Orientation in a Trail Network by Exploiting its Geometry for Swarm Robotics

Heiko Hamann, Marc Szymanski, and Heinz Wörn
Institute for Process Control and Robotics
Universität Karlsruhe
76128 Karlsruhe, Germany
Email: {hamann, szymanski, woern}@ira.uka.de

***Abstract*— Two control algorithms for a swarm robot are presented that enable it to orientate itself by using information from the geometry of trail bifurcations within a trail network. The development of these algorithms was inspired by the behavior of Pharaoh's ants as reported by Jackson et al. [4]. The performance of the robot is analyzed in a large number of embodied experiments with different bifurcation angles. The reactive behavior implemented by simple rules is sufficient to accomplish this task using a robot of limited capabilities. The frequency of correct reorientations is maximized when the trail bifurcation angle is 60 degrees, as found in natural networks.**

I. Introduction

The hardware used in swarm robotics is in general characterized by its boundedness concerning computational power, memory, energy, communication abilities, accuracy, diversity and number of sensors and actuators [2], [8]. Despite all this imprecision, useful and complex behavior can still be generated by using rules that are both probabilistic and simple. Nature, especially social insects, served as the biggest source of inspiration for such behaviors.

The work of this paper was inspired by the behavior of ants that extract polarity information out of the geometry of their trail networks [3], [4]. Our aim is not primarily a direct application of the presented algorithms in swarm robotics but a proof of concept. Starting from an observed natural phenomenon, that cannot be explained by biologists yet, we tried to mimic at an abstract level basic concepts of the environment, the ant's sensors, and its behavior. The most important part was, however, unknown which is the algorithm causing this observed behavior. The only clue that was available, was that a solution exists. However, it is unknown whether ants have other capabilities involved in this trail following experiment that the used robot does not have. The effectiveness of this approach might be a benefit for both communities, biologists and swarm roboticists.

In the following the robot is supposed to move on trails that enable it to orientate itself. The emergence of this trail system is not addressed in this paper and could be the result of a self-organized process as it is observed, for example, in social insects: "Most trail-laying ants produce complex trail networks branching throughout their foraging environment" [4]. A schematic representation of such a trail network is given in Fig. 1. Thus, in the ideal case we have a tree structure and every bifurcation has two branches leading to the foraging area and one leading to the nest entrance. In this work the robot behavior at a single bifurcation of this kind will be investigated as it was done by Jackson et al. [4]. A probabilistic reorientation procedure is sufficient since there are usually several consecutive bifurcations, i.e. there are several chances to reorientate.

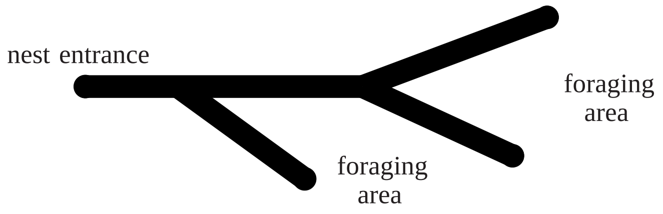


Fig. 1. Schematic representation of a trail network.

Besides the inspiration by nature the algorithms were found out in a traditional trial-and-error fashion because of the lack of other applicable top-down approaches. However, this is unendurable and first steps to more sophisticated top-down approaches have been proposed recently [5]–[7]. The use of online learning techniques for such a swarm robot would have been a research topic on its own, although an interesting one. The option of developing or learning a controller in simulation was consciously ignored because there are too many important subtleties that are complicated to simulate in this scenario, e.g. highly variable turning angles or intrinsic inhomogeneities in the trail on which the found robot behavior relies as discussed in section III.

Please note that the purpose of this work is not to produce a realistic model of the Pharaoh's ant. Whether the reorientation behavior of this ant is reactive or makes use of memory is an open question. In a previous biological work the reorientation behavior of the Pharaoh's ant is also modeled with agents acting reactively [3].

In the next section we briefly describe the swarm robot "Jasmine", which is a respectable byproduct of the European I-SWARM project, while the main aim of this project is to develop a robot in sizes of one magnitude below Jasmine's (about $3\times3\times3$ mm^3 compared to about $3\times3\times3$ cm^3). In section III we will describe the algorithms in detail and in section IV we present the results.

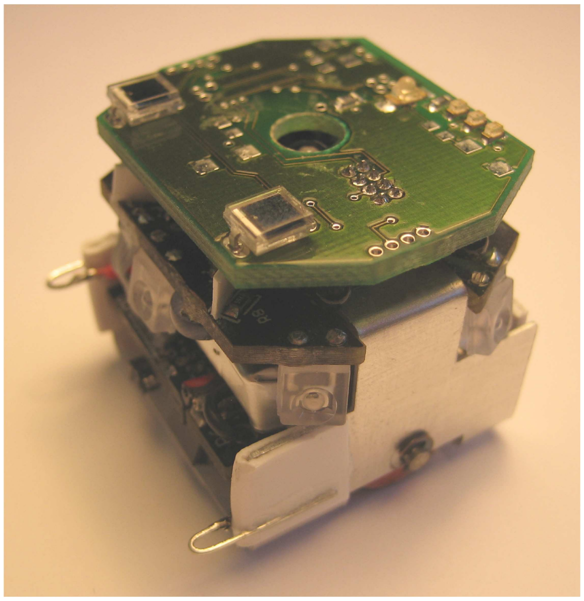


Fig. 2. The Jasmine swarm robot.

II. Materials and Methods

A. The Swarm Robot "Jasmine"

The swarm robot Jasmine (see Fig. 2 and [1]) was developed especially for swarm robot research. Despite its small size of about $30 \times 30 \times 30$ mm^3, it has good local communication abilities and a far distance scanning and distance measuring sensor. The good communication abilities result from six infra-red sensors and emitters arranged around the robot with a displacement of 60 degrees. These sensors are also used for short distance measurements. The far distance measuring sensor is hooked to the front of the robot. Two differentially driven wheels give this micro-robot a high manoeuvrability at a high speed. Generation two, that was used in this paper, has an Atmel Mega 8 micro-controller with 1 Kbyte RAM and 8 Kbyte Flash. Two LiPo battery packs provide 7.2 V for up to two hours of motion.

Generation three of the robot, that was not used here, has an Atmel Mega 168 micro-controller with 1 Kbyte RAM and 16 Kbyte Flash. Now a single LiPo battery pack is sufficient with the same endurance and optical encoders are available that allow odometric measurements in the mm-range.

Different from some other swarm robots Jasmine supports only local communication. Long distance communication via radio frequency is not implemented and does not correspond with the views of the construction team about swarm robot capabilities.

In this work we used an optional sensor board with two photodiodes that can be fixed on top of the robot. Here the board just sends permanently an averaged value of the measured brightness encoded in one byte per sensor to the robot's controller. These values are the average of 50 sensor measurements. In the experiments here, a video projector that projects the trails onto the arena was the only light source. The

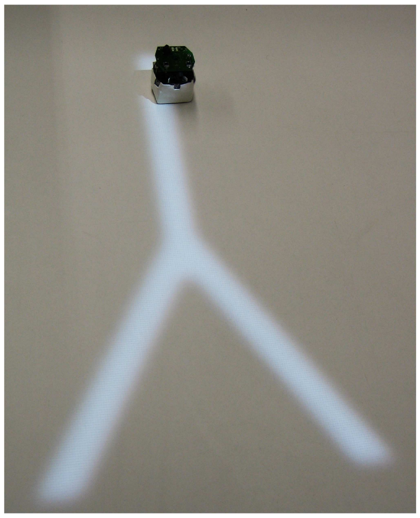


Fig. 3. The Robot Jasmine and the projection of the trail.

sensor board gives values of the full interval from 0 (dark) to 255 (bright) depending on the brightness of the part of the projected image that is shed on the photodiodes. In the following the currently measured value of the left and the right photodiode will be referenced by L and R.

Because of its design for low cost and the artificial restriction to use no long distance communication the Jasmine robot might seem to be below the technological state of the art. However, this makes it a suitable testbed for even much smaller robots because besides the locomotion there are no conceptual problems for miniaturization.

B. Motion Description Language Two Extended ($\mathbf{MDL2}\epsilon$)

To program the robot we used $\mathbf{MDL2}\epsilon$ as it is presented in this proceedings [9]. The Programming of a swarm robot can be done very efficiently using $\mathbf{MDL2}\epsilon$ if all needed atoms (smallest abstract component in an $\mathbf{MDL2}\epsilon$-controller) have been already implemented. An $\mathbf{MDL2}\epsilon$ program is usually very compact and concise, which makes quick changes easy. The implementation work of the below algorithms was definitely sped up by using this programming framework.

C. Experimental Setup

The setup for the experiments is simple. The video projector that is mounted about 2.5 m above the arena projects the trail in white on the floor (see Fig. 3). At positions in the arena that are not directly below the projector the light beams down in angles smaller than 90 degrees which is displayed by the shadow of the robot and its displacement to the projection on the ground in Fig. 3. The geometry of the projected trail bifurcation for $\alpha = 90°$ is shown in Fig. 4 and the key situations of the robot approaching a bifurcation are shown in Fig. 5.

III. Two Algorithms for the Orientation in Trail Networks

In this section we present two algorithms, one that enables the robot to find the nest (we will refer to this as the *nest algorithm*) and one to find the food places (*food algorithm*) in

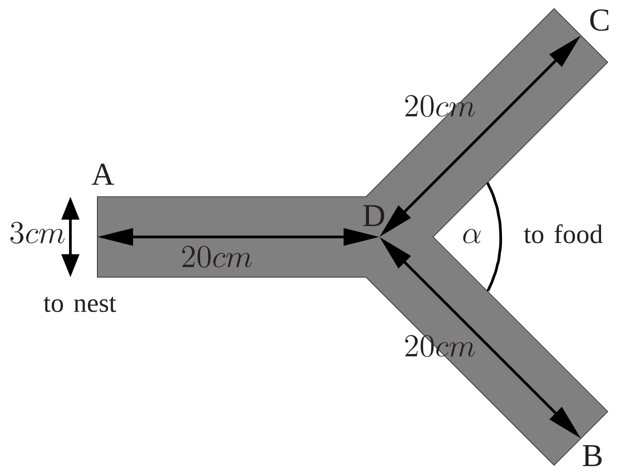


Fig. 4. Geometry of a trail bifurcation.

a trail network of trail bifurcations of given angles. In an implementation of a transportation scenario the robot could switch between these two algorithms using simple $\mathbf{MDL2}\epsilon$ mechanisms.

The trail following behavior is based on the input of the two photodiodes only. They are used to basically keep the robot on the trail and to identify the bifurcations. Three different events are relevant: First, at one sensor it is much brighter than at the other one: $|L - R| > \delta_{\text{diff}}$, for some threshold δ_{diff}. See Fig. 5(b) for a typical situation in the experiment where this holds. Second, at both sensors it is very dark: $(R < \delta_{\text{dark}}) \wedge (L < \delta_{\text{dark}})$, for some value $\delta_{\text{dark}} \ll 255$, see Fig. 5(c). Third, at both sensors it is very bright: $(R > \delta_{\text{bright}}) \wedge (L > \delta_{\text{bright}})$, for some value $\delta_{\text{bright}} > \delta_{\text{dark}}$, see Fig. 5(d). For simple straight-ahead driving on the trail it would be sufficient to rotate to the brighter side (in the following this will be called *correction turn*) if at one sensor it is much brighter than at the other one ($L > R + \delta_{\text{diff}}$ or $R > L + \delta_{\text{diff}}$). To reorientate correctly at the bifurcations in the network more sophisticated mechanisms are needed which will be discussed below.

A. The Food Algorithm

We say that walking to C or B in Fig. 4 takes you to the foraging area as it is true in most cases for trail networks of the Pharaoh's ant [3]. Hence in this section the robot should leave the bifurcation at C or B. Additionally, in case it moves in the wrong direction, i.e. it approaches a bifurcation from C or B, it should turn and leave the bifurcation either at C or B but not at A. A simplified flow chart of the algorithm is given in Fig. 6. The conditions are implemented as interrupts. Thus the execution of an action will be stopped if the superordinate conditions do not hold anymore. The values that were used for the thresholds are given in Table I.

The first condition checks if the light at one sensor is much brighter than at the other one. If $|L - R| > \delta_{\text{diff}}$ is true, a correction turn will be executed. Otherwise it is checked if it is quite dark at both sensors. Then a u-turn is performed and

TABLE I
BRIGHTNESS THRESHOLDS.

Threshold	Value
δ_{diff}	20
δ_{dark}	11
δ_{bright}	130

otherwise the robot keeps going straight.

Here the correction turn is, with up to about 45 degrees, bigger than the correction turn of the nest algorithm (see below). It should be noted that this turning angle is not constant. The rotation will be stopped immediately if the interrupt does not hold anymore, e.g. in case of a correction turn to the left the rotation is stopped if the interrupt condition $L > R$ is not true anymore. Even if the turn would be repeated in the exact same environmental setting, the angles could differ a lot since the version of the Jasmine robot that we used in this work (generation two) has no odometry.

In the case of coming from A the basic idea of this algorithm is to avoid approaching the bifurcation in a driving direction parallel to the line through A and D because this might trigger a u-turn if both photodiodes leave the trail almost synchronously. Coming from B or C we hope for this u-turn when the robot approaches the big angles ADB or ADC (for $\alpha \neq 120°$) as it is shown in Fig. 5(c).

B. The Nest Algorithm

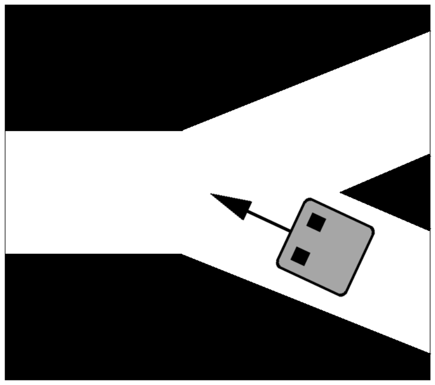
In the nest algorithm the correction turn is, with up to about 35 degrees, smaller than the correction turn of the food algorithm. The other change to the food algorithm is an extra condition that checks for a high brightness at both sensors. In that situation the robot turns up to 120 degrees. The rotation direction is chosen randomly with a 50% chance of a right and a 50% chance of a left turn.

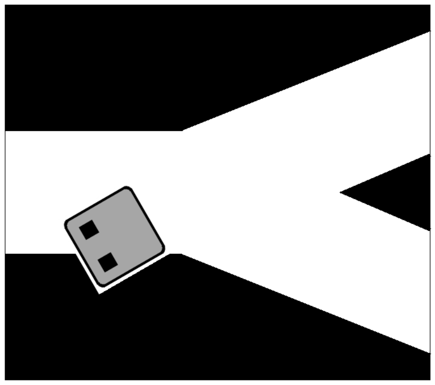
The basic idea is that the robot coming from A is nicely aligned to the trail such that it gets into the situation shown in Fig. 5(d). Then a bigger turn of up to 120 degrees followed by correction turns should bring it back to A. Coming from B or C the robot should overcome the bifurcation by correction turns and should not approach the position of brightest light.

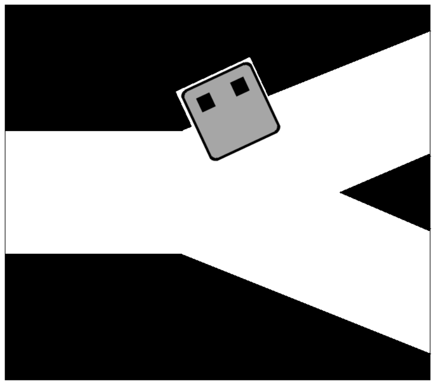
IV. Results

The setup for our experiments is shown in Fig. 4. In correspondence to the natural networks the direction to A is called "to the nest" and to B and C is called "to the food". The bifurcation angles were varied: $\alpha \in \{30°, 45°, 60°, 90°, 120°\}$. Each experiment starts with the robot being placed at A, B, or C positioned straight ahead and switched on. The experiment ends with one of six possible outcomes: The robot arrives at A, B, or C by having driven at least 20 cm on the trail before having possibly executed a u-turn, it executed a u-turn before having driven at least 20 cm (*failure*), it left the trail (*failure*), or it got caught in some repetitive behavior for more than two seconds, e.g. turning permanently.

For the food algorithm the definition of *correct turns* is as follows: The robot started at B or C (sine qua non) and arrived


(a) Approach of a bifurcation coming from the food place.


(b) Typical situation for a correction turn (here to the right).


(c) Typical position for a u-turn.

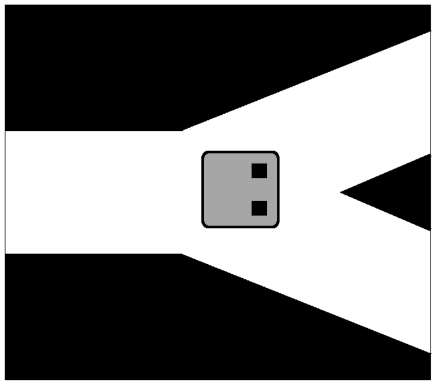

(d) Position of brightest light.

Fig. 5. Exemplary situations occurring when the robot (grey box with two little black boxes indicating the position of the photodiodes) approaches the trail bifurcation (in white).

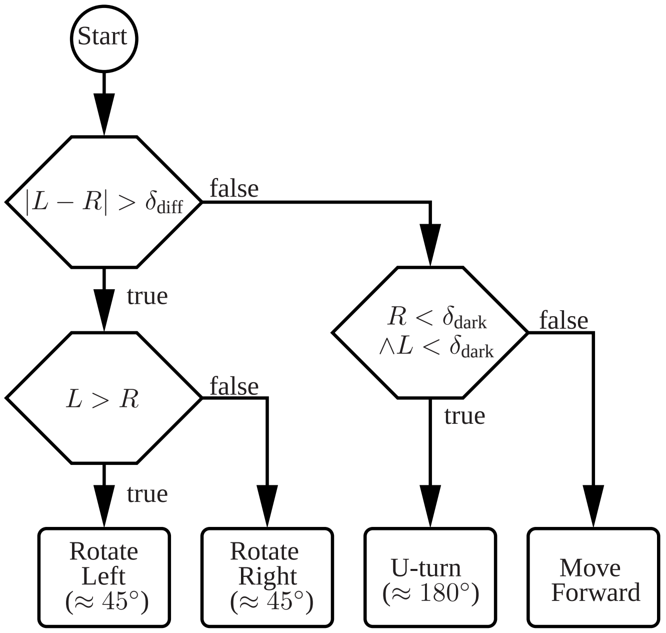

Start
$|L-R| > \delta_{\text{diff}}$
false
true
$R < \delta_{\text{dark}}$ $\wedge L < \delta_{\text{dark}}$
false
$L > R$
false
true
true
Rotate Left ($\approx 45°$)
Rotate Right ($\approx 45°$)
U-turn ($\approx 180°$)
Move Forward

Fig. 6. Simplified flow chart of the food algorithm.

at B or C again or it ended up in a repetitive behavior. This is a fair classification because the robot has recognized the bifurcation and with a more sophisticated algorithm it would be possible to minimize repetitive behavior and dropping the restriction of a purely reactive control it would be possible to avoid it at all. The definition of incorrect turns is this: The robot started at A (sine qua non) and arrived at A again or ended up in a repetitive behavior.

The definitions for the nest algorithm are defined in an analog way. Since fails occurred very seldom ($< 1\%$), they were not counted and experiments were repeated in case of fails.

For each, nest and food algorithm, we performed 50 experiments starting at B, 50 starting at C, and 100 starting at A for $\alpha \in \{30°, 45°, 90°, 120°\}$ and 100 at C, 100 at B, and 200 at A for $\alpha = 60°$. The results are shown in Fig. 8 and the ratio between the number of correct reorientations R_c divided by the number of incorrect reorientations R_i is given in Fig. 9(a). As long as $R_c/R_i > 1$ holds the algorithm is in principle effective. The average of both algorithms is clearly below the reported performance of the Pharaoh's ant [4]. The food algorithm performs much better than the nest algorithm. This is discussed below. But the nest algorithm is still effective for $\alpha < 75°$ although it might take a long time for a robot to find the nest. The peak of both algorithms is at $\alpha = 60°$ as observed in ants [4]. The noticeable drop in performance for $\alpha = 45°$ is induced by an incompatibility of the average correction turn angles and α. Although in many experiments for $\alpha = 45°$, at first, the robot turned correctly, it then approached the border of the straight trail paths in a low angle, which prevented it from executing another correction turn because of the threshold in the interrupt ($|L - R| > \delta_{\text{diff}}$). Thus it went often off the trail and did a u-turn.

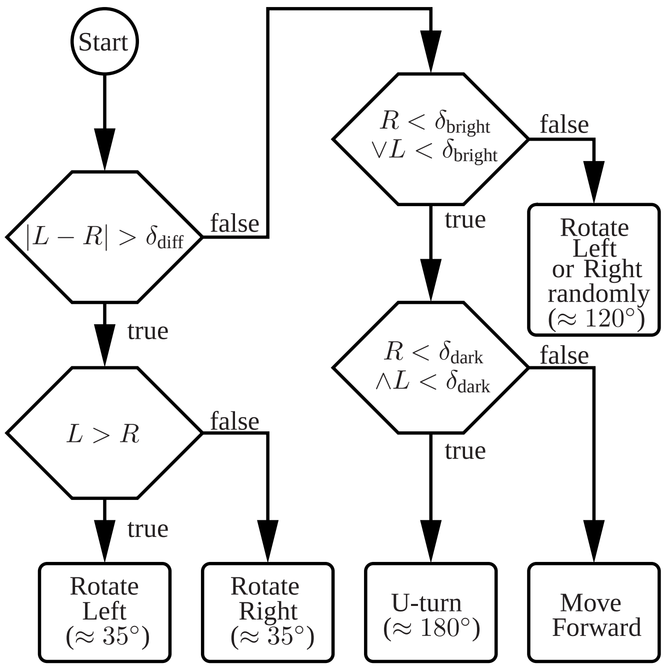
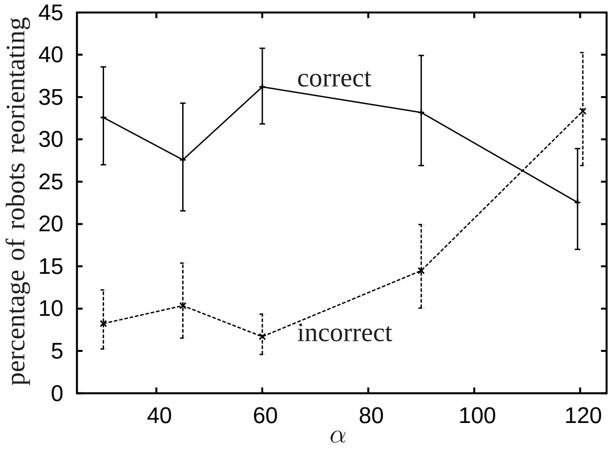

Start
$|L-R| > \delta_{\text{diff}}$
false
true
$L > R$
false
true
Rotate Left ($\approx 35°$)
Rotate Right ($\approx 35°$)
$R < \delta_{\text{bright}}$ $\vee L < \delta_{\text{bright}}$
false
true
Rotate Left or Right randomly ($\approx 120°$)
$R < \delta_{\text{dark}}$ $\wedge L < \delta_{\text{dark}}$
false
true
U-turn ($\approx 180°$)
Move Forward

Fig. 7. Simplified flow chart of the nest algorithm.

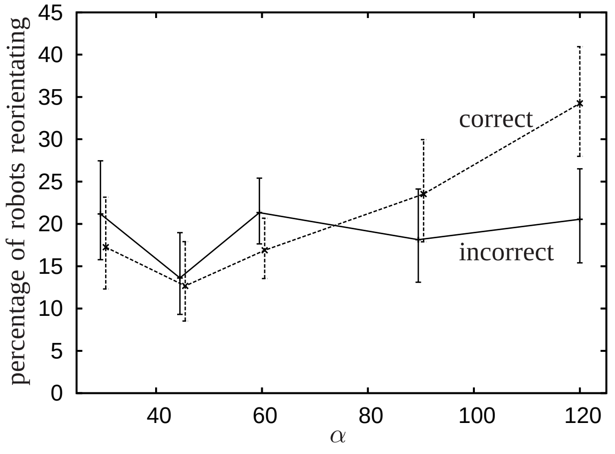
The case of $\alpha = 120°$ can serve as a sanity check for the data. Since there cannot be any polarity information encoded in a 120 degree bifurcation we assume that all three outcomes (A, B, and C; counting repetitive behavior as leaving at were it came from) are uniformly distributed, ignoring for simplicity that 180-degree-turns versus 120-degree-turns do not have to be equiprobable (here for $\alpha = 120°$ the ratio of 180-degree-turns to 120-degree-turns was about 1.5). Since we do the same amount of experiments starting at A as starting at B and C together, we get: $16.\bar{6}\%$ incorrect and $33.\bar{3}\%$ correct reorientations for the food task. That means we get a correct to incorrect ratio of $R_c/R_i = 2$ for granted. For the nest task we get vice versa: $R_c/R_i = 0.5$, which makes the nest finding a harder task. In Fig. 9(b) the ratios normalized to these values are given. This diagram shows clearly that both algorithms perform significantly better than and similarly well compared to random behavior.

V. Conclusion

At first, we presented the swarm robot Jasmine that was used to conduct the experiments. We showed that even for a very bounded hardware platform it is possible to solve complex tasks as reorienting in a network of trails depending on certain geometric properties of bifurcations. While the performance of the reorientation algorithm for finding food is good, the nest algorithm might possibly be improved heavily. The fact that on one hand it is easier for a robot moving totally random finding the food than finding the nest and that, on the other hand, the authors were not able to find a nest algorithm showing similar performance as the food algorithm might indicate that this is the more difficult part of the task or that the natural ant uses a different mechanism to return to the nest. However, this



(a) Food algorithm.

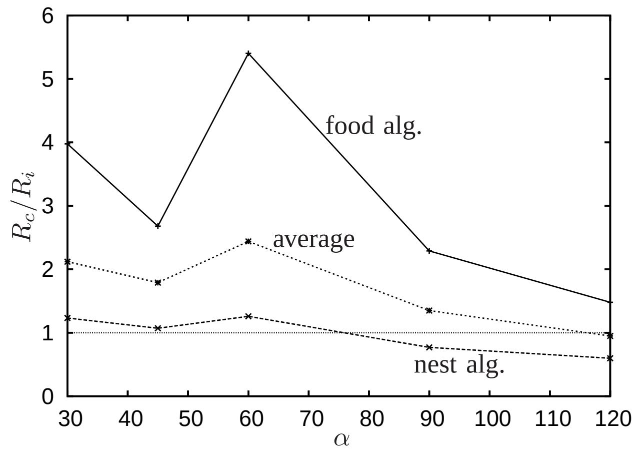


(b) Nest algorithm.

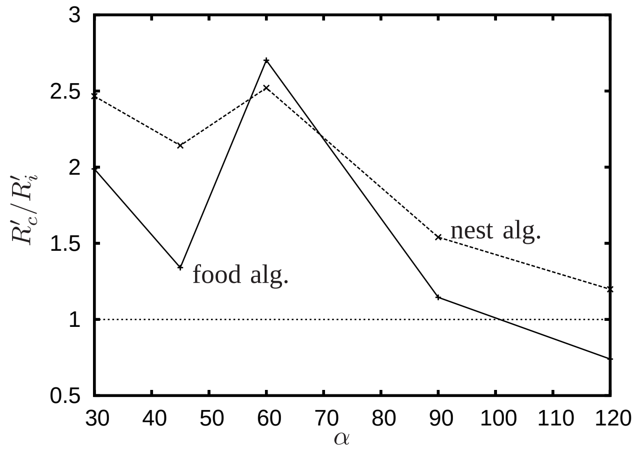
Fig. 8. Correct and incorrect reorientations for $\alpha \in \{30°, 45°, 60°, 90°, 120°\}$, error-bars indicate the 95% confidence interval (some values are shifted by $\pm 0.5°$ for better readability).

hypothesis can, of course, easily be falsified by presenting an algorithm with better performance.

The behavior described here is highly probabilistic which is a typical characteristic of natural and artificial swarms. Because of that for laymen watching our robot working its way through the trail network it might not look "intelligent" at all as it is true for following a natural ant's route. Such behavior is the only possibility to overcome the impreciseness of the individuals in a swarm, since highly efficient behaviors of the individuals ($R_c/R_i \to \infty$) cannot be achieved. However, because of the dependence on probabilities the algorithms presented here are hard to analyze. Only by directly measuring the robot's turning angles and tagging the triggering events, for example, it will be possible to find the probability distribution of the angles. This distribution is not explicitly specified in the algorithm but in the ensemble of the hardware platform and the software as it might be true for the natural ant. As already mentioned better approaches to master this unison of hardware and software are in need.



(a) For values $R_c/R_i > 1$ the algorithm is effective.



(b) Normalized, $R'_c/R'_i = 1$ is the ratio for random behavior.

Fig. 9. Ratio of correct to incorrect reorientations.

Acknowledgments

Hamann is supported by the German Research Foundation (DFG) within the Research Training Group GRK 1194 Self-organizing Sensor-Actuator Networks. Szymanski is supported by the European Union within the “Beyond Robotics” Proactive Initiative – 6th Framework Programme: 2003-2007 (I-SWARM project, Project Reference: 507006).

References

[1] Jasmine robot - project website, 2007. http://www.swarmrobot.org/.

[2] G. Caprari, P. Balmer, R. Piguet, and R. Siegwart. The autonomous microbot alice: a platform for scientific and commercial applications. In *Proc. of the Ninth Int. Symp. on Micromechatronics and Human Science*, pages 231–235, Nagoya, Japan, 1998.

[3] D. E. Jackson. *The Shortest Path is the One You Know*. PhD thesis, University of Sheffield, September 2005.

[4] D. E. Jackson, M. Holcombe, and F. L. W. Ratnieks. Trail geometry gives polarity to an foraging networks. *Nature*, 432:907–909, 2004.

[5] S. Kornienko, O. Kornienko, and P. Levi. Swarm embodiment – a new way for deriving emergent behavior in artificial swarms. In P. Levi, M. Schanz, R. Lafrenz, and V. Avrutin, editors, *Autonome Mobile Systeme*, pages 25–32, 2005.

[6] K. Lerman, C. Jones, A. Galstyan, and M. Mataric. Analysis of dynamic task allocation in multi-robot systems. *Int. J. of Robotics Research*, 25(3):225 – 241, 2006.

[7] A. Martinoli, K. Easton, and W. Agassounon. Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *Int. Journal of Robotics Research*, 23:415–436, 2004.

[8] J. Seyfried, M. Szymanski, N. Bender, R. Estana, M. Thiel, and H. Wörn. The i-swarm project: Intelligent small world autonomous robots for micro-manipulation. In E. Sahin and W. Spears, editors, *Swarm Robotics Workshop: State-of-the-art Survey*, pages 70–83, Berlin Heidelberg New York, 2005. Springer-Verlag.

[9] M. Szymanski and H. Wörn. Jamos - a mdl2e based operating system for swarm micro robotics. Appearing in this proceedings.