2011 IEEE/ASME International Conference on
Advanced Intelligent Mechatronics (AIM2011)
Budapest, Hungary, July 3-7, 2011

Avoiding deadlocks of mobile robots in narrow passages of environments populated with moving obstacles

Marija Đakulović, Borko Dimitrijević and Ivan Petrović

Abstract— **This paper presents a strategy for deadlock avoidance of mobile robots in narrow passages of environments populated with other moving objects. The proposed strategy detects deadlocks in narrow passages only by robot's perceptive sensors, i.e., no other communication means with moving objects is assumed. The strategy is based on the random multi-access algorithm for the network congestion avoidance. The strategy is implemented within our existing motion planning and control system for mobile robots and thoroughly tested by simulation and experimentally on the Pioneer 3DX mobile robot equipped with SICK LMS-200 laser range finder. The test results illustrate the appropriateness of the proposed strategy for resolving deadlocks in narrow passages.**

I. INTRODUCTION

An autonomous mobile robot is expected to perform goal directed tasks in dynamic environments populated with other moving objects such as mobile robots, animals and human beings. The mobile robot motion planning system has to avoid deadlocks or path conflicts with other moving objects. A deadlock is a possible situation in path planning in which a solution cannot be found, even though one exists.

The deadlock avoidance problem is usually solved as the part of the multi-robot path planning and motion coordination problem under assumption that the environment is populated only with multiple autonomous mobile robots. The deadlocks in such systems are typically caused by robots blocking each other's paths, and the planner being unable to find a solution in which robots move out of each other's way [1]. The existing methods for solving the problem of motion planning for multiple robots can be divided into two categories [2]: *centralized* and *decentralized approaches*.

The centralized approaches combine the configuration spaces of all individual robots into one composite configuration space, to which classical single-robot path planning algorithms are applied [3]. However, these planning approaches require computation time that is exponential in the dimension of the multi-robot configuration space.

Unlike the centralized approaches, in the decentralized approaches, the path for each robot is planned individually, followed by a certain strategy for resolving possible deadlocks with paths of other robots. The decentralized approaches are very efficient since they avoid hard combinatorial planning problems of centralized approaches, but they suffer from two main drawbacks: (1) they are incomplete in the sense that they do not guarantee finding a solution even if one exists; and (2) the resulting solutions are often not optimal. The decentralized approaches can further be classified as *coupled* or *decoupled*, depending on the level of coordination among the robots in resolving the deadlocks.

The coupled decentralized approaches plan and coordinate the paths of the robots explicitly in advance. The most popular are the prioritized planning approaches, which plan the paths in the configuration time-space for each robot in prioritized order, considering other robots of higher priority as moving obstacles at every point in time [4], [5], [6]. How to assign the priorities to the individual robots has a serious influence on whether at all a solution can be found and how long the resulting paths are [7].

The decoupled decentralized approaches relax multi-robot path planning giving emphasis to the coordination of robots' motions in real-time using reactive, behavior-based, or control-theoretic approaches. Approach based on the ant colony behavior [8] belongs to decoupled decentralized approaches. The ant colony approach first computes separate paths for individual robots without considering other robots paths in path planning. Then, prioritized rules are employed locally in coordinating the robots during movement to avoid deadlocks in paths. This approach makes the assumption that robots can detect each other by sensors and find out their positions and velocities. Thereupon, the higher priority is assigned to the robot with the higher velocity. Numerous behavior-based decoupled decentralized approaches apply decision theory for obstacle avoidance and deadlocks preventing [9], [10]. In [11] coordination graphs (introduced in [12]) are used for deadlock avoidance in the narrow passages. For each narrow passage a coordination graph is created. The main drawback of this method is high computational cost, which grows with the number of narrow passages.

The above described multi-robot path planning and coordination approaches assume that the environment is populated only with multiple mobile robots, which paths can be somehow coordinated. Therefore, they do not solve deadlocks caused by other moving objects such as human beings, animals and moving objects with unpredictable behaviors. Research of methods for robot's navigation among humans and for human-robot interactions has also been conducted. For example, the robot RHINO [13] is designed to give interactive tours through an exhibition in a densely populated museum. The robot RHINO has a full hierarchical structure of separate algorithms, which solve individual subtasks. The problem of deadlocks is solved by selecting alternative

This research has been supported by the Ministry of Science, Education and Sports of the Republic of Croatia under grant No. 036 – 0363078 – 3018.

M. Đakulović, B. Dimitrijević and I. Petrović are with Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, University of Zagreb, Croatia {marija.dakulovic, borko.dimitrijevic, ivan.petrovic}@fer.hr

978-1-4577-0837-4/11/$26.00 ©2011 IEEE

local path, or if deadlocks are successive and frequent, by calculating a new global path. Similar behavior has the tour-guide robot MINERVA [14].

This paper presents a new strategy which resolves deadlock problems in narrow passages generally, i.e. for any type of moving objects causing the deadlocks. In order to achieve the generality of the proposed strategy it is assumed that the robot cannot communicate with moving objects by any means except perceiving their motion by the on-board perceptive sensors. The proposed strategy for resolving deadlocks is based on the random multi-access algorithm for the network congestion avoidance [15]. Test results obtained with systematic simulations and experimentally with a real robot illustrate the appropriateness of the proposed strategy in resolving deadlocks in narrow passages.

The rest of the paper is organized as follows. Section II reviews used algorithms for environment map and robot representation, narrow passages detection and the robot motion planning. Section III presents the proposed deadlock avoidance strategy and Section IV test results. Section V concludes the paper.

II. Problem statement

Here we first present used occupancy grid map and robot representation. The algorithm for automatic creation of hierarchies of abstraction presented in our previous work [16] is used to identify narrow passages in the occupancy grid map. The part of the algorithm for narrow passages extraction is here revised. At the end of this section robot motion planning is shortly described.

A. Used occupancy grid map and robot representation

An occupancy grid map is created by approximate cell decomposition of the environment [17]. The whole environment is divided into squared cells of equal size e_{cell}, which are abstractly represented as the set of M elements $\mathcal{M} = \{1, \ldots, M\}$ with corresponding Cartesian coordinates of cell centers $c_i \in \mathbb{R}^2$, $i \in \mathcal{M}$. Each cell contains occupancy information of the part of the environment that it covers.

We use weighted occupancy grid map introduced in our previous work [18]. A weighted occupancy function $o(i) \in \{1, 2, \ldots, M_c + 1, \infty\}$, $i \in \mathcal{M}$ is used for representing the set of all obstacles in the environment noted as $\mathcal{O} = \{i \in \mathcal{M} \mid o(i) = \infty\}$ and unoccupied environment is represented by the set of unoccupied cells noted as $\mathcal{N} = \mathcal{M} \setminus \mathcal{O}$. Occupancy function is defined as follows:

$$o(i) = \begin{cases} \max\{1, (M_c + 2 - \min_{j \in \mathcal{O}} \|c_i - c_j\|_\infty)\} & \text{if } i \notin \mathcal{O} \\ \infty & \text{if } i \in \mathcal{O} \end{cases} \tag{1}$$

Described procedure generates the so-called safety cost mask around obstacles with smooth decrease of occupancy values from the obstacles towards the free space. The size of the safety cost mask is defined by the predefined integer number of cells M_{c}. The weighed occupancy grid map with $M_{\text{c}} = 4$ cells wide safety cost mask of the section of the experimental environment is shown in Fig. 1.

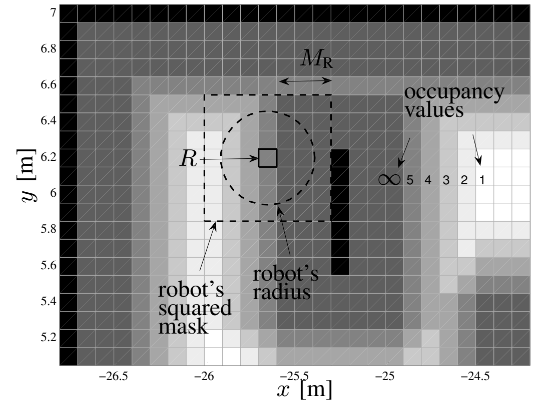

Fig. 1. The occupancy grid map with the safety cost mask ($M_{\text{c}} = 4$) and the robot squared mask ($M_{\text{R}} = 3$).

We assume that real shape of the mobile robot can be approximated by a circle of a radius r_r, which is very often used assumption in the literature. In that case the robot is represented by a squared mask in the grid map, within which the robot's real circular shape can be drawn. To allow the robot to be located within any unoccupied cell, all obstacles in the grid map are enlarged for the integer number of cells M_{R}, i.e. the robot is described by a squared mask of size $2M_{\text{R}} + 1$. Real shape of the robot and its squared mask in the occupancy grid map are depicted in Fig. 1. The robot's position is considered to be the cell R. The real obstacle placement is shown by black color.

B. Automatic detection of narrow passages

Narrow passages are extracted from the occupancy grid map by the algorithm introduced in our previous work [16]. The algorithm is here shortly revised.

Cartesian coordinates of the cell centers $(x, y) \in \mathbb{R}^2$ can be transformed into the integer coordinates of the occupancy grid map $(i, j) \in \mathbb{N}^2$ as $(i, j) := \left(\left\lceil \frac{x - x_0}{e_{\text{cell}}} \right\rceil, \left\lceil \frac{y - y_0}{e_{\text{cell}}} \right\rceil \right)$, where x_0 and y_0 are the smallest coordinates of the environment (the origin is the left bottom corner of the map). Therefore, the cell $n \in \mathcal{M}$ has real coordinates of the center $c_n \equiv (x_n, y_n) \in \mathbb{R}^2$, and integer coordinates $(i_n, j_n) \in \mathbb{N}^2$. Occupancy function is considered as a function of two variables $o : \mathbb{N}^2 \mapsto \{1, 2, \ldots, M_c + 2\}$. Notice that the value $M_c + 2$ is used instead of ∞. Three dimensional view of the function o is given in Fig. 2. Note that according to (1) occupancy values of two neighbor cells can differ maximally by one.

The narrow passage is defined as the set of cells in which local minimum along one axis of the function o is present, and value of the function o is larger than 1 (see Fig. 2). The largest width of such defined narrow passage is equal to $2(M_c + M_R)$. We have chosen for M_c to be equal to $M_R + 1$ and, therefore, the widest narrow passage will enable two robots to pass through it at the same time if both robots travels as close to the wall as possible. The candidate cells of local minima are determined for $\forall (i, j) \in \mathbb{N}^2$ according to

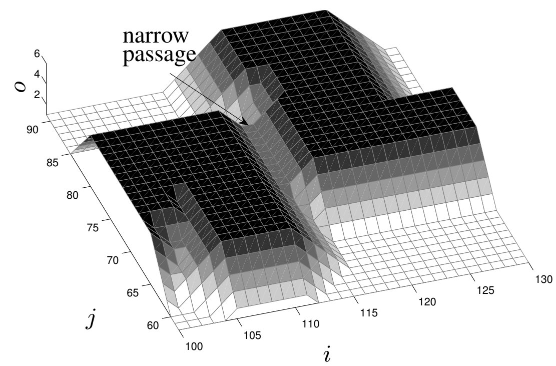

Fig. 2. The part of the environment in 3D view with the values of the occupancy function o on the x-axes.

the values of the second differential along i and j separately:

$$o_i''(i,j) = o(i-1,j) - 2o(i,j) + o(i+1,j) > 0, \\ o_j''(i,j) = o(i,j-1) - 2o(i,j) + o(i,j+1) > 0. \quad (2)$$

However, the second differential is not sufficient to find the local minima, i.e., the narrow passage, the following conditions must be taken into account, for clarity written only for the local minima along i-axis:

$$\begin{aligned} &\forall n \in \mathcal{N} \text{ with coordinates } (i_n, j_n) \in \mathbb{N}^2 \\ &\text{such that } o_i''(i_n,j_n) > 0 \text{ and } v \leftarrow o(i_n,j_n) > 1 \\ &\text{if } o(i_n-1,j_n) = v+1 \text{ and } o(i_n+1,j_n) = v+1 \\ &\text{or } o(i_n-1,j_n) = v+1 \text{ and } o_i''(i_n+1,j_n) > 0 \\ &\text{or } o(i_n+1,j_n) = v+1 \text{ and } o_i''(i_n-1,j_n) > 0 \\ &\text{then } \mathcal{U} \leftarrow \mathcal{U} \cup \{n\}, \end{aligned} \quad (3)$$

where the set of cells $\mathcal{U}$ is used to describe all narrow passages in the grid map, starting from the empty set. According to the conditions in (3) one or two neighbour cells are allowed to have minimal value along one axis. The results of finding narrow passages in the part of the environment represented by occupancy grid map with cost mask is shown in Fig. 3.

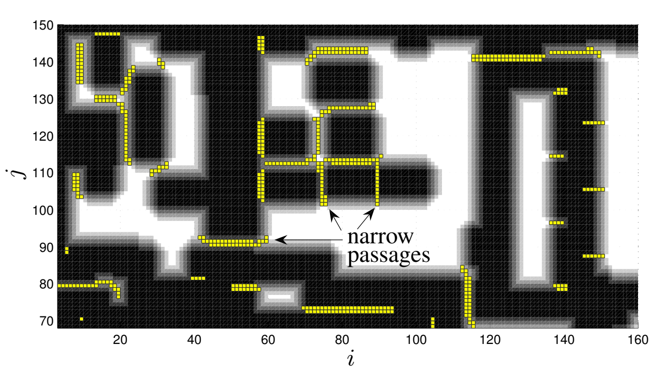

Fig. 3. The narrow passages determined in the occupancy grid map with cost mask.

C. Robot motion planning

The robot motion planning algorithm is composed of path planning and obstacle avoidance module. A list of global goals is given, which the robot must visit orderly. The path from the robot's position to the given goal is calculated by the D* algorithm [19] – the well known graph search algorithm capable of fast path replanning in changing environments. The path is replanned as the robot moves each time the robot's sensors detect nodes with changed occupancy values. The path is followed by the integration of the D* algorithm and the dynamic window obstacle avoidance algorithm, as described in our previous work [20]. It is assumed that the robot can not communicate with detected obstacles.

III. Deadlock avoidance strategy

The deadlock avoidance strategy searches the local area around a detected deadlock and by small movements tries to solve the deadlock. Of course, it is possible that no solution exists, so it is necessary to limit the number of retries.

The deadlock avoidance strategy is composed of three steps. The first step of the deadlock avoidance strategy is to detect a deadlock. To detect a deadlock the robot must observe the narrow passage by its sensor. In the deadlock situation the planner will declare that no path is found. The second step is to select a standstill position from which the robot will safely view the state of the narrow passage. This position is called the waiting position. The third step includes waiting for the random number of time intervals and then recalculating the path. The random time is chosen to avoid more robots to retry at the same time causing a possible new deadlock in their paths. This procedure is taken from the random multi-access algorithm for the network congestion avoidance [15]. These three steps of the proposed strategy are described in the following.

A. Detection of the deadlock situation

The robot by its sensors detects which cells in the grid map changed its occupancy from empty to occupied. All new obstacle cells are enlarged for the robot's dimensions. If these cells are the elements of the set $\mathcal{U}$, it means that newly discovered obstacle is in the narrow passage. The deadlock in narrow passage can occur in two cases:

(1) the obstacle in the narrow passage blocks the global robot's path (Fig. 4a).
(2) the robot is in the narrow passage and the obstacle blocks the global robot's path (Fig. 4b).

The case (2) describes a special deadlock situation in which a deadlock is not present in the narrow passages but must be processed since the robot in the narrow passage can cause the case (1) to another robot.

B. Selection of the waiting position

When the robot is halted in the deadlock situation it could block paths to other moving objects. Therefore, the safe position must be chosen from which the robot can safely view the position of the blockage in its path and at the same time allow passing other moving objects. Let assume that the cell in which the robot is currently positioned is noted by R, and the closest occupied cell in the blocked path to the cell R is noted by Z. The waiting cell P is selected according to two deadlock cases. In the case (1) the waiting cell P

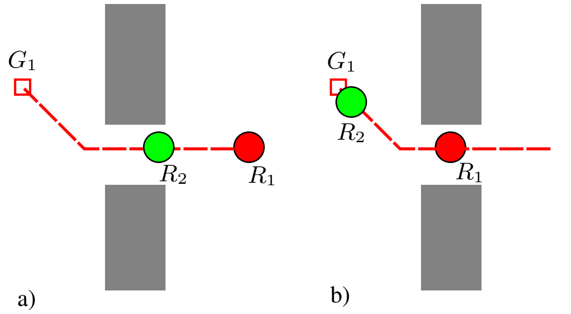

Fig. 4. The deadlock situation: a) the moving object R_2 in the narrow passage blocks the path of the robot R_1 (dashed line from R_1 to the goal G_1); b) the robot R_1 is in the narrow passage and the moving object R_2 blocks the path of the robot R_1.

is selected as any cell (at random) from the set of candidate waiting cells $\mathcal{A}$ (Fig. 5a). Starting from the empty set, the set $\mathcal{A}$ is determined by adding the cell $n \in \mathcal{N}$ if the following conditions are fulfilled:

$$\begin{aligned}&\forall n \in \mathcal{N}\\ &\text{if } \|c_n - c_Z\|_\infty \leq S_{max} \text{ and } \|c_n - c_R\|_\infty \leq \|c_Z - c_R\|_\infty\\ &\text{and } \forall m \in \mathcal{U}\ \|c_n - c_m\|_\infty > M_R \text{ and } \textbf{visible}(c_n, c_Z),\\ &\text{then } \mathcal{A} \leftarrow \mathcal{A} \cup \{n\}.\end{aligned} \tag{4}$$

The conditions in (4) say that candidate waiting cell is distanced from the obstacle cell Z for less than sensor maximal range S_{max} and must be at the right side of the passage (the second condition). Furthermore, it must not be near the narrow passage (for the robot's dimensions) and the cell Z must be visible from the candidate waiting cell, i.e., the line connecting c_n and c_Z must not intersect the real obstacle (non-enlarged obstacle). Infinity norm is used for simplicity of determining the area A.

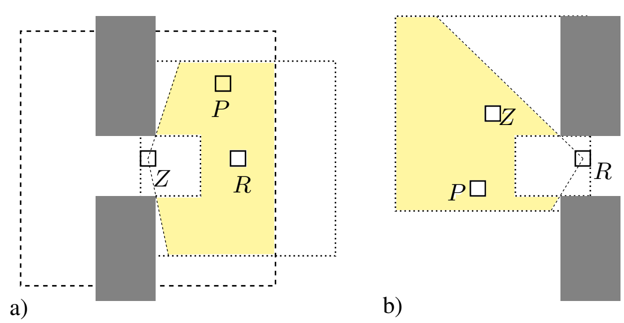

Fig. 5. The selection of the waiting cell P from the set A (shaded area): a) the occupied cell Z in the blocked path is in the narrow passage; b) the robot cell R is in the narrow passage and the occupied cell Z in the blocked path is not in the narrow passage.

In the case (2) the waiting cell P is selected similarly, but the set A is determined according to the constraint $\|c_n - c_Z\|_\infty < \|c_Z - c_R\|_\infty$ instead of the constraints $\|c_n - c_Z\|_\infty \leq S_{max}$ and $\|c_n - c_R\|_\infty \leq \|c_Z - c_R\|_\infty$ in (4) (Fig 5b). In very crowded areas it is possible that the cell P becomes occupied very soon. Then the procedure of the selection of the waiting cell P is repeated.

C. Waiting and retrying

The new path is calculated to the waiting cell P (this cell is set as the new local goal for the path planning module) and the robot is moved to its new local goal position. After reaching the local goal position the robot rotates until it achieves the orientation towards the occupied cell Z in the blocked global path. The robot waits in the waiting cell P for the random number of time intervals $T_z \in \{0, \ldots, T_{zmax}\}$. While the robot is waiting it is constantly viewing the state of the occupancy of the cell Z. The passage can become free at the time the robot travels to the waiting cell P. However, the waiting of random number of time intervals is necessary for solving a situation in which two robots controlled with the same algorithm, blocks each other paths. In that case the robots will move to theirs waiting cells and the passage will become free. Then, both robots will start moving to their global goals at the same time, and the deadlock will happen again.

At the waiting cell P, if the state of the passage is constantly occupied and there exist alternative global path to the global goal, after T_z numbers of time intervals the robot leaves the first global path to the global goal and takes the new one. If the state of the passage has become free after T_z numbers of time intervals, the robot calculates again the global path to the global goal, and start moving towards the global goal. If there is no alternative global path to the global goal and if the path is still blocked, after T_z numbers of time intervals it selects new random number from the twice as wide set of possible intervals $T_z \in \{0, \ldots, 2T_{zmax}\}$. Further, each new try doubles the set of possible intervals. After maximal preset number of tries N it leaves the first global path and chooses the next global goal.

IV. Test Results

Systematic robot simulations and experimental results obtained with a Pioneer 3DX mobile robot are chosen to illustrate the appropriateness of the proposed strategy in solving deadlocks in narrow passages. The laser range finder SICK LMS200 mounted on the robot was used for environment perception. It scans the environment in radial range of $\pm 90°$ with resolution of $1°$ and sends to the robot 181 uniformly distributed distances to the detected obstacles every 100 ms.

A. Simulation results

The simulation setup of two robots controlled by the same algorithm is shown in Fig. 6. There are presented four simulation snapshots. The robots follow their global paths to the given global goals: the green (light gray) robot follows the green (light gray) path, and the red (dark gray) robot follows the red (dark gray) path. When the robots detect each other by theirs sensors, the new global paths are calculated around detected obstacle cells (snapshot 1). Shortly after, the robots block global paths to each other (due to enlargement of obstacles for robot dimensions in the occupancy grid map). The details of this moment from the red and green robots' points of view are shown in Figs. 7a

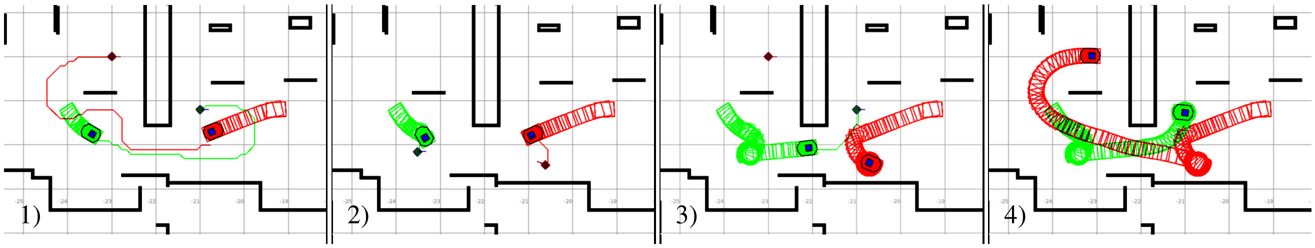

Fig. 6. Avoiding deadlock caused by two robots in the narrow passage.

and 7b, respectively. The initial paths are noted by dashed lines, and the new blocked paths are noted by solid line. Real obstacles are noted by black squares, and detected robots by red/green colored squares with black borders. Obstacles are enlarged for the robots' dimensions and safety cost values are calculated around them. The robot's position cell is noted by R and the closest occupied cell in the blocked path to the robot is noted by Z. The set $\mathcal{U}$ is recalculated according to (3) for the new obstacle configuration and is noted by small squares in narrow passages. The new candidate waiting positions are calculated according to (4) and noted by stars. The chosen candidate position in noted by P. The waiting

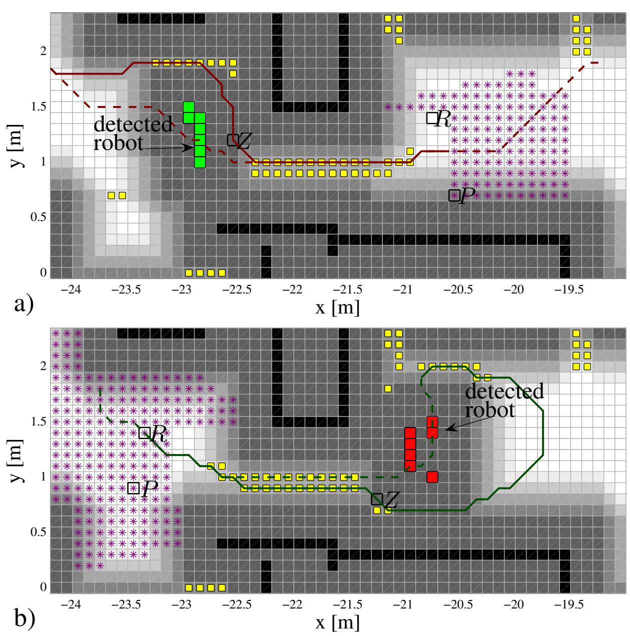

Fig. 7. Deadlock detection and selection of the waiting position in the occupancy grid map: a) blockage of the red robot's path; b) blockage of the green robot's path.

cells P are determined and given as new local goals (noted by rhombus in snapshot 2). The orientations of the local goals are pointed in the direction of the occupied cell in the blocked path (noted by small lines). After reaching the waiting positions, each robot waits random time interval. The result is that the green robot waits shorter time interval than the red one. Since the state of the passage has become free the green robot calculates again the global path to the global goal, and start moving towards the global goal (snapshot 3). From the red robot point of view, the passage is not free and it waits another random time interval from the twice as wider set of time intervals. After second waiting time interval, the passage becomes free and the red robot moves along recalculated path to its global goal (snapshot 4).

B. Experimental results

The experiment shows a case of real robot navigation in an indoor environment among moving objects – the people are passing through the doors, the doors are opening and closing. The experimental setup which the deadlock avoidance strategy must solve is composed of three parts: choosing the alternative global path, giving up from the current global path, and passing through the door that was closed.

The experimental environment is shown in Fig. 8 together with the robot's paths and trajectories and sensor readings. The robot uses the occupancy grid map of the environment in which all doors are open. During the experimental setup the passages 1 and 2 remain occupied, and the passage 3 changes its state from occupied to free, i.e. the closed door becomes open. Two global goals are given to the robot, named G_1 and G_2. The robot will give up from the global goal G_1 and will reach only the global goal G_2. The initial robot's global path to the global goal G_1 passes through the passage 2, which is blocked by two persons, and there exists the alternative global path (dashed line) through the passage 3, which has closed door.

The result of this experimental setup is as follows. The robot follows the global path P_1 until it detects by its sensors that the passage 2 is closed. Then, it goes to the waiting position and wait a random time interval. Since the passage 2 is constantly occupied the robot leaves the first global path and takes the alternative global path P_1'. While following the global path P_1', the robot detects that the passage 3 is occupied, waits at the waiting position, and chooses again the first global path P_1. This switching of paths P_1 and P_1' continues until maximal number of retries N is reached ($N = 4$ in the experiment). Then, the robot gives up from the global goal G_1 and chooses the new global goal G_2. The global path to the goal G_2 passes through the passage 1, which has closed door. The robot detects that the passage 1 is occupied and waits at the waiting position for a random time interval. Very soon, the door opens and after the waiting time interval the robot continues towards the global goal G_2. Traveled path is shown by devious line, and laser range data from different time instant by different colors, therefore, one can detect trails of doors opening in the passage 1 and people movement.

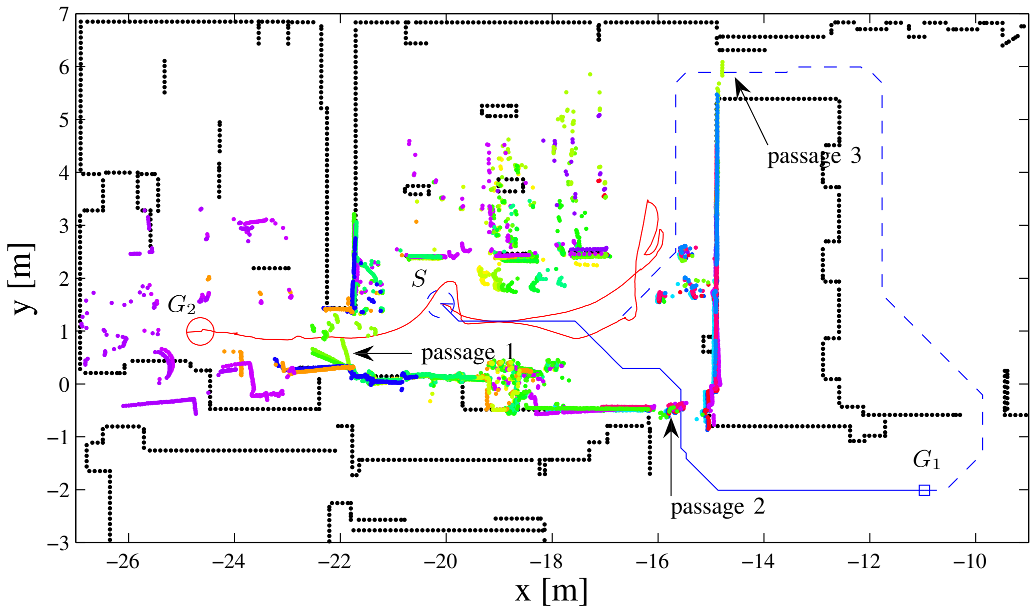

Fig. 8. Avoiding deadlocks in the real environment.

V. CONCLUSIONS

This paper presents a deadlock avoidance strategy, which ensures the long term moving of mobile robot in the environment populated with moving objects. The strategy resolves deadlocks in situations when the robot and some other moving obstacle (another robot, person) meet each other in narrow passages and block each others paths. Assuming the strictest constraint that the robot can not communicate with other moving obstacles but can only perceive their behaviors by its sensors, the proposed strategy based on the random multi-access algorithm for the network congestion avoidance, successfully solves deadlocks in narrow passages. Functionality of the proposed strategy is confirmed both by simulations and experiments on a real mobile robot in an office-like environment.

REFERENCES

[1] L. Parker, "Path Planning and Motion Coordination in Multiple Mobile Robot Teams," *Encyclopedia of Complexity and System Science*, 2009.

[2] J. Latombe, *Robot Motion Planning*. Dodrecht, Netherlands: Kluwer Academic Publishers, 1991.

[3] J. Lee, B. Lee, and M. Choi, "A real-time traffic control scheme of multiple AGV systems for collision free minimum time motion: a routing table approach," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, no. 3, pp. 347–358, 1998.

[4] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Algorithmica*, vol. 2, no. 1, pp. 477–521, 1987.

[5] S. Buckley, I. Center, and Y. Heights, "Fast motion planning for multiple moving robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1989, pp. 322–326.

[6] C. Oliver, M. Saptharishi, J. Dolan, A. Trebi-Ollennu, and P. Khosla, "Multi-robot path planning by predicting structure in a dynamic environment," in *Proceedings of the First IFAC Conference on Mechatronic Systems*, vol. 2, 2000, pp. 593–598.

[7] M. Bennewitz, W. Burgard, and S. Thrun, "Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots," *Robotics and Autonomous Systems*, vol. 41, no. 2-3, pp. 89–99, 2002.

[8] S. Liu, L. Mao, and J. Yu, "Path Planning Based on Ant Colony Algorithm and Distributed Local Navigation for Multi-Robot Systems," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, 2006, pp. 1733–1738.

[9] P. Pirjanian and M. Matarić, "A decision-theoretic approach to fuzzy behavior coordination," in *Proc. of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, 1999, pp. 101 –106.

[10] P. Pirjanian and H. Christensen, "Behavior coordination using multiple-objective decision making," in *Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, vol. 3209, 1997, pp. 78–89.

[11] Y. Li and K. Gupta, "Motion planning of multiple agents in virtual environments on parallel architectures," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2007, pp. 1009–1014.

[12] C. Guestrin, "Planning under uncertainty in complex structured environments," Ph.D. dissertation, STANFORD UNIVERSITY, 2003.

[13] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1-2, 2000.

[14] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, *et al.*, "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *The International Journal of Robotics Research*, vol. 19, no. 11, p. 972, 2000.

[15] C. Bordenave, D. Mcdonald, and A. Proutière, "Random multi-access algorithms: A mean field analysis," in *Proceedings of Allerton conference*. Citeseer, 2005.

[16] M. Seder, S. Jurić-Kavelj, and I. Petrović, "Automatic creation of hierarchical maps for indoor environments," in *Proceedings of the Fifth International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2008, pp. 19–24.

[17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, Massachusetts: MIT Press, 2005.

[18] M. Seder and I. Petrović, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 1986–1991.

[19] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'94)*, 1994, pp. 3310–3317.

[20] M. Seder, K. Maček, and I. Petrović, "An integrated approach to real-time mobile robot control in partially known indoor environments," in *Proceedings of the 32nd Annual Conference of IEEE Industrial Electronics Society (IECON 2005)*, 2005, pp. 1785–1790.