

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	3
2	Introduction	3
2.1	Swarm Robotic Systems	3
2.2	Swarm Taxis and Self-Healing	3
2.2.1	A Simple Task - Swarm Taxis	3
2.2.2	Achieving Swarm Taxis Mission	3
2.2.3	Coping Partial Robot Failure	4
3	Swarm Taxis Algorithms	4
3.0.1	α -Algorithm	4
3.0.2	β -Algorithm	4
3.0.3	ω -Algorithm	4
4	Self-Healing Algorithms	4
4.1	Single-Nearest-Charger Algorithm	5
4.1.1	Shared-Nearest-Charger Algorithm	7

1 Abstract

In our seminar work we are dealing with a way to give swarm robotic systems some more robustness. There are approaches to optimize the algorithms for their different tasks. The one we are coping with is a more general one. Self-healing can be adopted for a wider field of missions. Nevertheless we are coping with swarm taxis. Based on the paper "An Immune-Inspired Swarm Aggregation Algorithm for Self-Healing Swarm Robotic Systems" by Timmis, Ismail, Bjercknes and Winfield we are describing solutions to robot failures.

2 Introduction

2.1 Swarm Robotic Systems

Swarm behaviour is the decentralised coordination of many self-propelled entities. They got simple rules for their local interactions and communication which lead to an emergent behaviour (@see Wikipedia). When talking about swarm robotic systems we got a number of at least 5 robots with simple algorithms which only got their local knowledge. Those systems need to fulfill their task in a robust, flexible and scalable way.

2.2 Swarm Taxis and Self-Healing

One very simple tasks for robot swarms is moving towards a beacon, called swarm taxis. We will focus on swarms build to achieve this task. Timmis and Co worked with a swarm of 10 e-Puck robots.

2.2.1 A Simple Task - Swarm Taxis

E-puck robots are equipped with an omnidirectional light sensor, 3 proximity sensors, a short range, omnidirectional radio device and some LEDs. This is one possible set of tools for swarm taxis robots which need to go towards a light emitting beacon while avoiding obstacles on their way.

In their setting Timmis and Co identified three failure modes for a certain robot:

no failure everything works fine

total failure the robot completely stops and just behaves like an obstacle

partial failure the robot stops moving (e.g. due to low energy) but still sends signals

2.2.2 Achieving Swarm Taxis Mission

reaching target despite different robot failures
is called self-repair in terms of the swarm
there is no repair of certain robots

2.2.3 Coping Partial Robot Failure

reaching target despite partial robot failures
is called self-healing in terms of swarm robotics
meaning the repair of partially failed robots

3 Swarm Taxis Algorithms

3.0.1 α -Algorithm

α -threshold
limit of minimum neighbours
when less, reconnect
180°-turn
when reconnected, random heading

3.0.2 β -Algorithm

β -threshold
limit of minimum neighbors still having connection to a lost robot
when less, reconnect
same as in α -Algorithm

3.0.3 ω -Algorithm

ω -threshold
limit maximum of ticks since last avoidance behavior
when more, reconnect
turn to estimated swarm center
when reconnected, random heading

to calculate swarm center replaced wireless connection by timer and ring of IR-emitters

swarm even more connected than with β -Algorithm

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 Single-Nearest-Charger Algorithm

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.

We want to conclude, how robots will act, if one robot had failed. So if one robot has some malfunction and cannot move any further, 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. These signals are position data and data about the current state. All robots work with the same algorithm, 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 calls 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.

Algorithm 1 Single-Nearest-Charger Algorithm

```
1: procedure DONOR ENERGY
2:   Deployment: robots are deployed in the environment
3:   repeat
4:     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
5:   until forever
```

Due to problems with the recharge of the failing robots by docking on them (3 robots,

for 1 failed robot) we need a 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

```

4.1.1 Shared-Nearest-Charger Algorithm

An approach to avoid the failure, that one robot donor to one robot and both will fail. Is the idea of Shared-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 level is ok and one robot is closer to the failed robot than the others