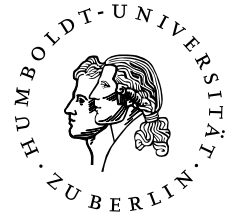


HUMBOLDT-UNIVERSITÄT ZU BERLIN
MATHEMATISCHE-NATURWISSENSCHAFTLICHE FAKULTÄT
INSTITUT FÜR INFORMATIK



Self-healing in swarm robotic systems

seminar work

by: Vincent Becker and Mark Schatz

Contents

1	Abstract	4
2	deinemudda	5
2.1	hart	5
3	robotbeacon	5
4	Self-Healing Algorithms	6
4.1	simple self-healing algorithms	6
4.2	Single-Nearest-Charger Algorithm	7
4.3	Chared-Nearest-Charger Algorithm	8

1 Abstract

swarm robotic systems and how we approach the problem of swarm taxis do it, just do it, make your dreams come true, YES you CAN!

2 deinemudda

2.1 hart

3 robotbeacon

4 Self-Healing Algorithms

We discussed the problem of anchoring and tried to solve this problem with the approach of swarm taxis algorithms. The swarm beacon taxis, tried to tell us the problem of, what can happen if we don't solve the problem of dead robots, who still sending signals.

4.1 simple self-healing algorithms

One approach is the idea of a Single-Nearest Charger. The idea behind this approach is that robots near the failed robot, send informations about their level of energy and distance away from the failed robot.

Single-Nearest-Charger Algorithm

We want to conclude, how robots will act, if one robot had failed. So if one robot has some malfunction and cannot move anymore, that would not be a problem for our case, because the robots would handle this failed robot as an object. Our Problem here is, that the robot can still send signals to other robots. This signals are position data and data about the current state. All robots work with the same algorithms, so everyone is trying to stay near to him, anchoring occurs.

(insert picture here, to demonstrate, simulation)

The Hypotheses behind this, is built as followed:

The use of single nearest charger mechanism (M2) for swarm beacon taxis does not improve the ability of the robots in the systems to reach the beacon when compared with M1 when more than two faulty robots are introduced to the systems

In contrast of the form of 'central sharing', robots must be able to distribute the collective energy resources owned by the group member. This means that we need a simple solution to share the energy between the robots. Here called the single nearest charger algorithm.

We assume the simplest rules, which are the position of the robot in the environment and the limit of energy transfer. Robots can donor or receive energy from only one robot at the same time. We say that the nearest robot acts as the donor if one failed robot is sending signals.

The issue about this idea, is the fact, that we can not control the amount of energy that is transferred from donor to receiver or vice versa. So the problem will be, that one failed robot get half of the energy of one donor, the donor now needs energy and a donor for himself. That brings another anchoring effect.

4.2 Single-Nearest-Charger Algorithm

Algorithm 1 Single-Nearest-Charger Algorithm

```
1: procedure DONOR ENERGY
2:   begin
3:   Deployment: robots are deployed in the environment
4:   repeat
5:     Random movement of the robot in the environment Signal propagation:
     Faulty robots will emit faulty signal Rescue: One of the healthy robots with the
     nearest distance (earliest arriving robot) will perform protection and rescue Repair:
     Sharing of energy between faulty and healthy robots according to algorithm 2
6:   until forever
```

due to problems with the recharge of the failing robots by docking on them (3 robs, for 1 failed rob) we need an solution, to not interfere with the other helpers so a constant energy flow can happen.

Algorithm 2 Algorithm for containment and repair for single nearest charger algorithm

```
1: procedure
2:   begin
3:   Evaluate  $pos_{self}(t)$ 
4:   Send  $pos_{self}(t)$  to peers
5:   Receive  $egy_{peer}(t - x)$  and  $pos_{self}(t - x)$  from peers
6:   for all  $egy_{peer}(t - x)$  do
7:     Evaluate  $egy_{peer}(t - x)$ 
8:     if  $egy_{peer}(t - x) < egy_{min}$  then
9:       Evaluate  $egy_{peer}(t - x)$ 
10:      Sort  $pos_{peer}(t - x)$  in ascending order
11:      Move to nearest  $pos_{peer}(t - x)$ 
12:     else
13:       Do not move to  $pos_{peer}(t - x)$ 
14:     end
15:   end
```

4.3 Chared-Nearest-Charger Algorithm

an approach to avoid the failure, that one robot donor to one robot and both will fail. Is the idea of Chared-Charging. The failed robot will send out information, about his location, in detail the distance between it and the failed robot and the actual energy level. If the energy level is above the threshold and the donors are in range of the failed one, then they will "help".

Visualisation