was found with the function (4) into interval $0.5 \leq \omega \leq 0.15$ from noise perturbation, as showed in Fig. 5.

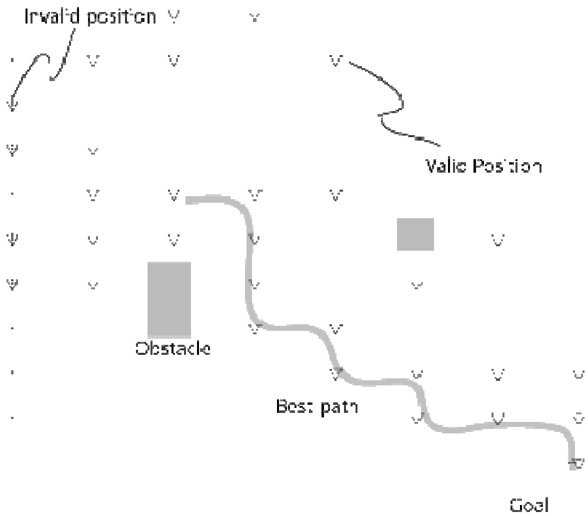

Fig. 5 Path planning with $0.5 \leq \omega \leq 0.15$, and $x = x + u(1 + \omega_u)$, $y = y - v(1 + \omega_v)$;

Taking the experimental environment with the obstacles unknown, the algorithm reach desired goal up to seven hundreds interactions and inferences computed, but only twenty different positions from more than one hundred positions, into the KB, the path calculated is to memorized like a start to goal, and the next travel only used the minimum disturbing motion (best-path), as showed in Table 3; the criteria used was the fuzzy Dijsktra's algorithm.

Table 3. The best-path trajectory occurred after thirty interactions with decrement by shortest path.

SE	S	SE	SE	SE	SE
+		+	+	+	+
-	-	-	-	-	-

Also was tested the function (3), having a good free-collision path, but in the same case from (1) the desired goal path failure, however it did not said that the functions wrong in other environment, goal or star point; only ensure that the a property mixture from their trough the algorithm process and with the shortest attempt to reach the success. Mobile robot behavior from each function implemented in fuzzy inference algorithms, in accord to start point up finishing the point represented in a comparative graphic, showing in Fig. 6. Besides from their behavior, the shortest path could see as quantity from point behind to the line representing the finished point.

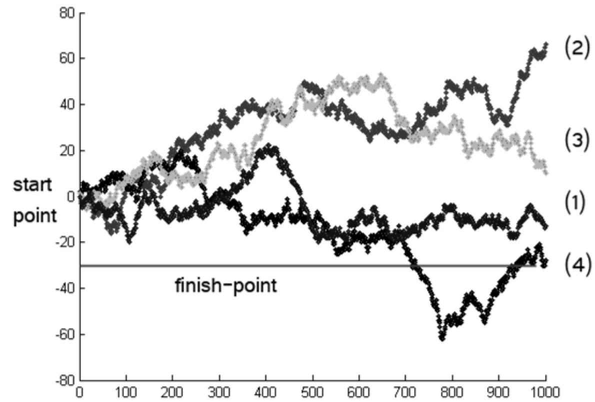

Fig. 6 Representation of fitness functions.

6.1 Real environment implement

A real environment was building in a specific place where floor tiles had been making from 0.3 m x 0.3 m., each floor tile represents a cell or state path, where took the cells like a reference from local position, showed in Fig. 7. Additionally the 3-Dx mobile robot, obstacles and wall behind it shows the simple space trajectory.

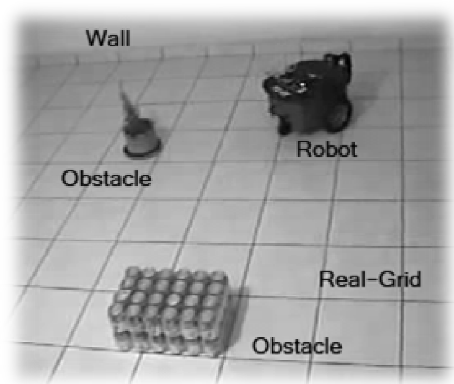

Fig. 7. Illustrative real environment implemented in an experimental space with different kind obstacles.

Such as simulation described, the algorithm as implemented in P3Dx mobile robot with the path computation, with a velocity from 0.1 m/s was experimented, and the goal was achieved with a deviation from 50 mm to x axis and 30 mm in y axis, however these measures were within valid tolerance. Different velocity from 0.5 m/s the path crashed with the environment, for this reason the path building results good in a low velocity, also was tested with a lowest velocity, but the deviation

ISSN: 1790-5117

ISBN: 978-960-474-054-3

change to 40 mm to x axis and 35 mm in y, respectively.

The last test, showed in Fig. 7, describe how the mobile robot had achieved the goal with lowest velocity $v = 0.05\ m/s$ and function (4) in a floor tile in a room with obstacle avoidance.

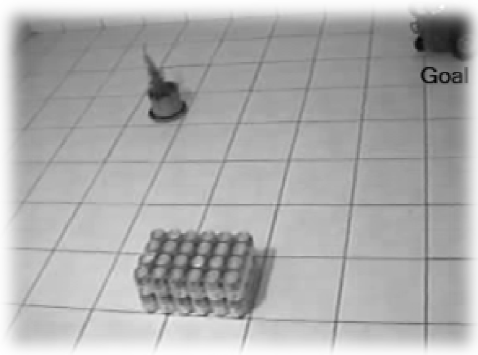

Fig. 8 Autonomous Navigation from 3Dx in a room with floor tile achieving to goal.

7. Conclusion

This paper showed a fuzzy algorithm for path planning to reach an exit loaded in unknown environment bounded with obstacles avoidance. The fuzzy algorithm implemented in a Pioneer 3-DX with the Arnl tools, the grid-map navigation from two dimensional spaces limited by floor tiles adjusted respect to the real environment used a set of inferences rules. The propose from path planning algorithm was to achieve a path for anywhere position or orientation exited with autonomous decision bounded in time and landmark motion as intelligent system. In this specific case the simulation required to have known the environment, but without known random position obstacles. The position reached by sonar sensor and defined target considered the odometric orientation and position in a grid-based with motion on cardinal and intermediate direction as long as the environment allowed to follow the path; so that, it only could move into valid position with free obstacles; the strategy represented by random paths, one to one at each floor tile and its fitness was evaluated only when the path achieve the desired goal. The simulation results illustrated the trajectory path that was possible with random obstacles, given a good enough strategy respect the sonar information in one time interval.

References

[1] T. Duckett, U. Nehmzow. "Mobile robot self-localization and measurement of performance in middle-scale enviroments", Robotics and Autonomous Systems 34, 2001, p.p. 117-129.

[2] U. Nehmzow, C. Owen. "*Robot navigation in the real world: Experiments with Manchester's forty two in unmodified, large environments*", Robotics and Autonomous Systems 33, 2000, p.p. 223-242.

[3] T. Duckett, U. Nehmzow. "*Mobile robot self-localization using occupancy histograms and a mixture of Gaussian location hypotheses*", Robotics and Autonomous Systems 34, 2001, p.p. 117-129.

[4] T.W. Manikas, K. Ashenayi, R. L. Wainwright. "*Genetic Algorithms for Autonomous Robot navigation*", IEEE Instrumentation & Measurement Magazine, 2007, p.p. 26-31

[5] D. A. Ashlock, T. W. Manikas, K. Ashenayi. "*Evolving A Diverse Collection of Robot Path Planning Problems*". IEEE Congress on Evolutionary Computation, 2006, p.p. 1837-1844.

[6] D. E. Koditschek. "*Autonomous Mobile Robots Controlled by Navigation Functions*". IEEE International Workshop on Intelligent Robot and Systems 1989, p.p. 639-645.

[7] J. Shen, Huosheng Hu. "A matlab-based Simulator for Autonomous Mobile Robots" University of Essex. UK.

[8] D. Calisi, L. Iocchi, D. Nardi. "A unified benchmark framework for autonomous Mobile robots and Vehicles Motion Algorithms (MoVeMA benchmarks)" University of Rome.

[9] K. H. Sedighi, K. Ashenayi, "Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm" University of Tulsa.

[10] A. Hermanu, T. W, Manikas, "Autonomous robot navigation using a genetic algorithm with an efficient genotype structure" University of Tulsa, Oklahoma

[11] R. Chatila, S. Lacroix, "Autonomous Mobile Robot Navigation for Planet Exploration The EDEN Project" LAAS-CNRS.

[12] T. Yoshida, A. Ohya, S. Yuta. "Autonomous Mobile Robot Navigation Using Braille Blocks in Outdoor Environment", University of

Tsukuba

[13] C. Buschmann, F: Müller, S. Fischer. "Grid-Based Navigation for Autonomous, Mobile Robots". Technical University of Braunschweig

[14] U. Nehmzow, C. Owen. "Robot navigation in the real world: Experiments with Manchester's forty two in unmodified, large environments", Robotics and Autonomous Systems 33, 2000, p.p. 223-242

[15] J. C. Latombe. Robot Motion Planning, Kluwer Academic Publisers, Boston, MA, 1991

[16] Kim, H.S.; Choi, W.H.; Kang, H.J.; Lee, K.C. "State estimation for autonomous guided vehicle using the extended Kalman filter"Control Conference, 2004. 5th AsianVolume 1, Issue, Page(s): 405 – 411.