Complex Structure Workspace Sweeping Using Semiautonomous Vacuum Cleaners

KRZYSZTOF SKRZYPCZYK
Department of Automatic Control
Silesian University of Technology
Akademicka 16, 44-100 Gliwice
POLAND
krzysztof.skrzypczyk@polsl.pl

AGNIESZKA PIERONCZYK
School of Engineering, Design and Technology
University of Bradford
Richmond Road, Bradford BD7 1DP
UNITED KINGDOM
agatpi@gmail.com

Abstract: The paper presents the study of design and synthesis of the system dedicated for sweeping large and complex spaces using a team of cleaning robots. The problems of coordination of activities of individual robotic-unit is discussed. Application of the Game Theory for creation of coordination algorithms is proposed. The paper deals also with design and implementation of algorithms of movement of autonomous (or semiautonomous) vacuum cleaner, that would be able to work inside of a flat, but not free of obstacles workspaces. In the paper several proposed algorithms of surface covering are discussed. The two algorithms were selected as the most appropriate and presented in this work. Experimental validation of efficiency of the proposed approach were done and the results of the validation are discussed in the paper.

Key-Words: - Mobile robotics, game theory, multi-robot, semiautonomous systems, vacuum cleaners, surface covering

1 Introduction

Domestic robots are robots which are used in homesteads. This group of robots is intended to perform tasks such as housekeeping as well as, serving as educational and entertainment robots. Despite of the fact, that the area of domestic robotics is much less funded than for example the military robotics – it has still been growing and developing rapidly, simultaneously increasing the level of human comfort and life. With reference to human kind history, people have always been looking for simplifications and improvements that would made their life easier. Housekeeping devices, especially domestic robots generate wide range of possibilities to achieve this goal. The issue is relatively up to date matter, specially nowadays – when robotics is so popular and creation of its applications became less expensive than any time before. The status that housekeeping robots can obtain in the future, can be compared to the status of all others domestic devices (e.g. oven, fridge, hairdryer) which are nowadays omnipresent at our homes, and create basics of each home's equipment. That can be done only if their performance, utility and prices would be optimized to accomplish a satisfactory trade off. Of course in order to design a properly working robotic-cleaner a number of problems related to mobile robotics must be solved. That is enough to mention path planning, data representation etc. [3,5,8,9].

Another aspect of application of robots to the task of indoor cleaning is using multiple cleaning devices. It is well known fact that larger number of simpler devices can perform complex tasks. When we consider the work of a team of robots, that are intended to perform some complex task (workspace cleaning for instance) the additional problem of coordinating actions (tasks) of individual robots needs to be taken into account. Wrong coordination may lead to ineffective task execution or even to inability of completion the task. The problem of coordination of multiple robots can be stated as a conflict situation between individual robotic-agents and can be modeled as a decision making problem. The Game Theory [1,4] seems to be a convenient tool for modeling problems of conflict nature. In this work the approach to coordination of multiple cleaning robots operating in a complex structured indoor environment is presented. The problem of coordination of elementary tasks is modeled as a *N*-person, noncooperative team decision problem. The hybrid architecture of a system that is designed to control the work of agents discussed.

Apart from the problem of team coordination there exist another, important issue to solve which is th problem of effective and robust surface covering b individual devices [11]. Therefore in the paper th aspect of design and implementation of an algorithr of movement of autonomous (or semiautonomous vacuum cleaner, that would be able to work inside of flat, but not free of obstacles workspaces is als considered.

The history of all robots which application was base on the vacuum cleaners cannot be fully ordered. Th cause of it is that most robots were created by th commercial companies, which official secrets have nc allowed to catalog all created prototypes. Fortunatel most of hovering robots, which have been sold o market till now, had been somehow described or a least mentioned in wide range of articles and website: That gave a possibility to make the historical outline c those robots.

First mentioned, was a prototype of the first sold o market robotic vacuum cleaner called trilobite [11] manufactured by the Swedish corporation Electrolu was available for purchase in 2001.

One year later, in 2002 the first generation of the most wide known, *the Roomba* [6] robotic floor cleaners has been on sale. Nevertheless, to date second and third generations of that robot have been produced. At once, with each generation new improvements were obtained. To begin with better brushes and larger dust bins, finishing with new algorithms of cleaning.

As was previously stated, design and implementation of a robotic vacuum cleaner is the very present issue. What is more, according to documentation and articles which addresses this group of devices – their algorithms should still be enhanced in order to obtain more immune for environment's disturbances and more efficient system. Additionally construction of the multiple vehicle, cleaning system considered in this paper has steel been an open issue.

2 System overview

The problem considered in the paper is two fold one. The first stage of research includes the study of construction of multiple robot system and coordination mechanisms. The second one is focused on design and evaluation of surface covering algorithms.

A general structure of the control system is presented in fig.1. The system can be split into two layers. The first one that is intended to be implemented on mobile platform (cleaning and collision avoiding algorithms) and the second, that provides information of robots location and communication between robots. The first one that is considered in this work has typical hybrid

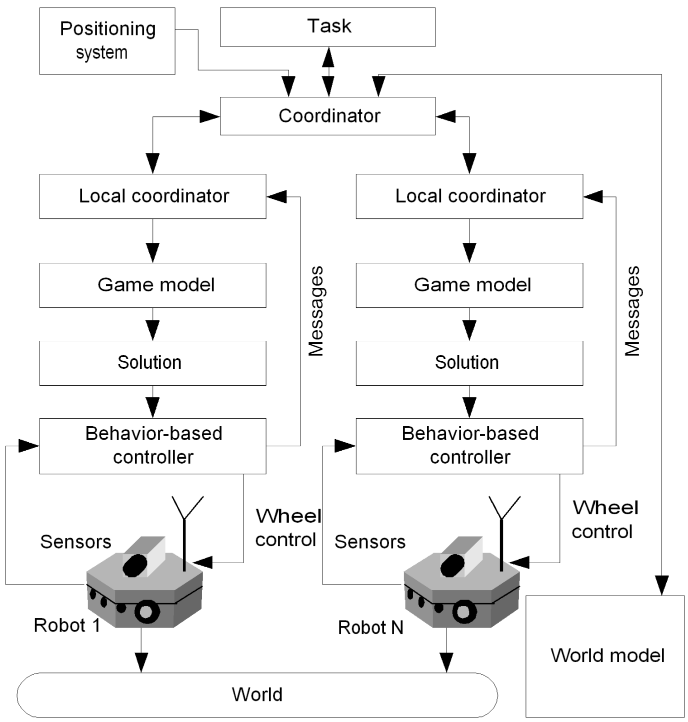

Fig. 1 Overview of the multi device cleaning system

structure. It consists in behavior-based motion controller that is responsible for executing elementary navigational tasks (modeled further by operators) and non-cooperative game based planner. The function of the planner is to choose from admissible actions the one that provide execution of a part of primary mission (clearing operation).

The workspace model

Robots are intended to operate inside a well structured, complex human made workspace. In order to simplify the navigation problem a partial knowledge about the environment is introduced to the system. In fig. 2a an exemplary office environment plan is presented. An overall workspace is divided into regions named *sectors*. Each sector represents an area occupied by a room, corridor or a part of a corridor. Moreover passages between rooms are distinguished and introduces to the model as *door-objects* called further *door* for simplicity. The workspace model is stored onto two layers: topological and geometrical. The first one is given by weighted graph:

$$M=(V,W) \quad V=\{v_1,v_2,\ldots v_M\},\ W\subset V\times V \tag{1}$$

which nodes represent objects of the environment: sectors V_S and doors V_D where $V = V_S \cup V_D$. On the topological level each object is described by a real number that is an "cluttering coefficient" c_i in case the object is a sector and by a probability of being opened o_j when the object is a door. The first coefficient reflects the number of objects placed inside of the given sector. On a geometrical level i-th sector is represented by coordinates of its top left corner (x_i^t, y_i^t), bottom right corner (x_i^b, y_i^b), and a center point (x_i^p, y_i^p). Similarly the j-th door-object is described by a circle of radius r_j and a center of the circle (x_j^p, y_j^p). The edges of the graph define relations of neighborhood between environmental objects related to vertices of the graph. Weighting factors fixed to the edges of the graph are related to some costs of moving robot from the i-th to the j-th vertex (object

It is worth of noticing that the model describes only invariable features of the workspace. The layout of objects (furniture, equipment) placed inside sectors is not known. Moreover it can undergo dynamic changes.

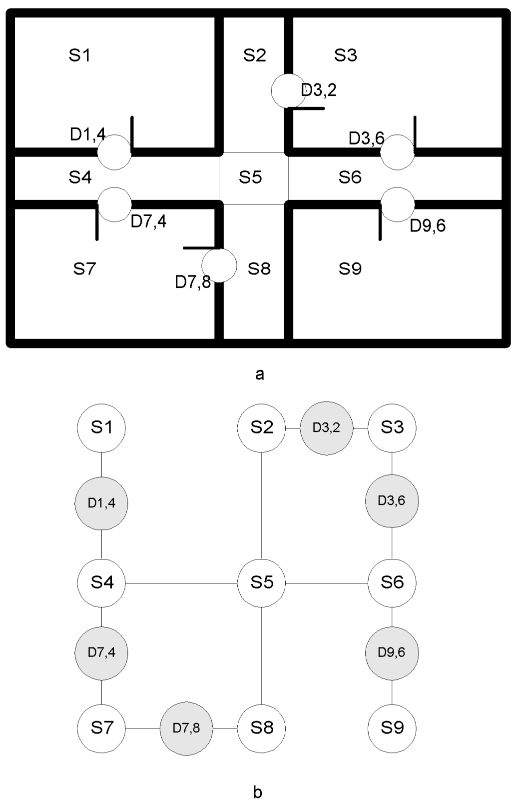

Fig. 2 An Exemplary Workspace Layout (a) and its Topological Model (b)

The planner

From the perspective of this work the planner is the core of the system. It consists of three modules: local coordinator, decision process modeling module and the solution computation one. The work of the planner can be briefly described in a following way. The local coordinator receives information of location of all the cleaning devices. Moreover it is provided with a world model and a primary task data. Depending on a type of the task, location of all of teammates and a state of the task execution, a model of a decision process is built. The model is in fact the cost function that depends on actions made by individual agents and a state of task completion. Next the problem is solved and the solution computed. The solution of the problem determines an elementary action which is optimal for a given agent from the point of view of primary task execution. Detailed description of the process of building the model (taking the cleaning task as an example) and the methods of solution shall be presented in further sections.

Task execution module

The role of this module is to execute elementary navigational tasks. It is designed using the behavior-based idea of control. It is composed of six behaviors that process the state and sensory information into proper set-points for motion controller which are values of linear and angular velocity. The coordination of behaviors activities is made by fixed priority arbiter. For the purpose of this article that is enough to consider the module as the one which is able to execute four different elementary tasks, represented by following operators:

- **FindDoor(D)** - the task of moving the robot inside the area of door-object D;
- **TraverseDoor(D,S)** -the task of going through the door-object *D* to the sector S;
- **Wait()** - the simple command that stop the robot;
- **GoTo(S1,S2)** - the task of moving robot from the sector *S1* to *S*2;
- **Clean(S)** - the task of cleaning the sector *S* by the robot;

3. Cleaning coordination strategy

3.1 The problem formulation

The task of complex structured workspace sweeping is similar to the exploration problem [2]. This task can be generally stated as a problem of visiting a given part of the workspace by teammates with a cost as low as possible.. In terms of this work the cleaning task is defined as visiting a part $M_V \subset V_S$ of the workspace M in a number of steps as small as possible. Here in this work, we model the problem as a sequence of one stage, non zero sum games in a normal form.

3.2 Game theory based approach

Let us first introduce a notation that will be used hereafter. The state of a team of robots is denoted by a set:

$$X = \{x_i\} \quad i = 1,2,...N, \quad x_i \in V \tag{2}$$

what is equivalent to the fact that there is the i-th robot inside the area described by the vertex x_i. The set of all possible actions of the robot described by operators is given by the set:

$$A = \{a_1, a_2, ...a_M\} \qquad 3)$$

where M is a number of all operators (in this work M=5). A set of possible actions of the i-th robot in the state x_i is defined by :

$$A_i(x_i) = \{a_1, a_2, ...a_K\} \tag{4}$$

and it is determined by precondition lists of individual operators. In our case they are as follows:

FindDoor(D)
preconditions = $\{x_i \in V_S,\ w_{x_i,D} \neq \infty\}$
TraverseDoor(D,S)
preconditions = $\{x_i = D \in V_D,\ w_{D,x_i} \neq \infty\}$
Wait()
preconditions = ϕ

GoTo(S1,S2)
preconditions = $\{x_i = S1 \in V_S,\ \mathrm{w}_{S1,S2} \neq \infty\}$

Clean(S)
preconditions = $\{x_i = S\}$

In the terms of the decision making process model of an action $a_k \subset A_i$ for $k = 1,2,...,4$ is a mapping:

$$a_k : x_i^n \to x_i^{n+1} \quad x_i^n \subset X \quad x_i^{n+1} \subset V \quad a_i \subset A_i \tag{5}$$

where x_k^n is the current state of the i-th robot, and x_k^{n+1} is a state the robot will be in as a result of the action a_k. The primary task of the team of robots is to visit and clean all objects defined by a set $M_G \subset V_S$. We introduce an auxiliary set defining objects that have already been cleaned and we denote it by $M_V \subset M_G$. Using this notation we can precisely formulate a goal of the team as satisfying the equality: $M_V = M_G$. The task of the planning algorithm is to choose for each robot one of the possible action, that applied to the robot will result in performing a part of the primary task. The problem of selection of proper action is in this work perceived as a game between individual robotic-players. The result of the game related to the defined task depends on decisions made by individual game participants. Moreover the task of exploration has a specific nature that can be classified as a team-work problem where all of the players (robots) want to optimize one performance index. Although the environment is in principle dynamic we model the problem of action planning as a sequence of static N-person games in a normal form. Therefore we need to define for each stage of the planning process the single cost function value of which depends on actions made by all of teammates and on the task completion state. We propose to define the cost function as a sum of three components:

$$I(a_1, ..., a_i, ...a_N) = -I_R + I_E + I_D \tag{6}$$

The first one is related to a some value of "reward" that is given to robots for cleaning part of the workspace and it is given by:

$$I_R(a_1,..a_i,...a_N) = \sum_{i=1}^{N} R_i(a_i)$$
$$R_i = \begin{cases} \frac{1}{k} R > 0 & if \quad \mathrm{x}_{\mathrm{i}}^{\mathrm{n+1}} \in M_G \cap x_i^{n+1} \notin M_V \\ 0 & otherwise \end{cases} \tag{7}$$

where R is a positive number that denotes the reward value. The k is a number of robots visiting the same object as a result of their actions. The value of the second component I_E is dependent on an amount of energy necessary to make an action a_i which is proportional to a cost of transition of robots between environmental objects defined by the model M:

$$I_E\left(a_1,..a_i,...a_N\right)=\sum_{i=1}^{N} w\left(x_i^n,x_i^{n+1}\right) \qquad (8)$$

Third component denotes cost of moving the robot to the nearest (in the sense of costs defined by W) unexplored object. Let us first denote a path of minimal cost between an object n and m as:

$$p_{\min}\left(n,m\right)=\left\{v_n,...v_k,...v_m\right\}\subset V \qquad (9)$$

and let the set of unexplored objects is given as $M_U=\left\{u_l\right\}=M_G\setminus M_V$. Then the cost I_D is given by:

$$\begin{aligned} &I_D\left(a_1,...,a_i,...,a_N\right)=\sum_{i=1}^{N} D_{\min,i} \\ &D_{\min,i}=\min_l\left[p_{\min}\left(x_i^{n+1},u_l\right)\right] \end{aligned} \qquad (10)$$

where $D_{min,i}$ is the cost of moving the i-th robot from the state x_i^{n+1} to the "nearest" unexplored object

4. Surface covering problem

4.1 The vacuum cleaner model

As a testbed for validation of proposed solution the miniature, laboratory mobile robot Khepera II [7] was used. The Khepera is presented in fig. 3. As a matter of fact, that implied also an usage of The MATLAB software, which was required to program the robot. Detailed specification of the Khepera II and is usage in the MATLAB can be found on the www.k-team.com web site. Nevertheless, the robot used in experiments is a complex device and it seems reasonable to highlight its main features. Thus, hardware and software description is presented below in order to introduce some essential knowledge, required for full apprehension of developed algorithms, that are afterwards presented.

Fig.3 The Khepera II robot as a model of vacuum cleaner [7]

The Khepera II is a small, differential wheeled mobile robot, designed as a scientific research and teaching tool at Federal Institute of Technology Lausanne (EPFL) in Switzerland. First of all, the robot is well suited for project's purposes because it renders rapid idea generation, prototyping and evaluation. In its User Manual can be found, that the robot allows also confrontation to the real world of algorithms developed in simulation for trajectory execution, obstacle avoidance, preprocessing of sensory information, hypothesis on behaviors processing [7].

4.2 The problem formulation

Another aim of the work is creation of surface covering algorithms for hovering robot that would be a part of discussed multiple device cleaning system. The algorithm should enable the unit to work inside of a flat, but not free of obstacles workspaces.

4.3 Proposed solution

In this section details on two designed surface covering algorithms are presented end discussed. The first is the simple random-walk based one. The second one is more complicated hybrid concept that merges two different approaches – random movement and spiral motion.

Random walk based algorithm

The first algorithm has been based on the idea of random movement (fig. 4). The robot is moving in forward direction till an obstacle is sensed, then it stops. Next, by comparing sensor readings decides in which direction to turn – left or right. Finally, by generating a random number decides how much to turn. The overriding aspect of this algorithm is a distance between the robot and obstacles. It has been done, by setting into its source code, a distance threshold value. Simply, when the distance is lower than the given threshold value, the robot takes an action in order to omit obstacles. As a result of this, the manner in which robot is moving depends strongly on its surrounding. Despite the fact, that the algorithm assumes ongoing turns from the obstacle (i.e. if obstacle is sensed on the left, it turns right) what may cause stacking in corners, the random turning banishes the issue. All in all, the algorithm fulfills the aim of the work, allowing the robot to move around a room avoiding obstacles. However, its efficiency should be compared with more sophisticated movement manner in order to examine its utility.

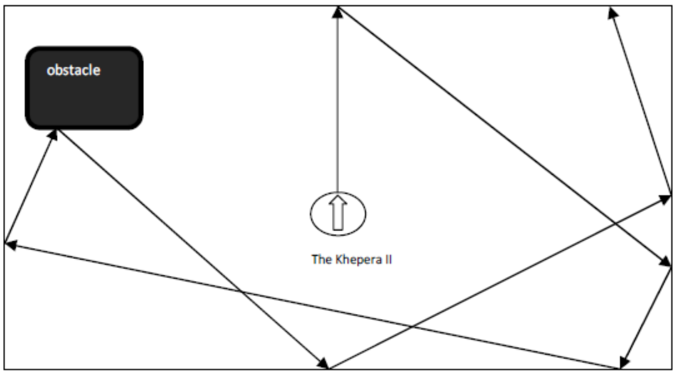

Fig. 4 The visualization of 8 iterations of robot's movement according to random walk based algorithm.

This working mode is presented below in a form of a flow chart (fig.5).

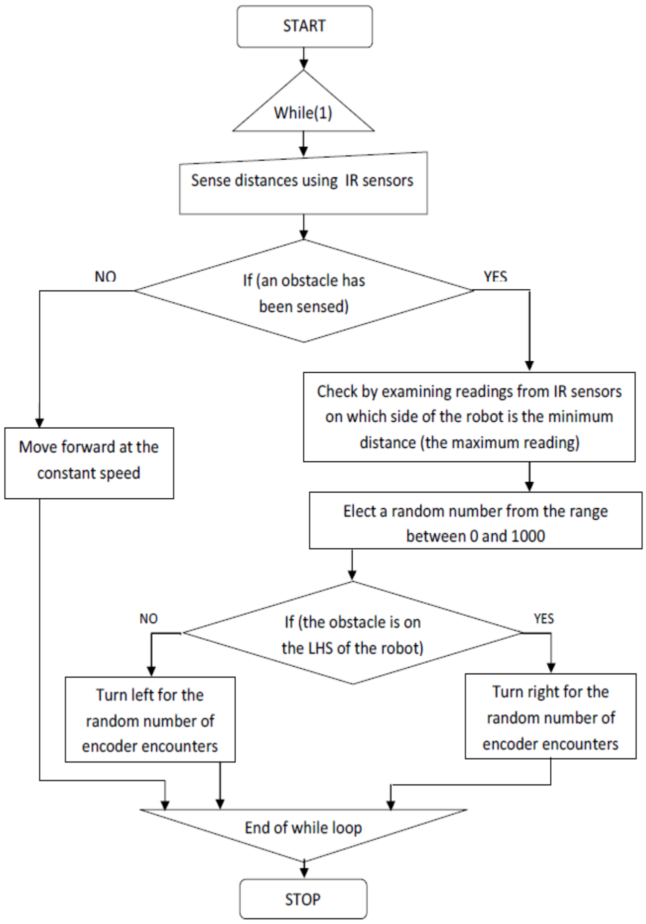

Fig. 5 The flow chart for the random walk based algorithm.

Hybrid algorithm

The second algorithm has been based on the idea of a spiral motion coupled with random turning (fig. 6). First of all, the robot checks if there is enough place to start moving spirally. If yes, the robot convolutes in a RHS direction, increasing a radius from centre point, the till an obstacle is sensed. When the obstacle is sensed, robot stops and compares sensor readings to decide in which direction to turn – left or right. After the decision is made, the robot turns a given direction for a previously allotted angle. Finally, in order to allow itself to begin convoluting, it moves away from the obstacle for a given range. Invariably, the overriding aspect of the algorithm is a distance between the robot and obstacles. It has been done likewise to algorithm I, by setting into its source code, a distance threshold value, thanks to which an action in order to omit obstacles can be taken. As a result of this, the manner in which robot is moving depends strongly on its surrounding characteristics.

The principle how the algorithm works is shown in the form of a simplified flow chart (fig. 7) that gives a full and clear view on properties of the algorithm.

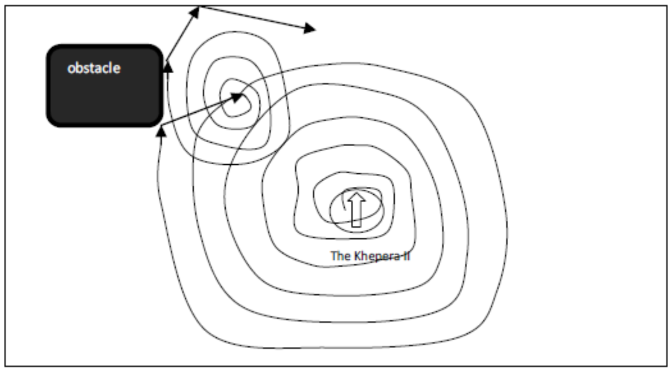

Fig. 6 The visualization of 3 iterations of robot's movement according to algorithm II

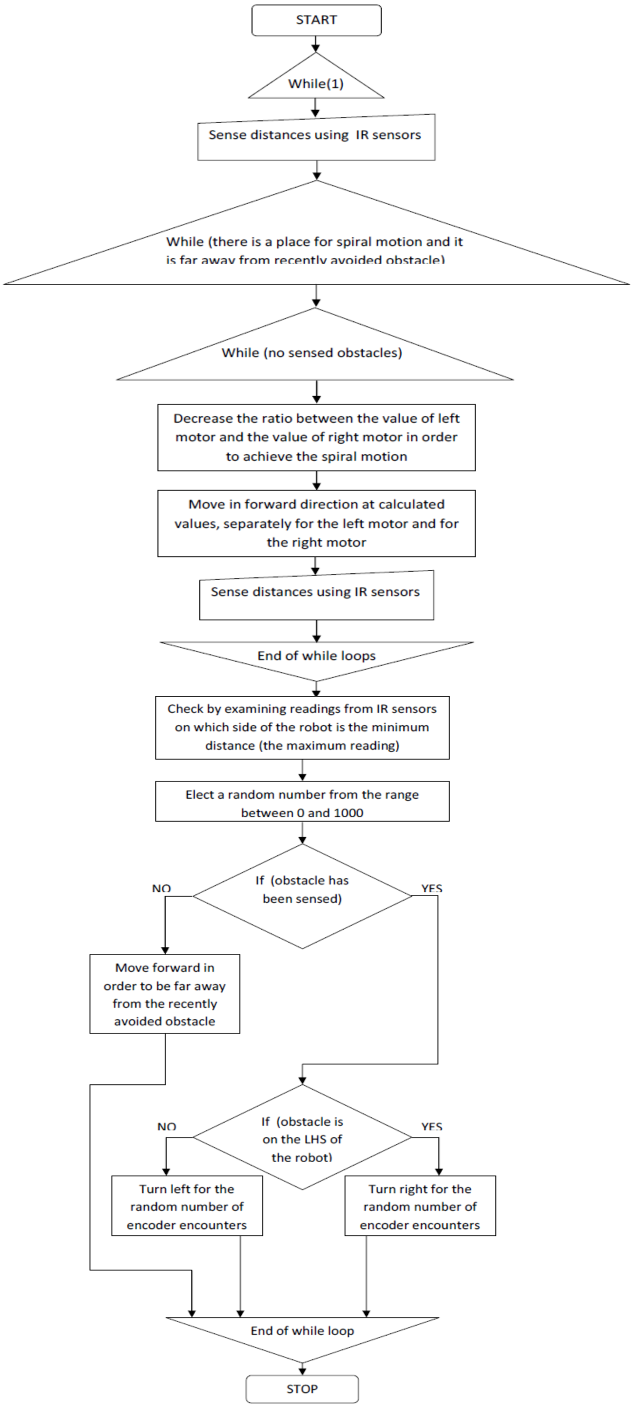

Fig. 7 The flow chart for the hybrid algorithm.

5. Simulation and experimental results

5.1 Coordination process

In order to show how the approach discussed in the paper works we present a result of an exemplary simulation. We implemented the method using a simulation environment The layout of the simulated workspace is shown in fig.8.

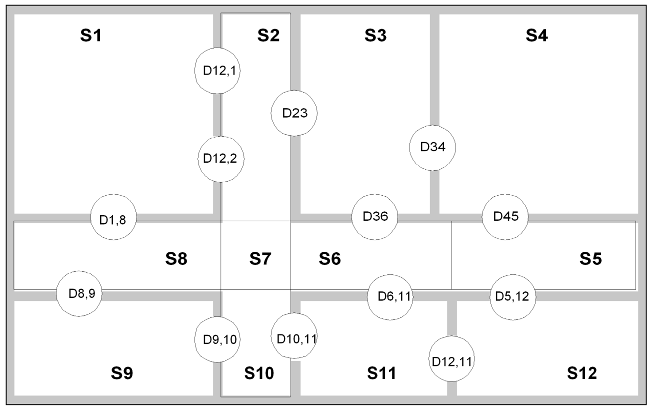

Fig. 8 The layout of a workspace used for a Simulation

It is an example of a typical office environment. It consists of twelve sectors $V_S = \{v_1, v_2, ...v_{12}\}$ that represent rooms and parts of corridors, and thirteen passages (doors) between rooms $V_d = \{v_{13}, ..., v_{25}\}$. The area of the workspace is of a size 15 × 10 [m] (width, height). Various pieces of furniture and equipment are placed inside of individual rooms. The robot model used for the simulation is described in section 4.1. We consider the following cleaning task. A team of three vacuum cleaners *(N*=3) is intended to clean the workspace what is equivalent to visiting all of the sectors. Thus the task is defined as $M_G = V_S$. Initially, vacuum cleaners are inside of sectors $X_{init} = \{v_4, v_9, v_2\}$ so $M_V = X_{ini}$. A sequence of operators that was used to perform the task is presented in the table 1 and the illustration of the simulated process is presented in fig. 9. The symbols *F*D, *T*D, *G*T, *W,CL* denote operators: *FindDoo*r, *TraverseDoo*r, *GoTo*, *Wai*t and Clean. We can see that algorithm works well, providing completion of the task stated above. One can notice that robot 1 perform only a small part of the task, exploring only one room. But it has to be taken into account, that the door D3,6 is modeled as the one that is almost for sure closed. That is the reason of this ”strange” task execution. Yet another aspect is worth of commenting. The algorithm presented in paper make only ”one step ahead” planning. It causes that the task execution may be not optimal. Using other planning algorithms we would obtain better or even optimal solution. But such an approach would be valid if the environment was static as well as we assumed perfect result of each action.

n	Robot 1	Robot 2	Robot 3
1	FD('D3,4')	FD('D9,10')	FD('D1,2_1')
2	TD(D3,4,S3)	TD(D9,10,S10)	TD(D1,2_1,S1)
3	CL(S3)	CL(S10)	CL(S1)
3	FD(D2,3)	FD(D10,11)	FD(D1,8)
4	TD(D2,3,S2)	TD(D10,11,S11)	TD(D1,8,S8)
5	CL(S2)	CL(S11)	CL(S8)
6	W()	FD(D11,12)	GT(S7)
7	W()	TD(D10,11,S12)	CL(S7)
8	W()	CL(S12)	GT(S6)
9	W()	W()	CL(S6)
10	W()	W()	GT(S5)
11	W()	W()	CL(S5)

Table 1 A Sequence of operators that provides completion of the workspace cleaning task

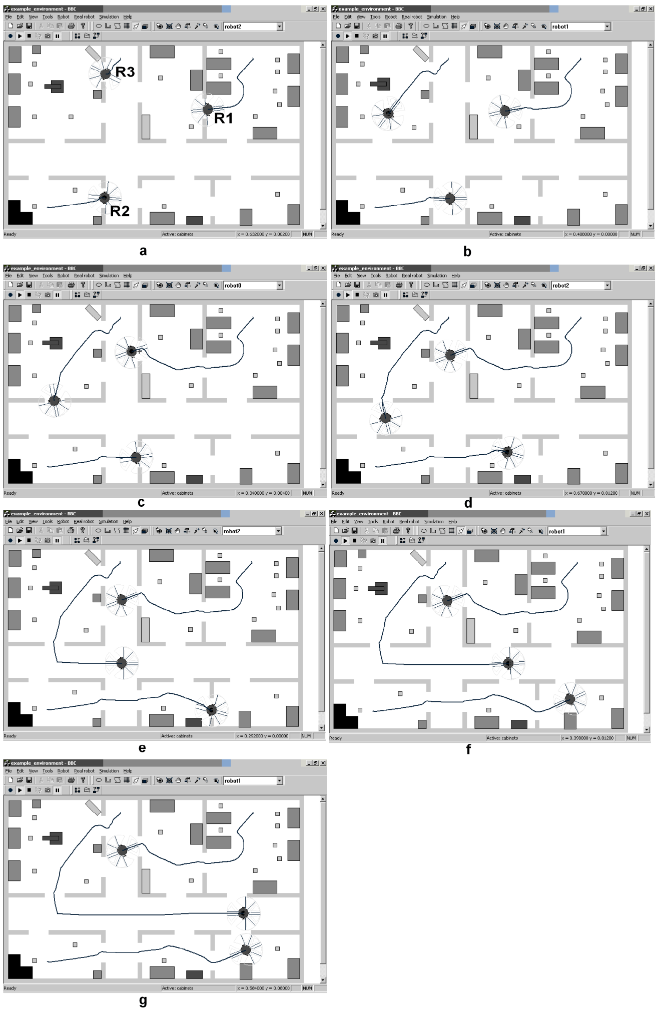

Fig. 9 Stages of realization of the cleaning task

5.2 Surface covering

Two described surface covering strategiesy have been tested if they are operating properly, what means that they fulfilled aim and all assumed objectives. All provided tests helped in speed and proximity sensors calibration. Finally, when proper work manners for both algorithms were obtained, their outputs gave appropriate and satisfactory results – for infinite period of time, the free from obstacles area of the workspace would be almost fully covered and obstacles would be avoided. However, it has to be mentioned that black or very reflective objects would not be avoided due to sensors insensitivity for that kinds of materials. At the same time, the prominent issue of all objectives was algorithm efficiency. The efficiency was seen as area coverage in time. Taking into account sensors and motors constraints it had been very important to keep illumination levels and initial conditions as similar as possible for each type of test. That assured the most applicable results from each simulation. Owing the fact that both algorithms involved random movement, their outputs were highly unpredictable. Nevertheless, based on numerous tests, following efficiencies for operation time equals 1 minute have been achieved (table 2).

Workspace dimensions: 400x500 mm; no obstacles	
Random Motion	Spiral Motion
1114 cm²	1430 cm²
Workspace dimensions: 800x500 mm; no obstacles	
Random Motion	Spiral Motion
1850 cm ²	2034 cm²
Workspace dimensions: 800x500x100 mm; one obstacle: 100 cm²	
Random Motion	Spiral Motion
1532 cm²	1296 cm²
Workspace dimensions: 800x500x100 mm; five obstacles sum: 150 cm²	
Random Motion	Spiral Motion
870 cm²	540 cm²

Table 2 Comparison of the efficiency of the two presented surface covering algorithms.

As it can be observed spiral motion algorithm has better efficiency coefficient for grater workspace dimensions and smaller number of obstacles. What is more, its efficiency depends strongly from the initial point. The best results were obtained for initial point being in the middle of the workspace, what allowed the robot to convolute with greater radius. On the other hand, when the initial point is closer to the workspace's boundaries, or when the workspace area is filled with obstacles – the random motion algorithm accomplishes better efficiency results than the spiral motion. To sum up, both algorithms that had been developed for the robotic version of a vacuum cleaner, have been checked and they worked properly. Regardless of the fact that their efficiencies varied within environmental changes, their empirically derived efficiencies were still satisfactory.

6. Conclusion

In this paper we discussed a problem of planning and coordination of tasks in a multi system.

The role of the discussed system is sweeping large and complex spaces using a team of cleaning robots. We considered a team of robotic-vacuum cleaners that was intended to perform a task cleaning of a human-made workspace of complex structure. We proposed both the hybrid architecture of a control system and method of coordination of multiple robotic agents. In the paper we also presented an algorithm of exploration of workspace. The core of the algorithm is the model of the process that is stated as a noncooperative game in a normal form. We applied the Nash equilibrium concept to generate a solution of the problem. Although the result of only one simulation was presented, we had made a numer of simulation experiments using both various parameters and workspace configurations. In all cases we obtained correct task execution. On the basis of simulations we carried out we can conclude that this algorithm works well and provides effective exploration of even very complex-structured environments. However, the algorithm can not guarantee optimal task performance. It is caused the algorithm uses only one-step-ahead planning method. But this approach on the other hand has other advantage - it allows to track dynamical changes of the environment and it does not need the assumption that a given action is always executed in a perfect way

The second issue discussed in the work was the design and implementation of algorithms of movement of autonomous (or semiautonomous) vacuum cleaner, that would be able to work inside of a flat, but not free of obstacles workspaces, have yet to be compared with the achieved outcome. Unquestionably, developed algorithms fulfilled this goal. What is more, the main objectives for the project were based on analysis of developed algorithms, their limitations and efficiencies. The Khepera II returned appropriate response with respect to both programs, keeping a distance to objects, as well as to the area coverage. The robot has not stacked accidently at any of the workspace recesses. As it has been mentioned before, although mechanical aspect of the project has been qualified to consider only different types of a testbed robots, environmental perception and physical characteristics have given some additional constraints influencing simultaneously on the project's final output. Those constraints together with project's output have been deeply analyzed and resulted in proposals for future developments. Despite of the fact that all the described system is in the design phase the preliminary results obtained are promising.

Acknowledgements

This work has been granted by the Polish Ministry of Science and Higher Education from funds for years 2008-20011.

References:

[1] Basar, M. and G.J. Olsder. 1982. *Dynamic Noncooperative Game Theory*. Mathematics in Science and Engeneering, Academic Press Inc. Ltd, London.

[2] Burgard, W. et al. 2000. "Collaborative Multi-Robot Exploration." *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco CA

[3] Ching Chang Wong et al., *An Intuitional Method for Mobile Robot Path Planning in Dynamic Environment,* WSEAS Recent Advances in Computers Computing and Communications, ISBN:960-8052-62-9, 2002, pp. 275-280

[4] A. Galuszka, A. Swierniak, Game Theoretic Approach to Multi Robot Planning, WSEAS Transactions on Computers, Issue 3, Vol. 3, July 2004

[5] D. Glavski et al, Robot Motion Planning Using Exact Decomposition and Potential Field Methods, Proc. of the 9th WSEAS International Conference on Simulation Modelling and Optimization, Budapest, Hungary, September, 2009 pp. 126-131

[6] iRobot Roomba, *The official polish iRobot Roomba webpage* Available: http://roomba.pl/

[7] K-Team, *Khepera II User Manual*, 1.1 version, Switzerland, page 11, 2002

[8] F.G. Martin, *Robotic Explorations A Hands-On Introduction to Engineering*, 1st ed. New Jersey, United States of America: Tom Robbins, 2001.

[9] P. J. McKerrow, *Introduction to Robotics,* 1st ed. Singapore: Addison-Wesley Publishing Company, 1991.

[10] K. Skrzypczyk, Behaviour Activity Based Maps Applied to Mobile Robot Navigation, Proc. of 13th WSEAS International Conference On Systems, Rodos, Greece, July, 2009, pp. 295-300

[11] Trilobite (2010), *Robotyka – The official Electrolux's Trilobite webpage* Available: http://www.electrolux.pl/node37.aspx?categoryid=6681

[12] I. R. Urlich, F. Monada, J. Nicoud, Autonomous vacuum cleaner. Available: ftp://lamiftp.epfl.ch/khepera/papers/ulrich_RAS96.

.