Experimental study of self–organized fault–tolerant behavior in robotic systems

R. Lafrenz[1], M. Schulé[1], M. Becht[1], M. Schanz[1], P. Molnár[2], J. Starke[3] and P. Levi[1]

[1]Institute of Parallel and Distributed High–Performance Systems, University of Stuttgart, Breitwiesenstr. 20–22, D–70565 Stuttgart, Germany, {name}@informatik.uni-stuttgart.de
[2]Center for Theoretical Studies of Physical Systems, Clark Atlanta University, 223 James P. Brawley Drive S.W., Atlanta, Georgia 30314, USA, pmolnar@cau.edu
[3]Institute of Applied Mathematics, University of Heidelberg, Im Neuenheimer Feld 294, D–69120 Heidelberg, Germany, starke@iwr.uni-heidelberg.de

1 Introduction

Concepts of self–organization, adapted from physics, chemistry or biology [4], [3], [7] will become more and more important for the implementation of suitable control mechanisms in the field of artificial intelligent systems and especially in the field of distributed autonomous mobile robotic systems. These concepts seem to be very promising to guarantee the required flexibility, robustness and fault–tolerance.

For the control of a multi–robot system consisting of autonomous mobile robotic units a dynamical system approach is used. This novel control method consists mainly of two ingredients. The first ingredient is an algorithm based on pattern formation principles adapted from physics capable to solve an important sub–class of combinatorial optimization problems, namely the so–called assignment problems [8], which occur often in flexible manufacturing systems and other operational areas. Due to the inherent self–organization process, this algorithm ensures fault–tolerant behavior. The second ingredient is a collision avoidance algorithm based on a behavioral force model [6], [10]. The present work is related to the previous ones and, where the experimental verification of the two–index assignment problem [9] and the $\mathcal{NP}$–hard three–index assignment problem [2] proposed in [6], [10] was done.

In the case of the $\mathcal{NP}$–hard three–index assignment problem elements of three sets have to be assigned in such a way that each element of set one is assigned to one and only one element of set two and one and only one element of set three minimizing hereby a given linear objective function. In manufacturing environments the elements could be for instance working stations or other targets (T) and two different types of robotic units (A and B), which have to cooperate at the targets, where the total working costs have to be minimized.

In [2] a scenario with no surplus of robotic units was considered and due to this restriction the system was not able to recover from the breakdown of one robot. In the case of a surplus of robotic units treated in this work, a fault–tolerant behavior of the system is achieved using the self–organization properties of the underlying selection equations. In the real robot experiments fixed targets and two different types of mobile robotic units are used. Because of the surplus of robotic units not all of them will be assigned to targets unless one or several of the assigned robotic units fail. If, for instance due to a breakdown, a robot is not able to reach the assigned target, the coupled selection equations will select one of the spare robotic units which will then move to the target instead of the broken one. This fault–tolerant behavior is immanent to the control mechanism and is a consequence of the implemented self–organization process.

2 Description of the Control Method

As mentioned before, the task to be solved is the assignment of autonomous mobile robotic units of different types A and B to fixed targets such that every target is served by exactly one robotic unit of each type A and B. The proposed mechanism consists of mainly two ideas: The first idea is to use a specific behavioral force model (see, e.g., [1] for an overview) for the navigation which controls the velocities ${}^A\mathbf{v}_i(t), {}^B\mathbf{v}_{i'}(t) \in \mathbb{R}^2$ of the robotic units A_i and $B_{i'}$ respectively [5]. The second idea is to solve the underlying assignment problem using coupled selection equations [8]. The appropriate combination of both ideas avoids collisions with obstacles and other robotic units on the one hand and guarantees a unique, always feasible and flexible (due to the self–organization process) assignment on the other hand. By using ordinary differential equations it is possible to analyze the dynamical behavior of the system and hence to predict some properties of the suggested control mechanism, e.g., the feasibility of the assignment.

2.1 Equation of Motion for the Robotic Units

The equation of motion for the velocity ${}^A\mathbf{v}_i$ of the robotic units A_i with $i \in \{1, \ldots, n_A\}$ is [10]:

$$\begin{aligned} {}^A\dot{\mathbf{v}}_i(t) = & \frac{1}{\tau}\left({}^Av_i^0\,{}^A\mathbf{e}_i^0(t) - {}^A\mathbf{v}_i(t)\right) + \sum_{i' \neq i} {}^A\mathbf{f}_{ii'}({}^A\mathbf{r}_{i'} - {}^A\mathbf{r}_i) \\ & + \sum_{i'} {}^A\mathbf{f}_{ii'}({}^B\mathbf{r}_{i'} - {}^A\mathbf{r}_i) + \sum_k {}^A\mathbf{f}_{ik}(\mathbf{x}_k - {}^A\mathbf{r}_i) \end{aligned} \quad (1)$$

Hereby the three summations have to be taken over the number n_A of robotic units of type A, the number n_B of robotic units of type B and the number n_o of obstacles which may be present in the working space of the robotic units. The equation of motion for the velocity ${}^B\mathbf{v}_i$ of the robotic units B_i with $i \in \{1, \ldots, n_B\}$ is defined analogue to (1). The parameter τ, the so–called relaxation time, is a measure how fast the actual velocity ${}^B\mathbf{v}_i(t)$ tends towards the intended velocity ${}^Bv_i^0\,{}^B\mathbf{e}_i^0(t)$. The parameters ${}^Av^0, {}^Bv^0 \in \mathbb{R}$ are the appropriate working speeds of the robotic units. The normalized vectors ${}^A\mathbf{e}_i^0(t), {}^B\mathbf{e}_i^0(t) \in \mathbb{R}^2$ represent the directions which are assigned to the corresponding robotic units by the coupled selection equations described below. ${}^{A}\mathbf{r}_i$, ${}^{B}\mathbf{r}_i$ and $\mathbf{x}_i$ are the positions of the robotic units A, B and the obstacles respectively, whereas the repulsive forces $\mathbf{f}$ of a robotic unit are caused by approaching an obstacle or another robotic unit.

Collision Avoidance. To ensure the collision avoidance required for instance in manufacturing environments, a behavioral force model is used. Therefore the short range force fields ${}^{A}\mathbf{f}_{ii'}(\mathbf{r}), {}^{B}\mathbf{f}_{ii'}(\mathbf{r}) \in \mathbb{R}^2$ are introduced in the equation of motion (1) for the velocities of the robotic units. The repulsive force acting on a specific robotic unit A_i or B_i with diameter d_i^{R} caused by adjacent robotic units or obstacles is defined by:

$$ {}^{A,B}\mathbf{f}_{ii'}(\mathbf{r}) = \begin{cases} (\tan g(\tilde{r}) - g(\tilde{r}))\, \frac{\mathbf{r}}{\|\mathbf{r}\|} & \text{for } 0 < \tilde{r} \leq \sigma \\ 0 & \text{for } \tilde{r} > \sigma \end{cases} \tag{2} $$

with $\tilde{r} = \|\mathbf{r}\| - \frac{d_i^{\mathrm{R}}}{2} - \frac{d_{i'}^{\mathrm{R},O}}{2}$ describing the distance between the robotic units and $g(\tilde{r}) = \frac{\pi}{2}\left(\frac{\tilde{r}}{\sigma} - 1\right)$. Here $\mathbf{r}$ is the difference vector between centers of the robotic unit A_i or B_i and adjacent robotic units or obstacles, $d_{i'}^{\mathrm{R}}$ and $d_{i'}^{\mathrm{O}}$ denote the diameters of these objects and $\|\cdot\|$ is the EUCLIDean norm. The used force (2) is, though piecewise defined, continuous and differentiable and is identical to zero if the distance between two objects is larger than the appropriate chosen range parameter σ. This property avoids unwanted stable points of the dynamical system (1) which could trap the robotic units and cause the system to fail.

Dynamic Selection of Targets with Underlying Three-Index Assignments. In the equation of motion (1) the destination vectors ${}^{A}\mathbf{e}_i^0$, ${}^{B}\mathbf{e}_j^0$ determine the intended moving direction for the robotic units of type A and B. They are defined through a kind of linear combination of the difference vectors of the robotic unit A_i to all targets

$$ {}^{A}\mathbf{e}_i^0 = \mathbf{\Phi}_{\gamma\delta}\left(\sum_k \left(\sum_j \xi_{ijk}\right) \mathbf{\Phi}_{\gamma'\delta'}\left(\mathbf{g}_k - {}^{A}\mathbf{r}_i\right)\right) \tag{3} $$

and of the robotic unit B_j to all targets

$$ {}^{B}\mathbf{e}_j^0 = \mathbf{\Phi}_{\gamma\delta}\left(\sum_k \left(\sum_i \xi_{ijk}\right) \mathbf{\Phi}_{\gamma'\delta'}\left(\mathbf{g}_k - {}^{B}\mathbf{r}_j\right)\right) . \tag{4} $$

The function $\mathbf{\Phi}_{\gamma\delta}(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|}$ normalizes the vector $\mathbf{x}$. The singularity $\mathbf{x} = \mathbf{0}$ cannot be reached due to stopping conditions of the robotic units at the targets. The coefficients $\left(\sum_j \xi_{ijk}\right)$ and $(\sum_i \xi_{ijk})$ dynamically select the targets for each of the robotic units. To ensure that each robotic unit selects one and only one target and every target is served by two robotic units, exactly one of each type A and B, the following coupled selection equations [8] are used:

$$ \dot{\xi}_{ijk} = \kappa \xi_{ijk}\left(1 + (3\beta - 1)\xi_{ijk}^2 - \beta\left(\sum_{i',j'} \xi_{i'j'k}^2 + \sum_{i',k'} \xi_{i'jk'}^2 + \sum_{j',k'} \xi_{ij'k'}^2\right)\right). \tag{5} $$

Non–negative initial values and $\beta > 1/3$ ensure for cubic tensors (ξ_{ijk}) that the system will always asymptotically reach a stable solution with one and only one non–vanishing element in each plane of the cubic tensor (ξ_{ijk}) which is equal to 1. That means in each plane which is defined by keeping one of the three indices fixed exactly one element is 1 while all other elements are equal to zero. A proof is given in [8]. This ensures always feasible assignments of robotic units to targets. The dynamical system (5) is a heuristic method to solve the $\mathcal{NP}$–hard three–index assignment problem. The solutions obtained by this method compare well with other approaches (see [11]). The parameter κ is needed to adjust the time–scales between the processes of navigation (1) and assignment (5). For the initial values of ξ_{ijk} the sum of the EUCLIDean distances of the initial positions $\mathbf{r}(0)$ of the robotic units of each type to the targets at the locations $\mathbf{g}_k$ is used:

$$\xi_{ijk}(0) = \alpha \left(1 - d^{AB}_{ijk} / \max_{i',j',k'} \left(d^{AB}_{i'j'k'} \right) \right) \tag{6}$$

with $\alpha > 0$ and $d^{AB}_{ijk} = \|{}^{A}\mathbf{r}_i(0) - \mathbf{g}_k\| + \|{}^{B}\mathbf{r}_j(0) - \mathbf{g}_k\|$. During the time evolution of the dynamical system (5) a competition and selection process takes place (see Figures 3 and 4). Due to the properties of the coupled selection equations that combination of the variables ξ_{ijk} wins the process, which ensures that the robotic units are assigned to the closest targets with respect to feasible solutions. This means, that one and only one element in each plane of the ξ–tensor reaches asymptotically the value 1 whereas all others decay to zero, which corresponds to the unique solution of the assignment problem.

3 Experimental Setup

To verify the self–organized control method the experimental setup shown in Fig. 1 is used. The used field was originally designed for RoboCup competitions[1]. The field is surrounded by white walls and its size is approximately 5 m × 10 m. Colored cylinders represent the targets T_i and black cylinders the obstacles both with diameter $d^{\mathrm{O}} = 30\ cm$.

3.1 Robotic Hardware

In the experiment four robotic units of the type Nomadic Super Scout are used and hence they have all the same hardware. These basically cylindrical robotic units have diameters of $d^{\mathrm{R}} = 42\ cm$ and a two wheel differential drive, which allows them to move with a maximum speed of $1\ \frac{\mathrm{m}}{\mathrm{s}}$ and an acceleration of $2\ \frac{\mathrm{m}}{\mathrm{s}^2}$. The basic sensors are wheel encoders and 16 ultrasonic transducers arranged around the outside perimeter of the robotic unit. The control of the motors and the processing of the sensory signals is performed by a Motorola 68HC11–based micro–controller board. To run the application software, like the distributed

[1] www.robocup.org

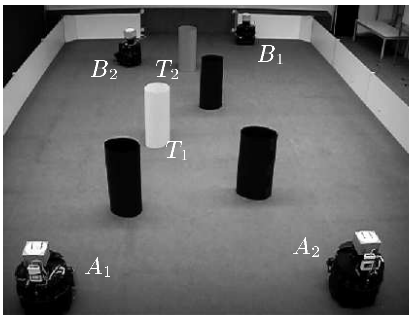

Fig. 1. Initial positions of the robotic units in the test area in the case with no surplus of robotic units.

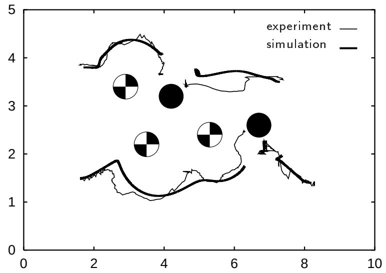

Fig. 2. Comparison between simulation and experiment: trajectories of the robotic units in the case with no surplus of robotic units.

control or the object recognition, each robotic unit is equipped with a Pentium 233 MHz board running LINUX. In addition the robotic units are equipped with wireless Ethernet.

3.2 Recognition of Targets, Obstacles, and other Robotic Units

For improved object detection we use a SICK[2] Laser Range Finder (laser scanner) which is mounted on the top of each robotic unit. This device yields distances for the front 180° with an angular resolution of up to 0.5°. With this sensor, all objects of appropriate height are detected and the relative position can easily be determined. These relative positions are the only informations which are necessary for the robotic units to calculate their velocity vectors, i.e., no world coordinate system is required for the suggested control method. We use the laser range finders to recognize the targets, obstacles, and other robotic units. In the current state of implementation we use a priori known target positions in order to distinguish between targets and other objects. In the experimental setup the obstacle avoidance does not distinguish between obstacles and robotic units. Objects within a certain radius of the expected position of a target are considered to be that target. In future experiments, we will use a calibrated color CCD–camera to determine the relative target position. In this case, no a priori knowledge is required.

3.3 Communication and Distributed Control

To demonstrate the distributed control the four on–board PCs of the mobile robotic units and two other computers which simulate the processing capabilities of the targets are used. Therefore communication between the robotic units

[2] Thanks to the SICK AG (www.sickoptic.com) for lending us some Laser Range Finders.

and computers is essential for the application of the presented control method. For the communication we use MICO, a freeware CORBA implementation. Each robotic unit calculates its own velocity vector based on the local sensor information and the current ξ–tensor. To stay flexible on the one hand and to reduce communication on the other hand, we have distributed the calculation of the ξ–dynamics only among the robotic units of type A. Each of these robotic units is responsible for only one plane of the ξ–tensor and sends this plane to both targets. After receiving the planes of both robotic units of type A a target calculates the six different sums occurring in (5) and sends them to each of the robotic units of type A. These sums are necessary to integrate eq. (5) numerically using a simple Euler–forward integration scheme and hence to update the plane of the ξ–tensor corresponding to the calculating robotic unit. This plane is sufficient for the calculation of their own velocity vector according to eq. (1). Now this plane is sent to each robotic unit of type B. To calculate its own velocity vector a robotic unit of type B needs information provided by both robotic units of type A. This procedure is repeated until the task is solved, i.e., all robotic units have reached their targets. This communication scheme implicitly synchronizes the system.

4 Results

In the previous work [2] the suggested control method based on the coupled selection equations (5) was verified by computer simulations and in a real robotic experiment (see Fig. 2). The task was the assignment and the movement of different types of robotic units to targets. This was done in such a way that each target was served by exactly one robotic unit of type A and one of type B. Furthermore the robotic units had to avoid collisions with randomly placed obstacles including other robotic units and also targets.
To demonstrate the fault–tolerant behavior a surplus of robotic units is used in the present work so that the system is able to recover from breakdowns of robotic units. Therefore we use two targets T_1 and T_2, two robotic units of type A and three robotic units of type B. Due to the surplus of robotic units of type B, the system is capable to recover from a breakdown of a robotic unit of that type. The time evolutions of the elements ξ_{ijk} of the ξ–tensor demonstrating the competition process of the different assignments are shown in Figures 3 and 4. Figure 3 corresponds to the case without breakdown of a robotic unit of type B, whereas the robotic unit B_1 has a breakdown after 10 s in the case of Fig. 4. Hereby, the spare robotic unit B_3 takes over the task which is indicated by the different winning elements of the ξ–dynamics. The corresponding trajectories of the robotic units in the working space are shown in Figures 5 and 6. In this simulation the initial values of the ξ–tensor corresponding to the robotic unit B_3 were not given by (6). Instead they are chosen very small so that they can definitely not win the competition process if all robotic units work well. That means that the robotic unit B_3 is indeed a substitute as one can easily read off from the Figures 5 and 6.

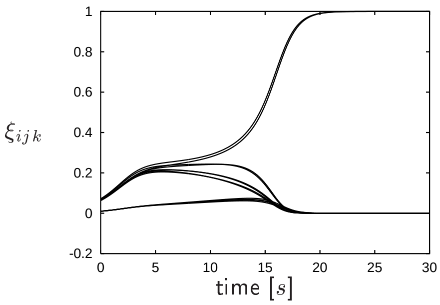

Fig. 3. Time evolution of the ξ_{ijk} in the case without breakdown. Winning elements: ξ_{121}, ξ_{212}.

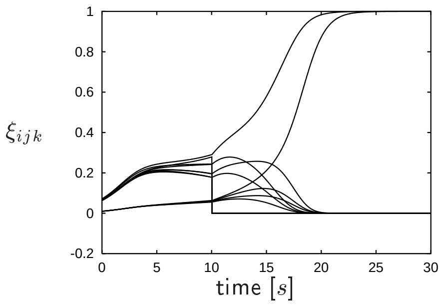

Fig. 4. Time evolution of the ξ_{ijk} in the case with breakdown after 10 s. Winning elements: ξ_{121}, ξ_{232}.

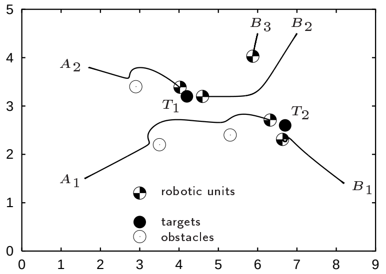

Fig. 5. Trajectories of the robotic units in the working area in the case of no breakdown.

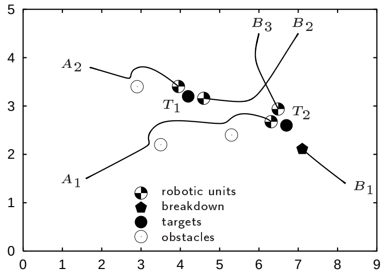

Fig. 6. Trajectories of the robotic units in the working area in the case of a breakdown of robotic unit B_1 after 10 s.

In Figures 7 and 8 the scenario was nearly identically but now the initial values of the ξ–tensor corresponding to the robotic unit B_3 were calculated by (6). As one can see the situation is, as expected, different. If no breakdown of robotic unit B_1 occurs (see Fig. 7) the values of the ξ–tensor corresponding to robotic unit B_2 lose the competition process because B_3 is initially closer to target T_1. In the case of a breakdown of B_1 the formerly spare robotic unit B_2 can take over (see Fig. 8).

From Figures 5 – 8 one can see, that finally each target is served by exactly one robotic unit of each type as required for the cooperative task the robotic units have to perform. During the motion of the robotic units collision avoidance was successfully done by the behavior–based force model of eq. (1) with the force terms defined in (2). One can clearly see the effect of the repulsive forces of the obstacles. As already mentioned in the previous work [2] it was necessary to use here a smaller value of the parameter κ of eq. 5 resulting in a longer time for the competition and selection process. To achieve a fault–tolerant behavior the selection process is completed after approximately 20 s (see Figures 3 and 4) with a total duration of the assignment of approximately 45 s.

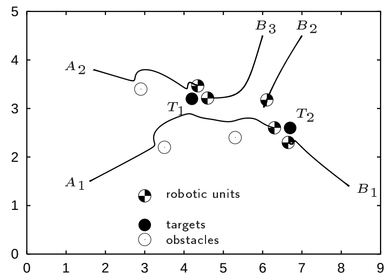

Fig. 7. Trajectories of the robotic units in the working area in the case of no breakdown.

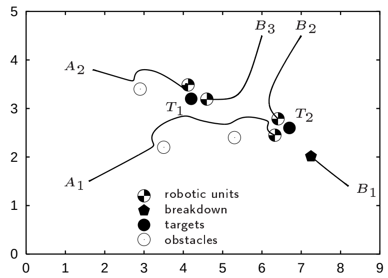

Fig. 8. Trajectories of the robotic units in the working area in the case of a breakdown of robotic unit B_1 after 10 s.

5 Conclusions and Outlook

The results show the verification of a fault–tolerant control method for a complex robot to target assignment scenario.
In future work parameter studies are planned to investigate the fault–tolerant behavior systematically. Furthermore specific situations which could appear in RoboCup scenarios will be considered. One may think of the selection between several offense and defense patterns of the robotic units realizing group behavior.

References

1. R. Arkin. *Behavior-Based Robotics*. The MIT Press, Cambridge, London, 1998.
2. M. Becht, T. Buchheim, P. Burger, G. Hetzel, G. Kindermann, R. Lafrenz, P. Levi, N. Oswald, M. Schanz, M. Schulé, P. Molnar, and J. Starke. Three–index assignment of robots to targets: An experimental verification. In *Proc. of Int. Conf. on Intelligent Autonomous Systems (IAS-6)*, 2000.
3. H. Haken. *Advanced Synergetics*. Springer Series in Synergetics. Springer-Verlag, Heidelberg, Berlin, New York, 1983.
4. H. Haken. *Synergetics, An Introduction*. Springer Series in Synergetics. Springer-Verlag, Heidelberg, Berlin, New York, 1983.
5. P. Molnár. *Modellierung und Simulation der Dynamik von Fußgängerströmen*. PhD thesis, Universität Stuttgart, Verlag Shaker, Aachen, 1996.
6. P. Molnár and J. Starke. Control of distributed autonomous robotic systems using principles of pattern formation in nature and pedestrian behaviour. submitted to IEEE Trans. on Systems, Men and Cybernetics: Part B, 1999.
7. G. Nicolis and I. Prigogine. *Self-Organization in Non-Equilibrium Systems*. Wiley, New York, 1977.
8. J. Starke. *Kombinatorische Optimierung auf der Basis gekoppelter Selektionsgleichungen*. PhD thesis, Universität Stuttgart, Verlag Shaker, Aachen, 1997.
9. J. Starke, T. Kaga, P. Molnár, M. Schanz, and T. Fukuda. Experimental study on self-organized control of distributed autonomous robotic systems. unpublished, 1999.
10. J. Starke and P. Molnár. Dynamic control of distributed autonomous robotic systems with underlying three-index assignments. to appear in the IEEE Proceedings of the IECON 2000, 2000.
11. J. Starke and M. Schanz. Dynamical system approaches to combinatorial optimization. In D.-Z. Du and P. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 2, pages 471 – 524. Kluwer Academic Publisher, Dordrecht, Boston, London, 1998.