GAME THEORY BASED TASK PLANNING IN MULTI ROBOT SYSTEMS

Krzysztof Skrzypczyk
Department of Automatic Control
Silesian University of technology
Akademicka 16, 44-100, Gliwice, Poland
e-mail:kskrzypczyk@ia.polsl.gliwice.pl

KEYWORDS

Game theory, motion planning, multi-agent systems

ABSTRACT

In the paper we discuss a problem of planning and coordination in a multi robot system. We consider a team of robots that performs a global task in a human-made workspace of complex structure. A hybrid architecture of the team motion control system is considered. The system can be split into two layers: the planner module and the behavior based collision free motion controller, that is designed to perform several elementary navigation tasks. The role of the planner is to plan and coordinate execution of elementary tasks by individual agents to obtain performance of global task. We presents the method of elementary tasks planning based on N-person game. The Nash equilibrium concept of the game solution is applied. An algorithm of multi robot workspace exploration is presented as an example of application of the proposed method. Simulation of the algorithm was carried out, and its result is presented and discussed in the paper.

INTRODUCTION

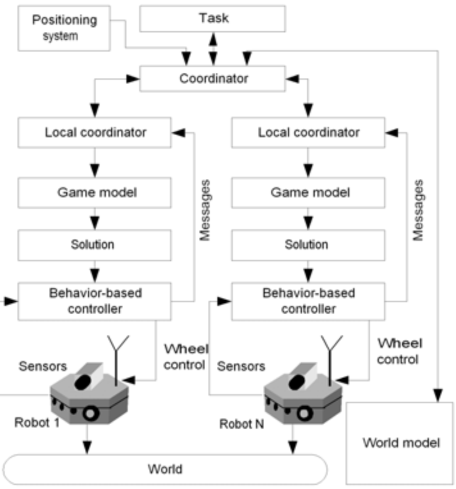
One of the fundamental requirements for mobile robot control system is its autonomy which is defined as an ability of operation without control of a human operator in an environment model of which is unknown and react a dynamical changes of this environment. Meeting these requirements implies the system has to cope with and solve many complex problems like task planning, collision free trajectory generation, operating on the basis of imprecise and uncertain information, environmental model building. Mobile robot control systems can be generally divided into three groups: deliberative, reactive (behavior-based) (Althaus and Christensen 2003; Arkin 1998; Skrzypczyk 2002) and hybrid systems. The most effective approach seems to be the third one that brings together advantages of reactive and deliberative systems (Fox et al. 1998; Shim et al. 2000). Operating by a robot in a real-world environment of a complex structure such as an office for instance, without any a'priori information about the environment usually leads to inefficient task execution. Therefore, if robot is designed to work inside definite workspace it is reasonable to introduce partial knowledge of the workspace to the control system (Althaus and Christensen 2003). When we consider a work of a team of robots, that are intended to perform some complex task (workspace exploration 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. Therefore a lot of attention has been paid to this problem (Burgard et al. 2000; Gerkey and Mataric 2002; Golfarelli and Meuleu 2000; Lawton et al. 2003; Sequierra and Ribeiro 2001; Schneider-Fontan and Mataric 1998). 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 is convenient tool for modelling problems of conflict nature. Therefore applications of game theory in the context of multi-agent coordination have been widely reported in the literature (Golfarelli and Meuleu 2000; LaValle 2000; Li and Payandeh 2001). Unfortunately most of the works lack the treatment of application aspect of game theoretical approach. They only consider the problem as a theoretical one, without taking into account limitations of a control system framework. In this paper we discus an approach to coordination of multiple robots operating in an environment of a complex structure, performing complex task which example is the environment exploration one. We model the problem of coordination of elementary tasks as a *N*-person, noncooperative team decision problem. We present a hybrid architecture of a system that is designed to control the work of agents that perform the stated task. In the end of this work we present and discus the simulation results of the proposed system.

SYSTEM OVERVIEW

A general structure of the control system is presented on the block diagram below (fig.1). The system can be split into two layers. The first one that is intended to be implemented on mobile platform 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 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 role of the planner is to choose from admissible actions the one that provide execution

Proceedings 16th European Simulation Symposium
György Lipovszki, István Molnár © SCS Press, 2004
ISBN 1-56555-286-5(book) / ISBN 1-84233-106-x(CD)

of a part of primary mission (in our case the mission is the workspace exploration). Of course the planner has to



Figures 1: A Diagram of the Hybrid, Multi-Robot Control System

take into account all possible actions of other agents and provide their proper coordination.

The world model

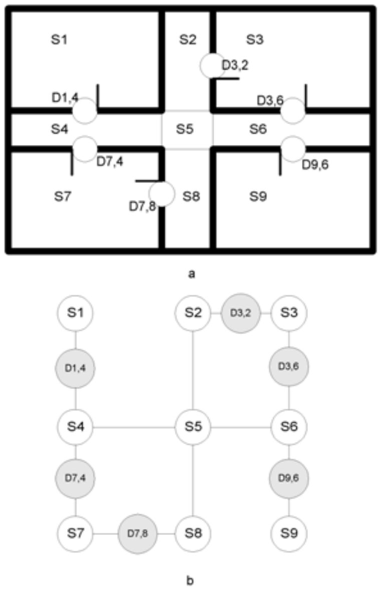
Robots are intended to operate inside a well structured, complex human made workspace. In order to simplify the navigation problem a partial knowledge about the environment is introduced to the system. In the fig. 2a an exemplary office environment plan is presented. An overall workspace is divided into 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:

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

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 $\left(x_i^t, y_i^t\right)$, bottom right corner $\left(x_i^b, y_i^b\right)$, and a center point $\left(x_i^p, y_i^p\right)$. Similarly the j-th door-object is described by a circle of radius r_j and a center of the circle $\left(x_j^p, y_j^p\right)$. 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). In this work we define the costs using some heuristic formulae:

$$w_{ij} = \begin{cases} L_{i,j}\left(1+c_i\right)\left(1+e^{-\alpha o_j}\right) & if\ \ v_i \in V_S \cap v_j \in V_D \\ L_{i,j}\dfrac{1}{o_i}\left(1+c_j\right) & if\ \ v_i \in V_D \cap v_j \in V_S \\ L_{i,j}\left[1+0.5\left(c_i+c_j\right)\right] & if\ \ v_i \in V_S \cap v_j \in V_S \end{cases} \qquad (2)$$

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.



Figures 2: An Exemplary Workspace Layout (a) and its Topological Model (b)

The behavior-based controller

The role of this module is to execute elementary navigational tasks. It is designed using the behavior-based idea of control (Arkin 1998). 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 more detailed description of this module refer to (Skrzypczyk 2002). 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 *S2*;

All of the operators are related to a task of collision free moving robot from a given initial location to a desired one which are specified by arguments of the operator. The difference between individual tasks lies in the set of parameters associated with a given operator that is sent to the controller. The parameters are priorities of individual behaviors as well as velocity limits.

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 of the robots. 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 an exploration task as an example) and the methods of solution shall be presented in further sections.

AN EXPLORATION PROBLEM FORMULATION

As we mentioned before, we want to present the method of coordination of multiple robots that provides completion of the task of exploration of the workspace the topological model of witch is known to the system. 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. Exemplary interpretation of the exploration task is delivering parts in a large plant by multiple robots. In terms of this work the exploration 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.

Modelling the Problem of Exploration

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

$$X = \{x_i\} \quad i = 1, 2, ...N, \quad x_i \in V \tag{3}$$

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 robot actions described by operators is given by the set:

$$A = \{a_1, a_2, ...a_M\} \tag{4}$$

where M is a number of all operators (in this work M=4). A set of possible actions of the *i*-th robot in the state x_i is defined by :

$$A_i(x_i) = \{a_1, a_2, ...a_K\} \tag{5}$$

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 = $\{x_i = S1 \in V_S, \; \mathrm{w}_{\mathrm{S1,S2}} \neq \infty\}$

In the terms of the decision making process model of an action $a_k \subset A_i$ is a mapping:

$$a_k : x_i^n \to x_i^{n+1} \quad x_i^n \subset X \quad x_i^{n+1} \subset V \quad a_i \subset A_i \tag{6}$$

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 all objects defined by a set $M_G \subset V_S$. We introduce an auxiliary set defining objects that have already been visited 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:

$$I(a_1, ..., a_i, ...a_N) = -I_R + I_E + I_D \tag{7}$$

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

$$
\begin{aligned}
&I_R\left(a_1,..a_i,...a_N\right)=\sum_{i=1}^{N} R_i\left(a_i\right) \\
&R_i=\begin{cases} \frac{1}{k}R>0 \quad if \quad x_i^{n+1}\in M_G \cap x_i^{n+1} \notin M_V \\ 0 \qquad\qquad\qquad otherwise \end{cases}
\end{aligned} \tag{8}
$$

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:

$$
I_E\left(a_1,..a_i,...a_N\right)=\sum_{i=1}^{N} w\left(x_i^n, x_i^{n+1}\right) \tag{9}
$$

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:

$$
p_{\min}\left(n,m\right)=\left\{v_n,...v_k,...vm\right\}\subset V \tag{10}
$$

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

$$
\begin{aligned}
&I_D\left(a_1,...,a_i,...,a_N\right)=\sum_{i=1}^{N} D_{\min,i} \\
&D_{\min,i}=\min_{l}\left[p_{\min}\left(x_i^{n+1},u_l\right)\right]
\end{aligned} \tag{11}
$$

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.

Solution

In the previous section we derived a model for a single stage of the exploration process. Now, we look for a solution of the problem defined above. The solution will be a set of actions of individual robots $S^*=\{a_{10},a_{20},...,a_{N0}\}$ that if performed lead to execution of a part of primary task. The problem of exploration is in principle cooperative one but taking into account a fact that robots can not communicate each other during an action execution implies that the problem has features of noncooperative one. Thus we try to find the solution for a single decision-making stage, considering the problem as a noncooperative one. One of the best known concept of solution of such problems is the Nash equilibrium one (Basar and Oldster 1982). Applying it to our problem, the solution is the equilibrium point if following inequalities are satisfied by the set of the decisions $\{a_{10},...a_{N0}\}$:

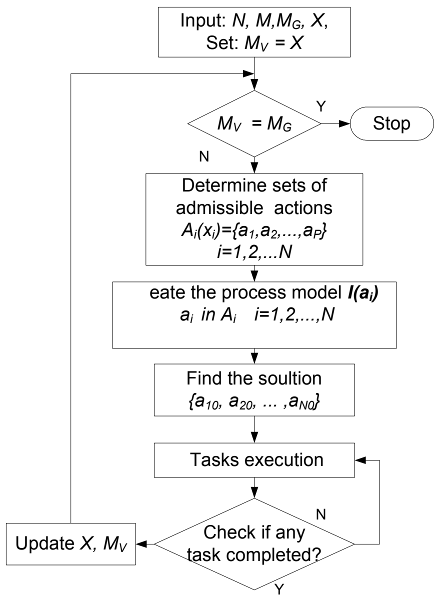
$$
\begin{aligned}
&I_1\left(a_{10},...a_{N0}\right)\le I_1\left(a_1,a_{20},...d_{N0}\right) \\
&\vdots \\
&I_N\left(a_{10},...a_{N0}\right)\le I_N\left(d_{10},d_{20},...d_N\right)
\end{aligned} \tag{12}
$$

In case of a team-problem, where $I_i = I,\ i=1,2,...,N$ the solution reduces to minimization of the cost function I. Thus we have:

$$
S^*=\{a_{10},a_{20},...,a_{N0}\} = \min_{a_1,...,a_N} I \tag{13}
$$

THE ALGORITHM OF EXPLORATION

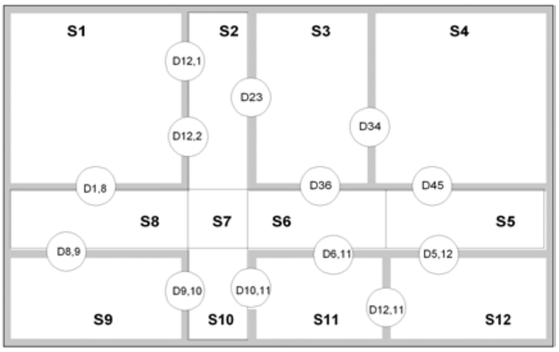
In this section we present an overall algorithm of workspace exploration for a team of robots. The block diagram of the algorithm is presented in the fig.3. First, the global task is stated by the set M_G and the state of the robot team X is determined using information of robots location. In the next step of the algorithm admissible actions of individual robots are determined taking into account the current state X of robots and environmental model M. Then, the model of the process is calculated according to equations (7)-(11). The solution of the problem is computed using formulas (12),(13) and it determines for each robot an elementary task (action) to be performed. The action code is sent to the motion-controller. A robot that executes an action as a first sends a message to the coordinator. That causes the state X of a team as well as the state of the primary task completion are updated. The process described above is repeated until the task is completed $M_G = M_V$.



Figures 3: The algorithm of exploration

SIMULATION EXAMPLE

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 *M.A.S.S.* (Multi Agent Systems Simulator) which was created and has been developed by the author. This application allows to create a model of a well-structured workspace and simulate the work of differentially-driven mobile robots inside of the modeled workspace. The layout of a workspace we modeled is shown in the fig.4.



Figures 4: 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 (fig. 5) and it is modelled by different cluttering coefficients c_j fixed to each sector. In the presented simulation they are equal C={0.7,0,0.1,0.6,0,0,0,0,0.2,0,0.3,0.2}. All of probabilities o_j that doors are opened are equal one except the door *D3,6* which is open with probability o_5=0.2. That corresponds to the fact that this door most time is closed. The robot model used for the simulation is typical differential driven one, with 8 range sensors placed around the disc-shaped platform of diameter 0.55[m]. We consider the following exploration task. A team of three robots *(N*=3) is intended to explore the workspace what is equivalent to visiting all of the sectors. Thus the goal of exploration is defined as $M_G = V_S$. Initially, robots are located inside of sectors $X_{init} = \{v_4, v_9, v_2\}$ so $M_V = X_{ini}$. In the fig.5 successive stages of simulation of exploration process are presented. A sequence of operators that was used to perform the task is presented in the table 1. The symbols *F*D, *T*D, *G*T, *W* denote operators: *FindDoo*r, *TraverseDoo*r, *GoT*o, *Wai*t. 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.

Table 1: A Sequence of Operators that Provides Completion of the Exploration Task

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	FD(D2,3)	FD(D10,11)	FD(D1,8)
4	TD(D2,3,S2)	TD(D10,11,S11)	TD(D1,8,S8)
5	W()	FD(D11,12)	GT(S7)
6	W()	TD(D10,11,S12)	GT(S6)
7	W()	W()	GT(S5)

SUMMARY AND CONCLUSION

In this paper we discussed a problem of planning and coordination of tasks in a multi robot system. We considered a team of robots that was intended to perform a task of exploration 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 simulation environment allowed us to verify this approach. Taking into account the results of simulations we can state that the approach presented in the paper seems to be promising. Therefore our future researches shall be focused on applying it to a real multi robot system.

Acknowledgments

This work has been supported by KBN Grant No. 3 T11A 012 26 in the year 2004.

REFERENCES

Althaus, P. and H.I. Christensen. 2003. “Behaviour coordination in structured environments.” *Advanced Robotics*, 17(7), 657–674.

Arkin, R.C. 1998. *Behavior-Based Robotics*. MIT Press, Cam-bridge, MA.

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

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

Fox, D.; W. Burgard; and S. Thrun. 1998. “A Hybrid Collision Avoidance Mathod For Mobile Robots.” *Proc. of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium.

Gerkey, B. and M.J. Mataric. 2002. “Pusher-watcher:An Aapproach to Fault-Tollerant Tightly-Coupled Robot Coordination.” *Proc. of the IEEE International Conference on Robotics and Automation (ICRA2002)*, Washington DC.

Golfarelli, M. and N. Meuleu. 2000. ”Multi Agent Coordination in Partially Observable environments.” *Proceedings of the 2000 IEEE Int. Conf. of Robot and Automation*, San Francisco, 2777- 2782.

LaValle, S. 2000. ”Robot motion planning:A game-theoretic foundation. Algorithmica.” 26(3), 430–65.

Lawton, J.R. et al. 2003. “A Decentralized Approach to Formation Maneuvers.” *IEEE Transactions on Robotics and Automation*, Vol 19, No 6, 933–941.

Li, Q. and S. Payandeh. 2001. ”On Coordination of Robotics Agents Based on Game Theory”, *ICRA 200*1, Seul, Korea, (private communication).

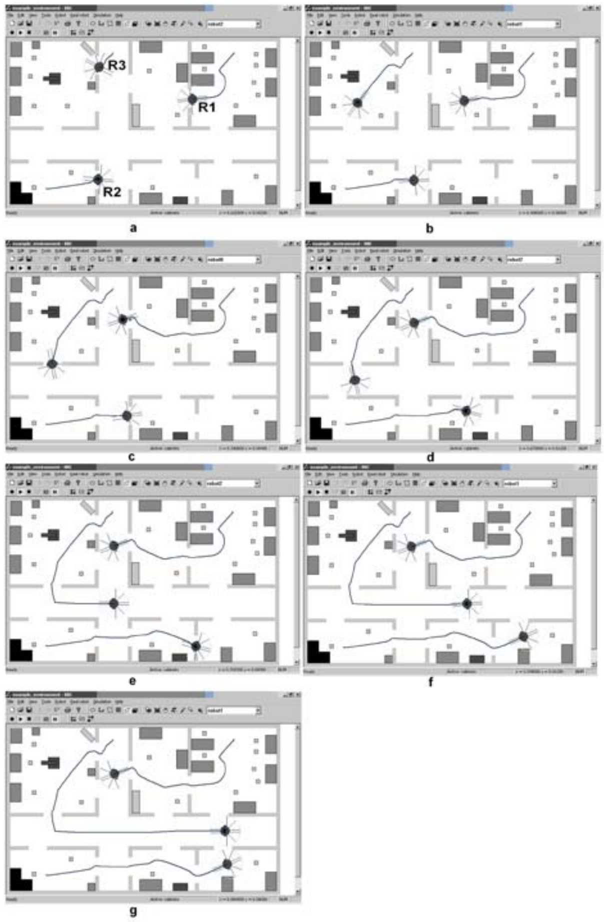
Sequiera, J. and M. I. Ribeiro. 2001. ”A Negotiation Model for Cooperation Among Robots”, *Proc. of European Control Conferenc*e, Porto, Portugal.

Shim, H.S. et al.2000. “A hybrid control structure for vision based soccer robot system.” *Int. J. Intelligent Automation and Soft Computing*, 6: (1), 89–101.

Schneider-Fontan, M. and M.J. Mataric. 1998. ”Territorial Multi-Robot Task Division.” *IEEE Transactions on Robotics and Automation*, Vol 15, N.5, 815-822.

Skrzypczyk, K. 2002. ”Behaviour-based approach to a synthesis of a mobile vehicle control system.” *IV-th Domestical PhD Workshop OWD*, Istebna-Zaolzie, Poland.

CHRISTOPHER T. SKRZYPCZYK was born in Tarnowskie Gory, Poland and went to the Silesian University of Technology in Gliwice, where he studied automatics and robotics and obtained his degree in 2000. In the same year he started his PhD studies at the Department of Automatic Control at the same university. His interests have been focused on mobile robots collision free motion planning and coordination methods in Multi Agent Systems based on the game the-ory. His e-mail is: `kskrzypczyk@ia.polsl.gliwice.pl`.



Figures 5: The result of the exemplary simulation