Int. J. Appl. Math. Comput. Sci., 2005, Vol. 15, No. 1, 89–97

CONTROL OF A TEAM OF MOBILE ROBOTS BASED ON NON-COOPERATIVE EQUILIBRIA WITH PARTIAL COORDINATION

KRZYSZTOF SKRZYPCZYK

Department of Automatic Control
Silesian University of Technology
ul. Akademicka 16, 44–100 Gliwice
e-mail: krzysztof.skrzypczyk@polsl.pl

In this work we present an application of the concept of non-cooperative game equilibria to the design of a collision free movement of a team of mobile robots in a dynamic environment. We propose the solution to the problem of feasible control synthesis, based on a partially centralized sensory system. The control strategy based on the concept of non-cooperative game equilibria is well known in the literature. It is highly efficient through phases where the solution is unique. However, even in simple navigation problems, it happens that multiple equilibria occur, which incurs a problem for control synthesis and may lead to erroneous results. In this paper we present a solution to this problem based on the partial centralization idea. The coordinator module is incorporated into the system and becomes active when multiple equilibria are detected. The coordination method includes a "fair arbiter" for the selection of an appropriate equilibrium solution. Simulation studies of the proposed methodology were carried out for 2, 3 and 5 robots, and their results are presented.

Keywords: multi-robot systems, motion planning, game theory

1. Introduction

In recent years a lot of attention has been paid to the problem of multiple robot motion planning. One of the tools for modeling and solving the problem of interactions between robots that share a common workspace is game theory. In the context of the application of game theory to mobile robot motion planning, several approaches have been reported in the literature. Golfarelli (1998) presents the collision avoidance problem faced by a population of "self-interested" interacting agents as a 2-player repeated game. He uses the criterion based on the "maxi-max" concept. The study by Li and Payandeh (2001) addresses the problem of planning motions of two robotic agents that perform cleaning up and collection tasks. In this study a multi-stage zero-sum game is proposed to model and solve the problem. A zero-sum game and the concept of a saddle point solution are proposed as a tool for sensor-based motion planning for mobile robots by Esposito and Kumar (2000). LaValle (1994; 2000) gives a method for analyzing and selecting time-optimal navigation strategies for n robots whose configurations are constrained to lie on a C-space road map. He uses the maximal Nash equilibria concept to find favorable strategies for each robot. The work by Chitsaz *et al.* (2004) presents an algorithm that computes a complete set of Pareto-optimal coordination strategies for two translating polygonal robots on a plane.

Of course, game theory is not the only method of coordination for multiple robots. There are at least two recent works advocating other approaches which are worth mentioning. The first one, by Belta and Kumar (2004), deals with the problem of generating optimal smooth trajectories for a group of fully mobile robots. They introduce a geometrical approach to control the motion of a rigid formation of a team of robots. In turn, Gerkey and Mataric (2002) present a method of dynamic task allocation for a group of mobile robots based on an auction method. Using this method, different types of cooperative behavior of a team of robots can be obtained.

Considering the game theoretical approach, the papers referenced above give a quite exhaustive range of game theory based approaches to multiple robot agent control. They show the advantages of this approach, which are (a) the avoidance of the computational explosion related to a full search of the configuration space and (b) a good intuition support coming from analogies to conflicting behavior in many areas. However, all the previous studies lack the treatment of one, but very important, aspect of this approach: the existence of multiple solutions to game problems. The existence of multiple solutions is common even for rather simple problems. Simulations show that the choice of a game equilibrium in such a situation is very important for the quality of the overall solution.

In this paper we propose a solution to the multiple equilibria problem based on the idea of partial coordination. The coordinator module is added to the navigation system and it becomes active when multiple equilibria are reported by moving robot agents. For the coordinator we will also interchangeably use the term arbiter. Apart from choosing the best one from among several equilibria, the arbiter is also to tackle the situation when no equilibria exist.

The partially centralized design allows for efficient and collision free control for robot motion. We present a series of simulations that prove the effectiveness of our approach. We investigate the influence of primary algorithm parameters on two differential drive vehicles.

2. Problem Formulation

The analyzed problem involves planning and controlling robot motions from their initial locations to their target locations while avoiding environmental obstacles. From the perspective of fast reactive control, the problem reduces to finding at each time moment (with the resolution of Δt) a set of controls that, when applied to robots, push them towards their goal locations. We consider the control of N mobile robots sharing the same workspace W with M moving obstacles inside. We make the following assumptions:

- The position and orientation vector $\boldsymbol{p}$ of each robot is known at each time moment $t_n,\ n = 1, 2, \ldots$.
- The coordinates of goal locations are known.
- The obstacles can be described by convex polygons, and the position of the center of mass of each obstacle is known.
- The robot actions (controls) are synchronized with the incoming sensory information.

Denote the state of the i-th robot by

$$p_i = [x_i\ y_i\ \Theta_i]\,, \quad i = 1, 2, \ldots, N. \tag{1}$$

The target location of the robot is denoted by

$$g_i = [x_{g,i}\ y_{g,i}]\,, \quad i = 1, 2, \ldots, N. \tag{2}$$

It is assumed that the sensory system supplies information about the location and heading of each team mate. As for the obstacles, it can supply only the location of each obstacle. Therefore, the heading of each obstacle must be extracted from information about the current and previous positions. Denote the state of the i-th obstacle by

$$b_j = \left[x_j\ y_j\ \hat{\Theta}_j\right], \qquad j = 1, 2, \ldots, M. \tag{3}$$

The heading of the obstacle $\hat{\Theta}_j$ is estimated as follows:

$$\hat{\Theta}_j = \arctan\left(\frac{x_j(t_n) - x_j(t_{n-1})}{y_j(t_n) - y_j(t_{n-1})}\right), \tag{4}$$

where t_n and t_{n-1} denote the current and previous time moments, respectively. In what follows, we shall consider the control of models of small, disk-shaped laboratory mobile robots. The diameter of their driving platforms is about 5 cm. The robots are differential drive vehicles which move by changing the velocities of the right and left wheels. For the purpose of our method, we define the control of the i-th robot as

$$u_i = [\omega_i\ v_i], \qquad i = 1, 2, \ldots, N, \tag{5}$$

where ω_i and v_i are respectively angular and linear velocities of the i-th robot, related to the velocities of the left and the right wheel by the formula

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = \begin{bmatrix} 1 & L/2 \\ 1 & -L/2 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{6}$$

Here v_L and v_R are linear velocities of the left and the right wheel, respectively, and L is the distance between the wheels.

With the above notation, the problem can be formulated as follows: At each discrete time moment ($t = n\Delta t$ for $n = 0, 1, 2, \ldots$), for each robot find control $u_i,\ i = 1, 2, \ldots, N$, which, applied to it, will lead to both a collision free and a goal oriented movement. We denote the set of controls generated at the moment of time t_n by

$$S(t_n) = \{u_{10}, u_{20}, \ldots, u_{N0}\}. \tag{7}$$

3. Collision Free Control and Game Theory

The problem of the coordination of multiple robots that share a common workspace can be perceived as a conflict situation between individual agents-robots. Each robot has its individual independent goal related to a point-to-point navigational task. The fact that robots share the workspace implies the necessity to analyze interactions between the robots. The existence of mobile obstacles inside the workspace makes the problem more complex. A convenient framework for modeling and solving problems of the conflict is game theory. In order to properly apply game theory to our problem, several issues have to be taken into account. These are listed below:

1. First, the interests (goals) of individual agents are not necessarily antagonistic. In other words, not every set of controls applied to the robots leads to a collision. Therefore, the game associated with robot control is a non-zero sum one.

2. Although the overall process of control is in principle dynamic, we understand it as a sequence of static problems. This assumption makes it numerically tractable.

3. In order to allow using algorithms for solving games for equilibria, we assume finite sets of discrete levels for possible control actions.

4. Problem Modeling

In this section we design a model of the decision making process in two stages: First, we have to obtain finite element sets of decisions which could be adopted by particular agents. Next, loss functions associated with each of the agents are defined.

4.1. Discretization of the Control Space

Discretize first both the angular and linear velocities:

$$\Omega_i = \left[\omega_i^1, \omega_i^2, \dots \omega_i^{K_i}\right], \qquad i = 1, 2, \dots, N, \quad (8)$$

$$V_i = \left[v_i^1, v_i^2, \dots, v_i^{L_i}\right], \qquad i = 1, 2, \dots, N, \quad (9)$$

where K_i and L_i are respectively the numbers of distcretization levels for the angular and linear velocities of the i-th robot.

The set of possible controls of the i-th robot is defined as the Cartesian product

$$U_i = \Omega_i \times V_i,$$
$$U_i = (\omega_i, v_i) : \omega_i \in \Omega_i \cap v_i \in V_i. \quad (10)$$

It is easy to notice that this method of discretization leads to large decision (control) sets and, in consequence, to a large problem size. Therefore, we propose a method based on using only one control variable, i.e., the angular velocity. We have

$$U_i = \left\{(\omega_i, v_i) : \omega_i \in \Omega_i \cap v_i = f(d_T, d_{\min})\right\}, \quad (11)$$

where $f(d_T, d_{\min})$ is a heuristic function used for determining a proper value of the linear velocity of the robot. Thus the size of the decision set U_i is equal to K_i. In order to define the formula of the heuristic function, we applied the following rules:

- The i-th robot moves at the velocity $v_{i,\text{opt}}$ when its distance to the nearest object $L_{\min,i}$ is greater than some treshold L_{R0}.
- The velocity is decreased when the nearest object is closer than L_{R0}.
- In proximity to the target the velocity is successively decreased to allow the so-called “soft landing” on the target location.

The heuristics above is encapsulated in the following expression:

$$v_i = \frac{1}{2} v_{i,opt} w_T \left(1 + w_R\right), \quad (12)$$

where w_T and w_R are respectively the coefficients of the influence of the target point and a nearest object

$$w_T = \frac{1}{1 + e^{-\alpha(L_{T,i} - L_{T_0})}},$$
$$w_R = \frac{1}{1 + e^{-\beta(L_{\min,i} - L_{R_0})}}, \quad (13)$$

with $L_{T,i}$ meaning the distance of the i-th robot to the target point. The coefficients α and β in (13) determine the sensitivity of the weighting factors in the neighborhood of the threshold distances L_0 and L_{R0}, respectively.

4.2. Loss Function

In this section we model loss functions associated with each of the agents. The loss of an agent is influenced by decisions made by other team mates and locations of obstacles. Therefore we write the loss function the of i-th robot as

$$I_i\left(d_1, d_2, \dots, d_N\right)$$
$$= f_i\left(d_1, d_2, \dots, d_N,\ b_1, b_2, \dots b_M\right) \quad (14)$$

for $i = 1, 2, \dots, N$, where $d_i \in \{1, 2, \dots, K_i,\}$ denotes the decision of the i-th agent that consists in selecting the d_i-th element from the decision set U_i defined by (11). To construct the function, we make use of the potential field method (Koren and Borenstein, 1991; Ge and Cui, 2002). We fill the robot’s workspace with an artificial potential field in which the robot is attracted to its target position and repulsed away from the obstacles. We compute the attractive and repulsive forces for the i-th robot as functions of decisions adopted by other robots. The attractive force applied to the i-th robot depends on a relative distance between the robot and its target. The value of the attractive force that would be applied to the robot as a result of its action d_i is given by

$$|\boldsymbol{F}_{a,i}(d_i)| = k_a \frac{1}{\hat{L}^2_{g,i}\left(d_i\right)}. \quad (15)$$

The direction of the force is interpreted in Fig. 1. The coefficient k_a in the above equation constitutes a gain that adjusts the influence of the attractive component to the resultant virtual force affecting the robot. In (15), $\hat{L}_{g,i}$ is the predicted distance between the i-th robot and its target location. It is computed as follows:

$$\hat{L}_{g,i}(d_i) = \sqrt{\left(x_{g,i} - \hat{x}_i^{d_i}\right)^2 + \left(y_{g,i} - \hat{y}_i^{d_i}\right)^2}, \quad (16)$$

where

$$\begin{aligned}\hat{x}_i^{d_i} &= x_i + v_i^{t_{n-1}} T_o \cos(\Theta_i + \omega_i^{t_{n-1}} T_o) \\ &\quad + v_i(\Delta t - T_o)\cos(\Theta_i + \omega_i^{t_{n-1}} T_o \\ &\quad + \omega_i^{d_i}(\Delta t - T_o)), \\ \hat{y}_i^{d_i} &= x_i + v_i^{t_{n-1}} T_o \sin(\Theta_i + \omega_i^{t_{n-1}} T_o) \\ &\quad + v_i(\Delta t - T_o)\sin(\Theta_i + \omega_i^{t_{n-1}} T_o \\ &\quad + \omega_i^{d_i}(\Delta t - T_o)),\end{aligned} \tag{17}$$

where $\omega_i^{d_i}$ denotes the angular velocity applied to the robot as a result of the decision d_i and v_i is the linear velocity of the robot set according to (12). We assume that there exists a delay in the system between the moment of receiving the sensory information and that of making a decision. Therefore we introduce to the model the time of delay T_0 which is the worst-case estimate of the time of computing and sending the information. The quantities $\omega_i^{t_{n-1}}$ and $v_i^{t_{n-1}}$ are previous controls applied to the i-th robot that still have an effect on the robot by the time T_0.

The repulsive force is a vector sum of forces generated by other team mates and obstacles:

$$\boldsymbol{F}_{\boldsymbol{r},\boldsymbol{i}}(d_1,\ldots,d_N) = \sum_{\substack{j=1\\ j\neq i}}^{N} \boldsymbol{F}_{\boldsymbol{r},\boldsymbol{i},\boldsymbol{j}}(d_i,d_j) + \sum_{j=1}^{M} \mathbf{F}_{b,\mathbf{i},\mathbf{j}}(d_i) \tag{18}$$

for $i = 1,2,\ldots,N$, where $\boldsymbol{F}_{\boldsymbol{r},\boldsymbol{i},\boldsymbol{j}}(d_i,d_j)$ is a predicted repulsive force generated by the j-th robot that affects the i-th robot. Similarly, the force generated by the j-th obstacle applied to the i-th robot is denoted by $\mathbf{F}_{b,\mathbf{i},\mathbf{j}}(d_i)$. The force $\mathbf{F}_{\mathbf{r},\mathbf{i},\mathbf{j}}(d_i,d_j)$ is computed as follows:

$$|\mathbf{F}_{\mathbf{r},\mathbf{i},\mathbf{j}}(d_i,d_j)| = \begin{cases} k_r\left(\dfrac{1}{\hat{L}_{ij}(d_i,d_j)} - \dfrac{1}{L_0}\right)^2 & \\ \qquad \text{if } \hat{L}_{ij}(d_i,d_j) < L_0, & \\ 0 \qquad\qquad \text{otherwise}, & \end{cases} \tag{19}$$

where L_0 is a limit distance of the influence of virtual forces and $\hat{L}_{i,j}(d_i,d_j)$ is the predicted distance between the i-th and j-th robots after applying by them their actions d_i and d_j. We compute it from

$$\hat{L}_{i,j}(d_i,d_j) = \sqrt{\left(\hat{x}_i^{d_i} - \hat{x}_j^{d_j}\right)^2 + \left(\hat{y}_i^{d_i} - \hat{y}_j^{d_j}\right)^2}. \tag{20}$$

The robot locations $(\hat{x}_i^{d_i}, \hat{y}_i^{d_i})$ and $(\hat{x}_j^{d_j}, \hat{y}_j^{d_j})$ are determined from (17). Similarly, we determine the force $\mathbf{F}_{b,\mathbf{i},\mathbf{j}}(d_i)$:

$$|\boldsymbol{F}_{b,\boldsymbol{i},\boldsymbol{j}}(d_i)| = \begin{cases} k_{rb}\left(\dfrac{1}{\hat{L}_{ij}(d_i)} - \dfrac{1}{L_0}\right)^2 & \text{if } \hat{L}_{ij}(d_i) < L_0, \\ 0 & \text{otherwise}, \end{cases} \tag{21}$$

where $\hat{L}_{i,j}(d_i)$ is the predicted distance of the i-th robot from the j-th obstacle after time Δt:

$$\hat{L}_{ij}(d_i) = \sqrt{\left(\hat{x}_i^{d_i} - \hat{x}_{o,j}\right)^2 + \left(\hat{y}_i^{d_i} - \hat{y}_{o.j}\right)^2}. \tag{22}$$

The estimated location of the j-th obstacle is determined from the current state $\boldsymbol{b}_j$ and the previous state $\boldsymbol{b}_j^{t_{n-1}}$ of the obstacle:

$$\begin{aligned}\hat{x}_{b,j} &= 2x_{b,j} - x_{b,j}^{t_{n-1}}, \\ \hat{y}_{b,j} &= 2y_{b,j} - y_{b,j}^{t_{n-1}}.\end{aligned} \tag{23}$$

The coefficients k_r in (19) and k_{rb} in (21) are gains that adjust the influence of repulsive components of the resultant force applied to the robot. Finally, the force that influences the i-th robot is the vector sum of attractive and repulsive forces:

$$\boldsymbol{F_i} = \boldsymbol{F_{a,i}} + \boldsymbol{F_{r,i}}. \tag{24}$$

The geometrical interpretation of the forces that influence the robot is presented in Fig. 1.

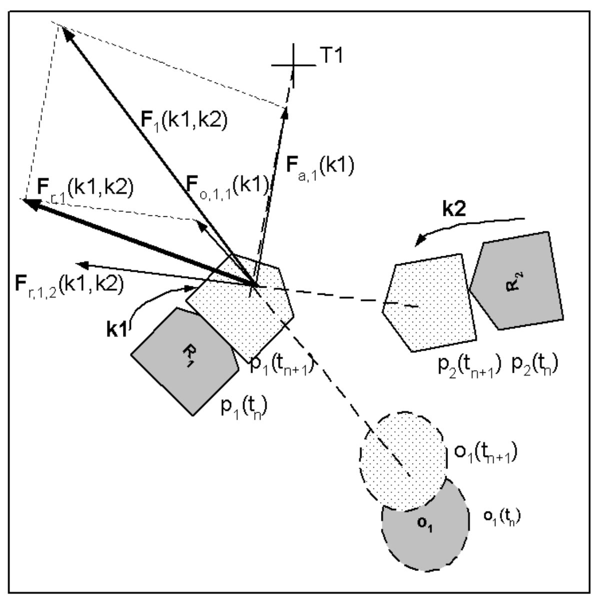


Fig. 1. Geometrical interpretation of virtual forces applied to a robot.

The virtual force applied to the robot moving in a given direction can be considered as a measure of safety.

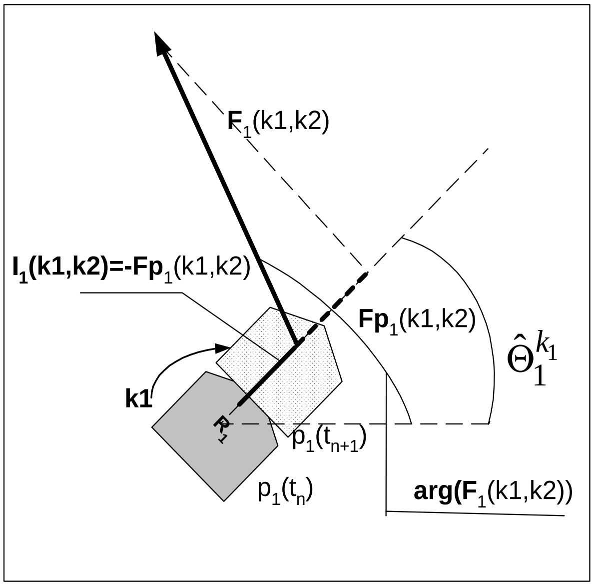


Fig. 2. Interpretation of the loss function.

The greater the force applied along a given direction, the greater the safety of the movement in this direction. Therefore, we employ the projection of the force vector (Fig. 2) onto the direction of the predicted movement of the robot to define the cost function of the i-th robot. Since we want to minimize the cost function, we determine it as follows:

$$I_i\left(d_1, d_i, \dots, d_N\right) = -\left|\mathbf{F_i}\right| \cos\left(\arg\left(\mathbf{F_i}\right) - \hat{\Theta}_i^{d_i}\right), \tag{25}$$

where

$$\hat{\Theta}_i^{d_i} = \Theta_i + \omega_i^{t_{n-1}} T_0 + \omega_i^{d_i}\left(\Delta t - T_0\right). \tag{26}$$

The potential field method in the form presented above has one serious drawback–it is prone to local minima. In such cases the value of the resultant virtual force is equal to zero and the solution is not determined. In our problem virtual forces are generated by moving objects and they are functions of possible actions of individual robotic agents. That makes the possibility that a given robot is trapped in a local minimum very small, so it is neglected in our deliberations.

5. Problem Solution

The key issue in the presented problem is to find, at each time moment $t = n\Delta t$, $n = 1, 2, \dots$, a set of decisions (7) which, when applied to robots, lead to a collision free, goal oriented navigation. Since we assume that the robots cannot communicate with one another within the period of time Δt, it seems reasonable to pose the problem as a non-cooperative one. The well-known concept of the solution is the Nash equilibrium (Basar and Olsder, 1982) defined by the following set of inequalities:

$$\begin{aligned} I_1\left(d_{10}, d_{20}, \dots, d_{N0}\right) &\le I_1\left(d_1, d_{20}, \dots, d_{N0}\right), \\ I_2\left(d_{10}, d_{20}, \dots, d_{N0}\right) &\le I_2\left(d_{10}, d_2, \dots, d_{N0}\right), \\ &\vdots \\ I_N\left(d_{10}, d_{20}, \dots, d_{N0}\right) &\le I_N\left(d_{10}, d_{20}, \dots, d_N\right). \end{aligned} \tag{27}$$

5.1. Partial Coordination

A problem occurs when (a) there are multiple Nash equilibria, and (b) there is no Nash equilibrium point. According to the author’s experience coming from many simulation studies, the situation (b) happens extremely rarely. Navigation problems must be specially constructed in order to encounter (b). Nevertheless, to make the control system complete, one can provide a solution to (b) by using the so-called $\min$–$\max$ (safety) strategy (Basar and Olsder, 1982):

$$\begin{aligned} d_{10} &= \min_{d_1} \max_{d_2,\dots,d_N} I_1\left(d_{10}, d_2, \dots, d_N\right), \\ &\vdots \\ d_{N0} &= \min_{d_N} \max_{d_1,\dots,d_{N-1}} I_N\left(d_1, d_2, \dots, d_{N0}\right). \end{aligned} \tag{28}$$

The case (a) occurs very often in practical problems and we have to choose the most appropriate equilibrium. In some cases we can use the theorem of equilibrium admissibility (Basar and Olsder, 1982). However, this theorem needs the assumption about equilibria comparability, which is rarely fulfilled in practice. In general, the problem of selecting a proper equilibrium is very complex (Maynard, 1982; Harsnyi, 1998; Masterson, 2000).

Here we propose to introduce an arbiter module which aims at selecting the most proper of multiple solutions. The block diagram of the system with the arbiter module is presented in Fig. 3.

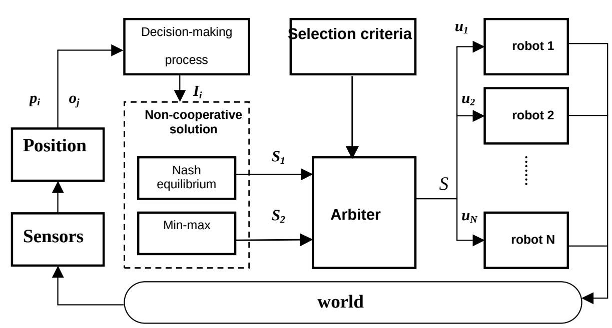


Fig. 3. Control system diagram.

Based on the sensory information provided to the system at successive time moments, the states of the robots and obstacles are extracted. Next, the decision-making problem is modeled according to the rules presented in Section 4. The solution is computed based on the Nash equilibrium (27) or the min–max concept (27). The diagram of the algorithm of the arbitration process is presented in Fig. 4.

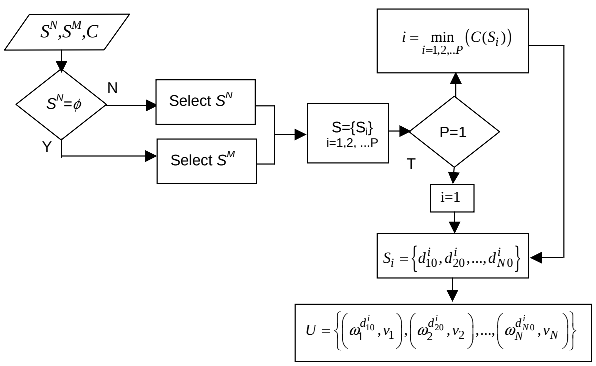


Fig. 4. Algorithm for the arbitration process.

The input data for the arbiter module are sets of solutions S^N computed according to (27) and solutions S^M computed according to (28):

$$S^N = \{d_{i0}^n\}, \quad S^M = \{d_{i0}^m\}, \quad n = 1, 2, \ldots, P_N,$$
$$m = 1, 2, \ldots, P_M, \quad i = 1, 2, \ldots N.$$

Additionally, the selection criterion is provided as a performance index C. If the set S^N is not empty, then it is selected as a preferable one $(S = S^N)$. If there are no Nash equilibria (the set S^N is empty), the arbiter switches to min–max solutions $S = S^M$. The next step is to check if the solution is uniqe $(P = P_N$ or $P = P_M = 1)$. If it is, the control of robots is performed according to

$$U = \left\{ \left(\omega_1^{d_{10}^n}, v_1\right), \left(\omega_2^{d_{20}^n}, v_2\right), \ldots, \left(\omega_N^{d_{N0}^n}, v_N\right) \right\}, \tag{29}$$

where $n = PN = 1$ or $n = m = PM = 1$. Otherwise $(P \leq 1)$, and the arbiter has to choose between multiple solutions. We introduce additional selection criteria stated as a performance index $C(S_n)$ where $S_n = \{d_{io}^k\}, \quad i = 1, 2, \ldots, N, \quad k = 1, 2, \ldots P$, and $S_{\mathrm{n}} \subset S$. The criterion is the minimal total cost for all robots and a uniform distribution of costs among the robots:

$$\begin{aligned} C(S_n) &= \sum_{i=1}^{N} \left(I_i(S_n) + \left| I_i(S_n) - \bar{I} \right| \right), \\ \bar{I} &= \frac{1}{N} \sum_{i=1}^{N} I_i(S_n). \end{aligned} \tag{30}$$

On that basis the arbiter chooses a solution $S^* = S_n$ that minimizes the index $C(S_n)$:

$$S^* = S_n = \min_{n=1,2,\ldots,P} \left(C(S_n) \right). \tag{31}$$

Finally, the robot control is determined from (29).

The solution method presented in this section allows us to generate a unique control of robots and provides both collision free and goal oriented navigation of multiple robots. Unfortunately, the solution we obtain may not be globally optimal. Morover, it is hard to prove its optimality at all. This is caused by the fact that the solution is influenced by a number of parameters. Additionally, there is no explicit optimization criterion stated and the optimality of the solution can be considered in many ways. On the other hand, the generated trajectories (several of them are presented in the next section) are smooth and collision free, and in this sense the solution can be considered as satisfactory.

6. Simulation Studies

6.1. Influence of the Time of Delay

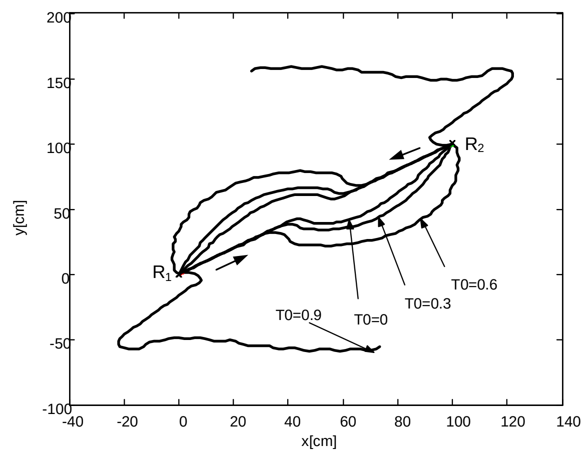
In this section we investigate the influence of including the time of delay in the process of modeling on the resulting trajectories. The first experiment shows the effect of modeling without considering the time of delay (cf. Fig. 5(a)). We assume that there exists a delay in the system which is simulated and it can be expressed as

$$T_0^{\mathrm{true}} = \frac{1}{2} T_0 \left(1 + \delta t\right), \quad \delta t \in [0, 1],$$

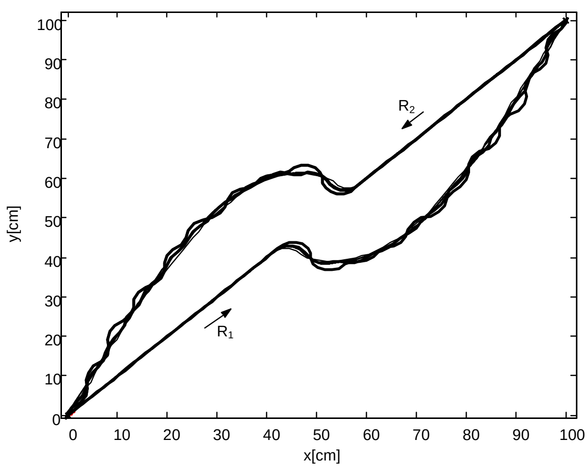
where T_0 denotes a maximal analysed time of the delay in the system and δt is a normally distributed random variable. The simulations were made for 5 values of $T_0 = 0,\ 0.3\Delta t,\ 0.6\Delta t, 0.9\Delta t$ and $\Delta t = 0.1\,$[s].

We applied five-element decision sets $U_1 = U_2 = \{(-225, v_i), (-112.5, v_i), (0, v_i), (112.5, v_i), (225, v_i)\}$, where v_i is determined in accordance with (12). The preferable linear velocities are set as $v_{1,\mathrm{opt}} = v_{2,\mathrm{opt}} = 25\,$[cm/s].

The unmodeled delay has a great influence on the system performance. If the delay time is $T_0^{\mathrm{true}} > 0.9\Delta t$ versus the modeled time $T_0 = 0$, then the robots are not able to reach their targets (Fig. 5(a)). If we introduce the knowledge of the true delay (the worst-case estimation of the time of delay) to the trajectory design, i.e., we set $T_0 = T_0^{\mathrm{true}} = 0.9\Delta t$, the robots will reach their targets (Fig. 5(b)). Decreasing $T_0 = T_0^{\mathrm{true}}$ results in shorter and smoother trajectories.



(a)



(b)

Fig. 5. Comparison of simulation results made for two robots in the case when T0 was not modeled (a) and when it was (b).

6.2. Enlargement of Decision Sets

Now we examine the influence of the size of decision sets on the control process. We check the system for up to 17 possible decisions. The rest of the parameters are identical to those of the previous section. Figure 6 presents the comparison of the trajectories obtained with the use of 2, 5, 9 and 17-element decision sets.

The trajectories in Fig. 6 do not differ very much. Therefore, in order to evaluate the quality of the obtained trajectories, we use the following performance index:

$$J_i = \frac{1}{H}\sum_{n=1}^{H}\left(u_i(t_n) - u_i(t_{n-1})\right)^2$$

for $i = 1, 2, \ldots, N$, where H is the simulation horizon, and $u_i(t_n)$ and $u_i(t_{n-1})$ denote successive controls applied to the i-th robot. In Fig. 7(a) the dependence of the J_i values on the number of possible robots decisions obtained for Robot 1 is presented. Increasing the number of decisions does not improve the control quality very much. On the other hand, the computation time increases much more than the index J decreases Fig. 7(b). Therefore, it is possible to set the number of decisions to a reasonably small value and obtain good control quality.

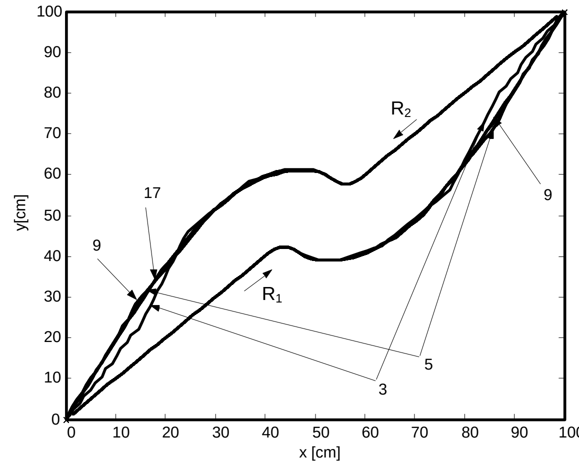


Fig. 6. Result of enlarging the size of decision sets. Simulations were carried out for 3, 5, 9 and 17 possible decisions.

6.3. Experiments with Moving Obstacles

Now we want to show how the system works in the presence of moving obstacles and for a larger number of controlled robots. Figures 8 and 9(a) present the execution of collision free navigation tasks for 3 and 5 robots, respectively. In both cases two moving obstacles were introduced, denoted by o_1 and o_2. The decision sets $U_1, \ldots, U_5$, as well as the rest of the parameters, are the same as before. The velocities of the obstacles are $v_{o_1} = v_{o_2} = 20$ [cm/s] and $\Delta t = 0.2$ [s]. In order to show that the obtained trajectories are really collision free, the time plot of the distance from the closest object was made for each robot, cf. Fig. 9(b). We can see that the minimal distance is not smaller than 13 cm which, in comparison with the size of the simulated robots ($r = 5$ cm), provides the safety of the trajectories.

7. Conclusion

In this paper a methodology based on non-cooperative games in a normal form was used for motion planning of a team of autonomous mobile robots that operate in a dynamic environment. The idea of the artificial potential field was applied to model the game between individual

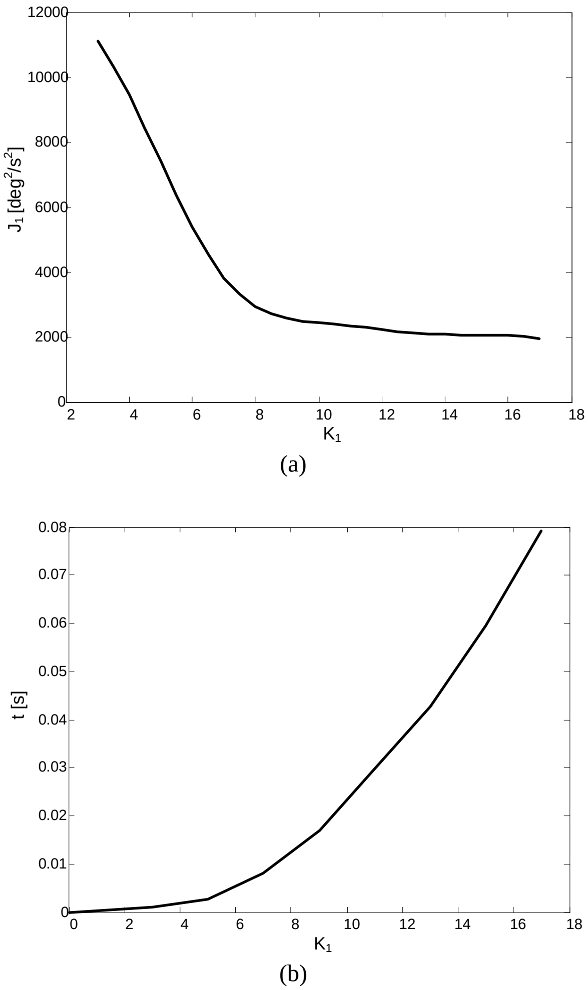


Fig. 7. Comparison of the control quality for Robot 1 defined by the performance index J_1 (a) and the computation time (b).

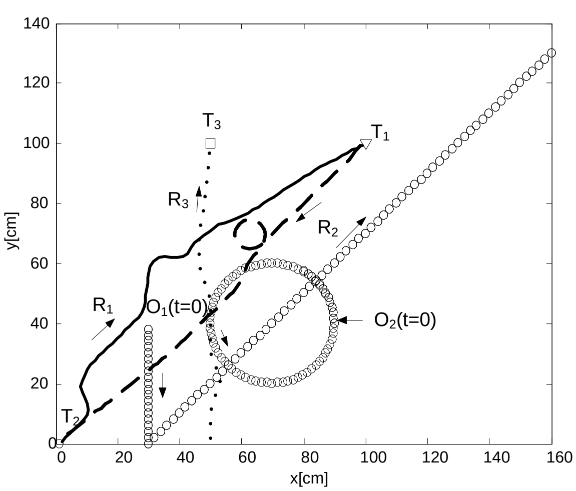


Fig. 8. Simulation results for three robots and two obstacles.

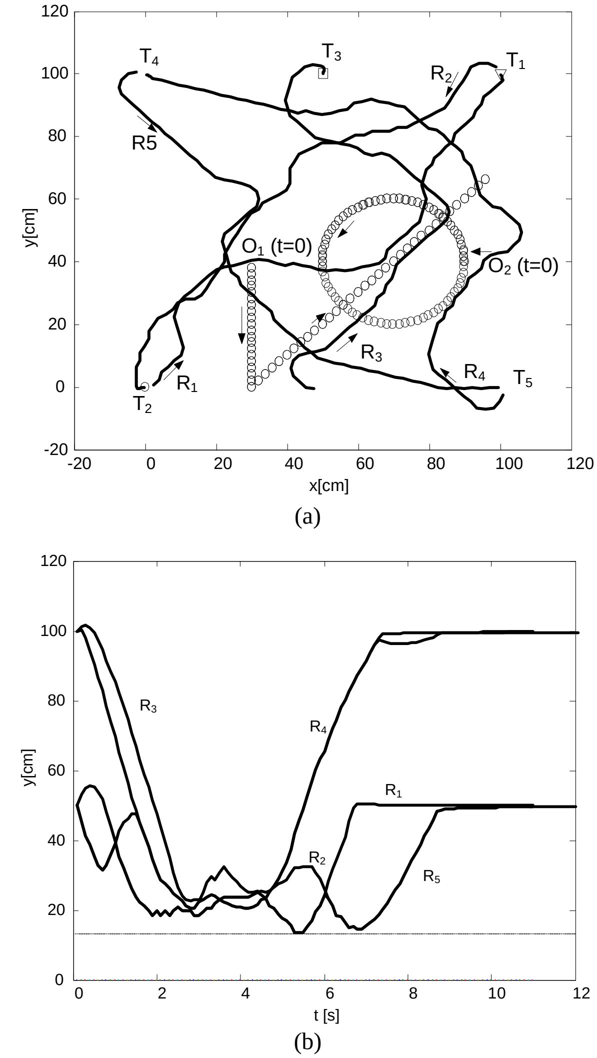


Fig. 9. Collision free trajectories obtained in simulations for 5 robots and 2 obstacles (a), and the evolution of the distance from the nearest object (b).

agents. The solution of the game based on an equilibrium concept was applied. The solution of the non-cooperative problem might not be unique. Therefore, a method of determining the unique solution that is further used to control a robot is needed. We proposed an approach based on partial coordination. The arbiter module was included to choose from among multiple solutions the one which gives a “fair” (uniform) distribution of costs. The simulations that were carried out prove the effectiveness of the presented approach. Moreover, the time of computing the solution is small enough to consider the method as a real-time control one and there is ongoing research devoted to the application of the elaborated approach to a laboratory setup consisting of two mobile robots.

References

Basar M. and Olsder G.J. (1982): *Dynamic Noncooperative Game Theory*. — London: Academic Press.

Belta C. and Kumar V. (2004): *Optimal motion generation for groups of robots: A geometric approach*. — ASME J. Mech. Des., Vol. 126, No. 1, pp. 63–70.

Esposito J. and Kumar V. (2000): *Closed loop motion plans for mobile robots*. — Proc. IEEE Int. Conf. *Robotics and Automation*, *ICRA*, San Francisco, CA, pp. 1020–1025.

Ge S.S and Cui Y.J. (2002): *Dynamic motion planning for mobile robots using potential field method*. — Autonomous Robot., Vol. 13, No. 3, pp. 207–222.

Gerkey B. and Mataric M.J. (2002): *Sold!: Auction methods for multi-robot coordination*. — IEEE Trans. Robot. Automat., Vol. 18, No. 5, p. 758–786.

Golfarelli M. (1998): *A game theory approach to coordination in MAS*. — Proc. 13-th European Conf. *Artificial Intelligence*, Brighton, UK, pp. 610–611.

Harsanyi J.C. and Selten R. (1998): *A General Theory of Equilibrium Selection in Games*. — Massachusetts: MIT Press.

Koren Y. and Borenstein J: (1991): *Potential Field Methods and their Inherent Limitations for Mobile Robot Navigation*. — Proc. IEEE Conf. *Robotics and Automation*, Sacramento, CA, pp. 1398–1404.

LaValle S. (2000): *Robot motion planning: A game-theoretic foundation*. — Algorithmica, Vol. 26, No. 3, pp. 430–465.

Li Q. and Payandeh S. (2001): *On coordination of robotic agents based on game theory*. — (private communication).

Masterton-Gibbons M. (2001): *An Introduction to Game-theoretic Modelling*. — Student Mathematical Library, Vol. II, American Mathematical Society.

Maynard Smith J. (1982): *Evolution and the Theory of Games*. — Cambridge: Cambridge University Press.

Received: 28 January 2004