Experimental Study on Self-organized and Error Resistant Control of Distributed Autonomous Robotic Systems

J. Starke

Interdisciplinary Center for Scientific Computing and
Institute of Applied Mathematics, University of Heidelberg
D-69120 Heidelberg, Germany
starke@iwr.uni-heidelberg.de

T. Kaga

Department of Micro System Engineering
Nagoya University, Nagoya 464-8603, Japan
kaga@robo.mein.nagoya-u.ac.jp

M. Schanz

Institute of Parallel and Distributed Systems (IPVS)
D-70569 Stuttgart, Germany
michael.schanz@informatik.uni-stuttgart.de

T. Fukuda

Center of Cooperative Research in Advanced Science
and Technology
Nagoya University, Nagoya 464-8603, Japan
fukuda@mein.nagoya-u.ac.jp

Abstract

The assignment of distributed mobile autonomous robots to targets, which occurs for instance as an important task in flexible manufacturing environments, is solved by using a self-organization approach motivated by pattern formation principles in biological, chemical, and physical systems. Similar to observations in many natural systems, such as ant tribes, the pattern formation of colored shells or convection patterns in the Rayleigh–Bénard problem of fluid dynamics, the self-organization principles lead to a robust and fault tolerant behavior where the patterns or structures recover from disturbances. The considered problem is the dynamic assignment of a number of robots to given targets where the mobile robots have to move to the targets in order to perform some tasks there. Hereby, each robot uses only local information (i.e., no world coordinate system is necessary). The underlying mathematical problem of the robot–target assignment is the so-called two-index assignment problem from combinatorial optimization. The approach used guarantees always feasible solutions in the assignment of robotic units to targets. As a consequence, for scenarios with only convex obstacles with large enough distances to each other, no spurious states cause the assignment process to fail. The error resistant control method for distributed autonomous robotic systems is demonstrated by several experiments with mobile robots. These results are compared and supplemented with computer simulations.

KEY WORDS—self-organization, selection equation, distributed robotic systems, assignment problems

The International Journal of Robotics Research
Vol. 24, No. 6, June 2005, pp. 465-486,
DOI: 10.1177/0278364905053239

1. Introduction

As in many engineering fields, robotic systems become more and more complex over time. One example is the concept of flexible manufacturing systems that are able to automatically produce products with large variety (Kusiak 1985; Feh 1999; Simeu-Abazi and Sassine 2001). These systems are required to reschedule and reconfigure themselves often to fulfill the current production condition. The large number of changing conditions increases the possibility of the system to fail. Larger varieties often imply smaller quantities for each manufacturing configuration. In order to maintain reasonable setup times and pre-production costs, the manufacturing systems have to be modular. As a consequence, the complexity of these systems increases further. Examples for such modular concepts are hierarchical multi-agent systems (Brooks 1986), dynamically reconfigurable cellular robotic systems (CEBOTs) as proposed in Fukuda and Nakagawa (1988), Fukuda et al. (1991), Fukuda, Kawauchi, and Asama (1992), Fukuda and

Ueyama (1994), "Tetrobot" modular robotic systems (Lee and Sanderson 2000), cooperative agent teams (Asama, Matsumoto, and Ishida 1989; Matarić 1992, 1995; Parker 1993, 1994; Arkin and Bekey 1997; Salido-Tercero et al. 1997; Schneider-Fontán and Matarić 1998; Bojinov, Casal, and Hogg 2002; Schultz, Parker, and Schneider 2003), behavior-based robotic systems (Parker 1996; Balch and Arkin 1998; Matarić 1998) and emergent manufacturing systems (Vaaria and Ueda 1998a, 1998b). The modularity of flexible manufacturing systems and frequently changing conditions, as mentioned above, create the necessity for fault-tolerant and robust control mechanisms (Barraquand, Langlois, and Latombe 1992; Brafman et al. 1997; Kavraki and Latombe 1998; Goldberg and Matarić 2002; Jung and Sukhatme 2002). Due to its robustness proven in many biological, chemical, and physical systems, a promising approach is to combine potential field and force models (Latombe 1993; Balch and Hybinette 2000; Parker 2002) with specifically constructed dynamical systems (Molnár and Starke 2001; Feddema, Robinett, and Driessen 2003) based on self-organization principles.

In conventional rule-based systems or technical systems using if-then-else structures, it is in principle possible to implement fault-tolerant behavior. However, it is difficult to cover all cases that are necessary for a complete solution strategy for recovering from dead-locks. In contrast to these conventional technical systems controlled by if-then-else structures, the functional structures in natural systems, such as, for example, in ant tribes, show fault-tolerant and flexible behavior, which makes them very robust against external interference and disturbances. Such fault-tolerant, flexible, and robust behavior is caused by the interaction of simple rules, which lead to the self-organized emergence of functional structures. The expression "self-organization" is used because the structure is not enforced by an external control but emerges from the interaction of the simple rules of the system itself. Inspired by such biological systems, a self-organization approach is used in the following to control distributed robotic units, and to generate the desired fault-tolerant, flexible, and robust behavior.

In a first attempt towards self-organized flexible manufacturing with sound mathematical foundation, the work presented here focuses on one particularly important problem of distributed autonomous robotic systems, which has recently attracted the interest of many researchers (see, for example, Lueth et al. 1998; Parker, Bekey, and Barhen 2000; Asama et al. 2002); the assignment of mobile robotic units to a certain number of targets to which the robots have to move is a common problem in industrial manufacturing systems. A simplified form of this problem is addressed in this paper. For instance, in transportation systems, automatic guided vehicles (AGVs; Pu and Hughes 1994) have to be assigned to loading and delivering stations.

To guarantee a fault-tolerant mode of operation of the system even in the case of failures, a dynamical system approach that leads to a self-organized dynamical behavior of the robotic units is used. This proposed control mechanism consists of two parts: the assignment part and the navigation part. In order to adapt the control method to several problems, it is important to extract the underlying mathematical structure. Many of these problems can be mapped onto the so-called assignment problem of combinatorial optimization (see, for example, Burkard 1979). To solve the two-index assignment problem by a self-organization process, a specific dynamical system approach is used: the coupled selection equations (Starke 1997; Starke and Schanz 1998). This is motivated by pattern formation principles in physical, chemical, and biological systems (Nicolis and Prigogine 1977; Haken 1983a, 1983b). In contrast to traditional optimization methods such as the Hungarian method (Papadimitriou and Steiglitz 1982) for solving the assignment problem, this dynamic approach allows for a solution process that is resistant against breakdowns of robots. The abilities to recover from breakdowns are investigated in detail in this paper. The navigation part of the proposed control method has to prevent collisions with the targets, obstacles, and other robotic units. In the following, a behavioral-based force model (Molnár and Starke 2001) is used, motivated by behavioral patterns of pedestrian crowds. Alternatively, the assignment part can be combined with other path-planning methods for the navigation, as described in Section 3.3.

To demonstrate the applicability of the proposed self-organization approach not only for computer simulations but also for real-world situations, additional experiments were carried out for verification. These experiments were realized with three autonomous mobile robotic units of the type Mark V (Cai et al. 1995; Cai 1997) using vision-based target recognition and distance sensors for collision avoidance. The fault-tolerant behavior is demonstrated in the experiments by a triggered breakdown of one of the robotic units.

2. Description of the Problem

To demonstrate the advantages and capabilities of the presented self-organization approach, we consider the following example of a robot–target assignment problem, which can be regarded as part of an industrial manufacturing process. In a planar (or three-dimensional) working area, mobile robotic units have to be assigned to some targets in order to perform tasks at the targets. Hereby, each target has to be served by one and only one robotic unit, and each robotic unit has to be assigned to one and only one target. The working area with the initial configuration of robots, targets and obstacles is illustrated in Figure 1. It is not important whether the working area is confined or not. The assignment has to be done in such a way that the total working costs of the robotic units at the targets are minimal. Characteristic for this optimization problem is that the decisions of all robots are coupled with each

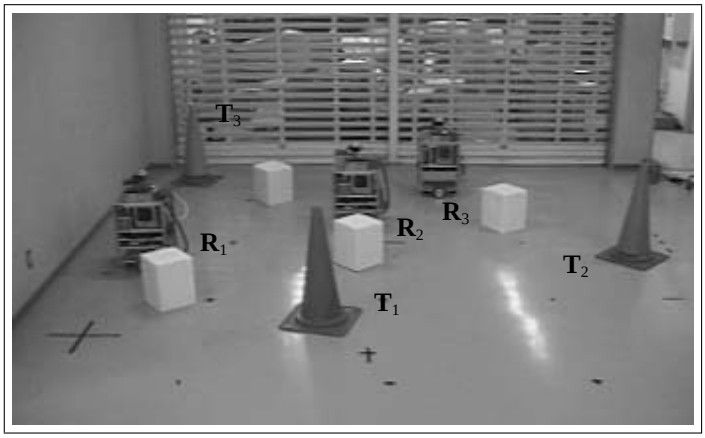

Fig. 1. Picture of the initial configuration of the robotic units $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{R}_3$, targets $\mathbf{T}_1$, $\mathbf{T}_2$, and $\mathbf{T}_3$, and obstacles in the working area.

other due to the constraints that each target has to be served by one and only one robot and each robot has to serve one and only one target which makes it a non-trivial problem. In order to select the most suitable target, each robotic unit has to recognize the direction of each target. The robots are equipped with vision systems to detect the directions to the targets. Although world-coordinate information, which could be a prior given knowledge or obtained for instance by any positioning system similar to the global positioning system (GPS), would simplify the experiment, we want to demonstrate the capability of this method using only local information.

A system of coupled selection equations assigns the robotic units to the targets. Hereby, it can be mathematically guaranteed (Starke 1997; Starke and Schanz 1998) that each target is served by one and only one robotic unit if the number of targets is equal to the number of robots. Once the robotic units have reached their targets they may perform any required task there, using additional equipment such as manipulators or welding tools. As already mentioned, these coupled selection equations are motivated from pattern formation principles in physical, chemical, and biological systems where nature shows us very robust, fault-tolerant, and flexible working behavior. Using these selection equations to control distributed robotic systems allows for transferring these properties of being robust, fault-tolerant, and flexible to the distributed robotic systems, which will be shown in the following sections.

The navigation system of the robotic units is based on the formalism of behavioral force models: propelling force terms steer the robots to their selected target, and repulsive force terms avoid collisions with obstacles or other robotic units. These propelling force terms are defined by the coupled selection equations so that for large times there is one and only one propelling force for each robotic unit. The repulsive forces of the (convex) obstacles have a finite range in order to avoid that, far away from these obstacles, these terms add up to unwanted stable stationary points of the dynamical system, which could trap the robots, and cause the system to fail.

3. Implemented Control Method

The robot–target assignment and the movement of the robotic units in the two- or three-dimensional space is controlled by using the combination of the coupled selection equations and a specific behavioral force model that has been suggested in Molnár and Starke (2001) as a purely theoretical approach disregarding experimental requirements. First, it is necessary to adapt the approach to the real sensor equipment, measuring the distances to the surfaces of the obstacles or other robots instead of the distances to center points of these objects. Secondly, the real movement of the robots has to be synchronized with the applied control method with respect to the robot velocities.

This specific combination of the differential equations for the selection process of targets and the behavioral force model shows self-organizing behavior similar to many phenomena in natural sciences (Nicolis and Prigogine 1977; Haken 1983a, 1983b) ensuring a fault-tolerant and error-resistant working principle. This self-organization is, for example, the pattern formation of rolls or hexagons in fluid dynamics or concentration patterns such as spirals or target patterns in chemical or biological systems. These pattern formation processes are based on the selection of simple basic patterns, so-called modes. This selection process leads to the fact that these patterns are very robust so that they rebuild themselves if parts of them are destroyed. In the treated robot–target assignment

problems, the assignment that is represented in the following section as a permutation matrix corresponds to such a pattern. Due to this correspondence, the robust and error-resistant properties are transferred to the field of robotics.

It has to be pointed out that the coupled selection equations could be combined with other potential field methods (Krogh 1984; Khatib 1986; Arkin 1987; Connolly, Burns, and Weiss 1990; Barraquand, Langlois, and Latombe 1992; Guldner and Utkin 1996) or other path-planning methods (Latombe 1993; Marchese 2002) as well, so that using the specific behavioral force model in the present work is just a first example and can be developed further. Nevertheless, due to the carefully constructed forces, the specific behavioral force model used here has some nice properties compared to other behavioral force models, which can produce traps and are summarized, for example, in Koren and Borenstein (1991) and Arkin (1998). These properties will be explained in more detail in the following section.

3.1. Equation of Motion for the Robotic Units

The equation of motion for the velocities of the robotic units consists of the already mentioned specifically constructed behavioral force model, where the coupled selection equations are an essential part to compute a destination vector for each robot as a result of the robot–target assignment problem. This destination vector is used to obtain a propelling force towards the selected target. It should be mentioned that, due to the unique assignment by the coupled selection equations, the propelling force will finally tend to one and only one of the targets. This guarantees not only that each target is served by one and only one robot and each robot serves one and only one target, but also that the propelling forces produce no traps for the robots. The property that the coupled selection equations produce in the long time run feasible assignments only is proven in Starke (1997). Furthermore, repulsive short-range forces are used to avoid collisions with obstacles or other robotic units. Considering cases with convex obstacles, these short-range forces avoid building traps, which is a common problem in most behavioral force models where small forces far away from the obstacles sum up to these traps (Arkin 1998; Latombe 1993).

This equation of motion (Molnár and Starke 2001) for the velocity $\mathbf{v}_i(t) \in \mathbb{R}^2$(or $\mathbb{R}^3$) of the robotic unit $i \in \{1, ..., n^{\mathrm{r}}\}$ is defined by

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}_i(t) = \frac{1}{\tau}\left[v_i^0\mathbf{e}_i^0(t) - \mathbf{v}_i(t)\right] + \sum_{\substack{i'=1\\ i'\neq i}}^{n^{\mathrm{r}}} \mathbf{f}_{ii'}^{\mathrm{r}}\left[\mathbf{r}_{i'}(t) - \mathbf{r}_i(t)\right] + \sum_{k=1}^{n^{\mathrm{o}}} \mathbf{f}_{ik}^{\mathrm{o}}\left[\mathbf{x}_k - \mathbf{r}_i(t)\right]. \quad (1)$$

Here, the first term represents the propelling force towards the selected target, the second term gives the repulsive force to avoid collisions between the robots, and the last term describes the repulsive force to avoid collisions with the obstacles. The parameter τ is a so-called relaxation constant, which is a measure for the time that the absolute value of the velocity of the robotic unit i needs to tend to the default value v_i^0. The normalized vector $\mathbf{e}_i^0(t) \in \mathbb{R}^2$ (or $\mathbb{R}^3$) represents the direction to the destination, which is dynamically selected by the coupled selection equations, and the vector terms $\mathbf{f}_{ii'}^{\mathrm{r}}$, $\mathbf{f}_{ik}^{\mathrm{o}}$ represent repulsive forces in order to avoid collisions of the robotic unit i at the location $\mathbf{r}_i(t)$ with other robotic units i' or with n^{o} obstacles located at the positions $\mathbf{r}_{i'}(t)$ or $\mathbf{x}_k$, respectively. Targets have to be considered as obstacles as well, unless the robotic units selected them as their destination.

The main point of the presented control method is the determination of the direction to the destination, which leads in the end to the selection of a specific target for the robotic unit i, and is governed by a competitive self-organized assignment procedure that is defined as a dynamical system called the coupled selection equations.

3.1.1. Selection of Targets

The destination direction $\mathbf{e}_i^0(t)$ of the robotic unit i in eq. (1) is defined by a linear combination of specifically normalized difference vectors to the available targets j at the positions $\mathbf{g}_j$

$$\mathbf{e}_i^0(t) = \mathbf{N}_{\gamma\delta}\left[\sum_j \xi_{ij}(t)\mathbf{N}_{\gamma'\delta'}\left(\mathbf{g}_j - \mathbf{r}_i\right)\right] \quad (2)$$

in dependence of the dynamic variable $\xi_{ij}(t)$, which is given by the coupled selection equations defined below.

The function

$$\mathbf{N}_{\gamma\delta}(\mathbf{x}) = \frac{1}{\|\mathbf{x}\| + 1/(\gamma\|\mathbf{x}\| + \delta)} \cdot \mathbf{x}, \quad (3)$$

with $\gamma, \delta > 0$, serves here as a normalization procedure of the vector $\mathbf{x}$ like

$$\mathbf{N}(\mathbf{x}) = \frac{1}{\|\mathbf{x}\|}\mathbf{x}. \quad (4)$$

The reason for using eq. (3) and not eq. (4) is that it avoids the singularity of the factor $1/\|\mathbf{x}\|$ at $\mathbf{x} = \mathbf{0}$. This allows us to use standard numerical solvers for the dynamical system of the proposed control method. Additionally, eq. (3) leads to a deceleration of the robot speed if they approach a target position such as some given point on the floor (which is not an obstacle where one has repulsive forces from the collision avoidance). This is illustrated in Figure 2.

The definition of the destination $\mathbf{e}_i^0(t)$ represents a normalized time-dependent linear combination of all normalized direction vectors $(\mathbf{g}_j - \mathbf{r}_i)$ of robotic unit i to the targets j. In order to guarantee that a target is served by one and only one robotic unit, a competition process among the time-dependent

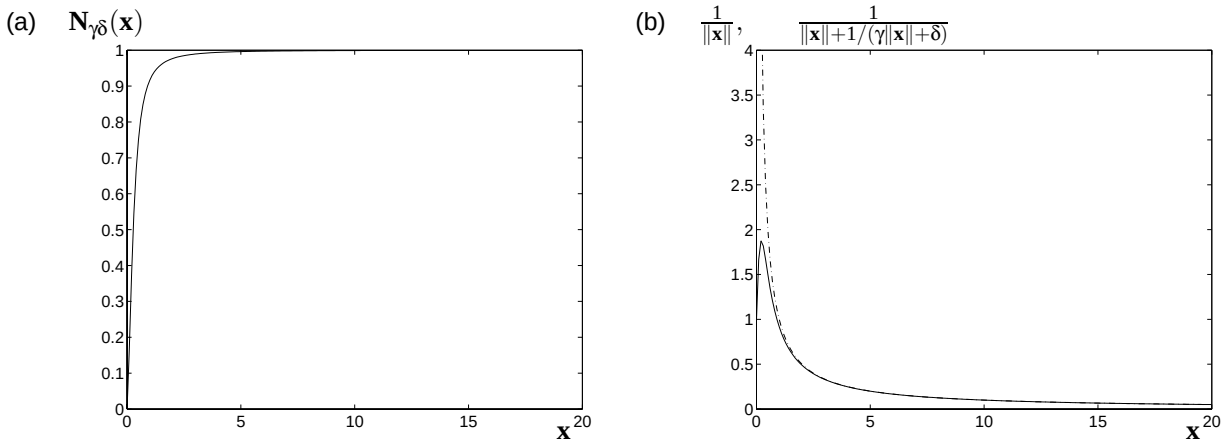

Fig. 2. (a) Plot of the used normalization function (3). To simplify the illustration the special case for $\mathbf{x} \in \mathbb{R}$ (i.e., the one-dimensional case) is shown. For small $\mathbf{x}$, the function $\mathbf{N}_{\gamma\delta}(\mathbf{x})$ goes to zero so that the robots reduce their speed if they approach a target position. The slope of that behavior can be adjusted by γ and δ. (b) The factor $1/\|\mathbf{x}\|$ (dashed line) shows a singularity at $\mathbf{x} = 0$ while the factor $1/[\|\mathbf{x}\| + 1/(\gamma\|\mathbf{x}\| + \delta)]$ is bounded and goes to zero for small $\mathbf{x}$.

coefficients $\xi_{ij}(t)$, according to a certain optimization criterion, is necessary, which leads finally to a unique assignment. The evolution of the destination vector in dependence of the weights $\xi_{ij}(t)$ is visualized in Figure 3.

The competition between the time-dependent coefficients $\xi_{ij}(t)$ is governed by the coupled selection equations (Starke 1997; Starke and Schanz 1998)

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi_{ij} = \kappa\xi_{ij}\left(1 - \xi_{ij}^2 - \beta\sum_{i'\neq i}\xi_{i'j}^2 - \beta\sum_{j'\neq j}\xi_{ij'}^2\right), \quad (5)$$

which represent a self-organized selection process, leading to the selection of one and only one direction vector and hence to the required unique assignment of the robotic unit i to the specific target corresponding to that direction vector. The dynamics of the coupled selection equations (5) is given here with a time-scaling κ, which has to be chosen appropriately to adjust the intrinsic time-scale of eq. (5) to that of eqs. (1).

Non-negative initial values of eq. (5) and $\beta > 1/2$ ensure for square matrices (ξ_{ij}) that the asymptotic dynamics always ends in stable fixed points, which correspond to permutation matrices. This means that there is one and only one non-vanishing element, which is equal to 1 in each row and in each column of the matrix (ξ_{ij}). A proof is given in Starke (1997) and is based on the fact that eq. (5) is a gradient flow of the type

$$\frac{\mathrm{d}}{\mathrm{d}t}\xi_{ij} = -\frac{\partial V}{\partial\xi_{ij}}, \quad (6)$$

where it is shown for the underlying potential function V that all minima that represent stable fixed points of eq. (5) correspond to permutation matrices. See also Starke and Schanz (1998) for further details. Due to this fact, each robotic unit is assigned to one and only one target and each target is considered by one and only one robot. Each permutation matrix, which represents a possible feasible solution, corresponds to a basin of attraction. Only if the initial state is changed such that it crosses a separatrix (i.e., the border between two basins of attraction) does the system converge to another permutation matrix corresponding to the neighboring basin of attraction.

A common optimization criterion would be the minimization of the total working costs corresponding to the specific problem. For reasons of visualization, we use as total working costs the total covered distance caused by the assignment of the robotic units to the targets, which is the sum of the individual initial distances of the robotic units to the targets. Selection equations can be considered as a "winner takes all" strategy, which selects always the largest initial value. Hence we use as initial values of the ξ_{ij} of the coupled selection equations (5) the linear transformed initial Euclidean distances of the robotic units i located at the positions $\mathbf{r}_i(0)$ to the targets j at the locations $\mathbf{g}_j$:

$$\xi_{ij}(0) = 1 - \frac{\|\mathbf{r}_i(0) - \mathbf{g}_j\|}{\max_{i',j'}\left[\|\mathbf{r}_{i'}(0) - \mathbf{g}_{j'}\|\right]}. \quad (7)$$

This leads to a heuristic optimization method to minimize the total path length. The solutions obtained by this method are not always global optimal but compare very well to other heuristics (Starke and Schanz 1998). The convergence to permutation matrices is fast enough for all practical purposes. Even for a large number (>100) of robots because of the good-natured numerical properties of eq. (5) the numerical solution is obtained within seconds.

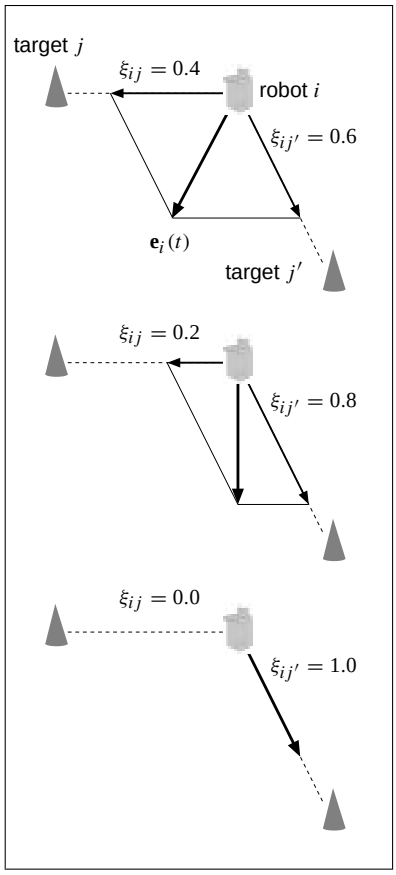

Fig. 3. Visualization of the dynamic selection process. At the beginning (upper picture) the destination vector $\mathbf{e}_i^0(t)$ points towards a linear combination of the weighted difference vectors $\xi_{ij}(t)\mathbf{N}_{\gamma'\delta'}\left(\mathbf{g}_j - \mathbf{r}_i\right)$ to the targets. After a while (lower picture), only one of the targets is selected by the coupled selection equations and the destination vector points to that target only.

3.1.2. Collision Avoidance

As already mentioned, a behavioral force model is implemented in the equation of motion (1) for the velocities of the robotic units in order to avoid collisions among the robotic units themselves as well as between them and the obstacles. Considering robotic unit i, we use the following model. Around each of the other robotic units at the locations $\mathbf{r}_{i'}$ with diameter $d_{i'}^{\mathrm{r}}$ or obstacles at the locations $\mathbf{x}_{i'}$ with diameter $d_{i'}^{\mathrm{o}}$, the repulsive forces

$$\mathbf{f}_{ii'}^{\mathrm{r,o}}(\mathbf{r}) = \begin{cases} -\left[\tan g(\tilde{r}) + g(\tilde{r})\right] \frac{\mathbf{r}}{\|\mathbf{r}\|} & \text{for} \quad 0 < \tilde{r} \leq \sigma^{\mathrm{r,o}} \\ 0 & \text{for} \quad \tilde{r} > \sigma^{\mathrm{r,o}} \end{cases} \tag{8}$$

with

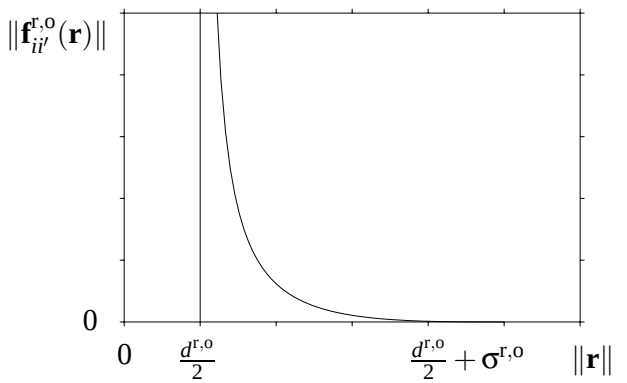

Fig. 4. Plot of the Euclidean norm of the used short-range force $\mathbf{f}_{ii'}^{\mathrm{r,o}}(\mathbf{r})$ defined in eq. (8).

$$\tilde{r} = \|\mathbf{r}\| - d_i^{\mathrm{r}}/2 - d_{i'}^{\mathrm{r,o}}/2 \quad \text{and} \quad g(\tilde{r}) = \frac{\pi}{2}\left(\frac{\tilde{r}}{\sigma^{\mathrm{r,o}}} - 1\right)$$

exert on the robotic unit i. The vector $\mathbf{r}$ is hereby the difference vector of robotic unit i to the other robotic units or obstacles and $\|\cdot\|$ is the Euclidean norm. The usage of repulsive forces such as eq. (8) to solve the problem of collision avoidance is common in the field of robotics. In contrast to many other behavioral force models, the force (8) used here has the great advantage that it is, despite its piecewise definition, continuous and differentiable, and that it is identical to zero if the distance between two objects is larger than a certain threshold $\sigma^{r,o}$. Figure 4 shows a plot of the Euclidean norm of the used short-range force. The finite range of the behavioral interactions avoids adding up small residuals of $\mathbf{f}^{\mathrm{r,o}}$ far away from the obstacles, which could cause unwanted stable points of the dynamical system and trap robots that could cause the system to fail. To be sure that no traps occur in the total system, convex obstacles have to be considered only with large enough surface–surface distances to each other (i.e., distances larger than $2\sigma^{\mathrm{o}} + 2\sigma^{\mathrm{r}} + d^{\mathrm{r}}$ so that the robots can pass in between two obstacles without feeling the collision avoidance forces). This distance condition is in some sense equivalent to avoiding the construction of a non-convex obstacle (also known in robotics as box canyon) by two or more convex ones. In some rare cases, it is possible to imagine that other robots in combination with convex obstacles produce non-convex traps under certain conditions. However, this problem is common to all local methods and can be handled by a combination of either world-knowledge or heuristic strategies to escape from such traps.

3.2. Adaption of the Equation of Motion to the Experiment

This self-organized control method for autonomous mobile robots using self-organizing principles adopted from nature

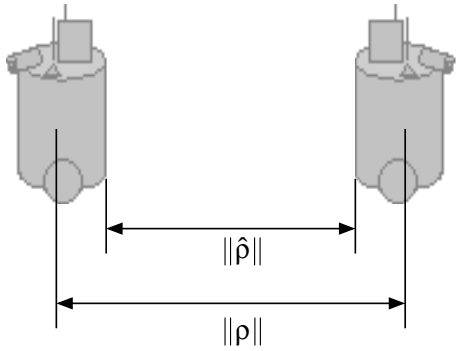

Fig. 5. $\|\varrho_{ik}\|$ denotes the distance between the centers of a robot i and an obstacle (or other robot) k while $\|\hat{\varrho}_{ik}\|$ denotes the measured distance between the surfaces of both objects.

was suggested for the first time in Molnár and Starke (2001). This was done on the basis of a theoretical framework and computer simulations. However, an experimental validation of the suggested control method using real autonomous mobile robots is still lacking. Hence, this paper explicitly addresses this verification. To be able to experimentally validate the suggested approach, it is necessary to adapt the equation of motion (1), due to the fact that the proximity sensors of the robots measure the distance from their own surface to the surface of other robots or the obstacles and not the distance between the central points.

Hence the equation of motion (1), with eq. (8) formulated with distances $\|\varrho\|$ between the central points, has to be transformed to the directly measured variable $\|\hat{\varrho}\|$. Both distances are visualized in Figure 5. With the measured difference vector

$$\hat{\varrho}_{ik} := \frac{\|\hat{\varrho}_{ik}\|}{\|\varrho_{ik}\|}\varrho_{ik} \tag{9}$$

to the surface of the other robots or obstacles and using

$$\hat{\mathbf{f}}_{ik}(\hat{\varrho}_{ik}) := \mathbf{f}_{ik}(\varrho_{ik}) \tag{10}$$

$$= \mathbf{f}_{ik}\left(\frac{\|\varrho_{ik}\|}{\|\hat{\varrho}_{ik}\|}\hat{\varrho}_{ik}\right) \tag{11}$$

we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}_i(t) = \frac{1}{\tau}\left[v_i^0\mathbf{e}_i^0(t) - \mathbf{v}_i(t)\right] + \sum_{\substack{i'=1\\ i'\neq i}}^{n^{\mathrm{r}}}\hat{\mathbf{f}}_{ii'}^{\mathrm{r}}\left[\hat{\varrho}_{ii'}^{\mathrm{r}}(t)\right] + \sum_{k=1}^{n^{\mathrm{o}}}\hat{\mathbf{f}}_{ik}^{\mathrm{o}}\left[\hat{\varrho}_{ik}^{\mathrm{o}}(t)\right]. \tag{12}$$

Here, $\hat{\varrho}_{ii'}^{\mathrm{r}}(t)$ and $\hat{\varrho}_{ik}^{\mathrm{o}}(t)$ are the measured distance vector of the surface of the robot i to the surface of another robot i' or obstacle k, respectively.

In the experiments presented here, the repulsive forces are not distinguished as those for obstacles and robots (i.e., $\hat{\mathbf{f}}_{ii'}^{\mathrm{r,o}} = \hat{\mathbf{f}}_{ii'}$). Furthermore, each of the robotic units is equipped with several distance sensors that recognize the obstacles which can be the other robotic units. Hereby, different sensors of a robotic unit may recognize different obstacles. Using the same behavior of collision avoidance for robots and obstacles, it is not necessary to distinguish the corresponding force fields. From eq. (12) it follows

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}_i(t) = \frac{1}{\tau}\left[v_i^0\mathbf{e}_i^0(t) - \mathbf{v}_i(t)\right] + \sum_{\substack{j=1\\ j\neq i}}^{n^{\mathrm{r}}+n^{\mathrm{o}}}\hat{\mathbf{f}}_{ij}\left[\hat{\varrho}_{ij}(t)\right] \tag{13}$$

with

$$\hat{\mathbf{f}}_{ij}(\hat{\varrho}_{ij}) := \begin{cases} \hat{\mathbf{f}}_{ii'}^{\mathrm{r}}\left[\hat{\varrho}_{ii'}^{\mathrm{r}}(t)\right] & \text{for } j \in \{1, ..., n^{\mathrm{r}}\} \\ \hat{\mathbf{f}}_{ik}^{\mathrm{o}}\left[\hat{\varrho}_{ik}^{\mathrm{o}}(t)\right] & \text{for } j \in \{n^{\mathrm{r}}+1, ..., n^{\mathrm{r}}+n^{\mathrm{o}}\}. \end{cases} \tag{14}$$

If each obstacle is recognized by one and only one of the sensors, the statement $\sum_{k=1}^{n^s}\delta_{kj} = 1$ with the Kronecker symbol δ_{kj} is true for all obstacles j. Using in addition the fact that all $\hat{\mathbf{f}}_{ij}$ are zero if there is no obstacle in the range σ_j of the sensors, the repulsive forces in eq. (13) can be written as

$$\sum_{\substack{j=1\\ j\neq i}}^{n^{\mathrm{r}}+n^{\mathrm{o}}}\hat{\mathbf{f}}_{ij}\left[\hat{\varrho}_{ij}(t)\right] = \sum_{\substack{j=1\\ j\neq i}}^{n^{\mathrm{r}}+n^{\mathrm{o}}}\sum_{k=1}^{n^s}\delta_{kj}\hat{\mathbf{f}}_{ij}\left[\hat{\varrho}_{ij}(t)\right] = \sum_{k=1}^{n^s}\hat{\mathbf{f}}_{ik}\left[\hat{\varrho}_{ik}(t)\right] \tag{15}$$

where

$$\delta_{kj} = \begin{cases} 1 & \text{if } \quad k = j \\ 0 & \text{otherwise,} \end{cases} \tag{16}$$

and n^s is the number of sensors installed in each robot i. The arrangement of the indices k has to be done in such a way that $\delta_{kj} = 1$ if the obstacle j is recognized by sensor $k = j$. In the case of all obstacles being outside the sensing ranges σ_j covered by the sensors of each robotic unit, all $\hat{\mathbf{f}}_{ij}$ are equal to zero and hence no repulsing force results for that robot, and eq. (15) holds as well. One finally obtains

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}_i(t) = \frac{1}{\tau}\left[v_i^0\mathbf{e}_i^0(t) - \mathbf{v}_i(t)\right] + \alpha\sum_{k=1}^{n^s}\hat{\mathbf{f}}_{ik}\left[\hat{\varrho}_{ik}(t)\right] \tag{17}$$

for the equation of motion. In eq. (17) α is an additional parameter to adjust the repulsive forces, which avoid the collisions with obstacles or other robots. These repulsive forces are visualized in Figure 6. Compared to eq. (1) the equation of motion (17) can be calculated directly by using the measured difference vectors $\hat{\varrho}_{ik}$ to the boundaries of the obstacles. If the assumption that each obstacle is recognized by one and only one of the sensors is not fulfilled (e.g. one obstacle is recognized by several sensors), the magnitude of the repulsive force can be adjusted using the parameter α or alternatively a normalization.

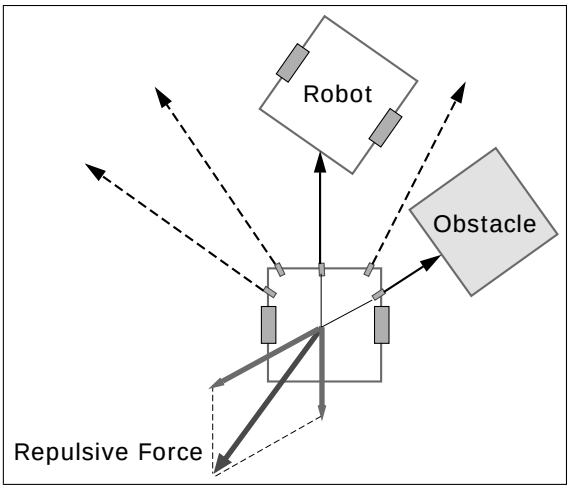

Fig. 6. Linear combination of several repulsive forces to calculate the resulting force.

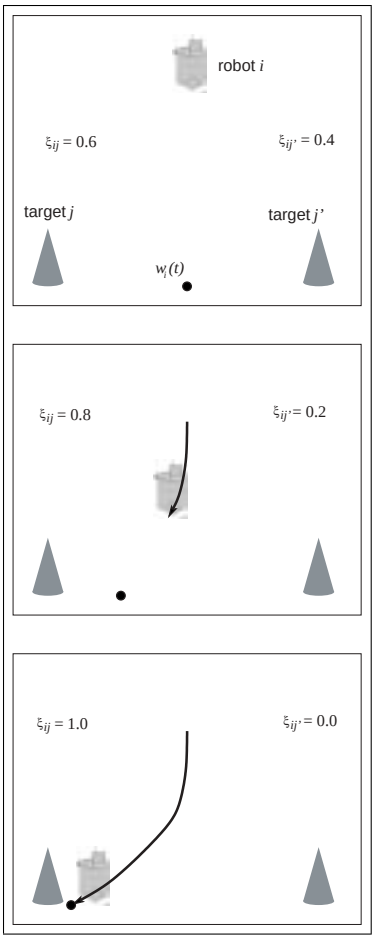

Fig. 7. Visualization of the intermediate target positions $\mathbf{w}_i(t)$. At the beginning (upper picture), the intermediate target position is the weighted center of mass of the targets while the picture series shows the convergence of the intermediate target positions $\mathbf{w}_i(t)$ to the selected final target positions $\mathbf{g}_j$ (lower picture).

3.3. Alternative Navigation of the Robots

It should be noted that in principle any path-planning method can be combined with the dynamical solution of the assignment problem by the coupled selection equation (5). Instead of the time-dependent direction vector (2) one can also use for the alternative path-planning dynamically adapted intermediate targets as a weighted linear combination

$$\mathbf{w}_i(t) \quad = \quad \sum_j \xi_{ij}(t)\mathbf{g}_j \tag{18}$$

of the target positions $\mathbf{g}_j$.

The current position $\mathbf{r}_i$ of each robot i and its corresponding intermediate target position $\mathbf{w}_i(t)$ can be considered as initial and final locations for the navigation. Using this information, any preferred path-planning method can be used to navigate around the obstacles. During the selection process governed by eq. (5), the matrix $[\xi_{ij}(t)]$ converges to a permutation matrix with elements 0 or 1, representing the final assignment. Therefore, each intermediate target $\mathbf{w}_i(t)$ tends to the selected final target positions $\mathbf{g}_j$, which is shown in Figure 7. If there are spare robots the approach (18) of the weighted linear combination of target vectors guides the spare robots to the origin. If it is preferred to stop the spare robots which are not assigned, one can of course use an approach similar to eq. (2).

4. Experimental Setup

4.1. Hardware of the Autonomous Robotic Units

Figure 8 shows the autonomous mobile robots called CEBOT Mark V (Cai 1997) which were used for the experiment. The robots have two differential wheels which are driven independently by separate DC motors. They equip DC batteries and can move without external power supplies. In order to demonstrate the decentralized task-solving properties of the presented algorithm, it is distributed among the robotic units. Each robot solves its part of the selection equations and equation of motion separately using an on-board CPU (Motorola 68040) and communicates the result to the other robotic units by using a wireless LAN with a transmission rate of 2 Mbps (Figure 9). The robotic units use proximity sensors for the detection of the obstacles and CCD cameras to find the targets.

4.2. Detection of Targets

For the implementation of the proposed algorithm, we have to ensure that the robotic units are able to recognize the

Fig. 8. The three autonomous mobile robots CEBOT Mark V used.

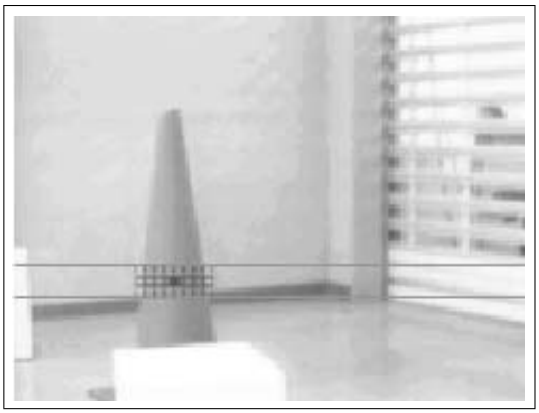

Fig. 10. An image from a CCD camera in the scanning phase.

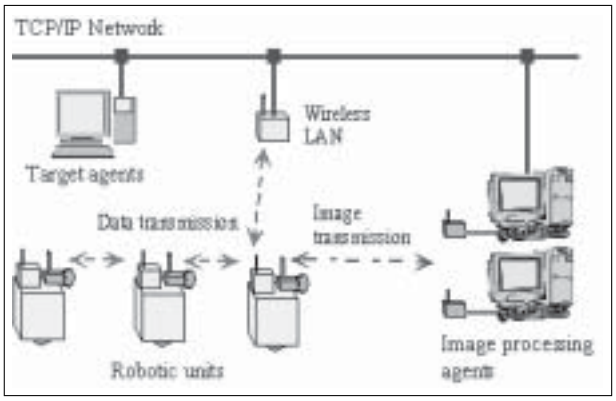

Fig. 9. System configuration for the target detection.

directions to the targets, and furthermore that they are able to distinguish between the targets in order to guarantee the uniqueness of the selection process. We used cones with different colors as targets and color CCD cameras as detection devices.

Figure 9 shows the system configuration for the target detection. Since image processing goes beyond the capabilities of the on-board CPU of the available robots Mark V, the detection process has to make use of an image processing agent, a software agent that resides on a stationary computer. First, a robot sends a request in the form of a video image to one of the image processing agents. After the transmitted image is processed, the agent sends the position of the target in the picture back to the requesting unit. Then, the robotic unit translates this information to the actual direction of the target. Unique ID numbers of the various components are included in the transferred messages, and ensure correct identification of images and targets. The image processing agent selects the video input signal corresponding to the requesting camera ID number and scans the transmitted image using a template matching method, which is implemented in the color tracking vision board TRV-CPW5 (FUJITSU). Clearly, we preferred an on-board image processing, but the available hardware of the robots was not capable of this. Nevertheless, the used setup allows for verification of the self-organized control as well.

In the experiment the targets are identified by their color: blue, green or red. 8×8 RGB pixel templates of the targets have been prepared and stored prior to the experiment. The image processing agent can identify the targets by calculating the correlation values between each template of the targets and the current CCD image. Figure 10 shows such an image in which the target is visible. In order to reduce the calculation time, the scan area is limited to the region between the parallel lines. The checkered area marks where the target is detected, i.e., the correlation of the pixels to the stored red template pattern exceeds a certain threshold.

If a target is detected in the image, the agent extracts the location of the center and the horizontal position of the checkered area and transmits the result to the requesting robotic unit. Using this information, the robotic unit is able to calculate the relative direction to the detected target. If no target is detected in the current image, the robotic unit rotates about 32°, which corresponds to the CCD camera's angle of view, and starts the scanning process over. This procedure repeats until all targets have been detected. An alternative approach, where the recognition process stops after a given time limit, can be implemented as well. Unrecognized or occluded targets are just left out in the assignment process by setting the corresponding ξ_{ij} values of such a target to zero. The coupled selection equations behave then as demonstrated below for the robot–target assignment with a breakdown.

4.3. Detection of Obstacles and Robots

One of the robots (R3) has ultrasonic sensors which can measure distances up to 2000 mm. The other two robotic units

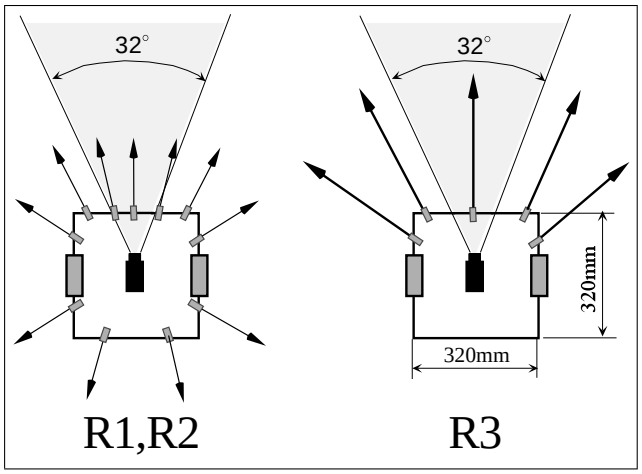

Fig. 11. The arrangement of the sensing devices is shown for the robotic units R1, R2, and R3. The 32° range of the CCD camera is marked as the shaded area and the sensing directions of the infrared and ultrasonic sensors are marked by arrows. The sensing range for R1 and R2 is 500 mm, and 2000 mm for R3.

(R1, R2) use infrared sensors with a scanning range of up to 500 mm. The detailed sensor arrangement is shown in Figure 11. This heterogeneous composition of the robotic system demonstrates that the proposed algorithm does not depend on ideal conditions, such as, for example, identical hardware.

Due to the discrete arrangement of the sensing devices, the robotic units are not able to detect the precise position of obstacles or other robotic units. Rather, each sensing device is calibrated to a particular range of angles, in which it detects objects, providing discrete information about the direction of the objects. In addition, the devices can measure the distances to the obstacles within their scanning range. This information is sufficient for the proposed algorithm to ensure collision avoidance in the experiment.

4.4. Distribution of the Control System

In the distributed environment, each robotic unit i maintains its own set of preferences $(\xi_{i1}, \xi_{i2}, \ldots, \xi_{in})$ using the Euler method

$$\xi_{ij}(t + \Delta t_\xi) = \xi_{ij}(t) + \Delta t_\xi \frac{\mathrm{d}}{\mathrm{d}t}\xi_{ij}(t) \tag{19}$$

for the integration of the coupled selection equations (5), with time-step Δt_ξ. Since the calculation is performed independently on each robotic unit, the updated preferences have to be shared among them. We use target agents, one for each target, to collect the preferences from the robots. The target agents are implemented as independent computational processes, which have a unique port number and IP address.

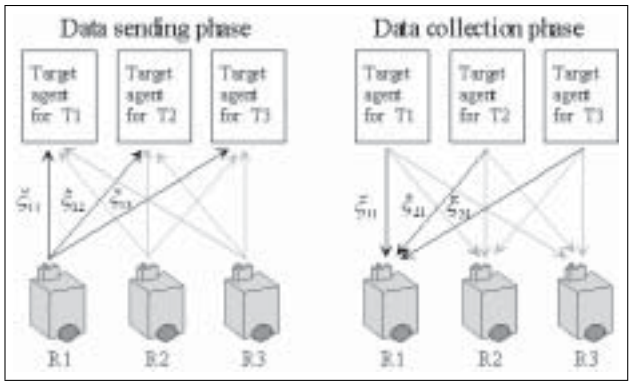

Fig. 12. Scheme of the distributed control.

The robotic units and the targets communicate via a wireless TCP/IP network. When the robotic unit i has calculated the values of $\xi_{ij}(t + \Delta t_\xi)$, it reports these preferences to all target agents. After the collection of all preferences, the target agents report these back to the robotic units (Figure 12). This implementation represents the proposed distributed control of the robots with the advantage of an error resistivity. The overall communication load depends on the number n^r of involved robots. It has to be noted that the suggested selection mechanism is very robust and fault-tolerant. Hence, the information update need not be done so frequently. In an average case, about 10 updates are enough. For each update, $2\,(n^r)^2$ real numbers have to be communicated in total. The low update rate indicates also that the communication need not be very reliable. Moreover, the lack of information or even wrong information at a certain time (caused by a temporary communication breakdown or error) will be compensated in the next step, when the information is provided again.

5. Experimental and Simulation Results

The following results show the self-organized solution of the presented target assignment problem. Experiments have been conducted in the configurations of three robots and three targets, and three robotic units and two targets. The latter case demonstrates the capability of the system to make use of spare units in case of breakdowns of one of the robots, i.e., the error resistance. In addition, the experimental results are compared with computer simulations.

5.1. Dynamic Assignment of the Robotic Units to Targets

The case of three robots and three targets was examined to show the dynamic assignment capability of the proposed algorithm. The initial arrangement of the robotic units, targets, and obstacles are shown in Figure 13. The initial distances

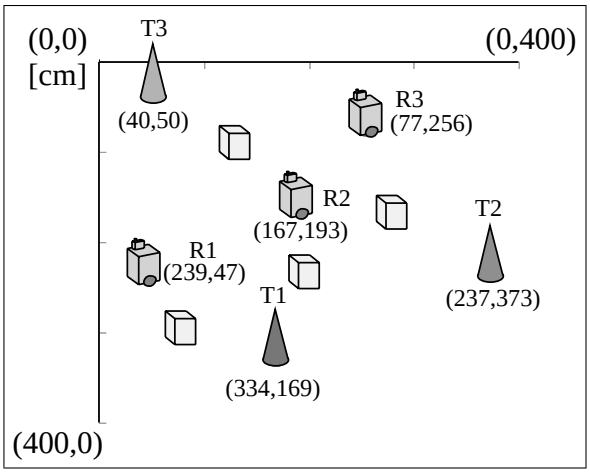

Fig. 13. Initial arrangement of the robots, targets, and obstacles. The targets are represented as cones, the robots as gray boxes, and the obstacles as white boxes. The positions of the obstacles are (303,80), (103,123), (245,190), and (168,292), respectively.

between units to targets are given by

$$(\|\mathbf{r}_i(0) - \mathbf{g}_j\|) = \begin{pmatrix} 155 & 326 & 199 \\ 167 & 193 & 191 \\ 271 & 198 & 209 \end{pmatrix}. \qquad (20)$$

Based on the distance matrix (20), eq. (7) gives the initial values for the coupled selection equations as

$$(\xi_{ij}(0)) = \begin{pmatrix} 0.525 & 0.000 & 0.390 \\ 0.488 & 0.408 & 0.414 \\ 0.169 & 0.393 & 0.359 \end{pmatrix}. \qquad (21)$$

With these initial values, the coupled selection equation (19) is calculated each 250 ms. The initial values $\xi(0)$ given in eq. (21) are distributed among the robots in the same way as the values $\xi(t)$ for all other times t, as described in Section 4.4.

The parameters used for the calculation of the coupled selection equation are shown in Table 1.

The velocity of each robotic unit is calculated by the Euler method from eq. (1) as follows:

$$\mathbf{v}_i(t + \Delta t_\mathbf{v}) = \mathbf{v}_i(t) + \Delta t_\mathbf{v} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}_i(t). \qquad (22)$$

To calculate for visualization purposes the position of the robotic units in world coordinates, the following integration scheme

$$\mathbf{r}_i(t + \Delta t_\mathbf{v}) = \mathbf{r}_i(t) + \Delta t_\mathbf{v}\, \mathbf{v}_i(t) \qquad (23)$$

is used. After six steps $\Delta t_\mathbf{v}$ the robotic units pause in order to perform a target scan process, and update their destination

Table 1. Parameter Setting for the Coupled Selection Equations

Integration step size	Δt_ξ (s)	0.25
Time-scaling	κ	0.45
Coupling strength	β	1.5

Table 2. Parameter Setting for the Motion Planning

Integration step size	$\Delta t_\mathbf{v}$ (s)	0.5
Relaxation time	τ (s)	2.0
Range for collision avoidance	σ^o (mm)	500
Diameter of obstacles	d^o (mm)	320
Default velocity of robots	v_i^0 (mm s^{-1})	200.0
Strength of repulsive forces	α	60.0

vector $\mathbf{e}_i^0(t)$ of eq. (17). The calculation of both the coupled selection equation and the motion planning is suspended during this target scan process. While the computational time for the numerical calculation of eqs. (17) and (5) can be neglected compared with the mechanical parts of most distributed robotic systems, the scanning of the targets is the time-determining process in the experiments. Table 2 shows the parameters used for the motion planning. The time-steps $\Delta t_\mathbf{v}$ for the motion planning were chosen larger than the time-steps Δt_ξ for the coupled selection equations due to the low precision of the mobile robot's steering. Clearly, it is necessary to ensure that the computed times in the two dynamical systems (5) and (17) are identical, and these have to be adjusted to each other depending on the different step size.

Figure 24 shows the experiment in a sequence of pictures. For the purpose of improved visualization, the trajectories shown in Figure 14 have been determined by using the dead-reckoning method (i.e., they were reconstructed by using the information of the number of rotations of the robot's wheels). The evolution of the preference matrix ξ_{ij} is depicted in Figure 15. On the left-hand side, coefficients are plotted of the number of iterations, and on the right-hand side over the elapsed time. By the plateaux in the graphs on the right, one can clearly see when the integration of the coupled selection equations was suspended in favor of the target finding process.

Figure 16 shows the result of a simulation of the proposed control system for the configuration in Figure 13. The trajectories of simulation (Figure 16) and experiment (Figure 14) fall fairly close together, even though the simulation assumes an ideal motion of the robots, and does not consider technical necessities such as the way our robotic units have to spin around in order to find their destination.

The simulation model has been introduced in Molnár and Starke (2001). The model assumes that the robotic units can sense any obstacle in range, and determine their distance and

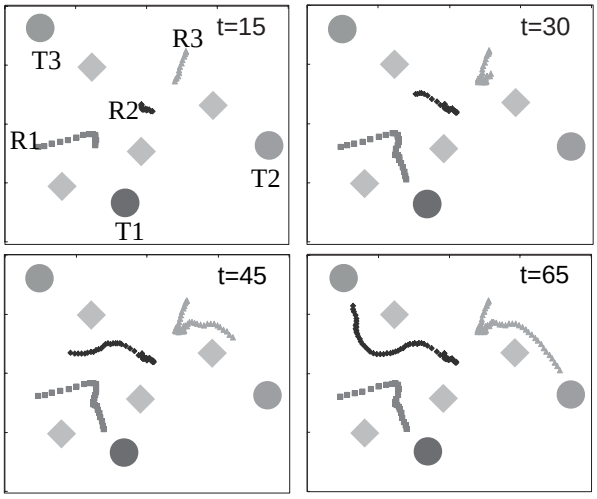

Fig. 14. Four snapshots of experimental trajectories of robotic units are shown for the assignment of three robots to three targets. The robots R2 and R3 are close enough at time $t = 30$ so that the collision avoidance force fields influence the moving direction, which results in a trajectory of R3 above the upper-right obstacle to the selected target T2.

Table 3. Parameter Setting for the Simulation

$\Delta t_{\mathbf{v},\xi}$ (s)	0.02
κ	0.45
β	1.5
τ	1.0
σ^o (m)	1.3
v_i^0 (m s^{-1})	1.2
α	1.0
d^o (m)	1.0

direction. The parameters used in the simulation are described in Table 3. In the simulation, the length-scale and the time-scale had to be adjusted in order to respect different sizes and to compensate for the timeouts caused by the robots through driving curves and sensing the targets. The preference matrix (ξ_{ij}) starts out with the same values as in the experimental setup. As a consequence, the system reaches the same assignment: unit R1 to target T1, R2 goes to T3, and R3 to T2.

5.2. Demonstration of the Error Resistivity by a Triggered Breakdown in the Experiment and Simulation

To demonstrate the error-resistant control, an experiment was carried out where three robotic units have to be assigned to

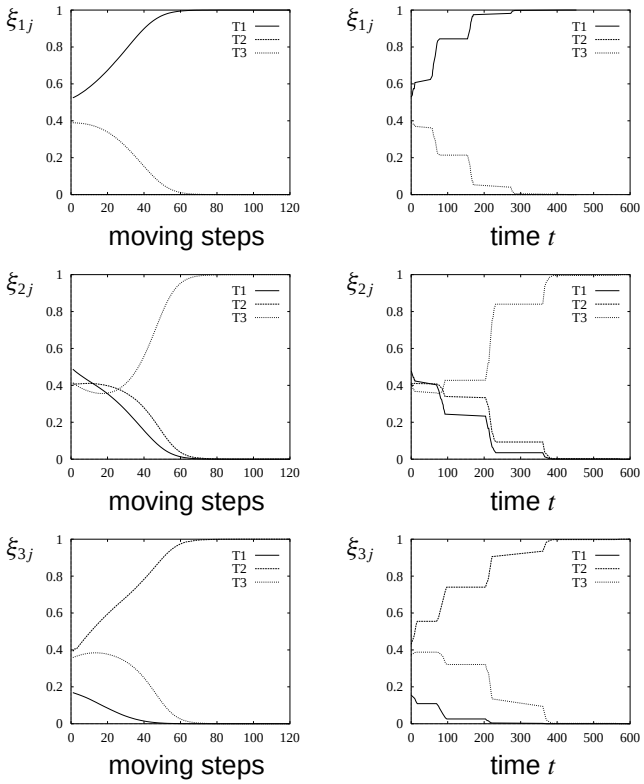

Fig. 15. The evolution of the preference coefficients. On the left-hand side ξ_{ij} is plotted over the number of iterations or moving steps. The graphs on the right show the evolution in real time. The plateaux on the right indicate the scanning phase where the robots do not move but search for the targets.

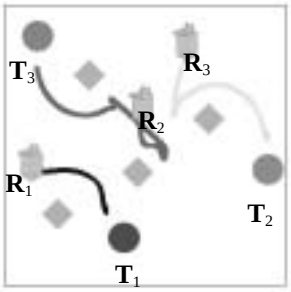

Fig. 16. Simulation of the experimental setup of Figure 13, which corresponds to the experimental result of Figure 14. As in the experiment, the robots R2 and R3 are close enough so that the collision avoidance force fields influence the moving direction; this results in a trajectory of R3 above the upper-right obstacle to the selected target T2.

only two targets. This will leave one unit out. The configuration can be thought of as having a spare unit for when one of the robots breaks down. The third experiment will actually demonstrate this situation.

We used T1 and T3 as the two targets. The initial arrangement is equivalent to Figure 13; only target T2 has been removed. The initial Euclidean distances or working costs of this arrangement are

$$(\|\mathbf{r}_i(0) - \mathbf{g}_j\|) = \begin{pmatrix} 155 & 199 \\ 167 & 191 \\ 271 & 209 \end{pmatrix}. \tag{24}$$

These are obtained from eq. (20) by removing the second column of the matrix (20), which corresponds to target T2, and are used to calculate the initial preference coefficients

$$(\xi_{ij}(0)) = \begin{pmatrix} 0.428 & 0.266 \\ 0.384 & 0.295 \\ 0.0 & 0.229 \end{pmatrix}. \tag{25}$$

To compute the coupled selection equation and the locomotion of the robotic units, the parameters of Tables 1 and 2 have been used.

Figure 25 shows the experimental result. The robot R1 was assigned to target T1, and R3 moved to target T3. The unit R2 started moving towards the targets, but as soon as all its preferences vanished, the robot stopped. However, the unit is still active and ready to move when a target becomes available (e.g. due to the breakdown of another unit). This is demonstrated in Figure 26. All three robotic units start in the same way as in the previous experiment. Then unit R3 was deactivated halfway to its destination, target T3. When the robot stopped, its preferences were set to zero. As a consequence, robotic unit R2 developed a preference for the newly available target T3, and filled in for the broken unit. The trajectories of the experiments with three robots and two targets without and with breakdown corresponding to the video sequences of Figures 25 and 26 are shown in Figure 17. These results clearly demonstrate for the proposed method the error resistivity and ability to recover from breakdowns of robotic units.

5.3. Comparison with Conventional Decision Making and Dependence of the Error Resistivity from the Speed of the Target Selection

As pointed out in the previous section, the proposed control method is capable of making use of spare units in the case of a breakdown of one of the assigned robotic units. The spare robot can take over the assignment from the broken robot as shown in Figure 17. To investigate the advantage of self-organized control compared to conventional decision-making algorithms for the robot–target assignment in more detail, problems of three robots and two targets with one breakdown were simulated. First, a fixed configuration was tested to understand the detailed process of the ability to recover. Secondly, the robot–target assignment of 50 different initial configurations for the positions of robots, targets, and obstacles were simulated to obtain more general statements.

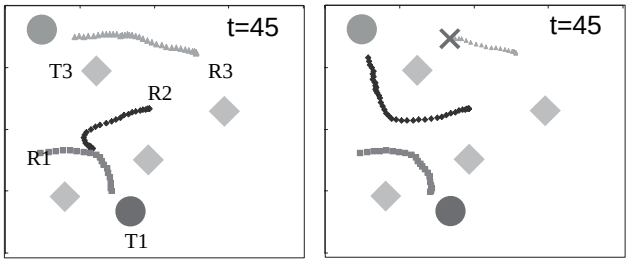

Fig. 17. Experimental trajectories of the robotic units in the case of two targets: left, without breakdown; right, with breakdown of robotic unit R3.

The time-scaling parameter κ of eq. (5) adjusts the intrinsic time-scale of the coupled selection equations to that of the equation of motion for the velocities (17). If it is guaranteed that breakdowns of robotic units or interruptions by tasks with high priority cannot occur, the assignment task can be solved with integer programming techniques such as the Hungarian method (Papadimitriou and Steiglitz 1982) and the robots can head off directly for the targets without any detours. A similar behavior can be achieved by the proposed approach with a large κ. In this case, the assignment is performed very fast and hence seems to be static, and the spare unit has to start moving from the initial point in the case of breakdowns. To minimize the assignment time, the spare robot should move to a location where it can cope with sudden breakdowns immediately. This can be achieved by an appropriate chosen value of the parameter κ, as shown in the following.

Figure 18 shows results of the experiments and the numerical simulations with $\kappa = 1.0$ and 0.1, respectively. The robot R3 stops moving after 15 steps, setting hereby its preferences ξ_{3j} to 0, which indicates the breakdown. Obstacles are removed in this experiment in order to clarify the relation between the preferences and the movement of the robotic units. Without breakdown, the robot R1 moves to the target T1 and robot R3 moves to target T2, and the preferences ξ_{21} and ξ_{22} of the spare robot R2 converge immediately to 0, and this robot stops moving. After the breakdown of the robot R3, the robot R2 starts moving again and reaches finally the target T2. In the case of $\kappa = 1.0$, robot R2 reaches the target after 63 steps, whereas in the case of $\kappa = 0.1$ it needs only 48 steps, since the speed of the selection dynamics is more appropriate (cf., Figure 18).

Figure 19 shows the number of steps of robot R2 that are necessary to reach the target T2 depending on the time-scaling parameter κ, on the one hand, and the occurrence of the

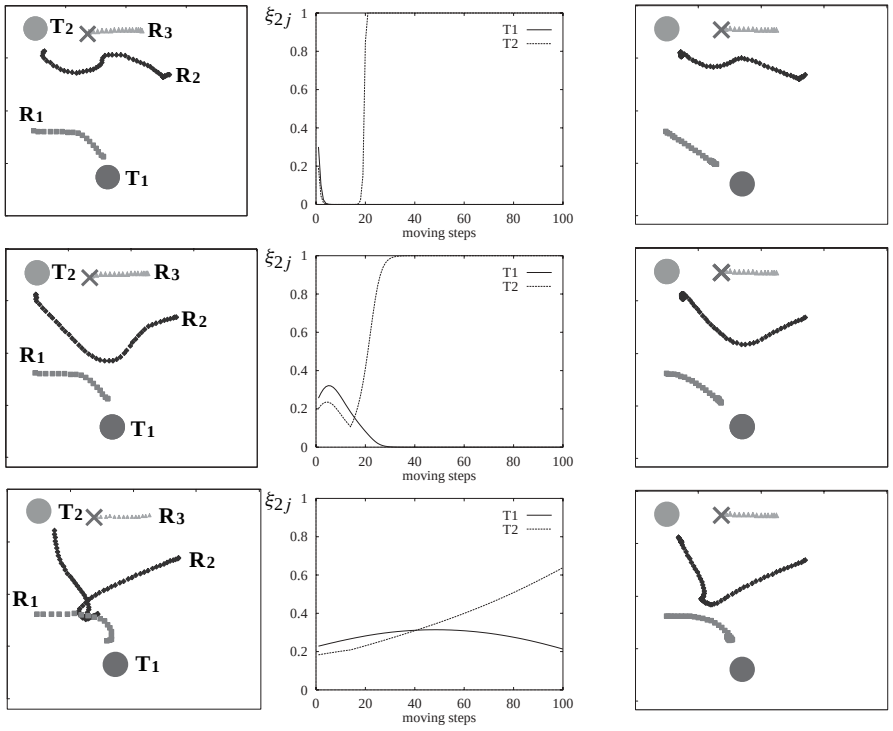

Fig. 18. Experimental trajectories (left), the evolution of the preferences of robot R2 and the simulated trajectories (right). The upper row was obtained for $\kappa = 1.0$, the middle row for $\kappa = 0.1$, the lower row for $\kappa = 0.01$.

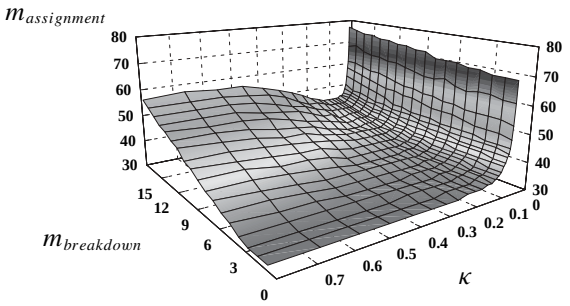

Fig. 19. Number $m_{assignment}$ of steps of robot R2 necessary to reach the target T2 depending on the time-scaling parameter κ and $m_{breakdown}$, the occurrence of the breakdown of robot R3.

breakdown of robot R3, on the other hand. As can be seen, the number of steps is minimal at the value $\kappa = 0.1$. Also very clear are the small values of $m_{assignment}$ for larger κ and small $m_{breakdown}$, because if the breakdown appears very early not much has occurred and the situation is similar to a new problem without breakdown. It has to be pointed out that the proposed control with a very large κ behaves like a control with a traditional decision-making process for selecting the targets, which is decided before the robots move at all.

To show this ability to recover and the error resistivity independent of the above specifically chosen initial configuration of robots and targets, the assignment of three robots to two targets with one breakdown occurring at randomly chosen times were simulated for 50 different initial configurations of robots, targets, and obstacles. The mean of the results of these 50 simulations with randomly chosen initial configurations is shown in Figure 20. These results are similar to those of the specific configuration shown in Figure 19, so that this ability to recover and the error-resistivity is a general property of the proposed self-organized control. The results show clearly that by using a properly chosen value for κ conventional decision-making algorithms, which behave similarly to the presented method for large κ, are clearly outperformed if a breakdown occurs. It is self-evident that this advantage of the self-organization approach only comes into operation if the considered system shows failures or breakdowns, which is very likely especially in complex systems.

5.4. Complex Scenarios with a Large Number of Robots

As pointed out in the introduction, especially complex scenarios where breakdowns are very likely require robust and fault-tolerant control methods, such as the proposed self-organization approach. In the following, the experimental

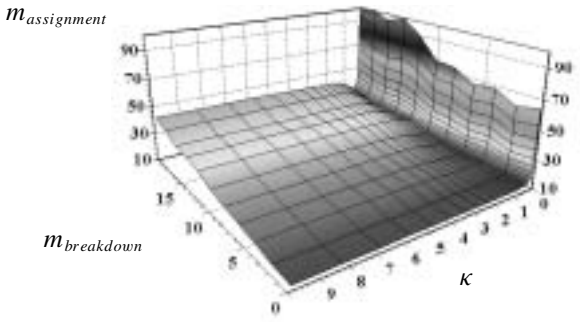

Fig. 20. Mean of the number $m_{assignment}$ of steps necessary for the robots to serve the targets over 50 initial configuration depending on the time-scaling parameter κ and $m_{breakdown}$, the occurrence time-step of the breakdown.

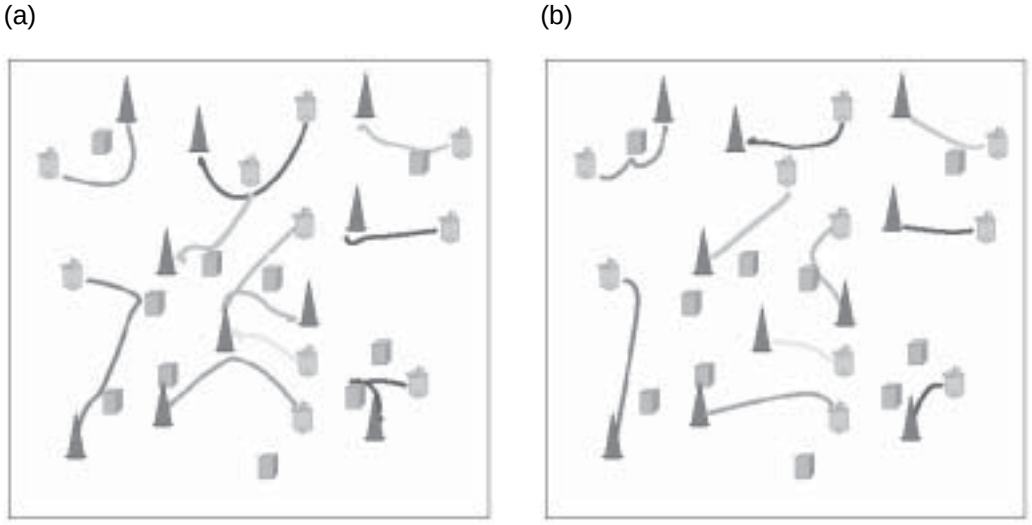

Fig. 21. Simulation result of the assignment of 10 robots to 10 targets. (a) Simulation with $\kappa = 0.4$. Due to the small value of κ, the robots move first to the direction of a kind of center of mass of the targets and then to the targets. (b) Simulation with $\kappa = 5.0$. Due to the large value of κ, the robots move nearly straight to the targets.

results shown above are supplemented with investigations of more complex scenarios. Due to the limited number of available autonomous mobile robots, the complex scenarios with a large number of robots were carried out in simulations.

To demonstrate the solution behavior of the proposed self-organized control mechanism, the assignment of 10 robots to 10 targets (Figure 21) and 35 robots to 30 targets with and without breakdowns for two different values of κ with fast and slow selection of the targets (Figures 22 and 23) were investigated in computer simulations. The results demonstrate the applicability of the presented self-organization approach even to complex scenarios.

6. Conclusions and Outlook

The performed experiments show that the presented dynamic control scheme for mobile autonomous robots using self-organizing principles adopted from physical, chemical, or biological systems works efficiently for real-world robotic experiments. Furthermore, the experiments demonstrate the fault tolerance of the control method with respect to breakdowns of robotic units. The suggested control method makes use of two dynamical systems. The first is a set of coupled selection equations, which solve the underlying assignment problem by a self-organized competition process. This

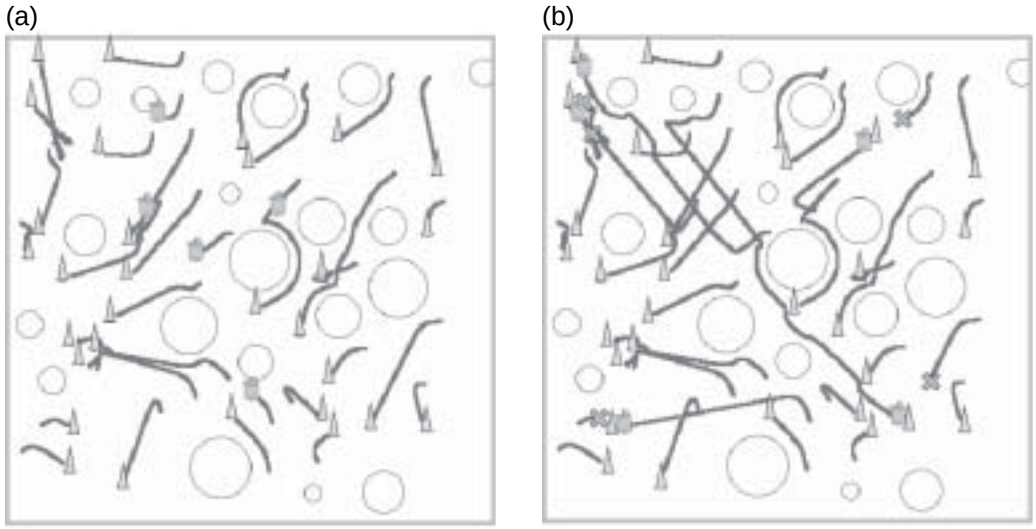

Fig. 22. Simulation result of the assignment of 35 robots to 30 targets with obstacles of different size with $\kappa = 1.0$ (a) without and (b) with breakdown. The five breakdowns are marked with "x". Due to the large value of κ, the selection process of the robots is carried out at the time of the breakdown, so that the ξ values of the remaining robots are equal to zero. This means that they have no more information about the targets, which causes wide detours for the recovering process. For better recovery behavior, smaller κ values have to be chosen, as shown in Figure 23.

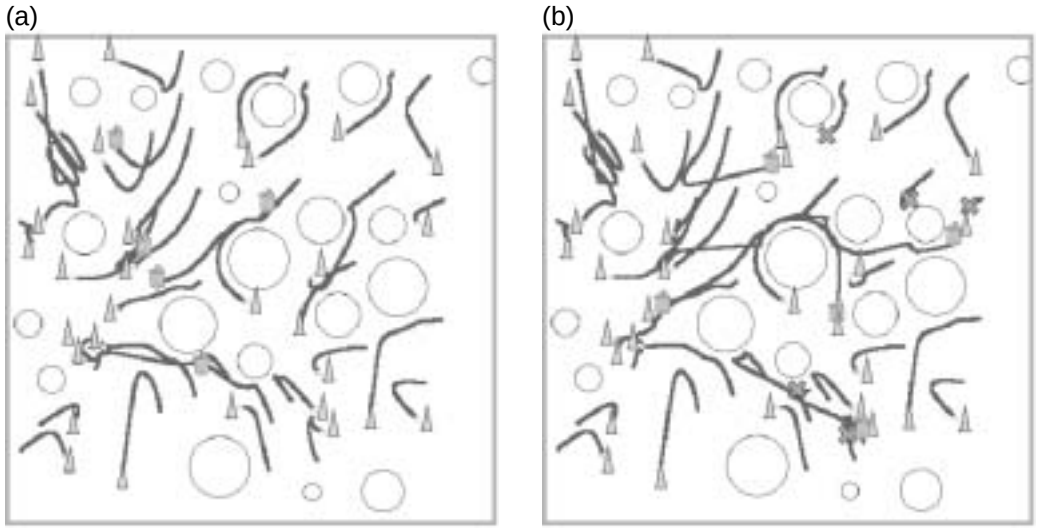

Fig. 23. Simulation result of the assignment of 35 robots to 30 targets with obstacles of different size with $\kappa = 0.4$ (a) without and (b) with breakdowns.

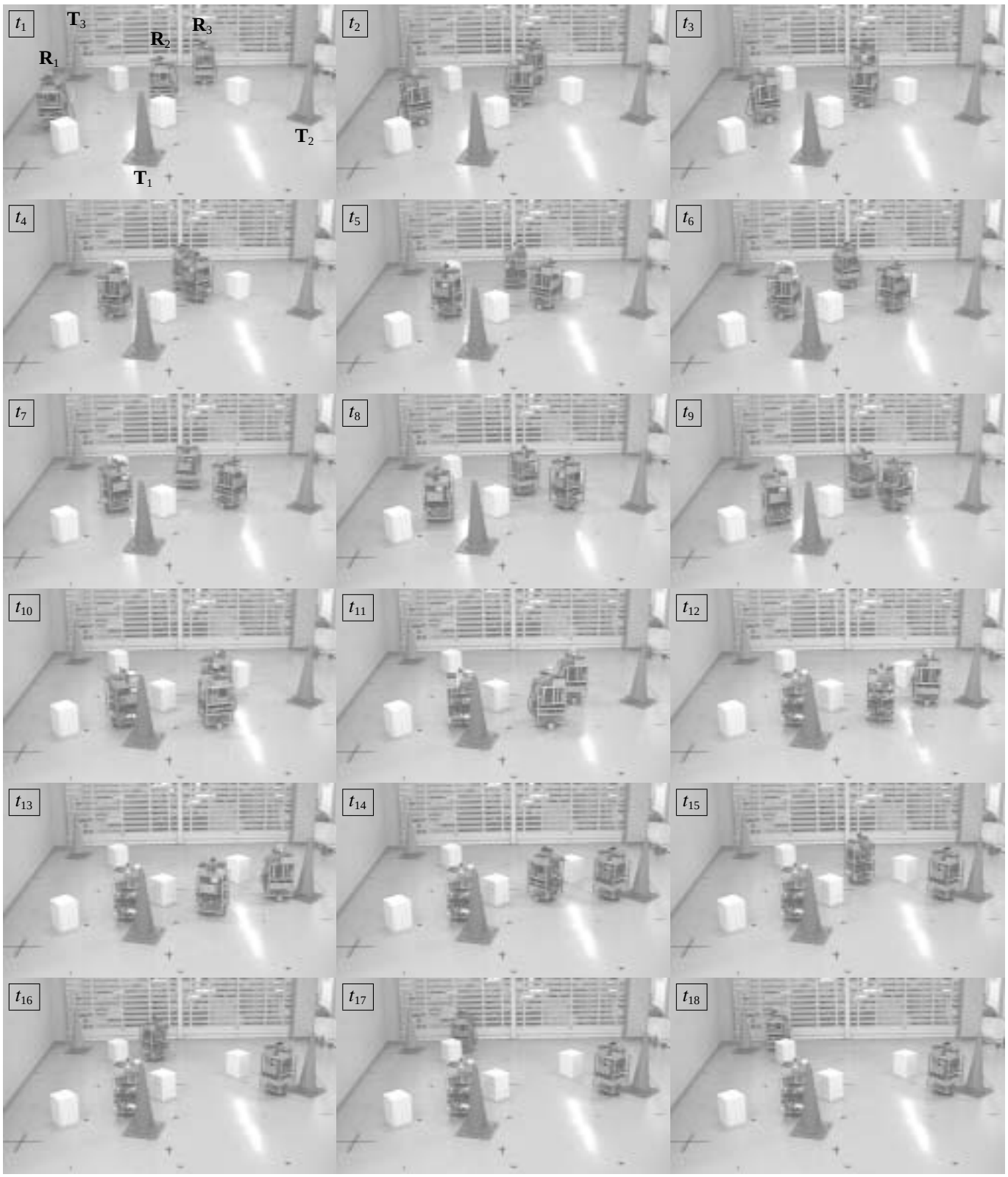

Fig. 24. The assignment of three robots to three targets. At the end unit R1 is assigned to target T1, R2 to T3, and R3 moves to T2.

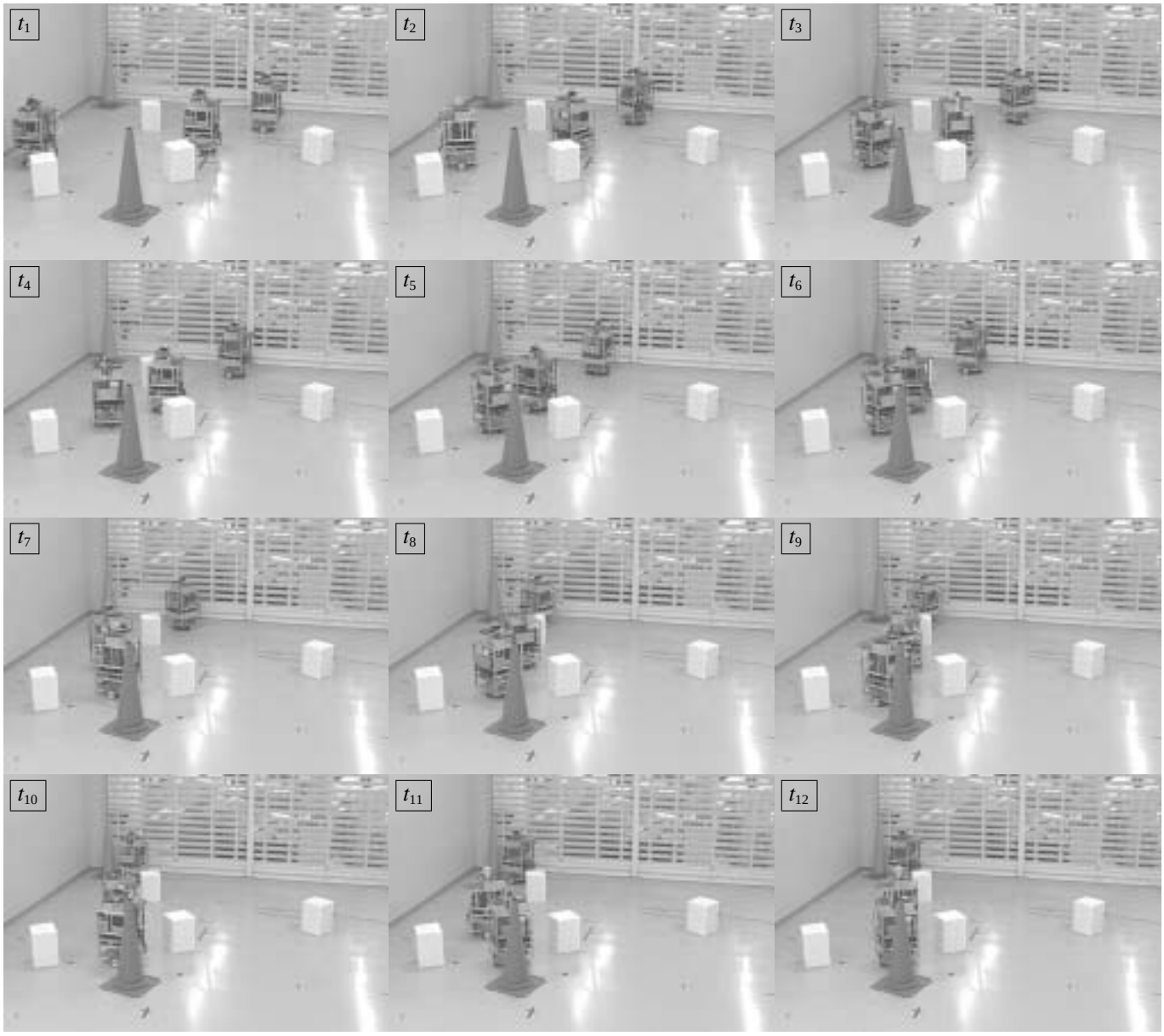

Fig. 25. The assignment of three robots to two targets. One of the units has no assignment, and stops.

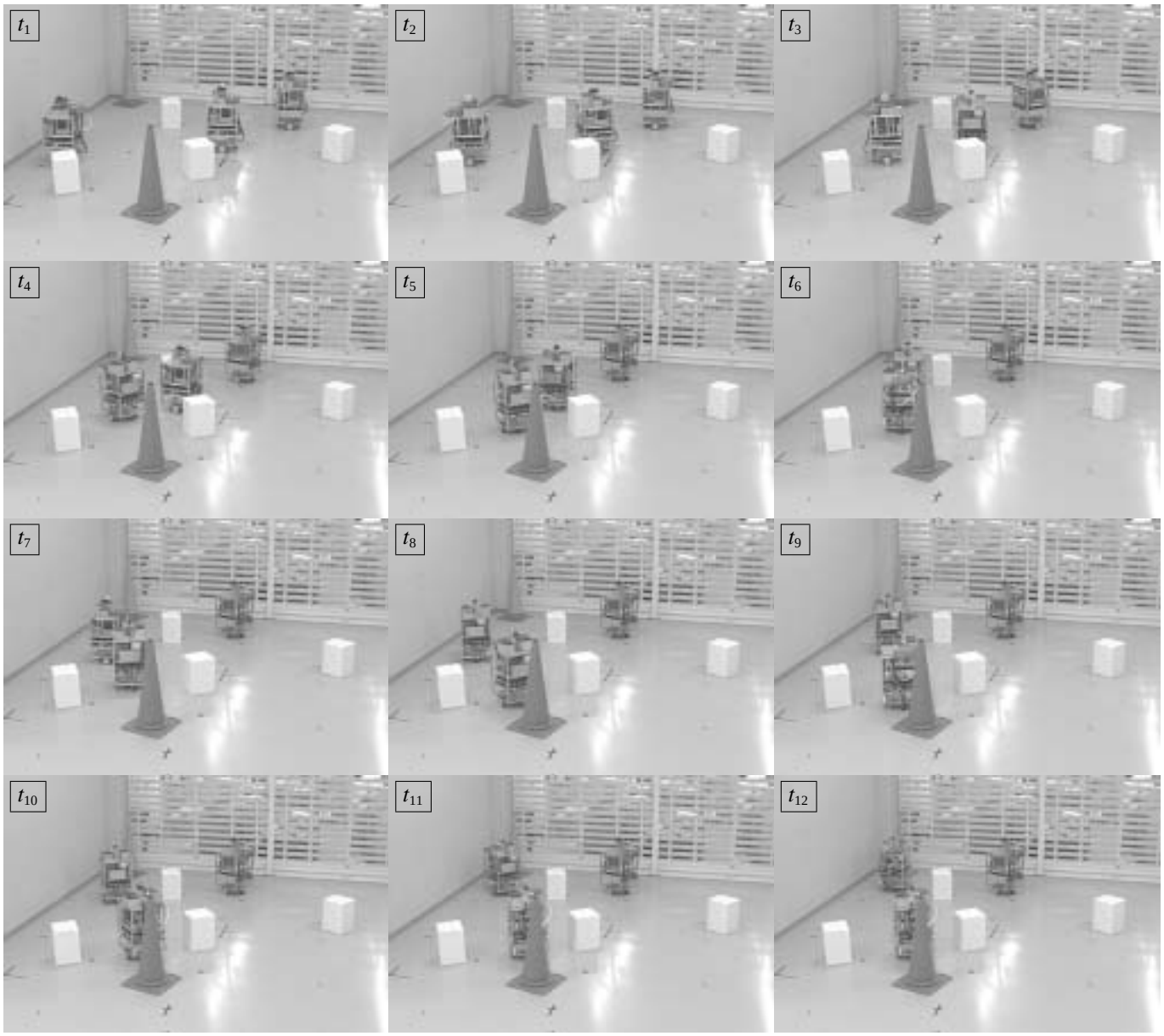

Fig. 26. The assignment of three robots to two targets with a simulated breakdown of unit R3 at time t_5. Robot R2, which has not been assigned to any target before, takes over.

dynamical system avoids spurious states and hence guarantees always feasible solutions of the assignment problem. The second system is a specific behavioral-based force model with smooth finite-range interaction functions. The combination of both dynamical system approaches results in the presented efficient and robust control system. As far as convex obstacles are considered only, the used force model avoids the trapping of robotic units, a common problem that is caused by summation effects of long-range interaction functions. It is clear that this model is not able to overcome the well-known phenomenon of trapping of robotic units if the obstacles are concave. However, instead of using the suggested behavioral-based force model, it is possible to apply other path-finding and collision-avoiding algorithms (e.g., Kavraki and Latombe 1998) in combination with the coupled selection equations, as described in Section 3.3.

Using the presented control system, the robots need not know their absolute location within a world coordinate system, but can navigate by sensing the targets and obstacles in their range (i.e., they use local knowledge only). The target sensing process can be replaced by an indoor positioning system similar to the satellite-based global positioning system (GPS) or by sharing the individual robot's information, which can be broadcast from robot to robot. This shared knowledge could avoid imprecise far distance target recognition.

Even though experimental studies can never consider all possible situations and will be in this sense always incomplete, the presented results demonstrate the applicability of the proposed self-organization approach to real-world problems. This is an important supplement to the sound mathematical foundations of the method. We consider the presented experiments as a first step towards real applications, such as flexible manufacturing systems or intelligent transport systems. The proposed self-organized control system might have potential for further various applications where, in complex systems, robustness and error resistance is of concern, for example in the airplane industry, autonomous maintenance and repair robots in remote locations, such as space stations, chip manufacturing, intelligent transportation systems and just-in-time-delivery control of trucks, and control of swarms of robots for search and rescue, usually in hostile and unknown environments.

Future experiments with a larger number of robotic units are planned to show the capability of the self-organized control for complex systems where breakdowns are very likely due to the large number of robots, and to study small group formation or swarm-like behavior. Our goal is to use robots with improved sensor capabilities, equipped for instance with laser scanners, 360° pan-tilt cameras or even radar, which would spare the robots from rotating in order to detect targets. Furthermore, it is planned to introduce time-dependent assignments, e.g., new targets appear over time or the targets are able to move in the environment. It is also possible that robotic units become available after they have finished their job at a certain target and can take part again in the competition process. Either way, it has to be ensured that the preferences (ξ_{ij}) do not lose their convergence and stability properties by considering these additives, which is a non-trivial task.

Acknowledgments

JS and MS are grateful to T. Fukuda for the invitation to Nagoya University and the hospitality there. The authors would like to thank P. Molnár for fruitful discussions.

References

Arkin, R. C. 1987. Motor schema based navigation for a mobile robot: an approach to programming by behavior. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Raleigh, NC, pp. 264–271.

Arkin, R. 1998. *Behavior-Based Robotics*, MIT Press, Cambridge, MA.

Arkin, R. C. and Bekey, G. A., editors. 1997. *Robot Colonies*, Kluwer Academic, Dordrecht (reprinted from *Autonomous Robots 4* 11).

Asama, H., Matsumoto, A., and Ishida, Y. 1989. Design of an autonomous and distributed robot system: actress. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tsukuba, Japan, pp. 283–290.

Asama, H., Arai, T., Fukuda, T., and Hasegawa, T., editors. 2002. *Distributed Autonomous Robotic System (DARS 5)*, Springer-Verlag, Berlin.

Balch, T. and Arkin, R. C. 1998. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation* 14(6):1–15.

Balch, T. and Hybinette, M. 2000. Social potentials for scalable multirobot formations. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, April 24–28, pp. 73–80.

Barraquand, J., Langlois, B., and Latombe, J.-C. 1992. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics* 22(2): 224–241.

Bojinov, H., Casal, A., and Hogg, T. 2002. Multiagent control of self-reconfigurable robots. *Artificial Intelligence* 142(2):99–120.

Brafman, R. I., Latombe, J.-C., Moses, Y., and Shoham, Y. 1997. Applications of a logic of knowledge to motion planning under uncertainty. *Journal of the ACM* 44(5):633–668.

Brooks, R. A. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* 2(1):14–23.

Burkard, R. 1979. Travelling salesman and assignment problems: a survey. *Annals of Discrete Mathematics* 4:193–215.

Cai, A. 1997. *Behavior Decision and Distributed Sensing on Cellular Robotic System*, Ph.D. Thesis, Nagoya University, in Japanese.

Cai, A., Fukuda, T., Arai, F., Ueyama, T., and Sakai, A. 1995. Hierarchical control architecture for cellular robotic system—simulations and experiments. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Nagoya, Japan, Vol. 1, pp. 1191–1196.

Connolly, C., Burns, J., and Weiss, R. 1990. Path planning using Laplace's equation. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Cincinnati, OH, pp. 2102–2106.

Feddema, J. T., Robinett, R. D., and Driessen, B. J. 2003. Designing stable finite state machine behaviors using phase plane analysis and variable structure control. *Journal of Intelligent Robotic Systems* 36(4):349–370.

Feh, T. 1999. Contingency management in flexible manufacturing systems using modal state logic. *Journal of Manufacturing Systems* 18(5):345–357.

Fukuda, T. and Nakagawa, S. 1988. Approach to the dynamically reconfigurable robotic system. *Journal of Intelligent and Robotic Systems* 1(1):55–72.

Fukuda, T. and Ueyama, T. 1994. *Cellular Robotics and Micro Robotic Systems*, World Scientific Series in Robotics and Automated Systems, Vol. 10, World Scientific, Singapore.

Fukuda, T., Buss, M., Hosokai, H., and Kawauchi, Y. 1991. Cell structured robotic system CEBOT: control, planning and communication methods. *Robotics and Autonomous Systems* 7:239–248.

Fukuda, T., Kawauchi, Y., and Asama, H. 1992. Dynamically reconfigurable robotic systems—optimal knowledge allocation for cellular robotic system (CEBOT). *Journal of Robotics and Mechatronics* 2(6):22–30.

Goldberg, D. and Matarić, M. J. 2002. Design and evaluation of robust behavior-based controllers. *Robot Teams: From Diversity to Polymorphism*, T. Balch and L.E. Parker, editors, A. K. Peters, Wellesley, MA.

Guldner, J. and Utkin, V. I. 1996. Tracking the gradient of artificial potential fields: sliding mode control for mobile robots. *International Journal of Control* 63(3):417–432.

Haken, H. 1983. *Synergetics, An Introduction*, Springer Series in Synergetics, Springer-Verlag, Berlin.

Haken, H. 1983b. *Advanced Synergetics*, Springer Series in Synergetics, Springer-Verlag, Berlin.

Kavraki, L. E. and Latombe, J.-C. 1998. Probabilistic roadmaps for robot path planning. *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, K. Gupta and A. P. del Pobil, editors, Wiley, New York, pp. 33–53.

Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research* 5(1):90–98.

Koren, Y. and Borenstein, J. 1991. Potential field methods and their inherent limitations for mobile robot navigation. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Sacramento, CA, pp. 1398–1404.

Krogh, B. 1984. A generalized potential field approach to obstacle avoidance control. Technical Report. SME-RI Technical Paper MS84-484, Society of Manufacturing Engineers, Dearborn, MI.

Kusiak, A. 1985. Flexible manufacturing systems: a structural approach. *International Journal of Production Research* 23(6):1057–1073.

Jung, B. and Sukhatme, G. S. 2002. Tracking targets using multiple robots: the effect of environment occlusion. *Autonomous Robots* 13(3):191–205.

Latombe, J.-C. 1993. *Robot Motion Planning*, 3rd edition, Kluwer Academic, Dordrecht.

Lee, W. H. and Sanderson, A. C. 2000. Dynamics and distributed control of modular robotic systems. *Proceedings of the International Conference on Industrial Electronics, Control and Instrumentation (IECON)*, Nagoya, Japan, October 22–28, pp. 2479–2484.

Lueth, T., Dillmann, R., Dario, P., and Wörn, H., editors. 1998. *Distributed Autonomous Robotic Systems (DARS 3)*, Springer-Verlag, Berlin.

Marchese, F. M. 2002. A directional diffusion algorithm on cellular automata for robot path planning. *Future Generation Computing Systems* 18(7):983–994.

Matarić, M. J. 1992. Designing emergent behaviors: from local interactions to collective intelligence. *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats 2*, Honolulu, Hawaii, December 7–11, pp. 432–441.

Matarić, M. J. 1995. Designing and understanding adaptive group behavior. *Adaptive Behavior* 4(1):50–81.

Matarić, M. J. 1998. Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior. *Trends in Cognitive Science* 2(3):82–87.

Molnár, P. and Starke, J. 2001. Control of distributed autonomous robotic systems using principles of pattern formation in nature and pedestrian behavior. *IEEE Transactions on Systems, Men and Cybernetics B* 31(3):433–436.

Nicolis, G. and Prigogine, I. 1977. *Self-Organization in Non-Equilibrium Systems*, Wiley, New York.

Papadimitriou, C. and Steiglitz, K. 1982. *Combinatorial Optimization – Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ.

Parker, L. E. 1993. Designing control laws for cooperative agent teams. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, GA, pp. 582–587.

Parker, L. E. 1994. Alliance: an architecture for fault tolerant, cooperative control of heterogeneous mobile robots. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Munich, Germany, pp. 776–783.

Parker, L. E. 1996. On the design of behavior-based multi-

robot teams. *Advanced Robotics* 10(6):547–578.

Parker, L. E. 2002. Distributed algorithms for multirobot observation of multiple moving targets. *Autonomous Robots* 12(3):231–255.

Parker, L. E., Bekey, G., and Barhen, J., editors. 2000. *Distributed Autonomous Robotic Systems (DARS 4)*, Springer-Verlag, Berlin.

Pu, P. and Hughes, J. 1994. Integrating AGV schedules in a scheduling system for a flexible manufacturing environment. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Diego, CA, pp. 3149–3154.

Salido-Tercero, J., Dolan, J., Hampshire, J. B., and Khosla, P. 1997. A modified reactive control framework for cooperative mobile robots. *International Symposium on Sensor Fusion and Decentralized Control in Autonomous Robotic Systems, Proceedings of the SPIE* 3209:90–100.

Schneider-Fontán, M. and Matarić, M. 1998. Territorial multirobot task division. *IEEE Transactions on Robotics and Automation* 14(5):815–822.

Schultz, A. C., Parker, L. E., and Schneider, F. E., editors. 2003. *Multirobot Systems: From Swarms to Intelligent Automata, Proceedings of the International Workshop on Multirobot Systems*, Washington, DC, March 17–19, Vol. II, Kluwer Academic, Dordrecht.

Simeu-Abazi, Z. and Sassine, C. 2001. Maintenance integration in manufacturing systems: from the modeling tool to evaluation. *International Journal of Flexible Manufacturing Systems* 13(3):267–285.

Starke, J. 1997. *Kombinatorische Optimierung auf der Basis gekoppelter Selektionsgleichungen*, Ph.D. Thesis, Universität Stuttgart.

Starke, J. and Schanz, M. 1998. Dynamical system approaches to combinatorial optimization. *Handbook of Combinatorial Optimization*, D-Z. Du and P. Pardalos, editors, Kluwer Academic, Dordrecht, Vol. 2, pp. 471–524.

Vaaria, J. and Ueda, K. 1998a. Biological concept of self-organization for dynamic shop-floor configuration. *Advances in Production Management Systems*, IFIP, pp. 55–66, Chapman and Hall, London.

Vaaria, J. and Ueda, K. 1998b. An emergent modeling method for dynamic scheduling. *Journal of Intelligent Manufacturing* 9:129–140.