Nasa Institute of Advanced Concepts
USRA Grant 07600-044

Self-Organized Navigation Control for Manned and Unmanned Vehicles in Space Colonies

Final Report

Péter Molnár
Clark Atlanta University

November 30, 2000

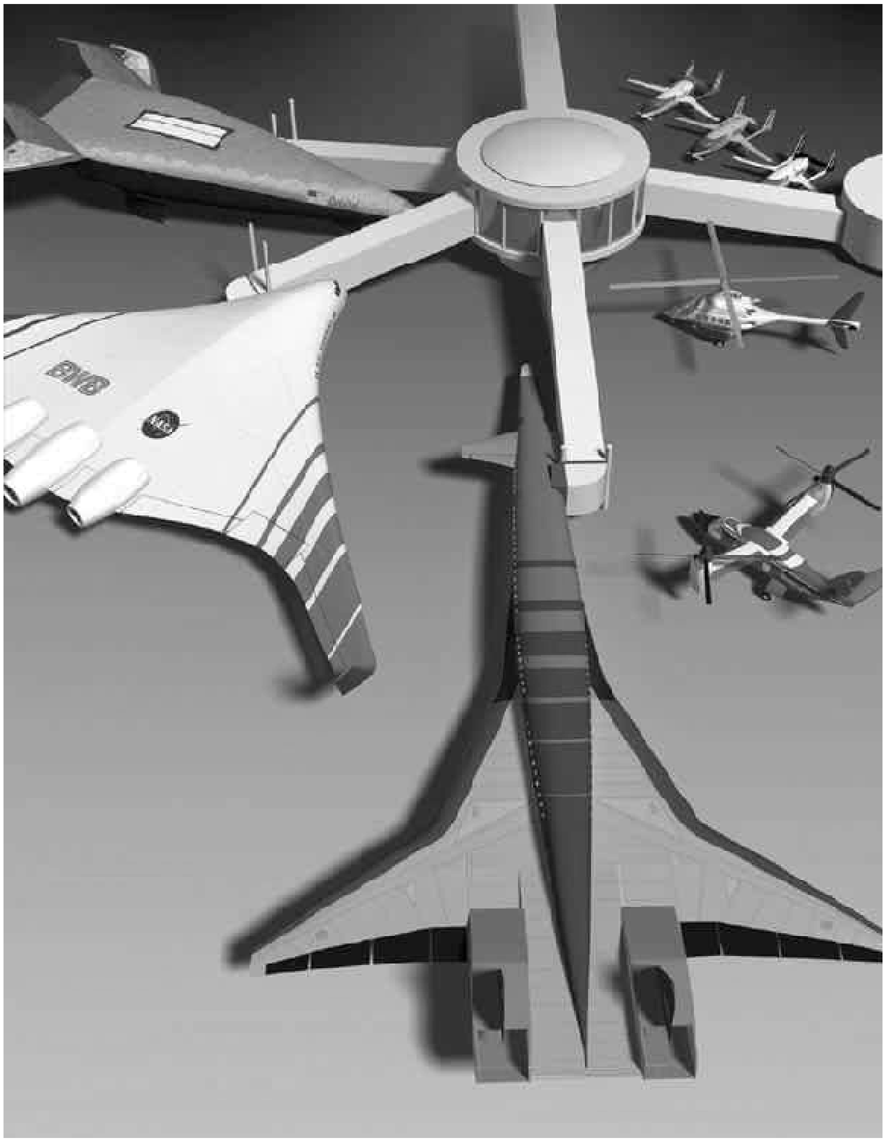


Figure 1: Space Port (source: NASA Strategic Plan 2000)

Contents

List of Figures

List of Tables

Summary

The objectives of the Phase I effort were to develop a preliminary mathematical and simulation framework for cooperative decision making and assess its feasibility within the context of autonomous mobile robots and distributed sensors. In particular, the results of Phase I effort have demonstrated the viability of using the coupled selection equations to implement robust and dynamic decision making.

We applied this formulation to various scenarios of task assignment between multiple mobile robotic agents, and were able to demonstrate

- robustness to device failures,
- resistance to communication loss, as well as the
- ability to form multi-dimensional teams in a real time manner.

Chapter 1

Concept of Self-organized Control

1.1 Areas of Application

The Self-organized Control System will benefit a number of sectors in the NASA Mission. We give a few examples here. The approach of using dynamical systems for planning and scheduling addresses two important aspects in particular:

1. **Redundancy:** In hostile environments, like space stations and colonies on distant planets, system failures and the loss of particular devices can affect the entire mission. In case of the break down of one device, other available units should fill in, or some device should change their previous assignment to fill the gap.

2. **Modular Components:** The use of smaller, more versatile units has the advantage of easy replacement and transportability to the space station. In recent years promising concepts of versatile robots, such as cellular robots, have been developed. These units can form teams in order to perform tasks such as moving large pieces of material, or hold them in place for assembly.

The computer simulations of various scenarios demonstrate the capability and feasibility of the self-organized control system based on Coupled Selection Equations. This approach can also be used for a different kind of autonomous agents, namely intelligent sensors.

1.1.1 Space Colonies

The construction and operation of space colonies will be a complex enterprise that requires a large number of devices, such as manned and unmanned vehicles and robots. A great deal of planning and scheduling is required to navigate those vehicles in an efficient manner. The coordination of the many different types of vehicles operating at a major airport illustrates some of the challenges facing the management of a space colony.

1.1.2 Space Port

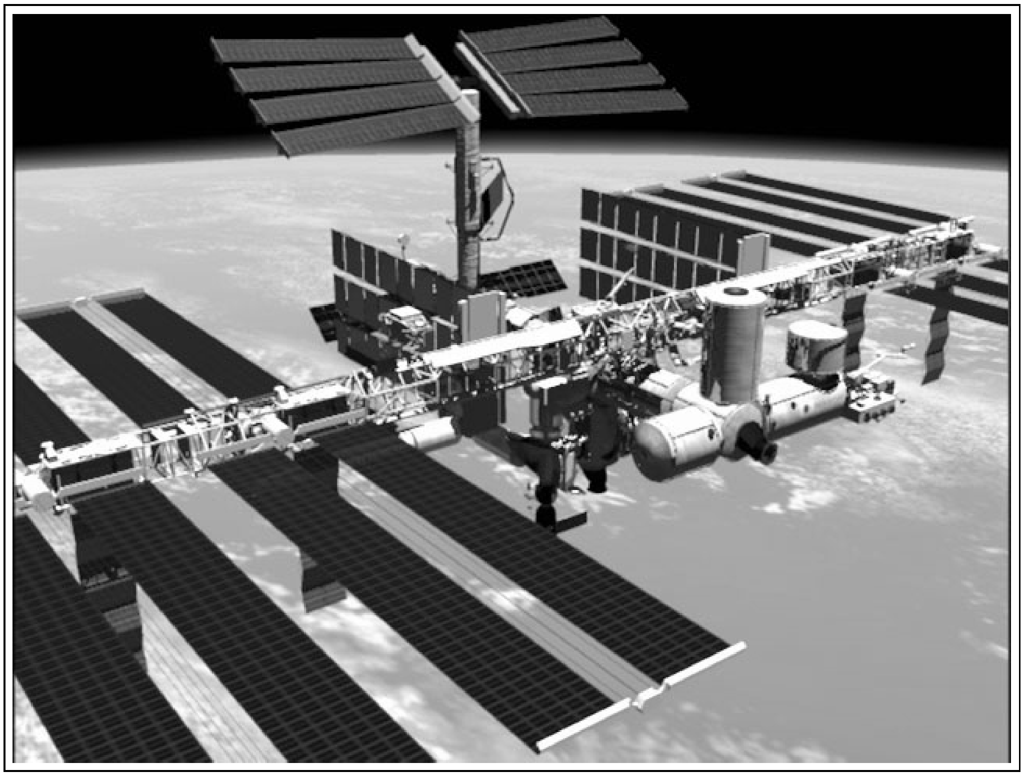


Figure 1.1: International Space Station (source: NASA)

At today's airports, all airplanes and ground vehicles must receive permission from the tower before they can travel to their destination. In contrast to this entirely centralized controller, self-organized planners make localized navigation decisions via communication between the network of mobile devices. Thereby, the controlling is distributed among the various units, and can be restricted within certain areas.

The self-organized control system will be able to handle both manned and unmanned vehicles to optimize resources. The entire systems will include a multitude of unmanned vehicle, or robots. However, the system can also assign space ships and shuttles to landing docks based on the availability of supporting devices and supplies at those locations.

1.1.3 Robots for Planetary Exploration

The self-organized control system can be implemented in robotic explorers such as the robot swimmers, proposed Physical Sciences Inc. [10]. The concept involves numerous small ($\approx$10 cm scale), autonomous, robotic swimmers which as individuals and as a cooperative group sense the ice, ocean, and sea floor environment and measure the physicochemical properties characterizing the presence of life on the Jovian moon Europa.

Using robots for explorations raises a number of concerns:

- the environment is not entirely know, therefore the system has to be able to adjust to unknown situations,
- the recovery of devices is usually not an option, therefore the failure or loss of some devices should not jeopardize the entire mission,
- and different types of devices, such as explorers, carriers and communication hubs have to operate in concert.

The conventional top-down control, i.e. the robots are individually controlled from a central (remote) location, is not feasible. Communication delays, the amount of information that has to be sent to the decision making site, and the complexity of systems with larger number of robots are prohibitive.

1.1.4 Space Stations

Space Stations like the International Space Station would greatly benefit from the cooperative robotics technology. Even though, stations of the near future are relatively small, they expose surfaces of several ten thousand square feet. The 70,000 sq. ft. sized solar panel of the International Space Station are critical to its functionality.

Swarms of robots, equipped with testing devices, could roam the solar panels, and issue early warning of damage or malfunction. They would cooperate with a different kind of robots that can repair the damage. An autonomous system could achieve this by allowing the scientists and astronauts aboard the station to concentrate on their primary mission. Controllers would be notified only when a problem could not be solved by the robot team. With increased sophistication, these robots could conduct repairs to the solar panels and to other parts of the station. A modular system consisting of self transforming autonomous robots could tackle almost any maintenance on the International Space Station, combining to increase their size, strength, usefulness. Even if the robots could not complete repairs they could assist any astronauts on repair oriented space walks.

Chapter 2

Mathematical Model

2.1 Coupled Selection Equations and Assignment Problems

Assignment of robotic units to targets requires a special selection mechanism. Traditionally combinatorial problems are solved with integer algorithms [8]. However recently several dynamical systems have been proposed to solve combinatorial problems. We use a system of Coupled Selection Equations (CSE) to solve assignment problems of dimension p. Hereby, p objects, each of a different population, have to be assigned. That can be, for example, $p-1$ different types of robotic units, each with different functionality or tools, which have to be combined to work together on one particular task. The population size for each dimension may vary, however only complete assignments should be allowed.

In general, an assignment problems of any dimension p can be represented as maximization problem of the total gain function, or the minimization of the total cost function respectively,

$$w^{\text{tot}} = \sum_{i_1,\dots,i_p} w_{i_1\dots i_p} \cdot \xi_{i_1\dots i_p} \tag{2.1}$$

with gain functions $w_{i_1\dots i_p}$ for each combination, $\xi_{i_1\dots i_p} \in \{0,1\}$ that complies to the conditions

$$\begin{aligned}
\sum_{\{i_1\dots i_p\}\setminus\{i_1\}} \xi_{i_1\dots i_p} &= 1 \qquad\qquad \forall i_1, \\
\sum_{\{i_1\dots i_p\}\setminus\{i_2\}} \xi_{i_1\dots i_p} &= 1 \qquad\qquad \forall i_2, \\
&\vdots
\end{aligned} \tag{2.2}$$

$$\sum_{\{i_1\dots i_p\}\setminus\{i_p\}} \xi_{i_1\dots i_p} = 1 \qquad \forall i_p.$$

In the two dimensional case, i.e., the assignment of exactly one robotic unit to each target, the dynamics of the coupled selection equations [7] is given by

$$\dot{\xi}_{ij} = \kappa\xi_{ij}\left(1-\xi_{ij}^2-\beta\sum_{i'\neq i}\xi_{i'j}^2-\beta\sum_{j'\neq j}\xi_{ij'}^2\right) \tag{2.3}$$

with $\beta > 1/2$, and a time scaling factor κ which adjusts the change per time compared to the equation of motion (2.10). Non-negative initial values

$$\xi_{ij}(0) := w_{ij} \qquad 0 \leq w_{ij} < 1 \qquad \forall i,j \tag{2.4}$$

ensure for square matrices (ξ_{ij}) that the system will always asymptotically end in a stable solution of permutation matrices, i.e., there is one and only one non-vanishing element which is equal to 1 in each row and in each column. A proof is given in [7]. Due to this fact, in the selection of the destination there is a one to one correspondence of robotic units to targets. In the case of a surplus of robotic units the coupled selection equations (2.3) ensure that there is not more than one target as destination for each of the robotic units. In reference to the boundary conditions of the maximization problem some of the sums in (2.2) would actually become zero. Because of the property of selection equations to select the largest initial value in the long time limit, these coupled selection equations can be used for assignment problems in combinatorial optimization [7]. Even though the global optimum is not necessarily reached, the outcome compares very well to other optimization algorithms [8].

If the problem requires to assign exactly two robots, each of a different type, to one target the coupled selection equations for a three dimensional tensor

$$\frac{d}{dt}\xi_{ijk} = \kappa\xi_{ijk}\left(1+(3\beta-1)\xi_{ijk}^2-\beta\left(\sum_{i',j'}\xi_{i'j'k}^2+\sum_{i',k'}\xi_{i'jk'}^2+\sum_{j',k'}\xi_{ij'k'}^2\right)\right). \tag{2.5}$$

with $\beta > 1/3$ have to be solved. Indices *i* and *j* denote robotic units of type *A*, and *B* respectively; targets are identified by index *k*.

In correspondence to equations (2.3) and (2.5) we can write the coupled selection equations for dimension *p* as

$$\dot{\xi}_{i_1\dots i_p} = \kappa\cdot\xi_{i_1\dots i_p}\left(1+(p\beta-1)\,\xi_{i_1\dots i_p}^2-\beta\left[\sum_{\{i'_1,\dots,i'_p\}\setminus\{i'_1\}}\xi_{i'_1,\dots,i'_p}^2+\dots+\sum_{\{i'_1\dots i'_p\}\setminus\{i'_p\}}\xi_{i'_1\dots i'_p}^2\right]\right) \tag{2.6}$$

with $\beta > 1/p$. For non-negative initial values the assignment emerges in the limit of large times as an asymptotically stable point that fulfills the conditions in (2.2).

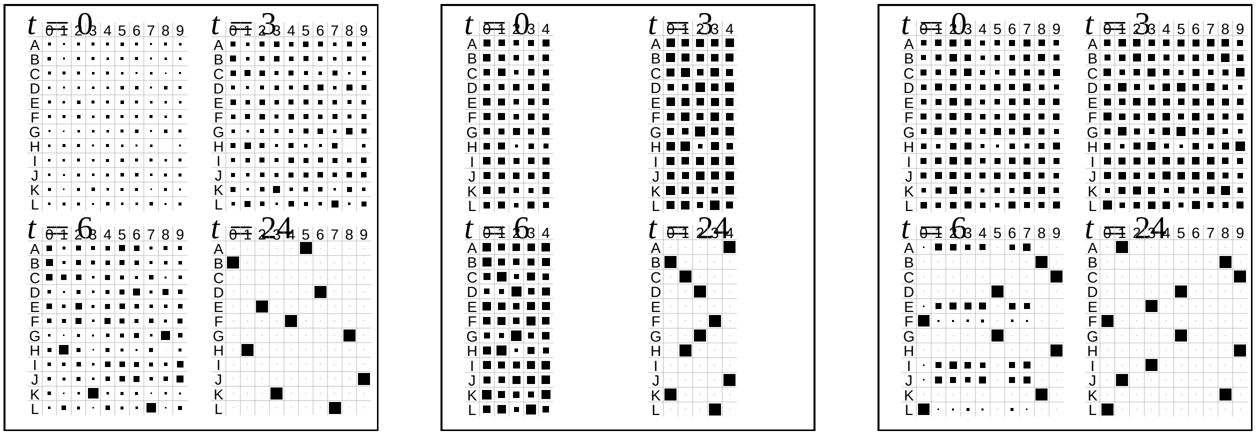


Figure 2.1: Two dimensional 12 x 10 assignment is shown on the Left. The values for ξ_{ij} are depicted as squares with the area proportional to value of that matrix element. Rows from 'A' to 'L' represent the robotic units, columns '0' to '9' the targets. The values might actually increase after initialization, before all but one in a row (or column) vanish. Since there are more robots than targets in this example the coefficients of two rows, 'C' and 'I', all are zero. The middle figure shows the development of a three dimensional 6 x 6 x 5 assignment problem. Two different types of robots, with 'A' to 'F' of one kind, and 'G' to 'L' of the other, have to be assigned with exactly one device of each kind to one target. Here the three dimensional tensor ξ_{ijk} is mapped onto a two dimensional matrix $\tilde{\xi}_{ik} = \sum_j \xi_{ijk}$ that represents the assignment of each robotic unit to a target individually, by summation of the coefficient of all the other robots. Again, two robots, 'E' of type one, and 'I' of type two, will not be assigned to any target. The figure on the Right shows a three dimensional 6 x 6 x 10 problem with a surplus of targets. The system performs as expected by always assigning exactly two object, one of each kind, to one target. Targets '2', '4', '6', and '7' show only null values in their columns.

Figure 2.1 shows the temporal evolution of two and three dimensional coupled selection equations.

For the initial values of ξ_{ij} we use the linear transformed EUCLIDean distance of the robotic units to the targets at $\mathbf{g}_j$

$$\xi_{ij}(0) \quad = \quad 1 - \frac{\|\mathbf{r}_i(0) - \mathbf{g}_j\|}{\max\limits_{i',j'}\left(\|\mathbf{r}_{i'}(0) - \mathbf{g}_{j'}\|\right)}. \tag{2.7}$$

Using this transformation, targets which are located closer to the robotic units than others obtain larger initial values, i.e., a larger preference for the final selection. From the definition follows directly that $\xi_{ij} \in [0,1]$ for all i, j. However, more sophisticated cost functions including manufacturing costs at the target's location, could be considered as well.

The time dependent matrix $(\xi_{ij})(t)$ depicts the preferences that each of the robotic units has for a certain target, or the preferences the targets have for being served by a specific robot.

2.2 Behavioral Force Model to Navigate the Robotic Units

A Behavioural Force Model is used to guide autonomous mobile robotic units through their environment. In the Behavioral Force Model any events, such as the presents of other robots or any obstacles in the way, are represented as force terms that yield the object to the typical behaviour of this situation.

In analogy to physical systems the intended change of the objects velocity vectors $\mathbf{v}_i(t) \in \mathbf{R}^2$ (or $\mathbf{R}^3$) is usually defined as [2] [5]

$$\frac{d}{dt}\mathbf{v}_i(t) = \frac{1}{\tau}\left(v_i^0\mathbf{e}_i^0(t) - \mathbf{v}_i(t)\right) + \sum_{i' \neq i} \mathbf{f}^{\mathrm{r}}_{ii'}(\mathbf{r}_{i'} - \mathbf{r}_i) + \sum_k \mathbf{f}^{\mathrm{o}}_{ik}(\mathbf{x}_k - \mathbf{r}_i) \tag{2.8}$$

and

$$\frac{r}{dt}\mathbf{r}_i(t) = \mathbf{v}_i(t) \tag{2.9}$$

even though no external forces are exerted onto the robotic units.

The direction to the destination of a robotic unit is given by a normalized vector $\mathbf{e}_i^0(t) \in \mathbf{R}^2$ (or $\mathbf{R}^3$) which points in direction of the target that has been assigned to this unit. Usually the robotic devices operate best at a certain speed $v^0 \in \mathbf{R}$ which is determined by the nature of the robotic units, similar to the constant walking speed we observe with most pedestrians. Within a certain relaxation time τ the units recover from any deviation of their path.

We simplify the dynamics of (2.8) and 2.9) by expressing the movement of the robots as a direct result of the destination vector $\mathbf{e}^0$ and the behavioral forces

$$\frac{d}{dt}\mathbf{r}_i(r) = v_i^0\mathbf{e}_i^0 + \sum_{i' \neq i} \mathbf{f}^{\mathrm{r}}_{ii'}(\mathbf{r}_{i'} - \mathbf{r}_i) + \sum_k \mathbf{f}^{\mathrm{o}}_{ik}(\mathbf{x}_k - \mathbf{r}_i) \tag{2.10}$$

The forces lose their meaning in the NEWTONean sense, though the representation of the dynamics in first order differential equations allows more reliable numerical integration.

The short range force fields for collision avoidance are defined by

$$\mathbf{f}^{\mathrm{r,o}}_{ii'}(\mathbf{r}) = \begin{cases} (\tan g(\tilde{r}) - g(\tilde{r}))\,\frac{\mathbf{r}}{\|\mathbf{r}\|} & \text{for} \quad 0 < \tilde{r} \le \sigma^{\mathrm{r,o}} \\ 0 & \text{for} \quad \tilde{r} > \sigma^{\mathrm{r,o}} \end{cases}$$
$$g(\tilde{r}) = \frac{\pi}{2}\left(\frac{\tilde{r}}{\sigma^{\mathrm{r,o}}} - 1\right) \tag{2.11}$$

with $\mathbf{r}$ being the distance between the centers of the robots and the obstacles, $\tilde{r} = \|\mathbf{r}\| - d^{\mathrm{r}}_{i}/2 - d^{\mathrm{r,o}}_{i'}/2$ being the distance between their perimeters, and $\|\cdot\|$ is the EUCLIDean norm, around each of the other units i' at the locations $\mathbf{r}_{i'}$ with diameter $d^{\mathrm{r}}_{i'}$ and obstacles at the locations $\mathbf{x}_{i'}$ with diameter $d^{\mathrm{o}}_{i'}$ (see Fig. 2.2). The distances between robotic units and obstacles can be adjusted with appropriate range parameters σ^{r} and σ^{o}. Figure 2.3 shows a plot of the used short range force.

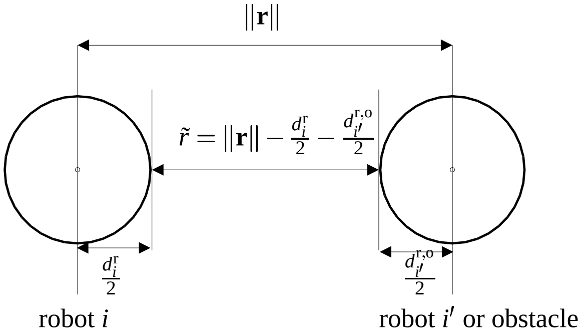

$\|\mathbf{r}\|$
$\tilde{r} = \|\mathbf{r}\| - \frac{d^{\mathrm{r}}_{i}}{2} - \frac{d^{\mathrm{r,o}}_{i'}}{2}$
$\frac{d^{\mathrm{r}}_{i}}{2}$
$\frac{d^{\mathrm{r,o}}_{i'}}{2}$
robot i
robot i' or obstacle

Figure 2.2: Definition of distance between a robot i and another robot i' or an obstacle.

The use of short range potential force avoids unwanted local minima or spurious states where the system could get stuck in. These repulsive forces can be calculated with the information from infrared or ultrasonic sensors that detect the proximity to an obstacle within a given range. Many mobile robotic units are equipped with a number of such sensors around their perimeter. To avoid

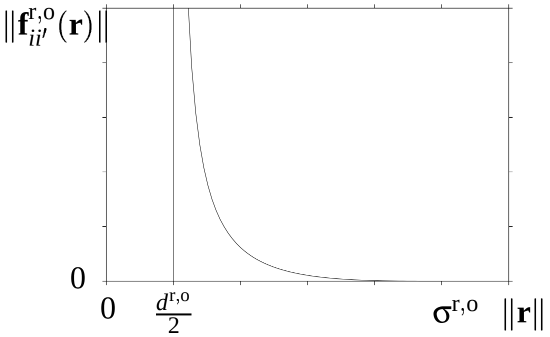


Figure 2.3: Plot of the used short range force $\mathbf{f}^{\mathrm{r,o}}_{ii'}(\mathbf{r})$ defined in (2.11).

the stagnancy of this multiple particles system at unstable stationary points, small fluctuations are added near stationary points.

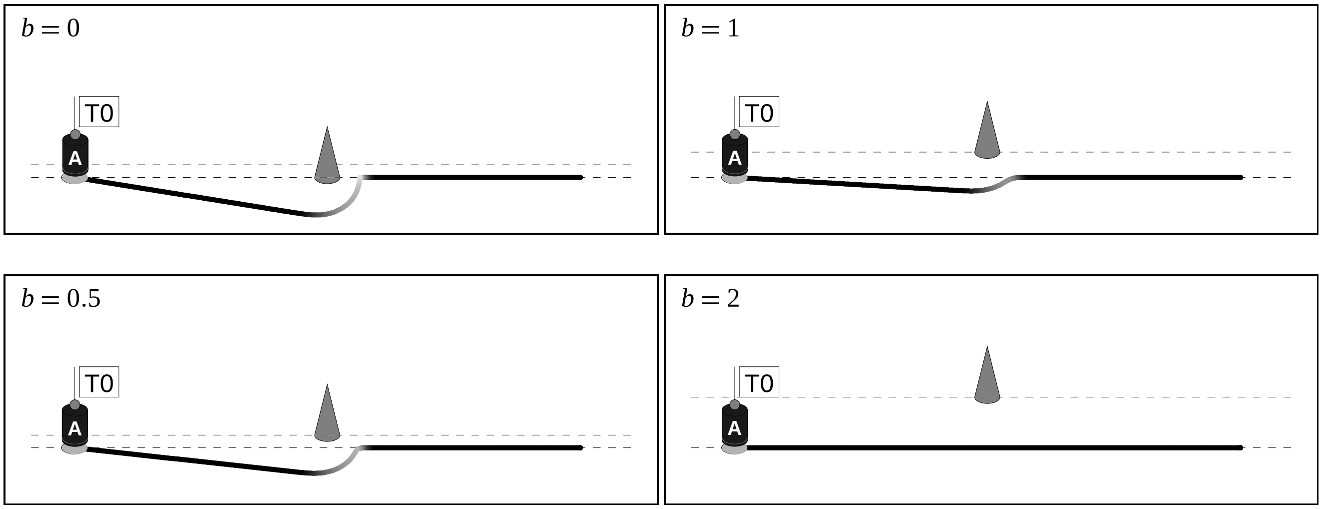


Figure 2.4: The trajectory of a single robot passing an obstacle is shown for various distances b from the obstacle to the robot's path. Collision avoidance is obtained by the repulsive force term in equation (2.11). In case of central collision, i.e. $b = 0$, the unstable equilibrium can be avoided by small fluctuations. The finite range of the repulsive potential has been chosen with $\sigma = 1$. For obstacles and robotic units with a diameter of one length unit each, the minimum distance for passing the obstacle without deviation is $b = 2$.

Trajectories as a result of the navigation model are depicted in Figure 2.4. The examples show a robotic unit passing an obstacle in various distances.

2.3 Integrated Decision and Navigation System

The preference that each of the robotic units has for the targets can be directly linked into the dynamics of the mobile units by choosing the destination vector

$$\mathbf{e}_i^0(t) \quad = \quad \mathbf{N}_{\gamma\delta}\left(\sum_j \xi_{ij}(t)\mathbf{N}_{\gamma'\delta'}\left(\mathbf{g}_j - \mathbf{r}_i(t)\right)\right) \tag{2.12}$$

as a linear combination of the difference vectors of the robotic unit i to all targets. This linear combination of the with $\xi_{ij}(t)$ weighted normalized difference vectors $\mathbf{N}_{\gamma'\delta'}\left(\mathbf{g}_j - \mathbf{r}_i\right)$ is demonstrated in figure 2.5.

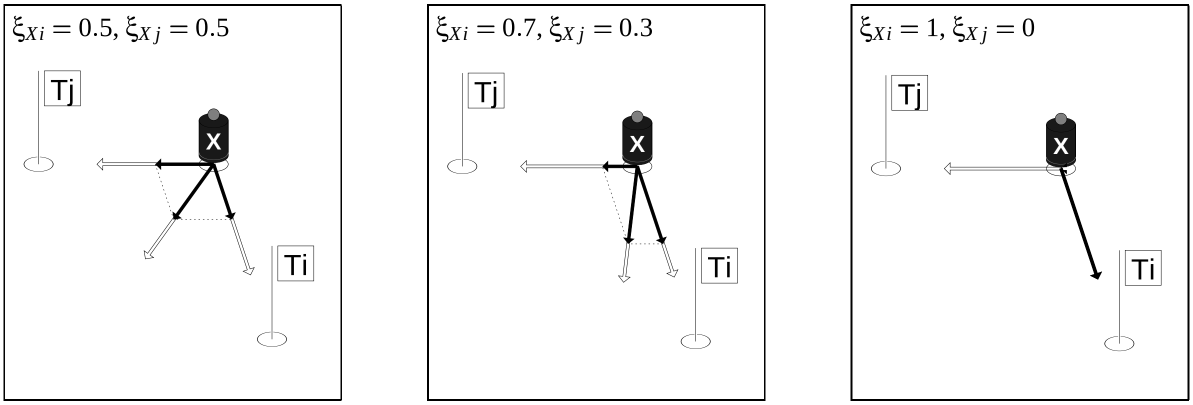

$\xi_{Xi} = 0.5, \xi_{Xj} = 0.5$
Tj
X
Ti
$\xi_{Xi} = 0.7, \xi_{Xj} = 0.3$
Tj
X
Ti
$\xi_{Xi} = 1, \xi_{Xj} = 0$
Tj
X
Ti

Figure 2.5: The robotic unit "X" stands close to two targets. Outlined arrows depict the unit vectors $\mathbf{e}_i^0$ to target "Ti" and $\mathbf{e}_j^0$ to target "Tj". Solid arrows represent the unit vectors multiplied by the coefficients $\xi_{Xi}\mathbf{e}_i^0$ and $\xi_{Xj}\mathbf{e}_j^0$ and their linear combination $\xi_{Xi}\mathbf{e}_i^0 + \xi_{Xj}\mathbf{e}_j^0$. The figures show the resulting orientation vector for various values of ξ.

The function $\mathbf{N}_{\gamma\delta}(\mathbf{x}) = \frac{1}{||\mathbf{x}||+1/(\gamma||\mathbf{x}||+\delta)} \cdot \mathbf{x}$ with $\gamma, \delta > 0$ mainly normalizes the vector $\mathbf{x}$ but avoids a singularity at $\mathbf{x} = \mathbf{0}$.

Once the dynamical system of coupled selection equations has reached a stable point, there will be exactly one destination vector left for each of as many robotic units as there are targets. All remaining units will stop moving as soon as all of their corresponding coefficients converge close enough to zero.

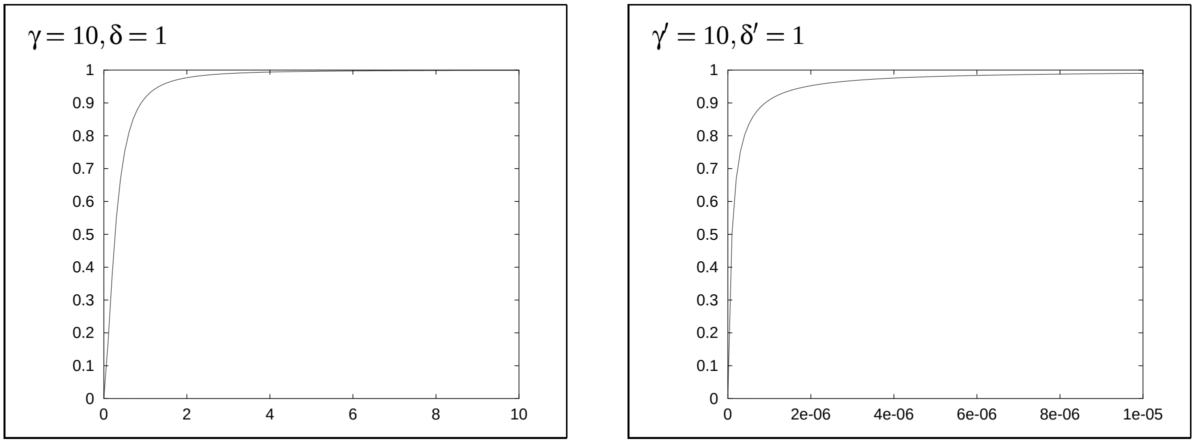


Figure 2.6: Function **N** is used to normalize vectors the the unit length without running into a singularity at **0**.

Chapter 3

Self-organized Task Assignment, Teaming

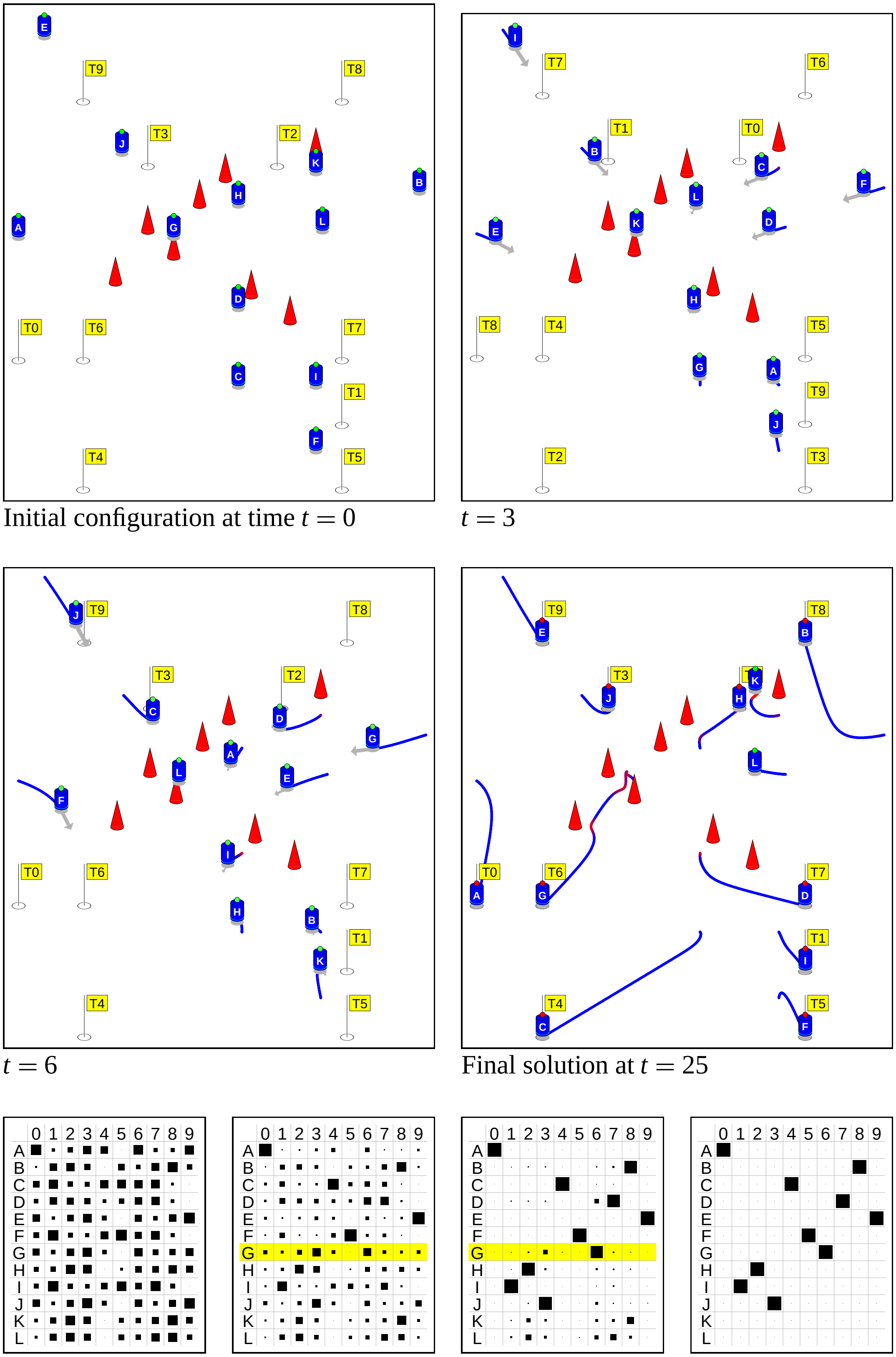
The distributed control system has been implemented in a simulation program. In this chapter we present some of the simulation results. We use one or two population of vehicles, with units of the same population are considered to be exchangeable.

3.1 Two-indexed Assignment

The first scenario, as depicted in Figure 3.1, could be an example for robots fueling spacecrafts after they arrived, or vehicles that pickup cargo and passangers. From the persepctive of the robot population, only the tasks that need the robots' services will be included in their decision process. Figure 3.1 shows ten pending tasks at different locations at the time $t = 0$. At this time we assume that twelve devices are available.

When the control process starts, the robots start "negotiating" about their assignments. They define their initial preferences, based on their position to the various targets. Then, those preference will be exchanged among the robots, and considered by them in their decision process.

The example shows two more robots than pending tasks. Two of those robots will realize that they are not needed, and suspend themselves until new tasks emerge.



Initial configuration at time $t = 0$

$t = 3$

$t = 6$

Final solution at $t = 25$

time $t = 0$

$t = 3$

$t = 6$

$t = 25$

Figure 3.1: Scenario: Robots deliver supply to the space crafts. The figure shows the position of the robots at various times. Below the corresponding state of the preference matrix (ξ_{ij}) is shown.

3.2 Heterogenous Robot Teams

Two different robot populations. For every task one of each has to be present. This could be the case for some repair job where one robot can hold the material while another one is equipped with a welding tool.

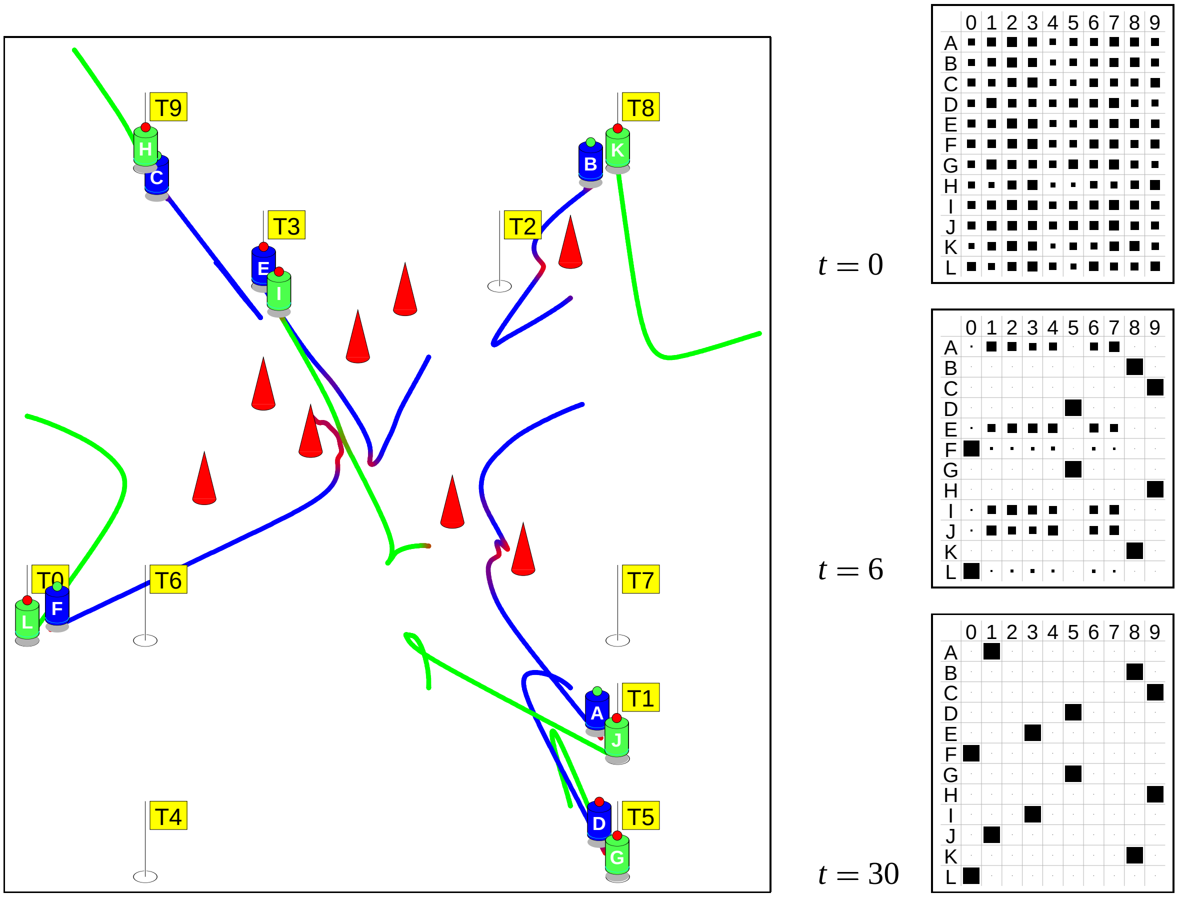
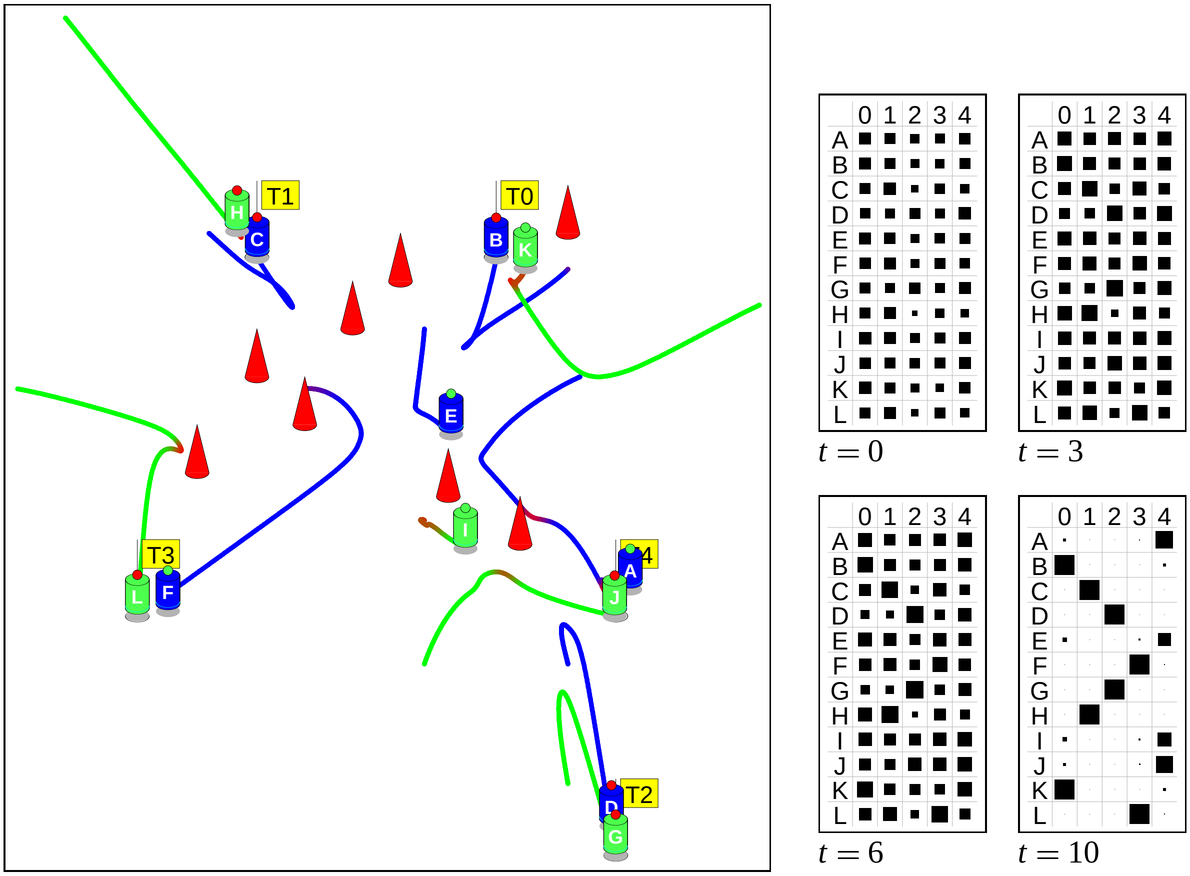


Figure 3.2: Three-indexed Problem



$t = 0$ $t = 3$

$t = 6$ $t = 10$

Figure 3.3: Three-indexed Problem (2)

Chapter 4

Robustness to Device Failures

4.1 Automatic Activation of Spare Units

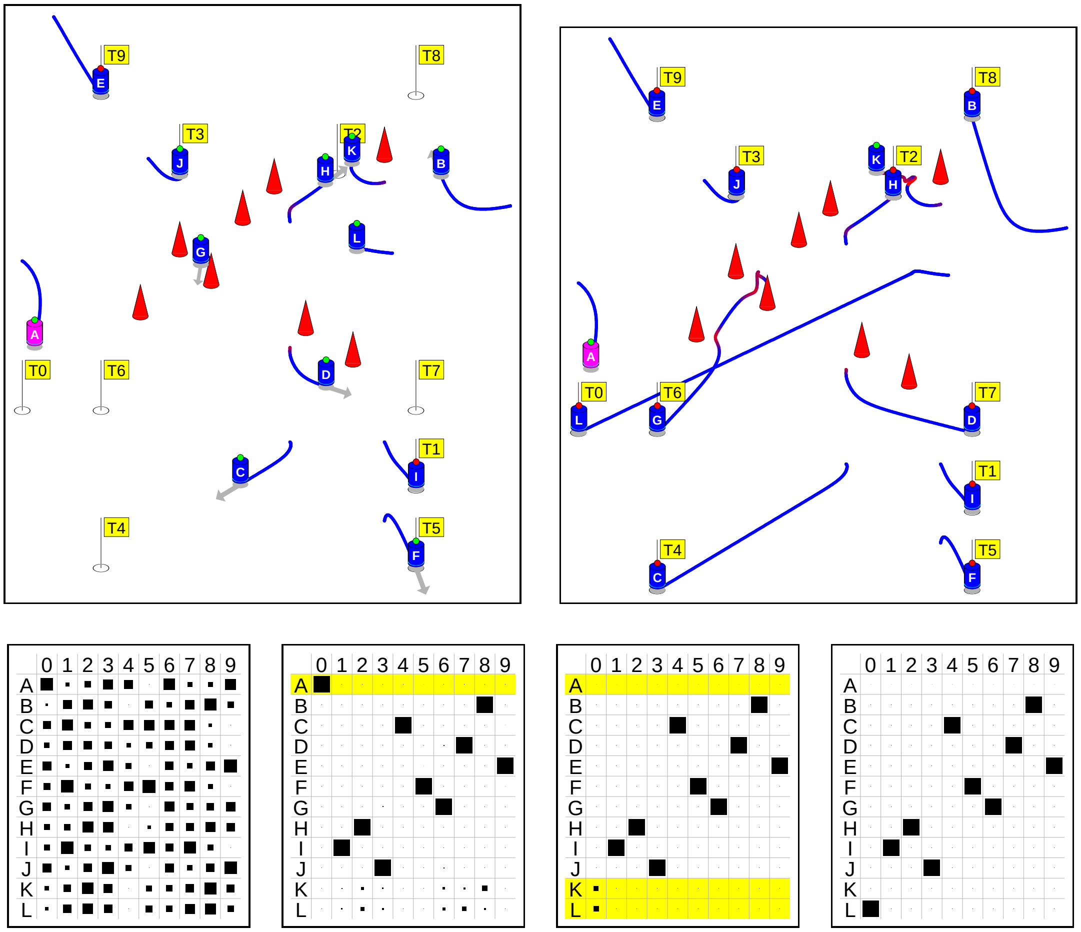
If more robots are available than there are pending tasks, some of the units will not persue a target, but stop and wait. The system offers an intrinsic functionality to activate these spare units in case one of the robots breaks down.

Figure 4.1 shows the progress of the robots up to a certain time, when one of the units fails. A breakdown of the robotic unit “A” at time $t = 7.4$ (left), and the self-organized recovery, i.e., replacement by unit “L” to provide target “T0” (middle) is shown. At the time the robot breaks down, it has already developed a preference for target “T0”. Once the failure is detected all coefficients ξ_{ij} that correspond to “A” are set to zero. The idle units “K” and “L”, in competition, gain preference towards the released target (right, highlighted rows), and start moving. Finally, robot “L” takes over.

4.2 Re-teaming

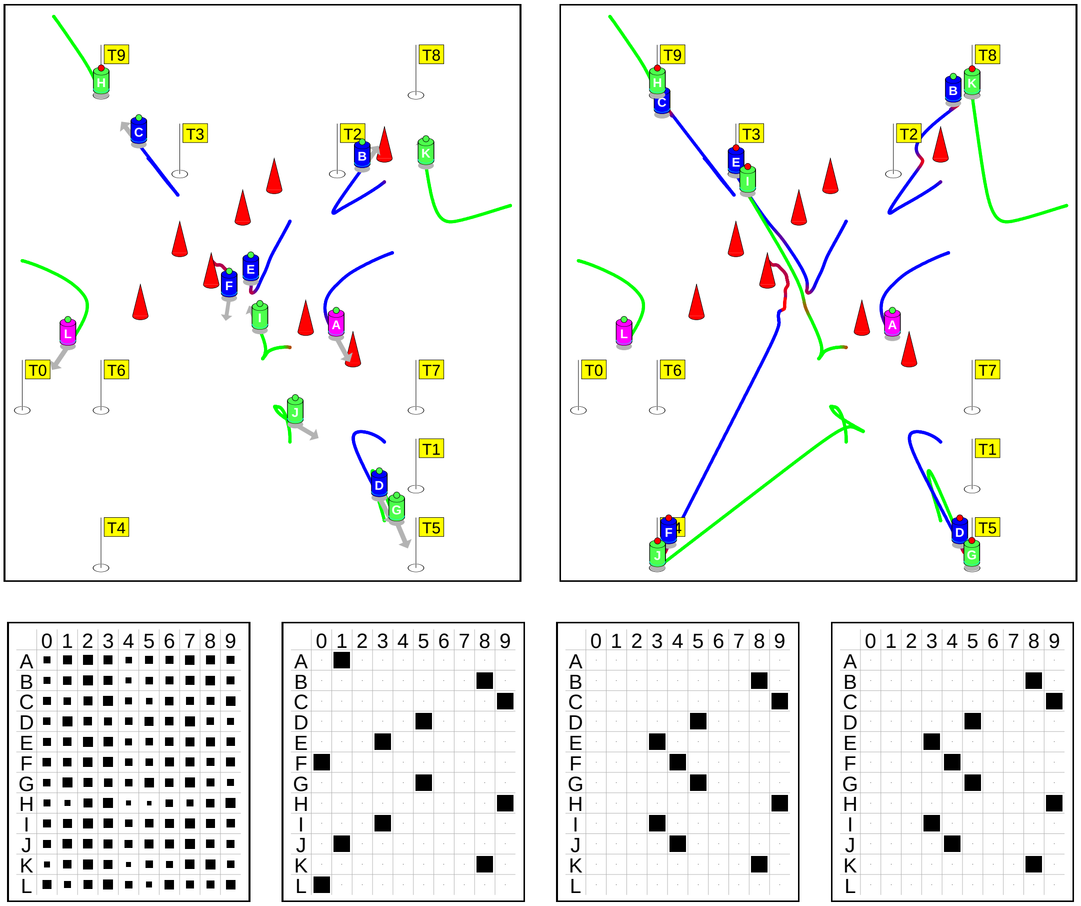
A greater challange to the system is when a robots of a team break down. It wouldn’t make sense for the remaining team-members to persue with the assignment since they would not be able to perform the tasks. Rather, robotic unit that a left in the broken team have to find new partners, and eventually persue a new assignment.

This scenario is shown in Figures 4.2 and 4.3.



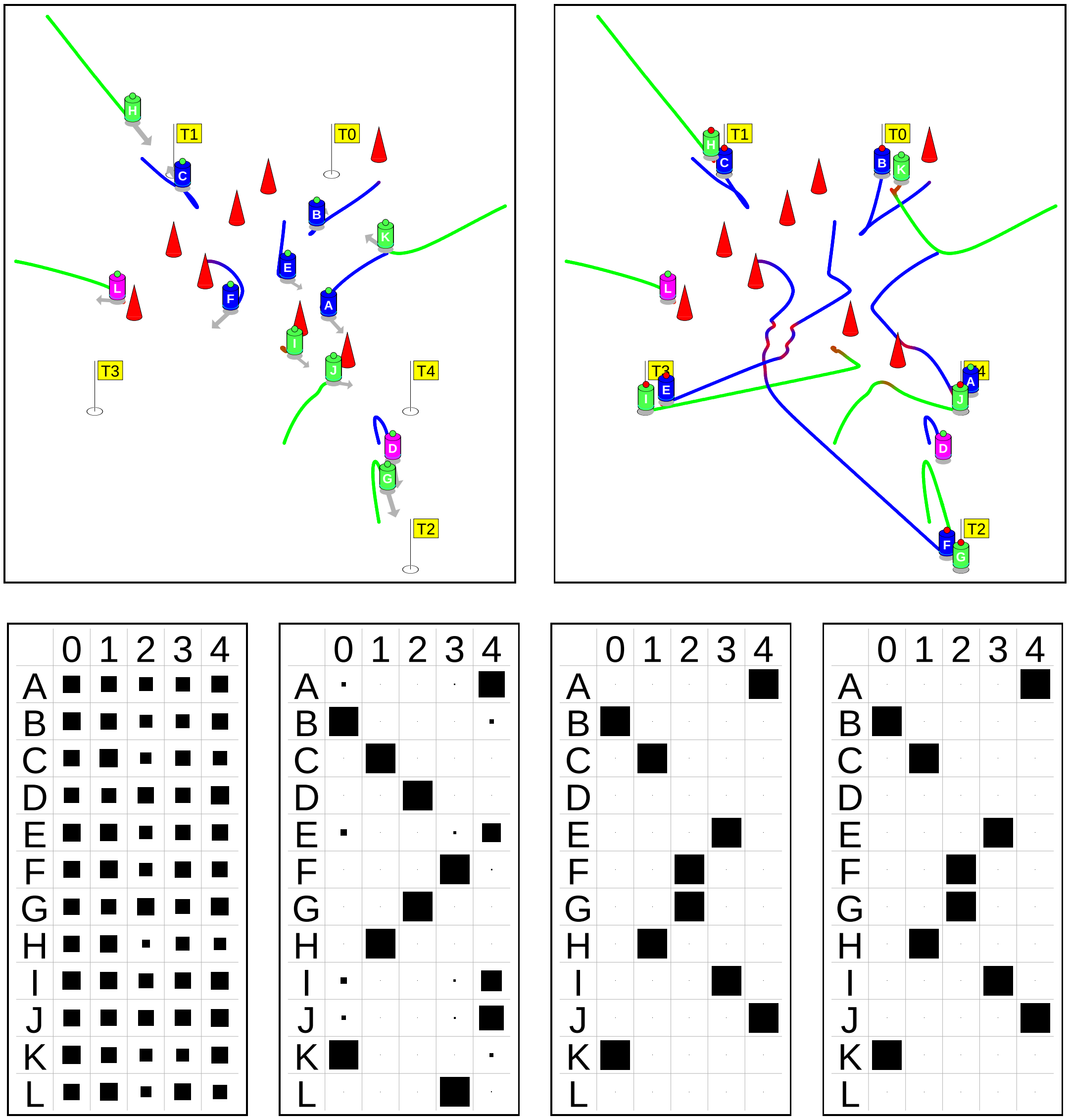
$t = 0$ $t = 7.5$ Failure $t = 15$ Recovery $t = 45$

Figure 4.1: Break Down of units ’A’ at $t = 7.5$. Once, the preferences of ’A’ are erased, units ’K’ and ’L’ become automatically active, and ’L’ will finally take over.



$t = 0$ $t = 10$ Failure $t = 20$ Recovery $t = 45$

Figure 4.2: Ten targets that require a team of two robots. At $t = 10$ the units 'D' and 'L' fail. The remaining robots of the broken teams have to find new partners.



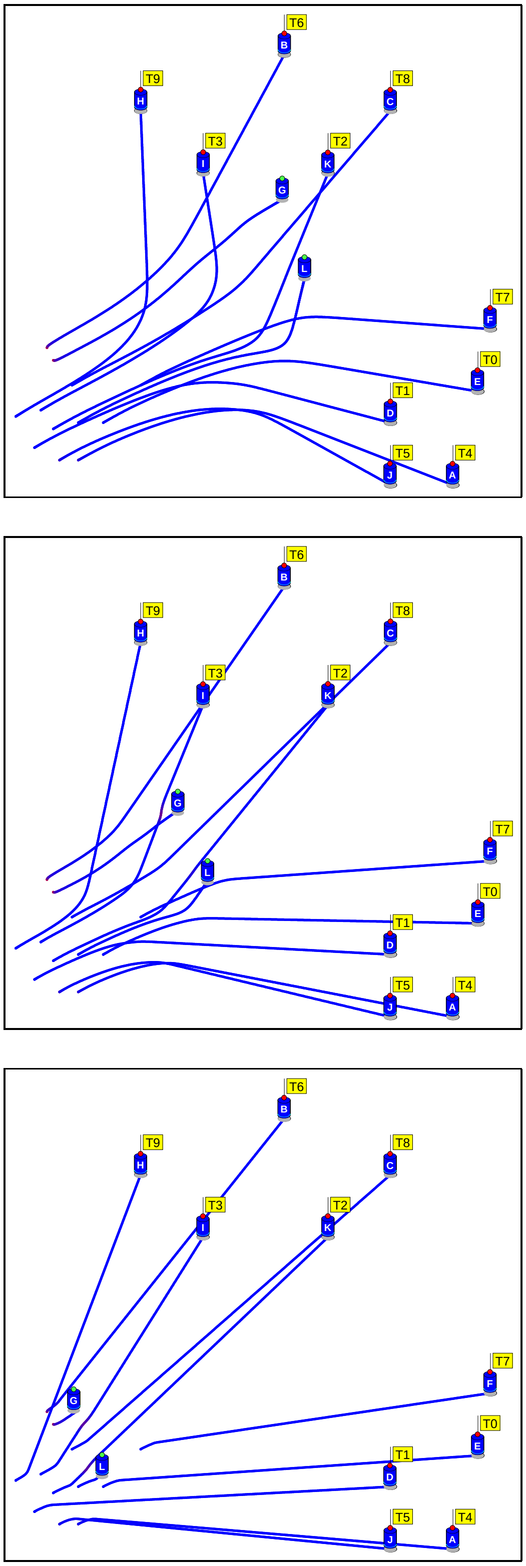
$t = 0$ $t = 10$ Failure $t = 12$ Recovery $t = 55$

Figure 4.3: Five targets that require a team of two robots. When two units break down, the remaining devices have to re-team.

4.3 Time Scaling

The time scales of the decision process (2.6) and the navigation process (2.10) are linked by the control paramter κ. This parameter determines how fast the decision process runs with respect to the navigation of the robots to their destination. In many situations it may be desirable to have the assignment quickly, and the let the units move straight to their destination (see Fig. 4.4c).

However, Figure 4.4 shows a scenario that could represent the delivery of goods from one central location (in the bottom left corner) to several destinations. The delivery off the goods is crucial, and the transport vehicles are at hight risks to fail. On extraterrestrial missions that may very likely be the case. The control parameter can be choose in a way that delays the decision process, while the vehicles are already moving in the general direction to the destinations. Before the assignments have been finalized, the robots have alreay a general idea about the outcome, based on the stronger elements of their preference vector. As a result, the vehicles are in a better position to change their course suddenly when one unit fails. One can send a certain number of additional transporters, just to ensure that all destination points receive their goods.



κ = 0.75

κ = 1.5

κ = 10.0

Figure 4.4: κ = 0.75, κ = 1.5 κ = 10.0

Chapter 5

Communication

An important part of the self-organized control system, as with any distributed system, is the aspect of communication: the vehicles have to receive information about pending tasks, and they have to exchange their intentions in form of preferences ξ_{ij}. Figures 5.1 and 5.2 show two concepts which differ in the way the robotic vehicles would be aware of pending tasks. In the first model (Fig. 5.1) target would announce its presence and service needs via broadcasting messages. E.g. a docking space craft could announce its demands, including its location, the amount of cargo etc.

The second model (Fig. 5.2 assumes that the tasks will be identified by the robots themselves, e.g. survaillance robots can send messages when they detect a problem to the maintenance and repair devices.

All communications can be implemented in multicast message passing, i.e. a message sent by one unit, will be received by all robots (in range). The robots broadcast their preferences periodically, therefore, a confirmation protocol that ensures that every message reaches the receiver is not required; the loss of information does not have a significant impact if the update messages are sent frequently.

The nature of the Coupled Selection Equations (2.6) allows to divide the system in local regions, and to implement a hierarchical communication structure (see Figure 5.3).

Equation (2.6) only considers positive preferences from the other robots. If the preferences of another robot are not known, they wont contribute to the sum, and wont have impact on the robots decision. Units that are out of range don't compete for the same tasks, which will in most cases be very reasonable, because they are to far away to get to the job-site.

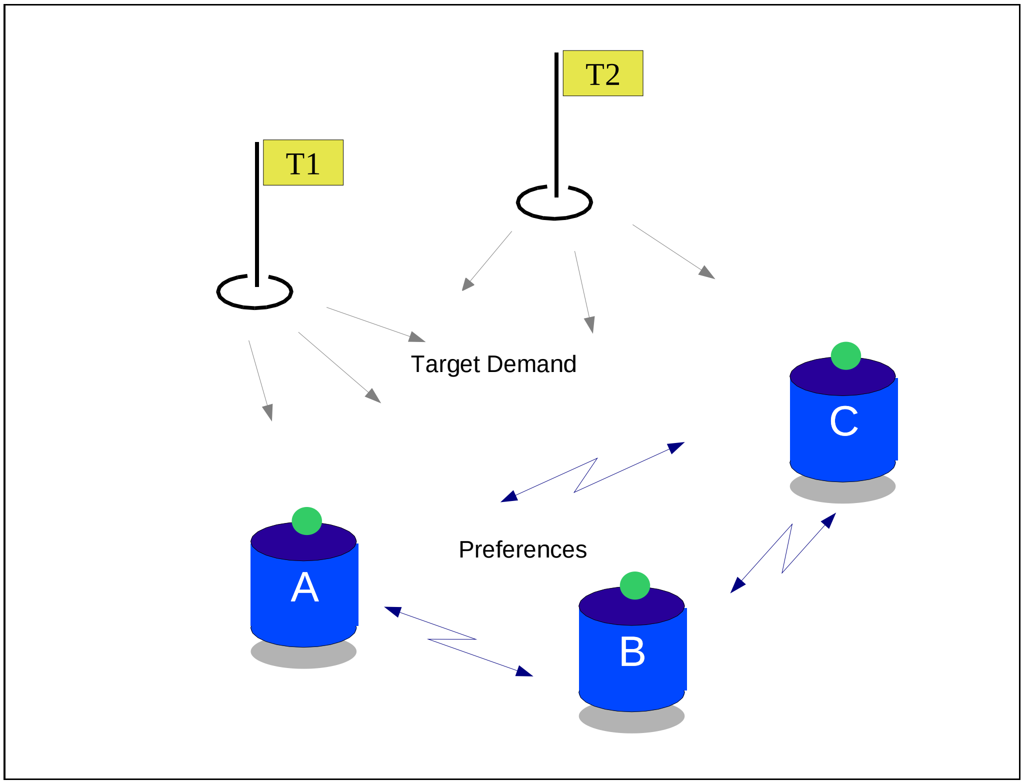


Figure 5.1: Multicast Communication, Target Demand: task requirements, location, Preferences: packages of the robots (ξ_i)-vectors. Robots learn about pending tasks by receiving target demand packages. They periodically broadcast updates of their preferences.

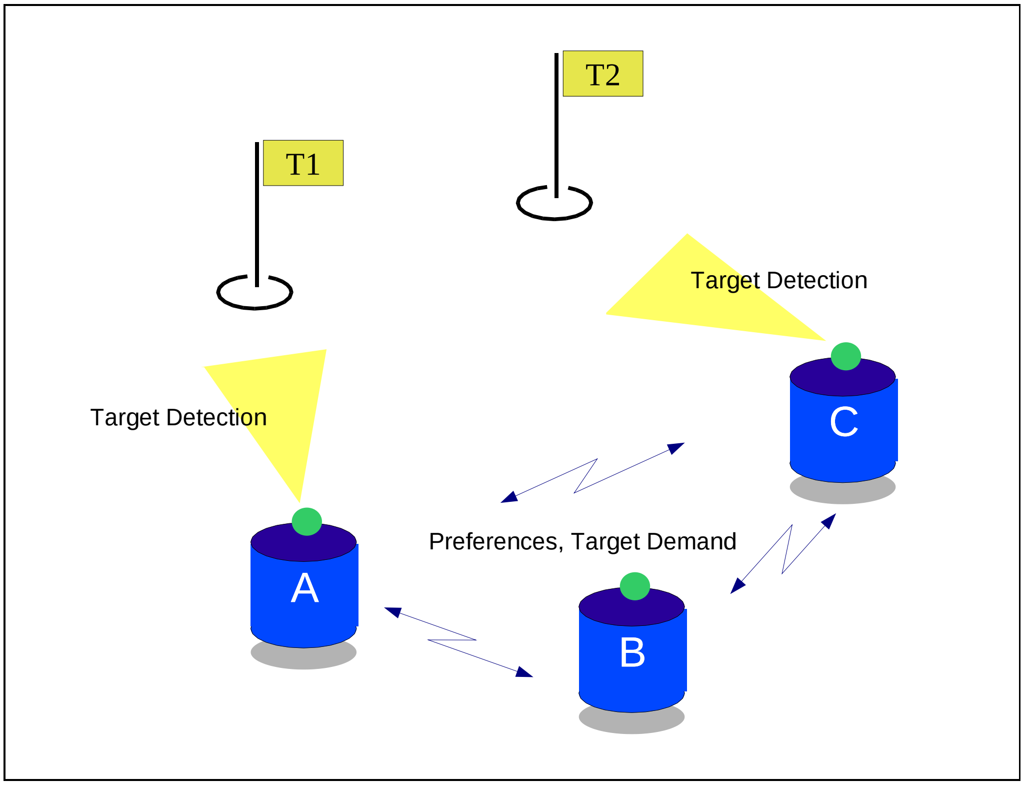


Figure 5.2: Multicast Communication, Target Detection: e.g. vision, Target Demand: task requirements, location, Preferences: packages of the robots (ξ_i)-vectors. Robots detect targets, and recognize their demands. They periodically broadcast updates of their preferences and information about targets that they have detected.

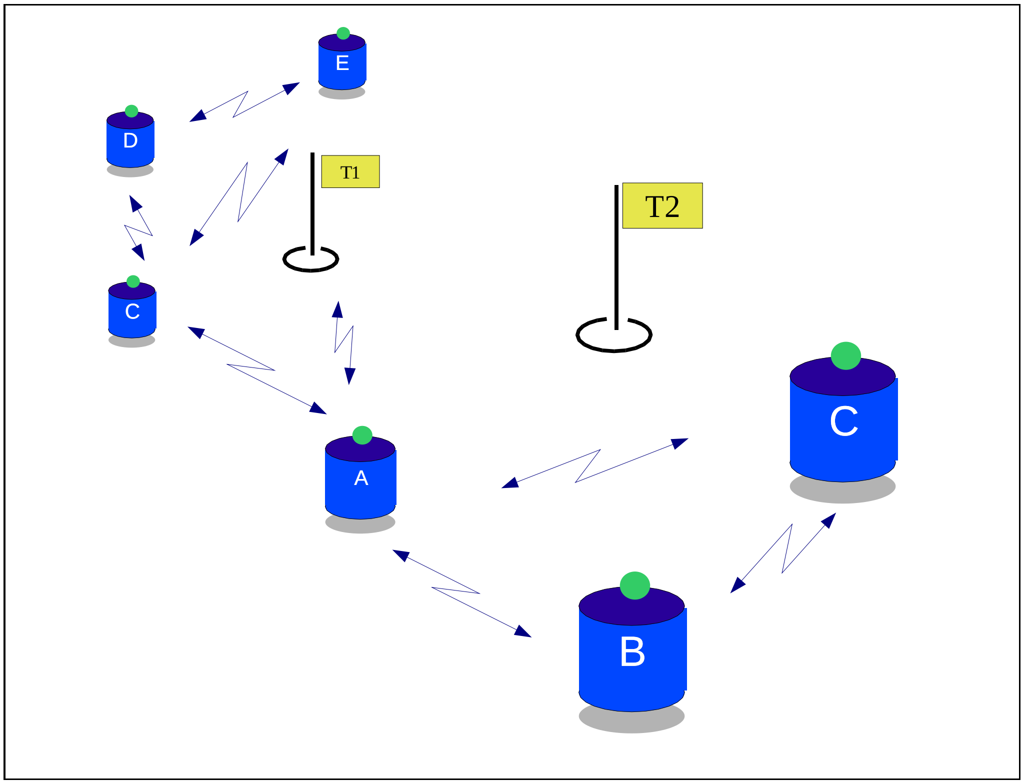


Figure 5.3: Limited range of communication. Robots consider only targets in their range. Selected units can aggregate preferences of robots in their immediate range, broadcast the information to other regions.

5.1 Demonstrating the Fault Tolerance

The follwowing shows results with simulated communication errors: messages will be randomly dropped with a given probability p. As a result, the representation of the decision state will eventually differ among the robots.

The simulation program is based on the equations (2.6) and (2.10) as described in section 2. The differential equations have been integrated by using the EULER-forward method with time steps $\Delta t = 0.05$ for the navigation part (2.10). The time scale of the coupled selection equations (2.6) is accelerated by factor $\kappa = 3$.

Every robot receives an update of the other robots' preferences in intervals of $\Theta = 1.00$. After $n = \frac{\Theta}{\Delta t}$ iterations the simulation program will copy the each robot's current preferences into the matrix (ξ_{ij}) of any other unit. Based on a random number, that will be generated for each transaction, and the package loss probability p, the program will decide whether the matrix elements will actually be copied. It is assumed that the robots communicate by sending data-packages using a non-confirmed protocol, such as UDP. Therefore, a data-package containing the recent preferences of one robot, might be received by some but not all devices.

Ten simulations have been run for each of ten values for p. Table 5.1 shows the elapsed time until every robot of the system reached a target and the final target assignment of the robots (A–J) for each of the simulation runs.

The collision avoidance maneuver of the robots extends the time they need to reach their destination. If a certain number of messages got lost, robots might choose a slightly different assignment which can actually result in shorter travel times, because the new path collides less often with obstacles. This explains why in the presented simulations the average elapsed time of ten simulation runs with 20% package loss probability is lower than the value in the 0% case (Fig. 5.4).

Higher rates of package losses will violate the constraint of assigning exactly one robot to every target. Simulations with package loss probability $p \geq 0.6$ may result in unfeasible assignments, where more than one robot approaches the same target (Figure 6.3.d).

The increase of these incidents with more frequent communication drop-outs is depicted in Figure 5.5. Even with no communication at all, robots still would approach a target. The target of their choice would depend on the initialization value $\xi_{ij}(0)$. However, several devices may select the same target, and some targets might be left out.

Our results show that the used distributed control [5] [9] is extremely stable and robust against communication drop-outs up to 50%.

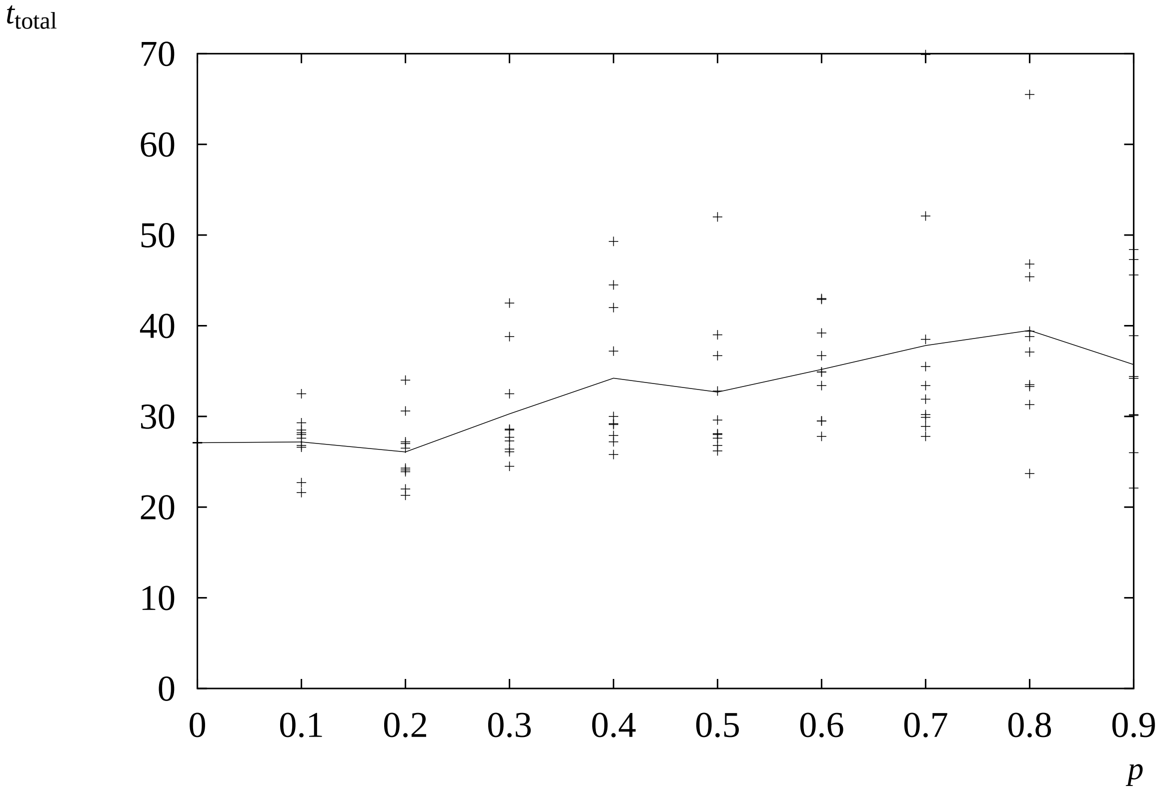


Figure 5.4: Elapsed times t_{total} until every robot reached a target. The solid line graph shows the average time over the package loss probability p.

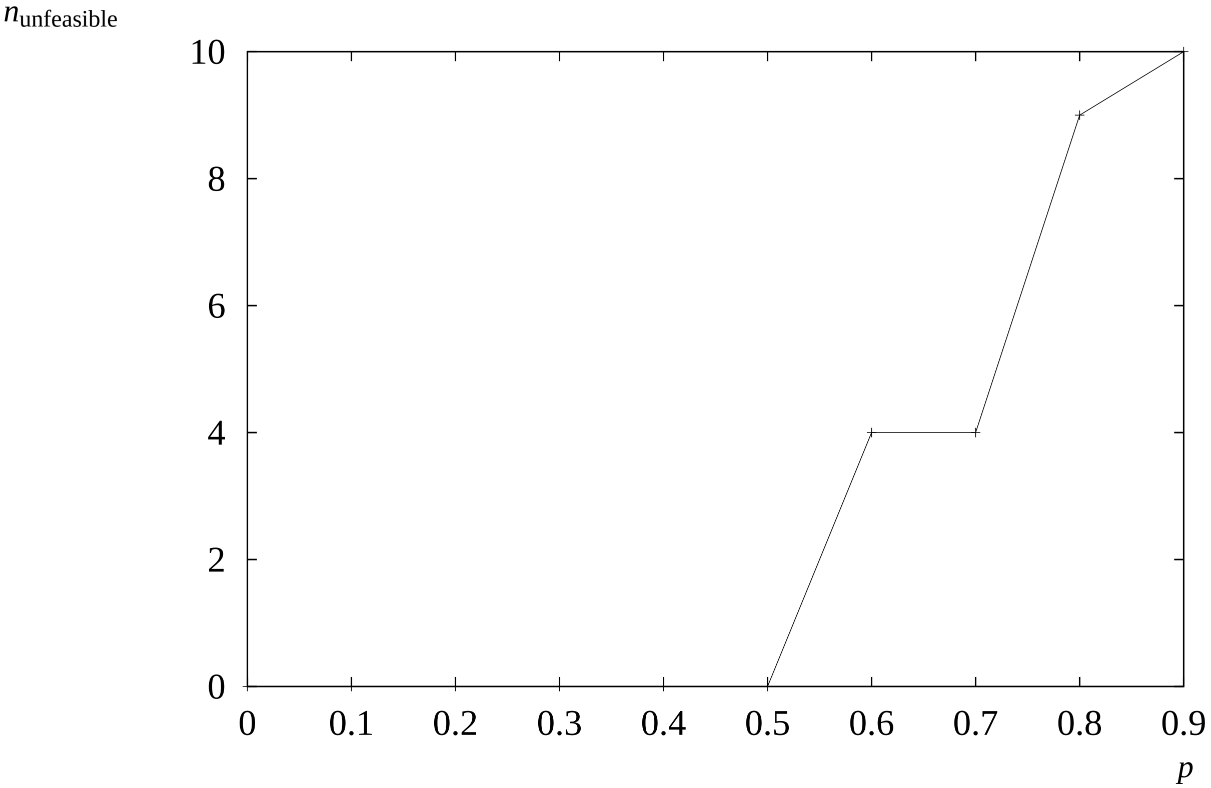
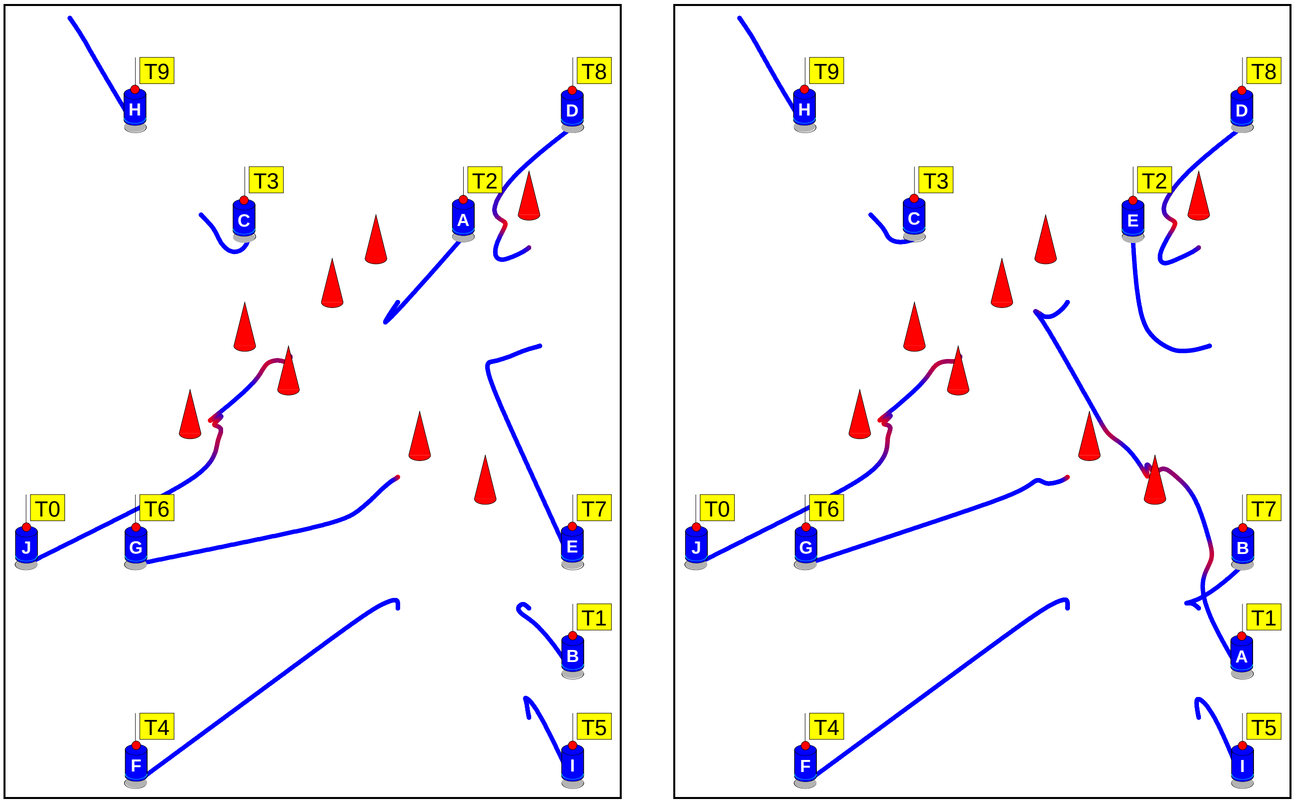


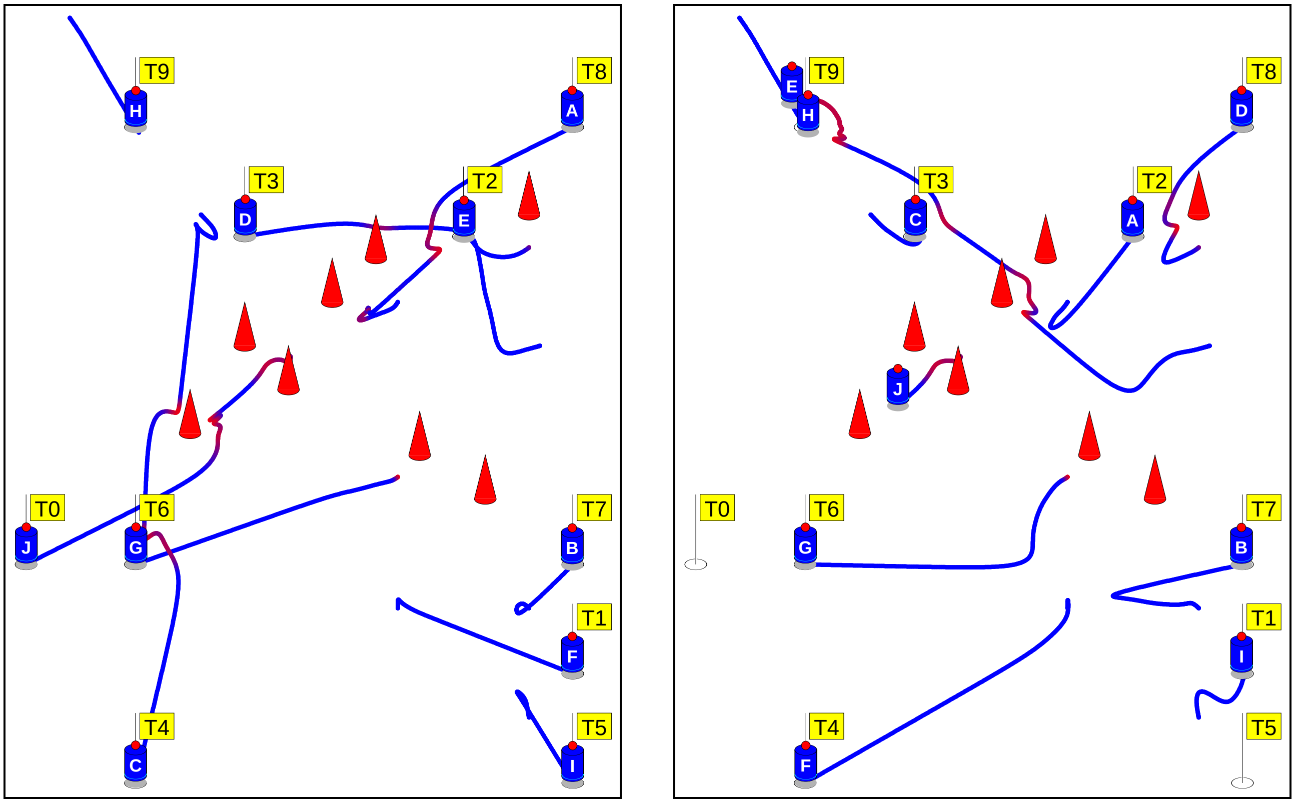
Figure 5.5: Number $n_{\text{unfeasible}}$ of unfeasible solutions in dependence of the package loss probability p. Unfeasible solutions are assignments that allow more than one robots to select the same target.



a. data loss probability $p = 0.0$

b. data loss probability $p = 0.2$

Figure 5.6: Simulation runs for different values of the data loss probability p: a. no message loss, the system finds its optimal assignment, at $t = 27.1$ each robot has reached a target; b. 20% of the messages between robots got lost, the elapsed time is 34.0;



c. data loss probability $p = 0.4$ d. data loss probability $p = 0.7$

Figure 5.7: c. 40% package loss, $t = 42.0$ and d. 70%, $t = 69.9$.

Table 5.1: Simulation results for different probabilities p for message loss. The table shows the elapsed time t_{total} until every robot of the system reached a target, and the final target assignment. Non-feasible assignments are written in brackets.

p	#	t_{total}	Assignment A–J
0	1	27.1	2 1 3 8 7 4 6 9 5 0
	2	27.1	2 1 3 8 7 4 6 9 5 0
	3	27.1	2 1 3 8 7 4 6 9 5 0
	4	27.1	2 1 3 8 7 4 6 9 5 0
	5	27.1	2 1 3 8 7 4 6 9 5 0
	6	27.1	2 1 3 8 7 4 6 9 5 0
	7	27.1	2 1 3 8 7 4 6 9 5 0
	8	27.1	2 1 3 8 7 4 6 9 5 0
	9	27.1	2 1 3 8 7 4 6 9 5 0
	10	27.1	2 1 3 8 7 4 6 9 5 0
0.1	1	29.3	2 1 3 8 7 4 6 9 5 0
	2	26.8	2 7 3 8 1 4 6 9 5 0
	3	28.0	2 1 3 8 7 4 6 9 5 0
	4	32.5	1 7 3 8 2 4 6 9 5 0
	5	26.6	2 1 9 8 7 4 6 3 5 0
	6	21.6	2 1 3 8 7 4 0 9 5 6
	7	28.5	2 1 3 8 7 4 6 9 5 0
	8	27.6	2 1 3 8 7 4 0 9 5 6
	9	28.2	2 1 3 8 7 4 6 9 5 0
	10	22.7	2 1 3 8 7 4 0 9 5 6
0.2	1	24.3	2 1 0 8 7 4 6 9 5 3
	2	23.9	2 1 3 8 7 4 0 9 5 6
	3	27.2	2 1 3 8 7 4 6 9 5 0
	4	30.6	7 1 9 8 2 4 6 3 5 0
	5	22.0	2 1 3 8 7 4 0 9 5 6
	6	27.0	2 1 3 8 7 4 6 9 5 0
	7	34.0	1 7 3 8 2 4 6 9 5 0
	8	21.3	2 1 3 8 7 4 0 9 5 6
	9	26.5	2 6 0 8 7 4 1 9 5 3
	10	24.1	2 7 3 8 1 4 0 9 5 6
0.3	1	26.1	0 1 3 8 2 4 7 9 5 6
	2	26.4	6 7 0 8 1 4 2 9 5 3
	3	27.3	2 1 9 8 7 4 6 3 5 0
	4	42.5	2 0 3 1 8 4 7 9 5 6
	5	24.5	4 1 3 8 7 6 5 9 0 2
	6	28.6	2 1 3 8 9 4 7 6 5 0
	7	38.8	2 1 3 8 0 4 7 9 5 6
	8	27.7	3 1 6 8 7 4 2 9 5 0
	9	28.5	3 1 0 8 2 4 7 9 5 6
	10	32.5	2 7 3 8 0 1 4 9 5 6
0.4	1	30.0	2 1 3 8 5 4 6 9 7 0
	2	42.0	8 7 4 3 2 1 6 9 5 0
	3	29.1	2 1 3 8 7 4 6 9 5 0
	4	37.2	3 1 0 8 4 2 6 9 5 7
	5	25.8	2 1 3 8 1 7 6 9 5 0
	6	27.9	2 1 0 8 6 4 7 9 5 3
	7	44.5	6 1 9 8 2 4 0 7 5 3
	8	29.2	0 7 3 8 1 4 6 9 5 2
	9	27.2	2 1 3 8 7 4 6 9 5 0
	10	49.3	9 1 3 8 2 4 6 7 5 0
0.5	1	36.7	1 4 3 2 8 6 7 9 5 0
	2	27.6	2 6 3 8 7 4 1 9 5 0
	3	28.1	7 1 3 8 6 4 2 9 5 0
	4	26.8	6 5 3 8 7 4 2 9 1 0
	5	39.0	2 7 0 8 4 1 6 9 5 3
	6	28.0	3 1 0 8 2 5 6 9 7 4
	7	26.2	3 1 9 8 7 4 6 2 5 0
	8	32.8	2 0 3 8 7 4 6 9 5 1
	9	52.0	8 7 3 0 2 4 9 1 5 6
	10	29.6	9 1 0 8 2 4 7 6 5 3
0.6	1	42.9	2 1 9 8 4 5 3 7 6 0
	2	29.5	7 1 2 8 6 4 0 9 5 3
	3	27.8	[7 1 0 8 9 4 2 8 5 3]
	4	36.7	1 4 3 8 2 0 7 9 5 6
	5	43.0	[7 5 4 2 8 3 6 9 8 0]
	6	33.4	[8 7 3 2 1 4 4 9 5 0]
	7	39.2	2 7 6 8 4 5 3 9 1 0
	8	29.5	[2 1 3 8 7 4 7 0 5 9]
	9	34.9	2 1 3 2 5 4 0 9 7 6
	10	34.9	9 7 3 8 2 1 4 6 5 0
0.7	1	52.1	7 4 9 8 3 1 0 6 5 2
	2	28.9	1 2 3 8 7 6 4 9 5 0
	3	31.9	8 1 0 6 7 4 2 9 5 3
	4	30.2	[7 8 8 3 2 4 1 9 5 6]
	5	38.5	3 7 0 8 2 6 4 9 5 1
	6	35.5	1 7 3 2 8 0 4 9 5 6
	7	33.4	[1 7 3 8 2 4 0 9 5 9]
	8	27.8	0 7 9 8 1 4 6 9 5 3
	9	69.9	[2 7 3 8 9 4 6 9 1 0]
	10	29.9	[8 1 9 3 7 4 6 0 5 3]
0.8	1	37.1	[3 0 9 1 2 4 8 6 5 9]
	2	23.7	[3 7 9 8 2 4 1 9 5 9]
	3	31.3	[2 1 3 8 1 4 0 9 5 6]
	4	38.8	[6 5 0 8 7 4 9 9 1 3]
	5	33.3	2 1 9 7 0 4 6 3 5 8
	6	39.4	[3 7 4 9 8 5 0 9 5 6]
	7	45.4	[5 1 8 8 2 3 0 9 2 9]
	8	65.5	[9 5 3 8 8 0 7 9 4 6]
	9	46.8	[5 1 3 9 8 7 6 9 5 4]
	10	33.5	[7 1 3 8 9 4 6 0 5 3]
0.9	1	34.2	[2 1 9 8 0 4 0 6 5 3]
	2	30.1	[2 1 8 8 7 4 4 9 5 0]
	3	48.4	[2 7 9 8 5 1 7 9 5 4]
	4	45.6	[2 1 9 7 8 4 7 9 5 0]
	5	34.4	[4 1 9 9 8 5 2 9 5 3]
	6	38.9	[2 7 9 8 5 6 4 7 5 0]
	7	30.2	[2 1 9 8 8 4 0 9 5 3]
	8	26.0	[2 4 9 8 2 4 7 9 5 3]
	9	22.1	[9 1 3 8 7 4 4 9 5 6]
	10	47.3	[4 1 2 8 8 5 7 9 7 3]

The success of the control system depends on the intervals between updates of the elements of the preference matrix (ξ_{ij}). Both factors, the frequency of data messages and the probability p of package losses contribute to the quality of the task assignment.

Frequent package losses in the communication channel can be compensated by shorter intervals between the broadcast of updates. In a technical implementation, robots should not send messages at a fixed rate, but rather randomly change the time between two transmissions. This can avoid periodical peeks in the data traffic, which yield to increased numbers of package collisions (on an wireless ethernet).

Chapter 6

Distributed Sensor Network

The concept of self-organized task assignment can also be applied for autonomous sensors. Like robots, these sensors are equipped with sensing devices, computation capabilities and some means of communication. They usually stay at the location they were deployed at.

A new paradigm in ground surveillance, e.g., consists of swarms of autonomous internetted sensors that can be used for target localization and environmental monitoring. The individual component is an inexpensive device containing multiple sensor types, a processor and wireless communication hardware, and a battery for power supply [1].

Scattered over a certain region, these devices are able to detect the direction or proximity of moving objects. Various triangulation methods are discussed in [6] for electro-magnetic sensors which can also be applied for acoustic and other sensors [3] One of the most limiting factor of the devices is the battery supply. In order to conserve power, these units should be able to adjust their activities to the current situations. Energy consuming signal processing should only be performed if the quality of the raw sensor data promises a significant improvement to the localization results.

The self-organized control system allows the devices to select the algorithms' complexity which balances the requirements for good localization performance and energy conservation. The devices make their selection autonomously, based on their own sensory data, information that they receive from other devices in the region, and the amount of energy they have left.

Unattended Grounds Sensors (UGS), e.g., may be able obtains a Direction of Arrival (DOA) estimate by an array of acoustic sensors, cameras or other type of detectors. When the UGS begins to estimate the DOA, it is able to determine a $q-$factor that represents the expected quality of the received signal. If the $q-$factor is high, the UGS continues processing to determine and communicate the DOA. Otherwise, the sensor uses the value of the $q-$factor to obtain a rough estimate of the range to the target.

We use the statistics for the $q-$factor as a function of Signal to Noise Ration (SNR) for a narrow-band acoustic signal at a frequency of 200Hz [4]. given target signal $s(t)$. The statistics were generated out of 1,024 simulation runs. Figure 6.1 shows first and second order statistics of the $q-$factor as a function of SNR. The sampled can be fitted to approximate the dependency of the statistics on SNR using analytic expressions. These analytic expressions are

$$\mathcal{E}\{q\} \approx 2.765 \times 10^4 \frac{(\mathrm{snr})^2}{110 + (\mathrm{snr})^2}, \tag{6.1}$$

and

$$\mathrm{STD}\{q\} \approx 812 e^{-2.6|\log_{10}(\mathrm{snr})-1|^{1.5}}. \tag{6.2}$$

Further, we assume the R^2 energy dissipation law where we set the nominal SNR at one meter from the target to be 60db, i.e., 10^6. This allows to express the statistics of q as a function of range.

6.1 Decision Making

It is critical that the sensors use their energy resources as efficiently as possible. For example, in case they are not in a reasonable range of a target, they should not proceed the signal processing steps, because the quality of the localization would not justify the invested amount of energy. Furthermore, if a number of other sensors are close so that they produce reliable data for the localization, additional information will not significantly improve the accuracy of the targets' estimated positions. In such a case, sensors can suspend, and conserve their resources.

The decision whether a sensor should actively participate in the surveillance process has to be made in consideration with the state of all the other sensors. This can be achieved by the management layer that oversees the sensor network. In our approach we use dynamical systems theory to implement a distributed problem-solver. The concept is based on a special family of dynamical systems, called selection equations.

The sensors share their estimated DOA with each other by broadcasting messages which include their measurements and the sensors preference for performing the task. The format of the messages can be of the format:

- Field 1: Sensor Identification,

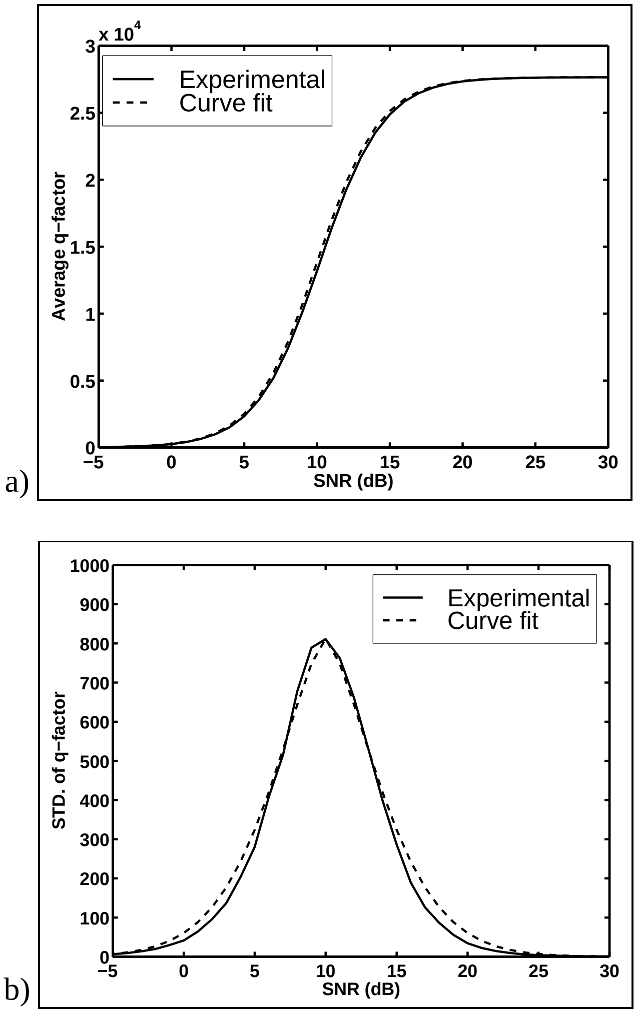


Figure 6.1: Measured statistics for time delay and $q-$factor estimates as a function of SNR: (a) mean of $q-$factor and (b) standard deviation of $q-$factor.

- Field 2: Sensor Position,
- Field 3: Estimated DOA,
- Field 4: Preference $\xi \in [0,1]$.

Each sensor keeps track of the message received from other sensors. Hereby, the exchange of messages does not require to be synchronous, nor does the entire system collapse if some messages get lost. The tolerance and resistivity of systems like this with respect to communication failures is similar to what has been discussed in section 5.1.

The sensors select their mode of operation based on the their preference coefficient ξ_i which includes the q-factor of the last measurements and the information gathered from other sensors. The development of the preferences is given by the dynamic equation

$$\frac{d}{dt}\xi_i = \xi_i \left(\alpha_i^2 - \beta \sum_{i' \neq i} \xi_{i'}^2 \right), \tag{6.3}$$

with $\alpha_i = \frac{<q>_i}{q_{\max}}$ and the scaling factor β, which weights the contribution of the other sensors. For constant conditions, the ξ will converge to a value between 0 and 1 in the time limit. (See Fig. 6.2). The dynamics of the system allow only a few values to grow while the majority of the parameters will be close to zero.

The preference value ξ is a result of two competing terms, the sensors own ability to perform the task, and the preferences of the other sensors. The dynamics of this value follows the dynamical behavior of the $q-$factor. However, even small $q-$factors can result in strong preferences when there no other suitable sensors around. On the other side, competition among the sensors is introduced to limit the number of active sensors at a time.

The sensors select their mode of operation based on their preference coefficient ξ:

$$\text{mode}_i = \begin{cases} \textbf{active} & \text{if} \quad \xi_i > \Theta \\ \textbf{observing} & \text{if} \quad \theta <= \xi_i <= \Theta \\ \textbf{suspended} & \text{if} \quad \theta > \xi_i \end{cases}, \tag{6.4}$$

where an **active** sensor performs DOA and contributes to the target localization process, an **observing** sensor stops its signal processing after determining the $q-$factor, and a **suspended** sensor turns into sleep mode for a given time for maximum energy conservation.

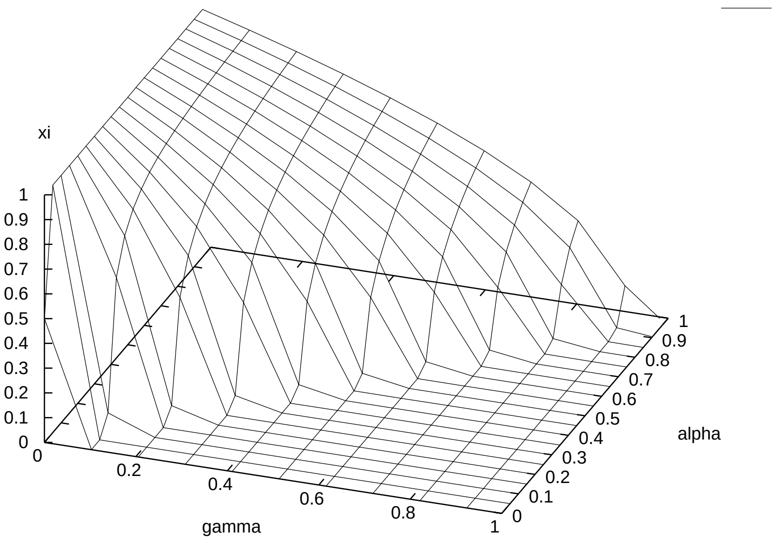


Figure 6.2: Hyperplane of stable solutions in the time limit of (6.3) for different values of α and $\gamma = \sum_{i' \neq i} \xi_{i'}^2$. For values $\gamma > 1$ the stable solution is always $\xi_i = 0$.

6.2 Simulation Results

The simulation shows 100 sensors that are randomly scattered within an area of 4 square kilometers. That is equivalent to 2.5×10^{-5} sensors per square meter in average.

We create a distribution of $q-$factors as a function of the distances between sensors and targets according to (6.1) and (6.2). Based on this distribution, a random number generator produces the simulated $q-$factor for each sensor.

With the scaling factor $\beta = 0.1$ in (6.3) and the threshold $\Theta = 0.7$ in (6.4) the average number of active sensors is between 4 and 5. This depends on the density of the sensors in a particular area. A scaling factor of $\beta = 0.05$ would allow 9 to 10 sensors in average to become active.

Figure 6.3 shows a sequence of snapshots of the simulation. The target, depicted in shape of an arrow, moves from the lower left corner of the picture to the right. At certain times, more than the desired four or five sensors are active. (See Fig. 6.3b-d.) However, some of those preferences drop quickly, due to the competition term in (6.3).

We have shown, that acoustic sensors can indeed evaluate the quality of their data, and that this can be used as criteria for sensor management. The proposed management system is a distributed decision process with each sensors computing a segment of the solution. The information, necessary

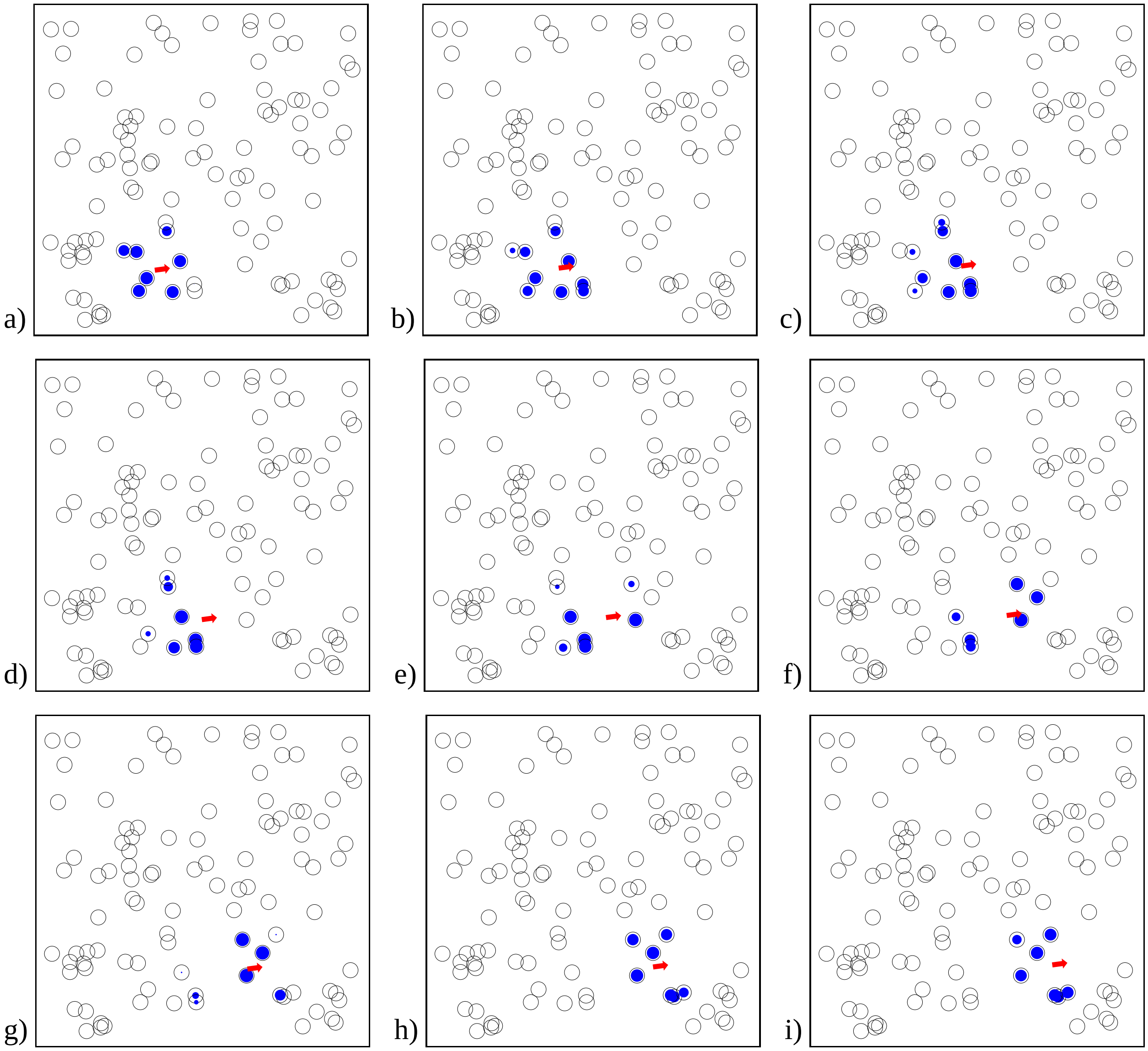


Figure 6.3: Snapshots of the simulation in time increments of 50 time steps (seconds). From a) $t = 850$ to i) $t = 1250$. The sensors are depicted as outlined circles. The size of the solid core corresponds to the sensors preference value ξ_i. The arrow indicates the target's position and velocity. (The size of the symbols is not in proportion to the scale of the plot.)

for this decision process adds only a small portion to the amount of data, that sensors exchange for target localization.

Chapter 7

Conclusion and Outlook

The objectives of this project were to develop a preliminary mathematical and simulation framework for cooperative decision making and assess its feasibility within the context of autonomous mobile robots and distributed sensors.

In particular, the results have demonstrated the viability of using the coupled selection equations to implement robust and dynamic decision making. When applied to a sample scenario of task assignment between multiple mobile robotic agents the formulation was able to demonstrate robustness to device failures, resistance to communication loss, as well as the ability to form multi-dimensional teams in a real time manner. These very encouraging results suggest a more comprehensive approach that can be used to support this Phase II proposal.

Coupled Selection Equations are a family of coupled differential equations with the specific characteristic that every element will converge to either 0 or one in the time limit. We can use the dynamics to describe combinatorial problems, such as selection processes. Boundary conditions, such as the requirement of a permutation, e.g. matching every element form one population with an element from the a second one, can be implemented in the design of the functions.

So far CSE's are known only for a few types of problems, which are presented here report. E.g. assignment problems with varying numbers of agents require specially designed equations. Future research will target different types of boundary conditions, and seek to design CSE's that satisfy these.

Bibliography

[1] Loren P. Clare, Gregory J. Pottie, and Jonathan R. Agre. Self-organizing distributed sensor networks. In *Proc. of SPIE,* volume 3713, pages 229–237, April 1999.

[2] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51:4282–4286, 1995.

[3] Lance M. Kaplan, Péter Molnár, and Qiang Le. Bearings-only target localization for an acoustical unattended ground sensor network. In preparation to be submitted to *SPIE AeroSense 2001*.

[4] P. Molnár, E. J. Lockett, and L. M. Kaplan. Self-organized task assignment for distributed sensors. In *SPIE*.

[5] P. Molnár and J. Starke. Control of distributed autonomous robotic systems using principles of pattern formation in nature and pedestrian behaviour. *IEEE Trans. on SMC: Part B*, 2001.

[6] Yaakov Oshman and Pavel Davidson. Optimization of observer trajectories for bearings-only target localization. *IEEE Trans. on Aerospace and Electronic Systems*, 35(3):892–902, July 1999.

[7] J. Starke. *Kombinatorische Optimierung auf der Basis gekoppelter Selektionsgleichungen*. PhD thesis, Universität Stuttgart, Verlag Shaker, Aachen, 1997.

[8] 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.

[9] J. Starke, M. Schanz, and H. Haken. Self-organized behaviour of distributed autonomous mobile robotic systems by pattern formation principles. In T. Lueth, R. Dillmann, P. Dario, and H. Wörn, editors, *Distributed Autonomous Robotic Systems 3*, pages 89 – 100. Springer Verlag, Heidelberg, Berlin, New York, 1998.

[10] Thomas W. Vaneck. A system of mesoscale biomimetic roboswimmers for exploration and search of life of europa. Technical report, Physical Sciences, Inc., 2000.