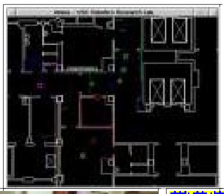
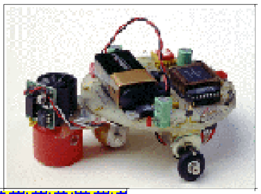
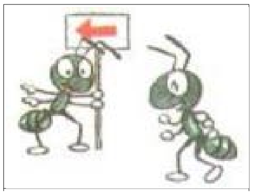
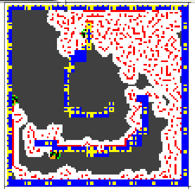
ICRA-03 Tutorial on Ant-Based Mobile Robots: Robust Navigation and Coverage with Single Robots and Robot Teams



Abstract

Ant robots are simple and cheap robots with limited sensing and computational capabilities. This makes it feasible to deploy teams of ant robots and take advantage of the resulting fault tolerance and parallelism. Ant robots cannot use conventional planning methods due to their limited sensing and computational capabilities. Rather, their behavior is driven by local interactions. For example, ant robots can communicate via markings that they leave in the terrain, similar to ants that lay and follow pheromone trails. In the past couple of years, researchers have developed ant robot hardware and software and demonstrated, both in simulation and on physical robots, that single ant robots or teams of ant robots solve robot-navigation tasks (such as path following and terrain coverage) robustly. Thus, ant robots might provide a promising alternative to the currently very popular probabilistic robot navigation approaches. Researchers have also developed a theoretical foundation for ant robotics, based on ideas from real-time heuristic search, stochastic analysis, and graph theory. This half day tutorial on the current state of the art in ant robotics is given by experts who will give an overview of this exciting area of mobile robotics without assuming any prior knowledge on the topic. It will cover all important aspects of ant robotics, from theoretical foundations to videos of implemented systems. To this end, it will bring together the various research directions for the first time, including theoretical foundations, ant robot hardware, and ant robot software. Its primary objective is to give non-specialists a comprehensive overview of the state of the art of the field, for example, to allow researchers and students to do research in the area and to allow practitioners to evaluate the current state of the art in ant robotics. Consequently, it is of interest to anyone who is interested in mobile robotics and artificial intelligence.

Speakers

The material for the tutorial will be provided by the five researchers listed below, all of which are experts on the topic of ant robotics. They have been selected to cover the theory of ant robotics as well as practical approaches, including ant robot software and hardware.

Sven Koenig (Georgia Institute of Technology, USA)

Sven Koenig became an assistant professor in the College of Computing at Georgia Institute of Technology after receiving his Ph.D. in computer science from Carnegie Mellon University. He is the recipient of an NSF CAREER award, an IBM Faculty Partnership Award, the Raytheon Faculty Fellowship Award from Georgia Tech, and a Fulbright Fellowship Award. He is currently on the editorial board of the Journal of Artificial Intelligence Research (JAIR) and will co-chair the International Conference on Automated Planning and Scheduling (ICAPS) in 2004. Sven's research centers around techniques for decision making (planning and learning) that enable situated agents to act intelligently in their environments and exhibit goal-directed behavior in real-time, even if they have only incomplete knowledge of their environment, limited or noisy perception, imperfect abilities to manipulate it, or insufficient reasoning speed. He has published over 60 papers on this topic in the artificial intelligence and robotics literature.

Israel Wagner (Technion and IBM Haifa Research Lab, Israel)

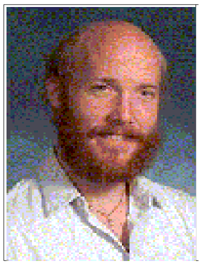
Israel Wagner received his B.Sc. degree in computer engineering from the Technion (Israel Institute of Technology), Haifa, in 1987, cum laude, an M.Sc. degree in computer science from Hebrew University, Jerusalem, in 1990, cum laude, and a Ph.D. degree in computer science from the Technion in 1999. He was a research engineer at General Microwave, Jerusalem, from 1987 until 1990, when he joined the IBM Haifa Laboratories as a member of the technical staff. Israel is currently an adjunct senior lecturer in the Computer Science Department at the Technion. His research interests include multi-agent robotics, manual and automatic VLSI design, computational geometry, and graph theory. He has published over 25 papers on these topics in the artificial intelligence, robotics, and VLSI literature and co-edited a special issue of the Annals of Mathematics and Artificial Intelligence on ant robotics in 2001.

Andrew Russell (Monash University, Australia)

Andrew (Andy) Russell received his B. Eng. and Ph.D. degrees in 1972 and 1976, respectively, from the University of Liverpool, in the U.K. After a brief period in the computer industry he took a position as Engineer working on robot design and applications in the Department of Artificial Intelligence at the University of Edinburgh, Scotland. For the past 19 years he has been a member of academic staff at the University of Wollongong and now at Monash University (both in Australia). Andy's research interests include robot tactile sensing, the design of robotic mechanisms and olfactory sensing for robots. He has written books on robot tactile sensing and odour sensing for mobile robots and published over 70 refereed conference papers and journal articles describing his work in intelligent robotics.

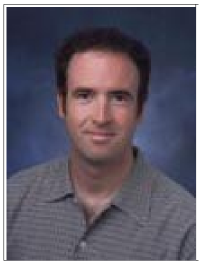
Andrew will give a video presentation.

David Payton (HRL Laboratories, USA)



David Payton is principal research scientist and manager of the Cooperative and Distributed Systems Department at HRL Laboratories in Malibu, California. He received his B.S. degree from UCLA in 1979 and his M.S. degree from MIT in 1981. David is currently principal investigator for the DARPA Pheromone Robotics project and is also involved in development of agent-assisted multi-user collaboration tools. After joining HRL Laboratories in 1982, David has been involved in numerous projects for the development of intelligent autonomous agents. This includes work on the DARPA Autonomous Land Vehicle project, the Unmanned Ground Vehicle and the development of behavior-based robot control. David has over 25 publications in the area and holds six patents.

Richard Vaughan (HRL Laboratories, USA)



Richard Vaughan received a D.Phil. in Computation from Oxford University in 1999, and a B.A. (Hons.) in Computing with Artificial Intelligence from Sussex University in 1993. He was a postdoctoral research associate at the Robotics Laboratory of the University of Southern California from 1998 to 2001, and is currently a member of the technical staff at HRL Laboratories. Richard's research concerns the mechanisms of intelligent behavior in individuals and groups of people, animals, robots and software agents. His projects emphasize dynamic autonomy, scalability, interaction and simplicity. The majority of his more than 20 publications concern the functional analysis and recreation of aspects of animal behavior, a methodology he calls "constructive ethology". His other papers are on simulation, interfacing and networking for robotics applications. Richard is a founding member of the Player/Stage project, which creates tools for robotics and sensor network research.

Sven Koenig
Georgia Institute of Technology
USA

Ant Robotics

Terrain Coverage

Sven Koenig

College of Computing
Georgia Institute of Technology

Overview

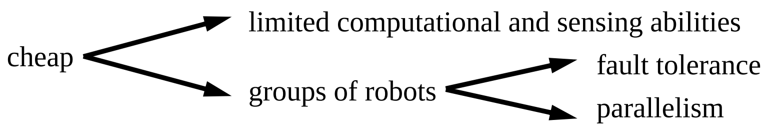
Overview:
One-Time or Repeated Coverage of Known or Unknown Terrain with Single Ant Robots or Teams of Ant Robots for
- Mine Sweeping
- Surveillance
- Surface Inspection
- Guarding Terrain

Structure:
- Motivation
- Theoretical Results
  - Real-Time Search
- Empirical Results
  - Simulation
  - Actual Robots

joint work with:
Jonas Svennebring, Boleslaw Szymanski (RPI), and Yaxin Liu

Motivation

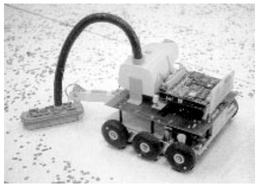
Motivation


cheap → limited computational and sensing abilities
cheap → groups of robots
groups of robots → fault tolerance
groups of robots → parallelism

DC06

Robomow

Cye


Koala

Motivation

You want to build a team of robots that cover terrain repeatedly, for example, to guard a museum at night.

The terrain could be initially unknown.
The terrain could change dynamically.
The robots have very noisy actuators or sensors.
The robots can fail.

Approach 1: POMDP-Based Navigation Architecture

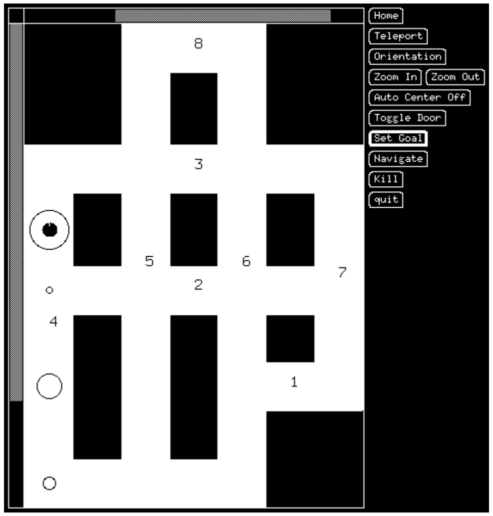
You want to build a team of robots that cover terrain repeatedly, for example, to guard a museum at night.

The terrain could be initially unknown.
The terrain could change dynamically.
The robots have very noisy actuators or sensors.
The robots can fail.

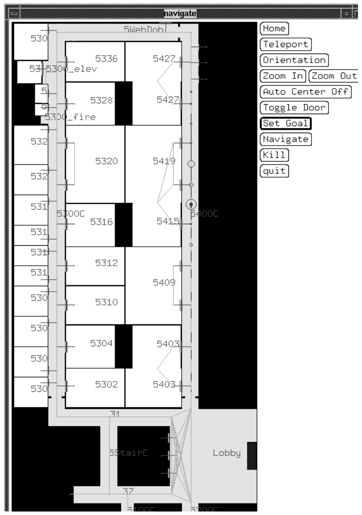
Probabilistic
Planning

R. Simmons and S. Koenig. Probabilistic Robot Navigation in Partially Observable Environments. In Proceedings of the International Joint Conference on Artificial Intelligence, 1080-1087, 1995.

Approach 1: POMDP-Based Navigation Architecture

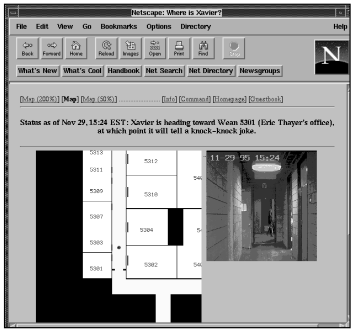


simulator



interface to Xavier

Approach 1: POMDP-Based Navigation Architecture



Status as of Nov 29, 15:24 EST: Xavier is heading toward Wean 5301 (Eric Thayer's office), at which point it will tell a knock-knock joke.

Xavier

Advantages of Navigation with POMDPs

- uniform, theoretically grounded framework
- maintains arbitrary probability distributions over the poses
- explicitly models all uncertainty using probabilities
- utilizes all available sensor data (landmarks, robot movements)
- robust towards sensor errors (no explicit exception handling required)

Approach 2: Ant Robotics

You want to build a team of robots that cover terrain repeatedly, for example, to guard a museum at night.

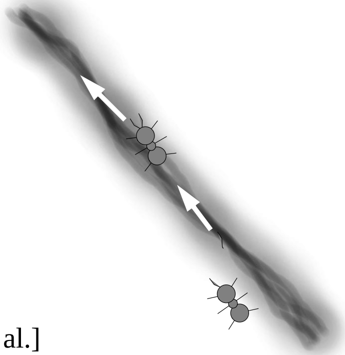
The terrain could be initially unknown.
The terrain could change dynamically.
The robots have very noisy actuators or sensors.
The robots can fail.

~~Probabilistic Planning~~

no location estimates!
no planning!
no direct communication!

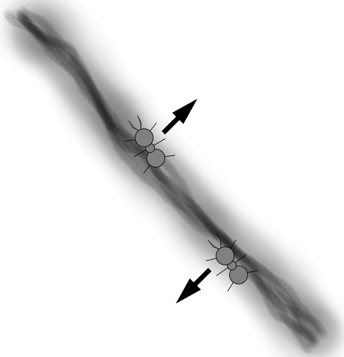
simpler hardware and software!

Approach 2: Ant Robotics
Path Following using Pheromone Traces



- short lived traces
  - alcohol [Sharpe et al.]
  - heat [Russell]
  - odor [Russell el at.]
- virtual traces [Vaughan et al.; Payton et al.]

Approach 2: Ant Robotics
Exploration and Coverage using Pheromone Traces

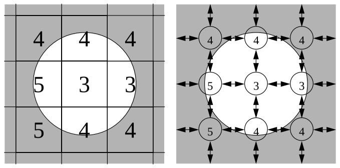


- long lived traces [Svennebring and Koenig]

Theoretical Results
(Real-Time Search)

Theoretical Results (Real-Time Search)

Node Counting



Initially, the u-values *u(s)* are zero for all states *s*.

1. *s* := start location
2. *s'* := a neighboring location of *s* with a minimal u-value
3. *u(s) := 1 + u(s)*
4. move the ant robot to location *s'*
5. go to 2.

Theoretical Results (Real-Time Search)

time step 0

0	0	0	■	0	0
0	0	0	■	0	0
0	0	■	■	0	0
0	0	0	0	0	0
0	0	0	0	0	0

time step 1

3	0	0	■	0	0
0	0	0	■	0	0
0	0	■	■	0	0
0	0	0	0	0	0
0	0	0	0	0	0

time step 2

3	1	0	■	0	0
2	0	0	■	0	0
0	0	■	■	0	0
0	0	0	0	0	0
0	0	0	0	0	0

time step 3

3	1	0	■	0	0
2	2	0	■	0	0
1	0	■	■	0	0
0	0	0	0	0	0
0	0	0	0	0	0

time step 4

3	1	0	■	0	0
2	2	1	■	0	0
1	2	■	■	0	0
0	0	0	0	0	0
0	0	0	0	0	0

time step 5

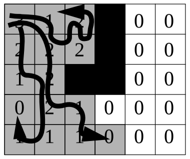
3	1	1	■	0	0
2	2	1	■	0	0
1	2	■	■	0	0
0	2	0	0	0	0
0	0	0	0	0	0

time step 6

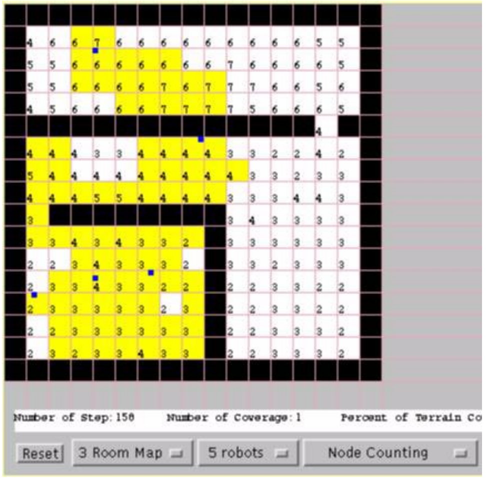
3	1	1	■	0	0
2	2	2	■	0	0
1	2	■	■	0	0
0	2	1	0	0	0
0	1	0	0	0	0

time step 7

3	1	2	■	0	0
2	2	2	■	0	0
1	2	■	■	0	0
0	2	1	0	0	0
1	1	1	0	0	0



Theoretical Results (Real-Time Search)

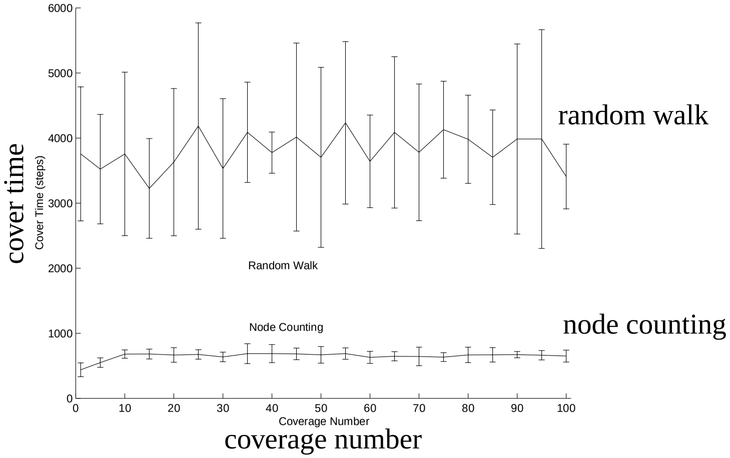


Number of Step: 158 Number of Coverage: 1 Percent of Terrain Co

programming: Jonas Svennebring

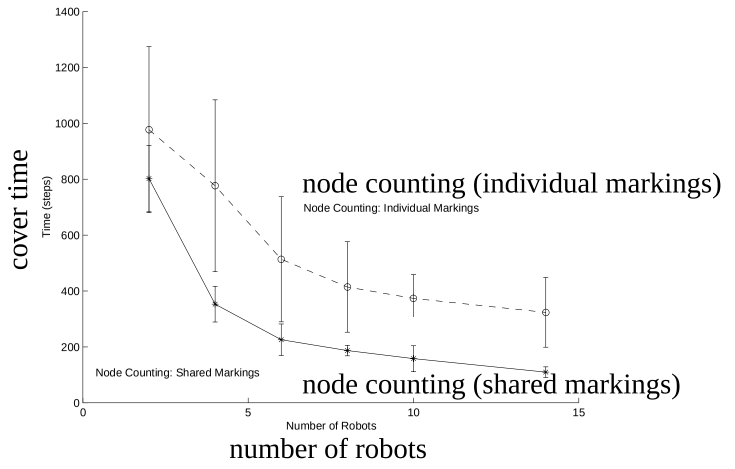
Theoretical Results (Real-Time Search)

The numbers (= markings) coordinate the ant robots!



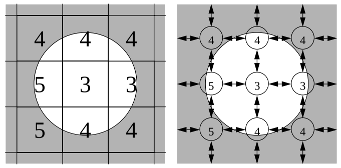
Theoretical Results (Real-Time Search)

The numbers (= markings) coordinate the ant robots!



Theoretical Results (Real-Time Search)

Additional Real-Time Search Methods



Initially, the u-values $u(s)$ are zero for all states s.

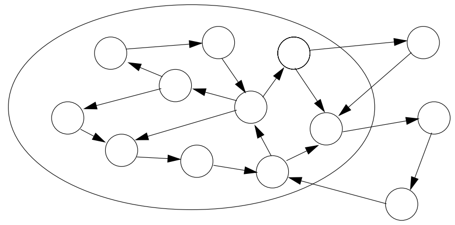
```
   1. $s$ := start location
   2. $s'$ := a neighboring location of $s$ with a minimal u-value
   3. $u(s) := 1 + u(s)$                          Node Counting
or    $u(s) := 1 + u(s')$                         Korf's LRTA* (= Wagner's VAW)
or    if $u(s) \leq u(s')$ then $u(s) := 1 + u(s)$   Wagner's Update Rule
or    $u(s) := max(1 + u(s), 1 + u(s'))$           Thrun's Update Rule
   4. move the ant robot to location $s'$
   5. go to 2.
```

Theoretical Results (Real-Time Search)

Theorem:

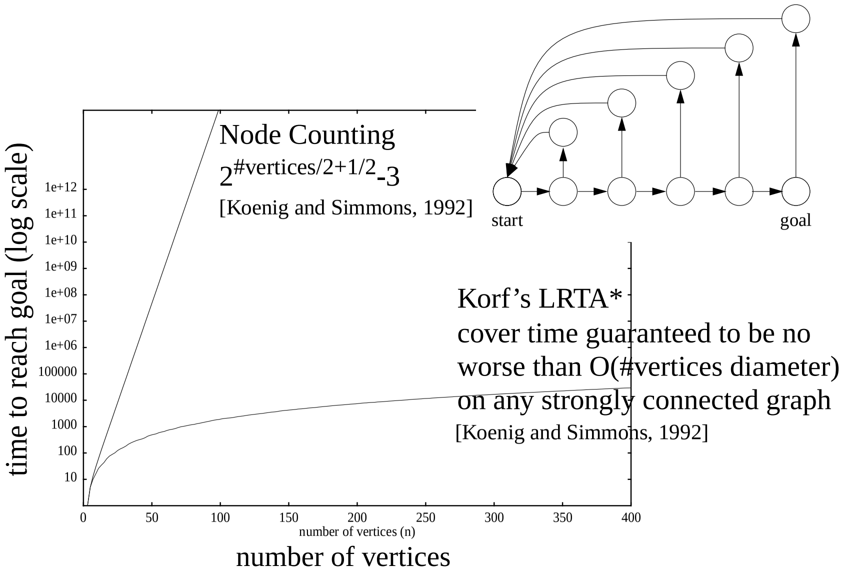
Teams of ant robots that all use the same real-time search method cover all strongly connected graphs repeatedly.

Proof:

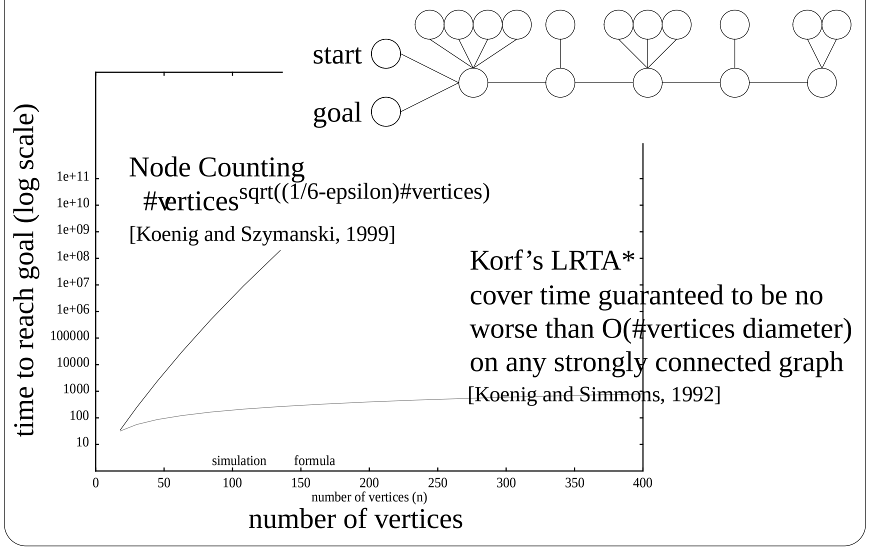


QED

Theoretical Results (Real-Time Search)



Theoretical Results (Real-Time Search)



Node Counting
#vertices$^{\text{sqrt((1/6-epsilon)\#vertices)}}$
[Koenig and Szymanski, 1999]

Korf’s LRTA*
cover time guaranteed to be no worse than O(#vertices diameter) on any strongly connected graph
[Koenig and Simmons, 1992]

Theoretical Results (Real-Time Search)

Initially, the u-values *u(s)* are zero for all states *s*.

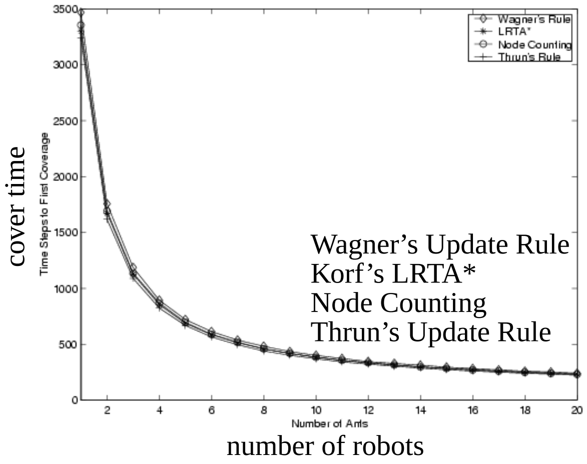
```
    1. $s$ := start location
    2. $s'$ := a neighboring location of $s$ with a minimal u-value
    3. $u(s) := 1 + u(s)$                                    exponential cover time
or     $u(s) := 1 + u(s')$                                   polynomial cover time
or     if $u(s) \leq u(s')$ then $u(s) := 1 + u(s)$          polynomial cover time
or     $u(s) := max(1 + u(s), 1 + u(s'))$                     polynomial cover time
    4. move the ant robot to location $s'$
    5. go to 2.
```

Theoretical Results (Real-Time Search)

Is the worst-case cover time of a single ant robot that uses node counting polynomial or exponential in the number of vertices (an adversary can choose the graph topology, the start vertex, the goal vertex, and the tie-breaking rule),

- if the strongly connected graphs are directed? yes (see 3 slides ago)
- if the strongly connected graphs are undirected? yes (see 2 slide2 ago)
- if the strongly connected graphs are undirected grids? unknown

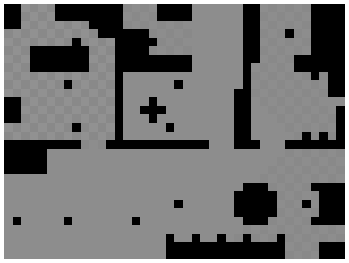
Theoretical Results (Real-Time Search)



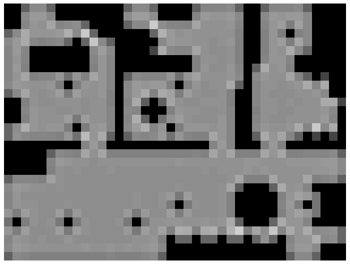
Wagner’s Update Rule
Korf’s LRTA*
Node Counting
Thrun’s Update Rule

[Koenig et. al., 2001]

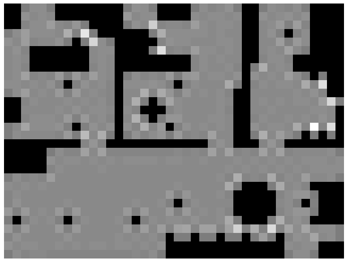
Theoretical Results (Real-Time Search)



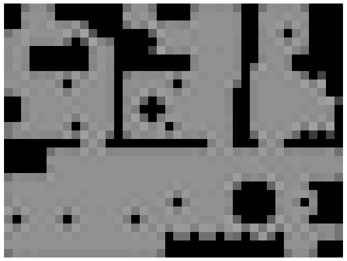
Node Counting



Korf’s LRTA*



Wagner’s Update Rule



Thrun’s Update Rule

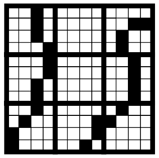
[Koenig et. al., 2001]

Empirical Results
(Simulation and Actual Robots)

BORG Lab

[Svennebring and Koenig, 2003]

Empirical Results (Actual Robots)

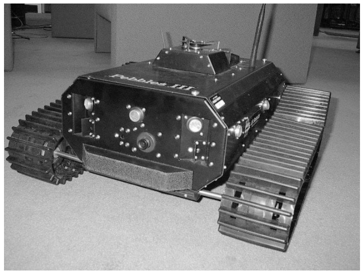


4	0	4
4	0	4
4	3	2

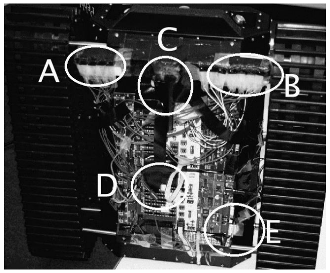
Ant robots that all use node counting are easy to implement!

Empirical Results (Actual Robots)

Ant Robot Hardware



Thanks to Ashwin Ram for the hardware.

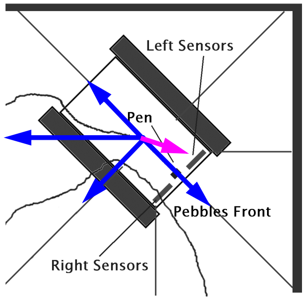


A: trail sensor
B: trail sensor
C: pen
D: micro-controller
E: RS232 interface

Empirical Results (Actual Robots)

Ant Robot Software

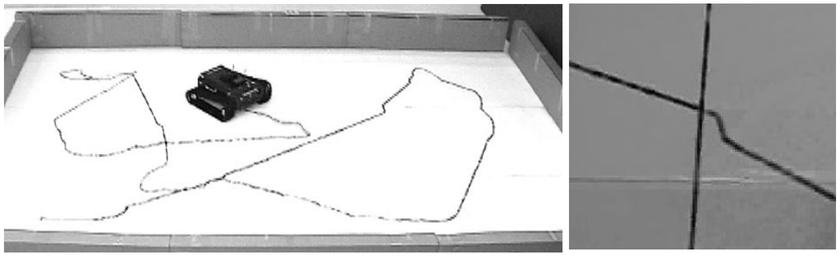
our ant robots use a schema-based navigation strategy with an obstacle avoidance behavior and a trail-avoidance behavior



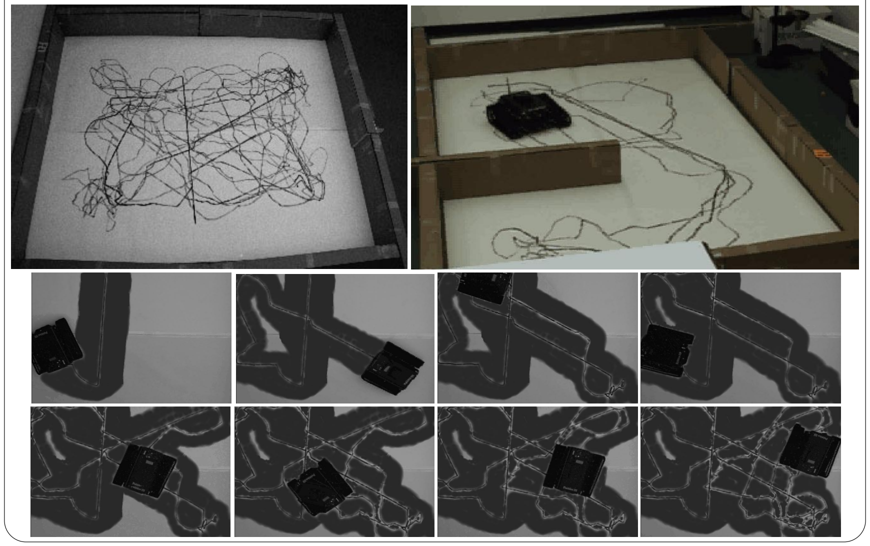
Empirical Results (Actual Robots)

Ant Robot Software

our ant robots use a schema-based navigation strategy with an obstacle avoidance behavior and a trail-avoidance behavior

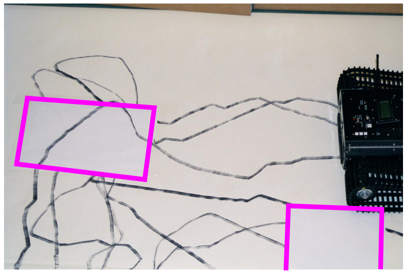


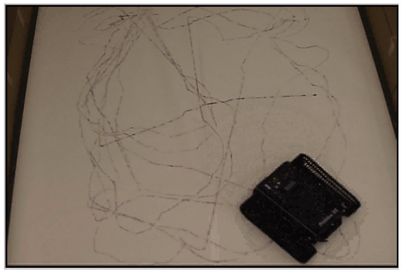
Empirical Results (Actual Robots)



Empirical Results (Simulation and Actual Robots)

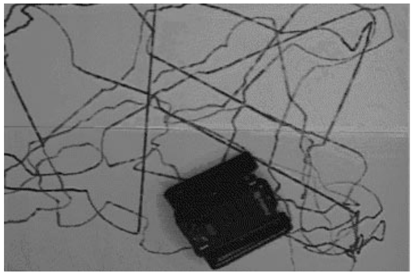
Our ant robots cover closed terrain even if
- they don’t know the terrain in advance or the terrain changes,
- some ant robots fail,
- some ant robots are moved without realizing this, or
- some trails are destroyed.


destroyed areas of trails

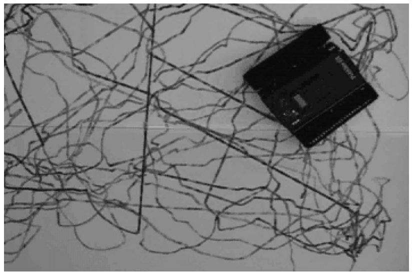

low-intensity trails

Empirical Results (Actual Robots)

The terrain gets saturated with trails over time.

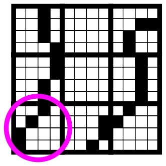


end of first coverage



end of third coverage

Empirical Results (Modeling with Real-Time Search)



randomly drop a drop of ink into this cell

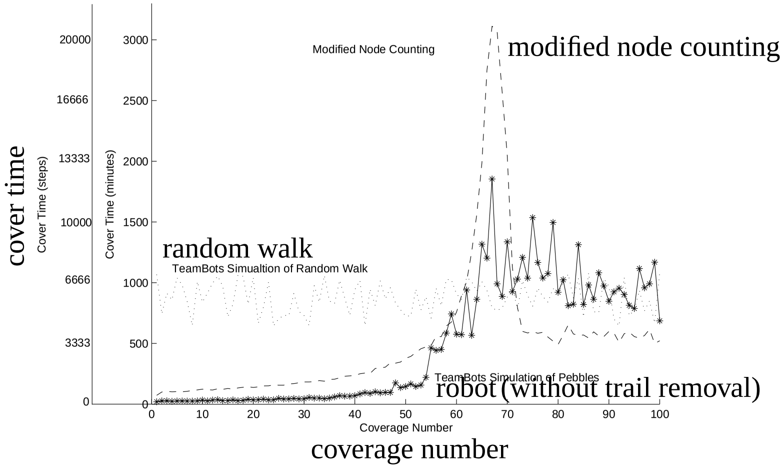
4	0	4
4	0	4
4	3	2

increase this number by one with probability (16-4)/16

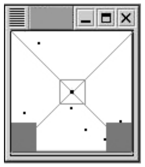
Initially, the u-values *u(s)* are zero for all states *s*.

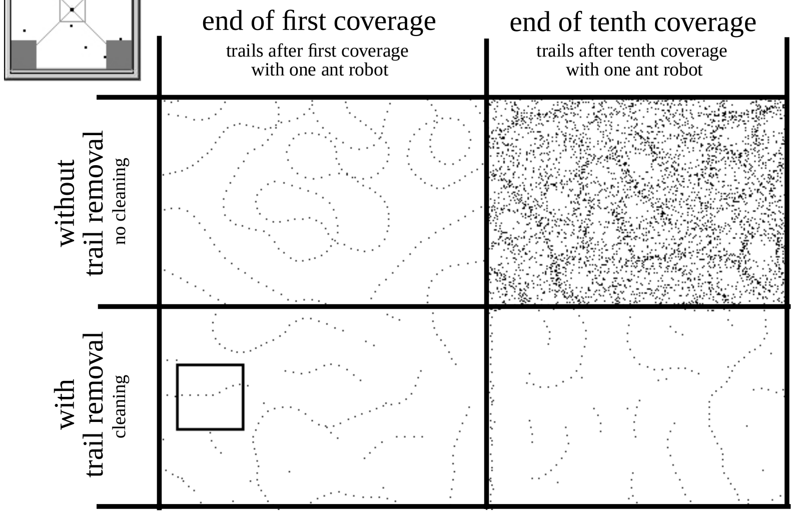
1. *s* := start location
2. *s'* := a neighboring location of *s* with a minimal u-value
3. with probability (170-u(s))/170 do: *u(s)* := *1* + *u(s)*
4. move the ant robot to location *s'*
5. go to 2.

Empirical Results (Simulation)

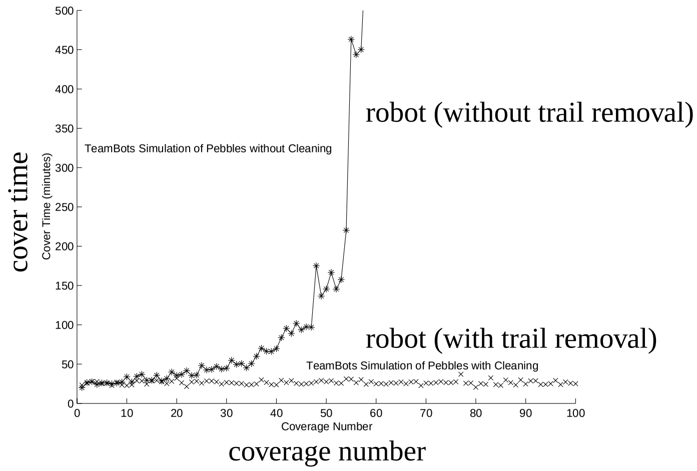


Empirical Results (Simulation)





Empirical Results (Simulation)



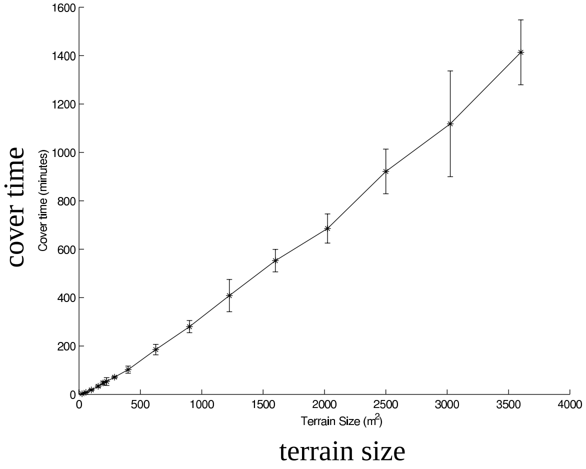
cover time

robot (without trail removal)

robot (with trail removal)

coverage number

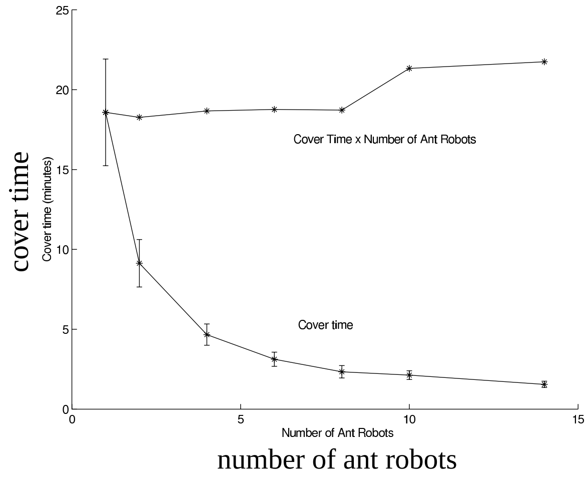
Empirical Results (Simulation)



cover time

terrain size

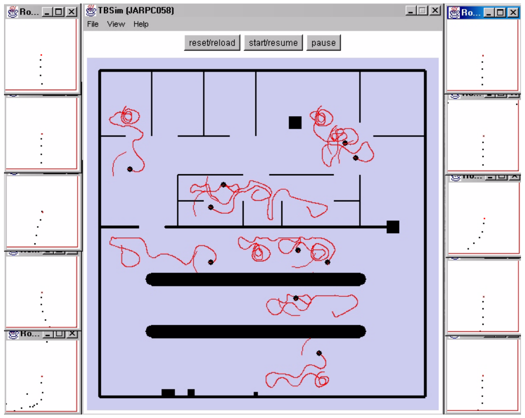
Empirical Results (Simulation)



cover time

number of ant robots

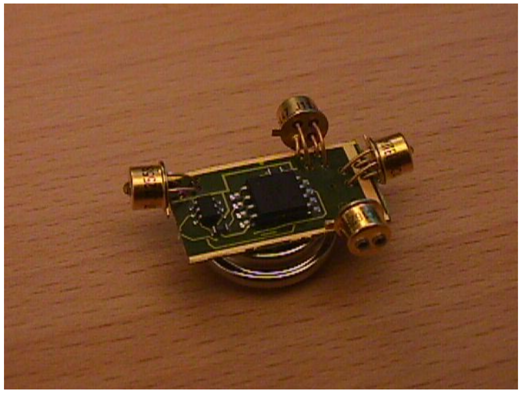
Empirical Results (Simulation)



85 hours
without any
ant robot
getting stuck

25 x 25 meters
10 ant robots

The Future



small infrared tranceivers as smart markers
(similar interesting work is performed at USC and other institutions)

Summary

Real-time search methods provide an interesting means for coordinating single ant robots or teams of ant robots that cover known or unknown terrain once or repeatedly. They leave markings in the terrain, similar to what some ants do. The ant robots robustly cover terrain even if the robots are moved without realizing this, some robots fail, and some markings get destroyed. The robots do not even need to be localized.

Selected Publications

Ant Robotics

[Svennebring and Koenig, 2002] J. Svennebring and S. Koenig. Building Terrain-Covering Ant Robots. Technical Report, GIT-COGSCI-2002/10, College of Computing, Georgia Institute of Technology, Atlanta (Georgia), 2002

[Koenig et. al., 2001] S. Koenig, B. Szymanski and Y. Liu. Efficient and Inefficient Ant Coverage Methods. Annals of Mathematics and Artificial Intelligence, 31, 41-76, 2001.

[Svennebring and Koenig, 2003] J. Svennebring and S. Koenig. Trail-Laying Robots for Robust Terrain Coverage. In Proceedings of the International Conference on Robotics and Automation, 2003.

Real-Time Search

[Koenig and Szymanski, 1999] S. Koenig and B. Szymanski. Value-Update Rules for Real-Time Search. In Proceedings of the National Conference on Artificial Intelligence, 718-724, 1999.

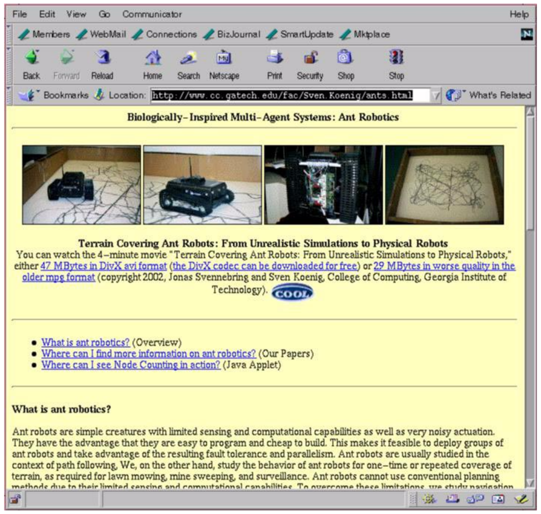
[Koenig and Simmons, 1996a] S. Koenig and R.G. Simmons. Easy and Hard Testbeds for Real-Time Search Algorithms. In Proceedings of the National Conference on Artificial Intelligence, 279-285, 1996.

[Koenig and Simmons, 1996b] S. Koenig and R.G. Simmons. The Influence of Domain Properties on the Performance of Real-Time Search Algorithms. Technical Report, CMU-CS-96-115, School of Computer Science, Carnegie Mellon University, Pittsburgh (Pennsylvania), 1996.

[Koenig and Simmons, 1995] S. Koenig and R.G. Simmons. Real-Time Search in Non-Deterministic Domains. In Proceedings of the International Joint Conference on Artificial Intelligence, 1660-1667, 1995.

[Koenig and Simmons, 1992] S. Koenig and R.G. Simmons. Complexity Analysis of Real-Time Reinforcement Learning Applied to Finding Shortest Paths in Deterministic Domains. Technical Report, CMU-CS-93-106, Computer Science Department, Carnegie Mellon University, Pittsburgh (Pennsylvania), 1992.

Additional Information


File Edit View Go Communicator Help
Members WebMail Connections BizJournal SmartUpdate Mktplace
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://www.cc.gatech.edu/fac/Sven.Koenig/ants.html
What's Related
COOL

Biologically-Inspired Multi-Agent Systems: Ant Robotics

Terrain Covering Ant Robots: From Unrealistic Simulations to Physical Robots
You can watch the 4-minute movie "Terrain Covering Ant Robots: From Unrealistic Simulations to Physical Robots," either 47 MBytes in DivX avi format (the DivX codec can be downloaded for free) or 29 MBytes in worse quality in the older mpg format (copyright 2002, Jonas Svennebring and Sven Koenig, College of Computing, Georgia Institute of Technology).

- What is ant robotics? (Overview)
- Where can I find more information on ant robotics? (Our Papers)
- Where can I see Node Counting in action? (Java Applet)

What is ant robotics?

Ant robots are simple creatures with limited sensing and computational capabilities as well as very noisy actuation. They have the advantage that they are easy to program and cheap to build. This makes it feasible to deploy groups of ant robots and take advantage of the resulting fault tolerance and parallelism. Ant robots are usually studied in the context of path following, We, on the other hand, study the behavior of ant robots for one-time or repeated coverage of terrain, as required for lawn mowing, mine sweeping, and surveillance. Ant robots cannot use conventional planning methods due to their limited sensing and computational capabilities. To overcome these limitations, we study navigation

http://www.cc.gatech.edu/fac/Sven.Koenig/ants.html

Israel Wagner
Technion and
IBM Haifa Research Lab
Israel

Ant Robotics

Cycles, Robustness, and Randomness

Israel A. Wagner

IBM Haifa Labs

and

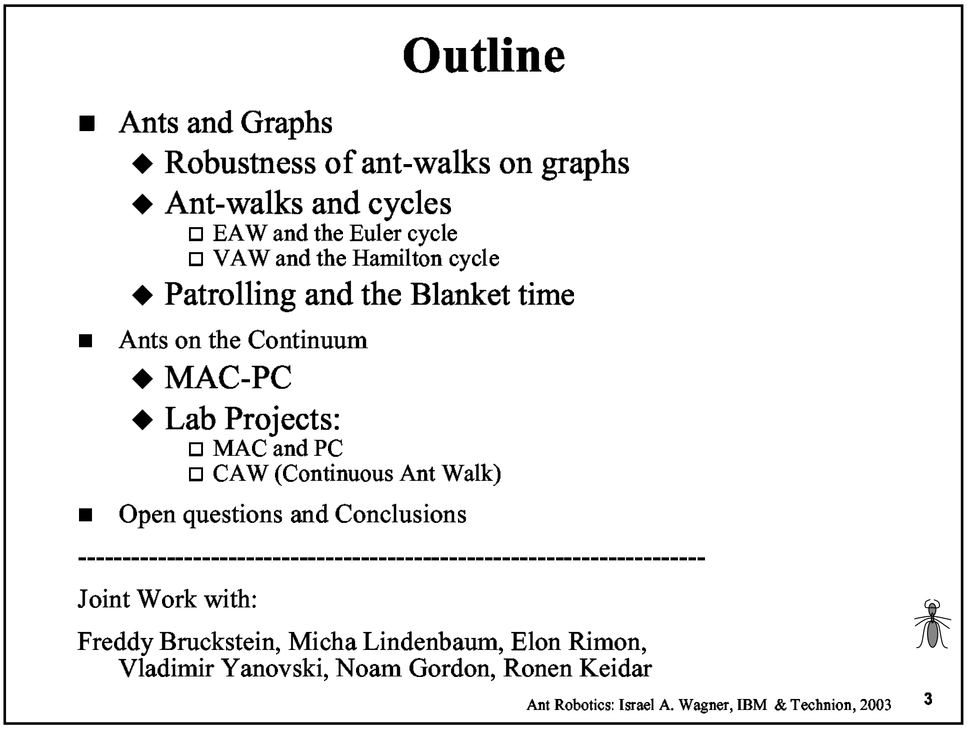
Intelligent Systems Lab

Computer-Science Department, Technion

http://www.cs.technion.ac.il/~wagner/

Motivation

- Observation: ants achieve impressive results, especially robustness, despite inherent limitations and frequent failures
- Questions:
  - What makes a group of simple, limited creatures much more efficient than any single one ?
  - Can we borrow ideas for robotics ?
- Our assumption:
  The ant's strength is related to its weakness:
  limited sensing, computing and communication implies:
  no single bit of information is critically important
- In the sequel: some examples of analysis, simulation and lab experiments, demonstrating the robustness of ant-robots

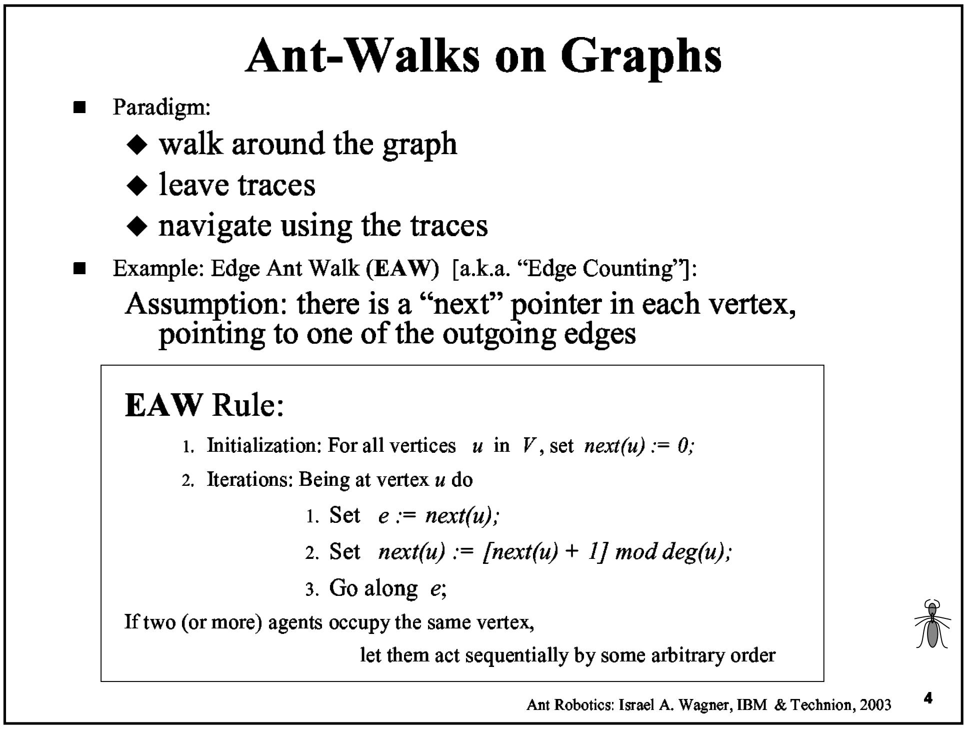


Outline

- Ants and Graphs
  - Robustness of ant-walks on graphs
  - Ant-walks and cycles
    - □ EAW and the Euler cycle
    - □ VAW and the Hamilton cycle
  - Patrolling and the Blanket time
- Ants on the Continuum
  - MAC-PC
  - Lab Projects:
    - □ MAC and PC
    - □ CAW (Continuous Ant Walk)
- Open questions and Conclusions

Joint Work with:

Freddy Bruckstein, Micha Lindenbaum, Elon Rimon,
Vladimir Yanovski, Noam Gordon, Ronen Keidar



Ant-Walks on Graphs

- Paradigm:
  - walk around the graph
  - leave traces
  - navigate using the traces
- Example: Edge Ant Walk (**EAW**) [a.k.a. "Edge Counting"]:

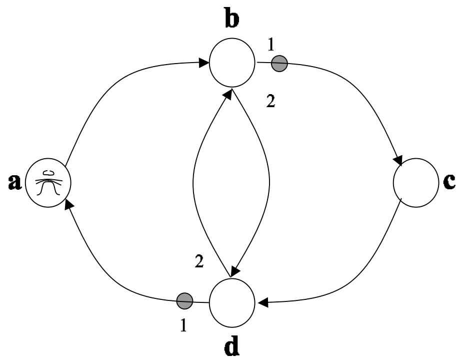
Assumption: there is a "next" pointer in each vertex, pointing to one of the outgoing edges

EAW Rule:

```
1. Initialization: For all vertices $u$ in $V$, set $next(u) := 0;$
2. Iterations: Being at vertex $u$ do
        1. Set $e := next(u);$
        2. Set $next(u) := [next(u) + 1] \bmod deg(u);$
        3. Go along $e$;
```

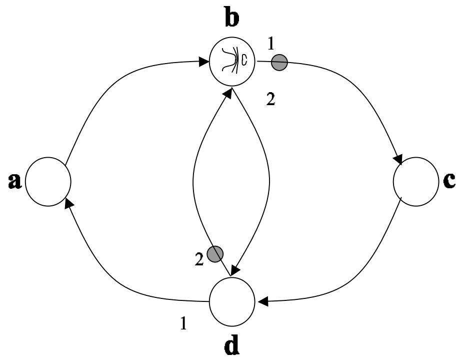
If two (or more) agents occupy the same vertex,
let them act sequentially by some arbitrary order

EAW - example



Path = *a*

EAW - example



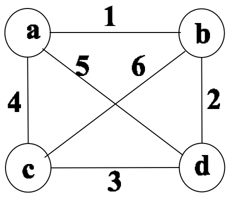
Path = *abcda{bdbcda}**

EAW and the Euler Cycle

- Assume an Edge Ant Walk on a graph $G(V,E)$
- Euler Cycle (**EC**) = a cycle in which each edge in E is visited *exactly* once
- Theorem:
  If ***G*** is Eulerian (*in-degree(u)=out-degree(u)* for all vertices *u*),
  then an **EAW** process converges to an Euler cycle
- How long does it take ?
  - ◆ $O(mn)$, where $m=|E|$, $n=|V|$
    [Bhatt, Even, Greenberg, Tayar (99)]
  - ◆ $O(mD)$, where $D=diam(G)$
    [Yanovski, Wagner, Bruckstein (99)]

EAW, *Euler* and *Extended Euler*

- What if ***G*** is non-Eulerian ?
- Extended Euler Cycle (**EEC**):
  a cycle in which all edges in E are visited
  the same number of times
- Example:
  a *non-Eulerian* graph:



an EEC of order 2: $\{123452635164\}^*$

(a cycle where each edge occurs exactly *twice*)

EAW, Euler and Extended Euler

- Theorem:
  The unified path of a group of ***k*** agents
  applying the **EAW** rule eventually enters
  an *Extended Euler Cycle* (of some order)

- How long does it take ?
  estimation with simulation: less than ***mD***

- Guess:
  time decreases with ***k***, but no proof yet.

- Application: *patrolling* a network, that is:
  continuously exploring all edges, as *uniformly* as possible

- Let us first consider a simpler method of (stochastic) patrolling …

Patrolling by a Random Walk

- An *simple random walk* on ***G(V,E),*** ruled by

$$\Pr(u \to v) = \begin{cases} \dfrac{1}{d(u)} & (u,v) \in E \\ 0 & \textit{otherwise} \end{cases}$$

is an *ergodic* process that eventually arrives at a *stationary distribution* **π** :

$$\pi(u) = \lim_{t \to \infty} p_t(u) = \frac{d(u)}{2m}$$

and the convergence (“mixing”) rate to π can be expressed by the <u>*blanket time*</u> ***B*** [Winkler , Zuckerman (96)]:

$$B = \text{first time } t \text{ for which } \forall u,v \; \frac{p_t(u)/p_t(v)}{\pi(u)/\pi(v)} < 2$$

Patrolling by a Random Walk

- Blanket time (edge version):
  The first time at which, for any two edges ***e, f*** in ***E***,

$$\frac{visits_t(e)}{visits_t(f)} \leq 2$$

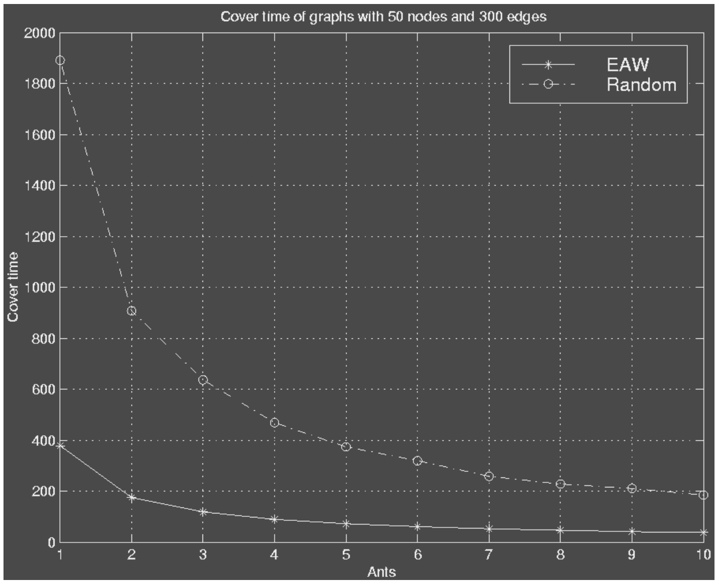
- by random walk:

  $E[B] = O(C\,(\log\log n)^2)$ (expected)

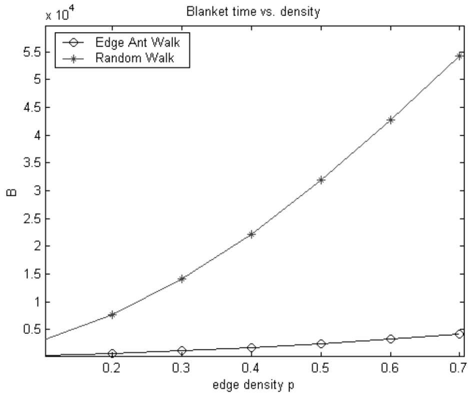
  where ***C*** is the expected cover time of ***G***
  [Kahn,Kim,Lovasz,Vu (FOCS'00)]
  Note: ***C*** can be as large as ***O(n³)*** [as in a *lollipop* graph]

- by **EAW**: *B < 2mD* (deterministic)

Simulations - MultiAgent Ant Walk



Simulations - Blanket time



(Averaged over 100 random(*p*) graphs with 20 vertices)

Hence:

Ant-Walk achieves a fast blanket-time, deterministically

(robustness against initial bad path)

VAW and the Hamilton Cycle

- Hamilton Cycle (**HC**) = a cycle in which each *vertex* in ***E*** is visited exactly once
- **HAM** problem:
  given a graph, does it contain an HC ?
  (a special case of the **Traveling Salesperson Problem)**
- **HAM** is Known to be ***NP*-complete**, hence not solvable in time ***poly(n)***, unless ***P=NP***
- Our **ant-inspired** approach (VAW, LRTA*):
  - repeatedly traverse the graph, marking the vertices
  - always go to the neighbor with minimum trace
- Can this “vertex” flavor of EAW lead to an HC ?

VAW and HAM

```
Rule Vertex-Ant-Walk (u vertex;)
    1. v := u's neighbor with minimal value of (s,t)
    2. s(u) := s(u)+1
    3. t(u) := t
    4. t := t+1
    5. go to v
end Vertex-Ant-Walk.
```

Analytic results:

- a Hamiltonian cycle is a *limit cycle* of VAW
- Some limit-cycles are *not* Hamiltonian

Simulation

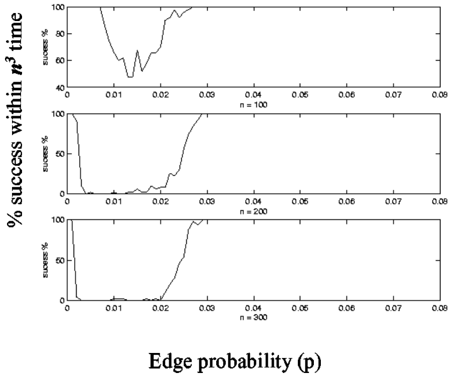
For most random Hamilton graphs, convergence to an HC is fast (within $\mathbf{n}^3$)

But – it depends on ***p*** –

the *edge probability*

Hence:

A possible heuristic for *HC*



Robustness: EAW on Faulty Graphs

- What happens to **EAW** if each edge fails with probability ***p*** ?
- Theorem:
  If each edge is prone to failure with probability ***p***,
  then

$$T^{EAW} \le mD\left[\frac{1+p}{1-p} + \frac{\log\Delta}{\log(1/p)}\right]$$

where

Δ = maximum vertex degree in ***G***

m = number of edges

D = diameter of ***G***

Covering a Continuous Domain

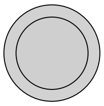
- A general covering problem:
  - Given: a set of *K* agents/robots, arbitrarily distributed in an unknown continuous region *R* of positive measure
  - Goal: determine a rule of motion for each robot, such that after following it for a period *T*, the region *R* is covered,
  - that is: find, for each robot *i*, a feasible set of points: $x_i(1), x_i(2), \ldots, x_i(T)$ such that

$$R \subseteq \bigcup_{i=1}^{K} \bigcup_{t=1}^{T} a(x_i(t))$$

  where *a(x)* is the area covered from point *x*.
  - In ant-robotics, same protocol applies to all robots
- Issues:
  - Algorithm
  - Bound on cover time with *K* agents, *T(K)*
  - Robustness to :
    - □ Sensing errors
    - □ Individual failures

Covering a Continuous domain

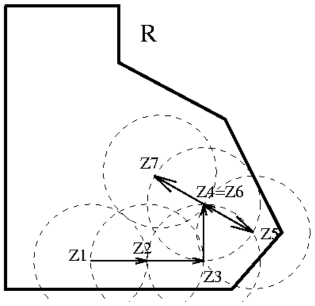
- **No floor marking or floor plan available**
- **The robot can sense obstacles, evaporating traces, and uncovered area (e.g. dirt)**
- **It can sense in the cleaning area and a little beyond.**



Mark and Cover (MAC) Algorithm:

(Inspired by *Ariadne Thread / DFS*)

- If there is an uncovered point in the sensing area:
  go there (and leave trace)
- Otherwise:
  use the trace to backtrack
- If nowhere to backtrack:
  stop



MAC - Analysis

- Observation:
  MAC is a **DFS** on the dynamically-built graph whose vertices are the circles' centers
- Lemma:
  Each *forward* step covers at least *a/2* new area
- Theorem:

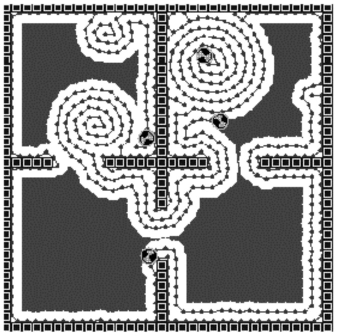
$$T^{MAC} \le 6.5\frac{A}{a}+\frac{2P}{r}$$

  where:
  - ◆ ***A** = area(R),* **a** = area covered by a robot in one step
  - ◆ **r** = robot's radius, ***P** = perimeter(R)*
- Note: the shape affects cover time via *P*

Continuous domain – cont.

- The robot returns to initial point. (+/-)
- Faster than predicted bound.
- Optimal path is NP hard.
- Sensitive to changes in floorplan.
- Cooperative behavior is present but speedup is lower.
- The resulting trace is a *spanning tree* of the region.

simulation



MAC vs. PC

- **PC** ("Probabilistic Covering") – a random walk in *R*
- Expected covering time is bounded:

$$E[T^{PC}] < 2\rho(A/a)\log(A/a)$$

where ρ is the maximum resistance in R,
if made of a material with unity sheet-resistance
(always less than the diameter)

- Hence:
  PC is *slower* than MAC but much more *robust* to:
  - Odometric / trail sensing errors
  - Calculation errors
- Our experience: MAC / PC hybrid is best

Theoretical Results: Upper bounds on cover time by ant-walks

domain	algorithm	marking	time
graph	VAW	vertices	$nd(1+p)/(1-p)$
	EAW	edges	$md(1+p)/(1-p)$
continuum	MAC	trail	$2P/r + 6.3(A/a)$
	PC	none	$2\rho(A/a)\log(A/a)$

k=#a(ge)nts, ρ=resistance, P=perimeter,
n=#vertices, m=#edges, d=diameter, p=failure prob.
A=total area, a=robot's area, r=robot's radius

Real Robots in Lab

We examined two ways to implement evaporating trails:

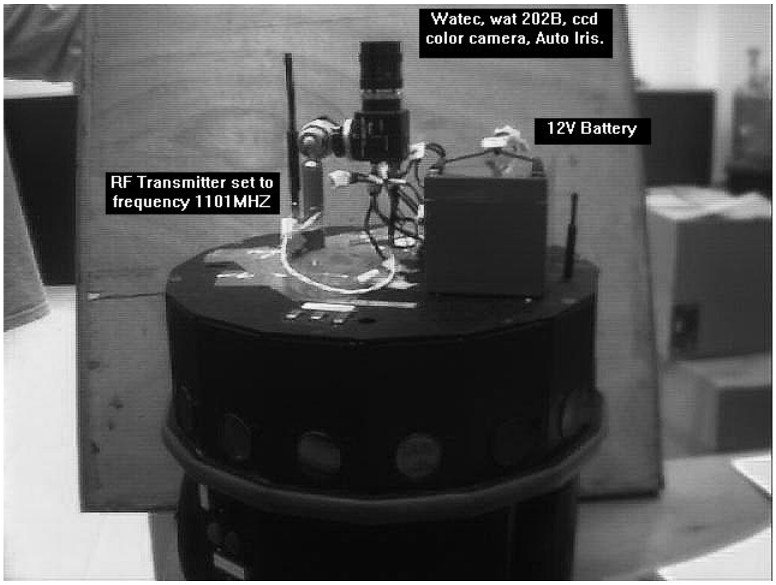
- Virtual traces in robot's memory:
  MAC robot with fiducials

  (IS Lab, CS Dept., Technion)

- Evaporating liquid on the floor:
  VAW robot with evaporating traces

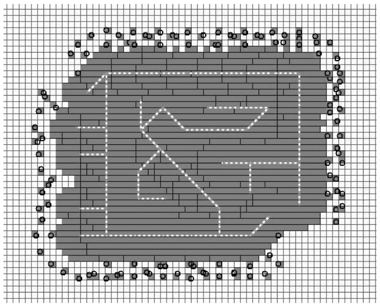
  (Mechatronics Lab, ME Dept., Technion)

MAC Robot – a close look

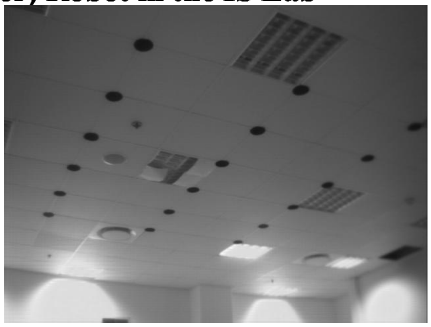

Watec, wat 202B, ccd color camera, Auto Iris.
12V Battery
RF Transmitter set to frequency 1101MHZ

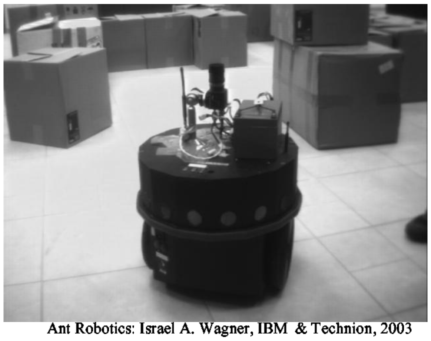
MAC (Mark And Cover) Robot in the IS Lab

- MAC **robot (NOMAD-based)**
- **Odometry-free navigation, using ceiling fiducials and memorized markings**



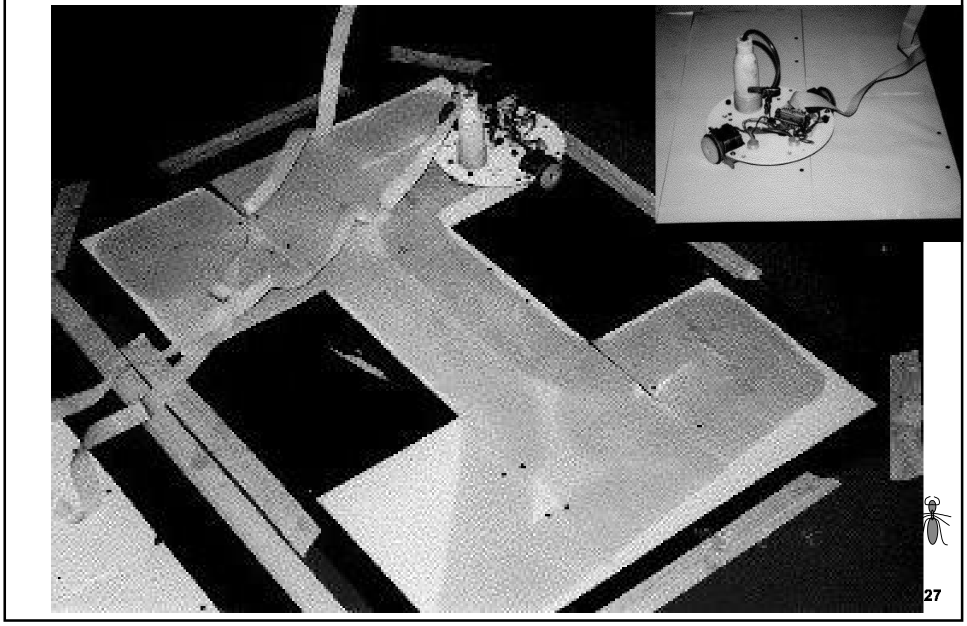
- **With:**
  Micha Lindenbaum,
  Yehuda Gan-El,
  David Yaron,
  Moran Shochat





A VAW Robot in the ME Lab

(with Elon Rimon, Yoav Gabrieli, Hod Katan & Guy Rogel)



Conclusion

VAW, EAW and MAC/PC algorithms show the advantages of Ant Covering, Cycling and Patrolling:

- **Covering without positioning**
- **Automatic, modular cooperation**
- **Simplicity enables redundancy and robustness to sensing errors, robot failures and environment changes**
- **Yet – many questions are still open …**

Some Open Questions

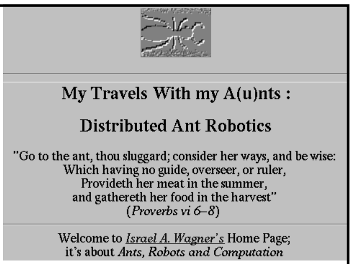
- **MultiRobot <u>speed-up</u>:**
  Upper bounds on *T(k)* for EAW, VAW and MAC

- **The effect of <u>randomness</u>:**
  choose the next step with probability
  (increase robustness to noisy sensors ?)

- **Apply to more complex <u>problems</u>:**
  e.g. as an an approach to cooperative problem solving
  (different from "Ant Colony Optimization" (ACO), founded by, Dorigo et al, using use global knowledge)

- **Shape <u>formation</u>:**
  bring the group to a desired configuration,
  without central control

Selected Publications



My Travels With my A(u)nts :

Distributed Ant Robotics

"Go to the ant, thou sluggard; consider her ways, and be wise:
Which having no guide, overseer, or ruler,
Provideth her meat in the summer,
and gathereth her food in the harvest"
(*Proverbs vi 6–8*)

Welcome to <u>*Israel A. Wagner's*</u> Home Page;
it's about *Ants, Robots and Computation*

- *Why the ant trails look so straight and nice ?*
  [A. M. Bruckstein, Mathematical Intelligencer 1993]
- *Smell as a computational resource – a lesson we can learn from the ant*
  [I. A. Wagner, M. Lindenbaum, A. M. Bruckstein, ISTCS 1996]
- *ANTS – A(ge)nts, Networks, Trees and Subgraphs*
  [I. A. Wagner, M. Lindenbaum, A. M. Bruckstein, FGCS 2000]
- *MAC vs. PC - Determinism and Randomness as Complementary Approaches to Robotic Exploration of Continuous Unknown Domains*
  [I.A. Wagner, M. Lindenbaum, A. M. Bruckstein, IJRR 2000]
- *A Distributed Ant Algorithm for Efficiently Patrolling a Network*
  [V. Yanovski, I. A. Wagner, A. M. Bruckstein, ANTS'2000]
- **Annals of Mathematics and Artificial Intelligence,**
  Special issue on *Ant Robotics*
  [I. A. Wagner, A. M. Bruckstein – Editors, 2001]
- And see also my URL for other publications, courses and on-line simulators:
  http://www.cs.technion.ac.il/~wagner/

David Payton
Richard Vaughan
HRL Laboratories
USA

Virtual Pheromone Robotics

David Payton
Richard Vaughan

HRL LABORATORIES

[payton,vaughan]@hrl.com

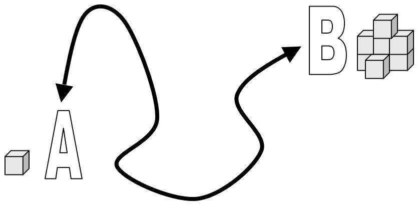
Overview

- We seek the benefits of Ant Algorithms while using more convenient robot communications technology
- We describe two systems that perform realistic tasks in unknown indoor environments
    1. An ant-like resource transportation team
    2. A swarm of simple robots that locates a target

  Both use conventional comms (serial IR and 802.11 wireless) to create environmental gradients

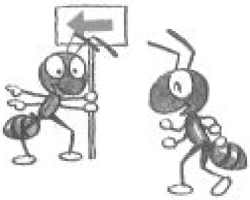
System 1:
Ant-like Resource transportation

- Team of N robots starts from a `home’ location A in an unknown environment
- Explore to find supply of resource at B
- Carry resource home and return for more
- Exemplified by foraging of ants & bees



Insect trail-following

- Ants lay & follow directional chemical trails - stigmergy
- Honey bees communicate parameters of successful missions to their sisters
- Pros: adaptive, robust, distributed, scalable
- Cons: not necessarily optimal, emergent, tricky mechanism

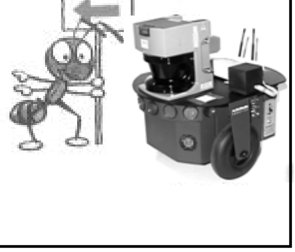


Trails in localization space

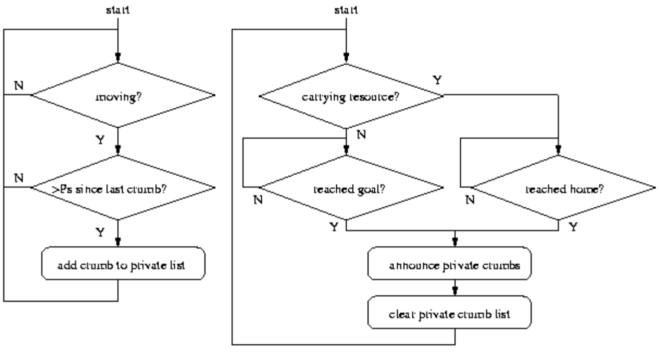
- Abstract away the chemical trail
- Communicate landmarks in shared localization space
- Will work as long as localization spaces are correlated to within the system's noise tolerance
- Pros: adaptive, robust, distributed, scalable + lightweight, simple
- Cons: not necessarily optimal, robots must be localized *(not trivial indoors, but many options exist)*

Crumb trails

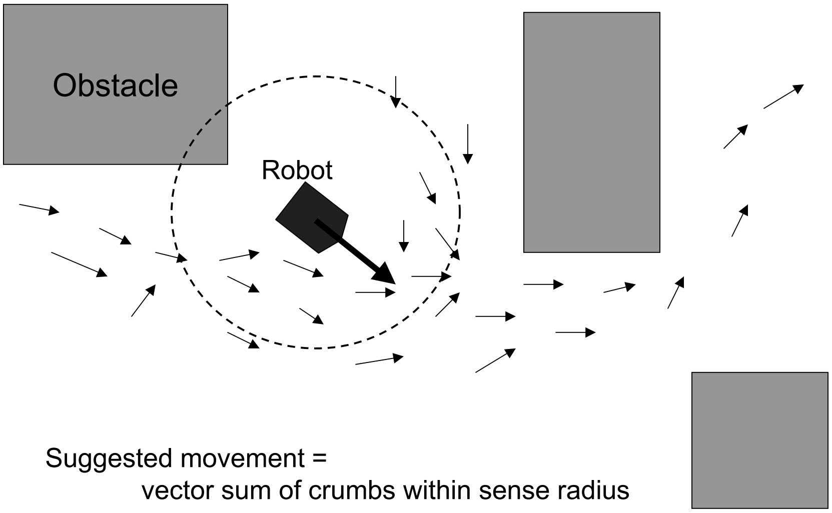
- A crumb is a landmark/signpost in localization space, indicating the direction to move next
- Trail represented by a linked list of "crumb" data structures
- Each crumb contains:
  - a position **(x,y)**
  - a suggested heading θ
  - a timestamp **t**



Trail laying algorithm

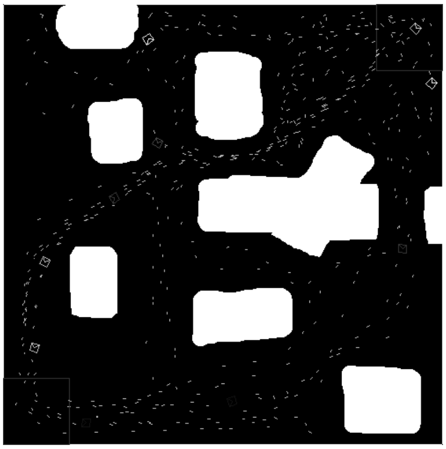


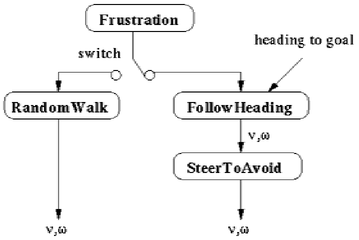
Trail reading algorithm



Suggested movement =
vector sum of crumbs within sense radius

Simulation study

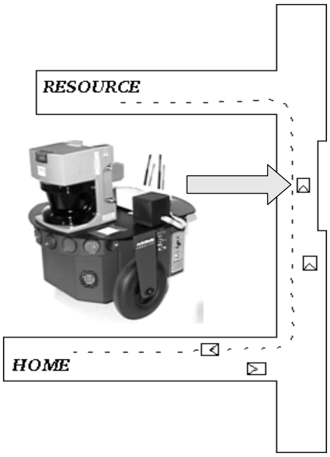




- Trail following outperforms random walk and directed search
- Tolerates significant localization error
- Emergent resistance to interference

Whistling in the Dark: Cooperative Trail-Following in Uncertain Localization Space
Vaughan, Stoy, Sukhatme & Matarić, Proc.Int.Conf Autonomous Agents 2000

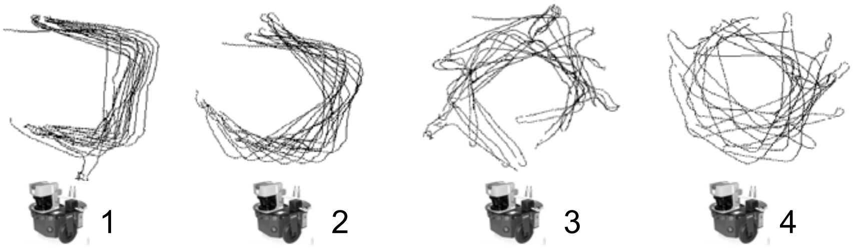
Real world evaluation



- Transfer the earlier results to real 4 x Pioneer 2 robot team in a structured environment
- Use trail-following with a different, specialized local navigation controller
- Communicate via wireless ethernet using UDP
- Localize using odometry
- How quickly will real-world odometric drift cause system failure?

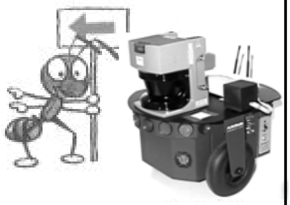
Results

- Four robots ran for 27 mins, completing 28 round trips of 60m each (total 1.68km) before accumulated odometry errors defeated system.
- The 4 robots performed 3 times better than a single ideal-performing robot.

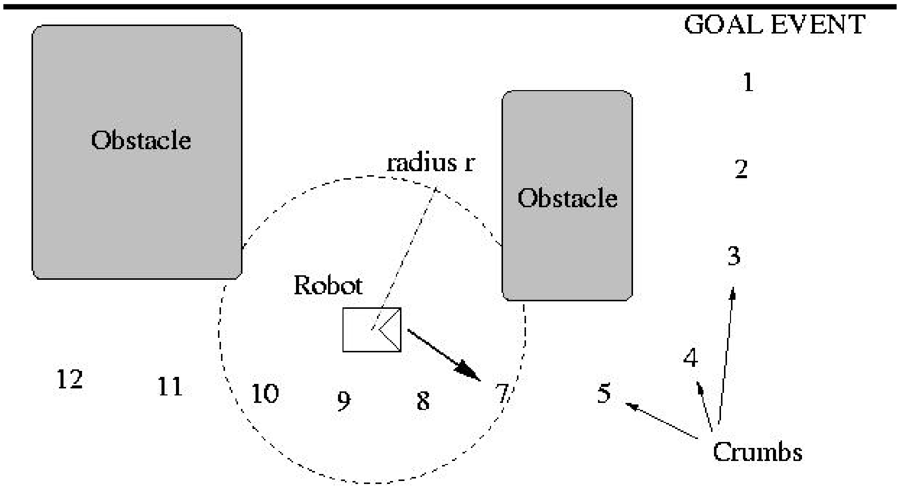


Self-correcting trails

- Robots agree on a set of task-relevant events (e.g. get-resource, drop-resource)
- A crumb is a landmark/signpost in localization space, indicating the distance to a goal event
- Each robot announces a crumb on the network every T seconds
- Received crumbs stored in a linked list, representing a 'trail'
- Each crumb contains:
  - an event name **E**
  - a position **(x,y)**
  - a distance-to-event estimate **d**
  - a timestamp **t**

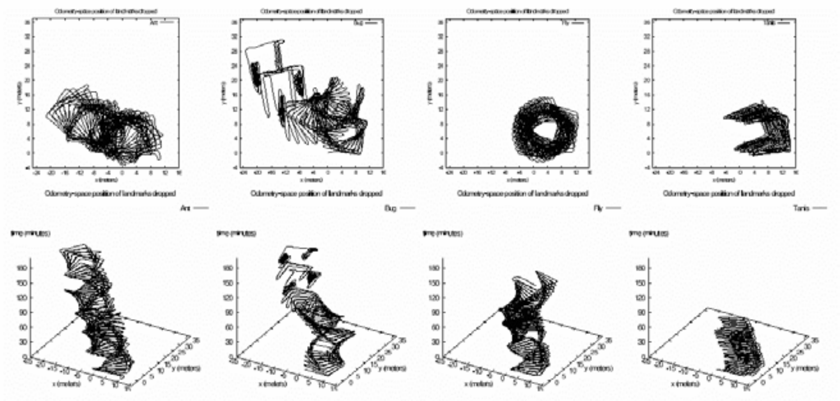


Using the trail



Suggested movement =
towards the crumb within *r* with lowest distance-to-goal

Results

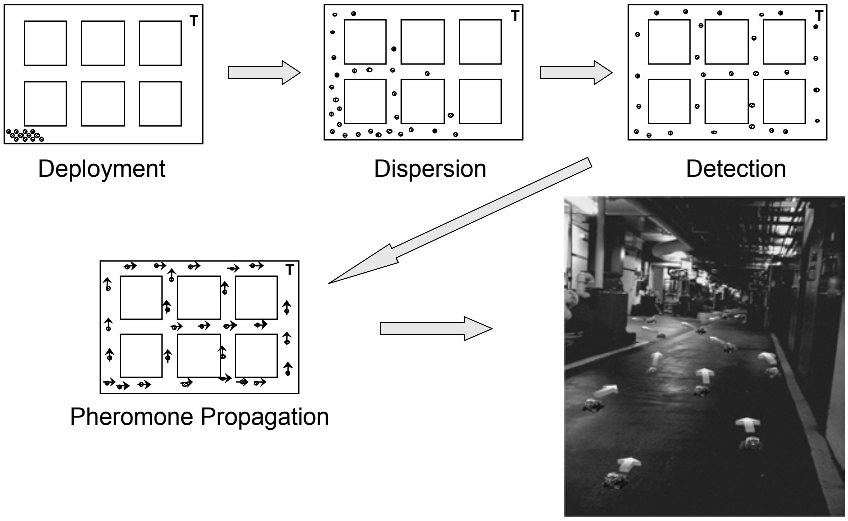


“LOST: Localization-Space Trails in Robot Teams”, Vaughan, Stoy, Sukhatme & Mataric, IEEE Transactions on Robotics and Autonomous Systems18:5 2002

Resource transport summary

- Demonstrated cooperative search and transport
  - 4 robots traveled 8.2km (5.1 miles) over 3hrs 10mins until batteries or other hardware failed
- The trail laying/following method
  - is robust wrt. real-world odometry error
  - scales to N sources, M sinks, O population
  - is independent of navigation strategy: therefore independent of robot hardware
  - Builds just enough world model for stigmergy
  - has low computation and bandwidth requirements (16 bytes/second per robot)

System 2: Swarm finds hidden target



Deployment

Dispersion

Detection

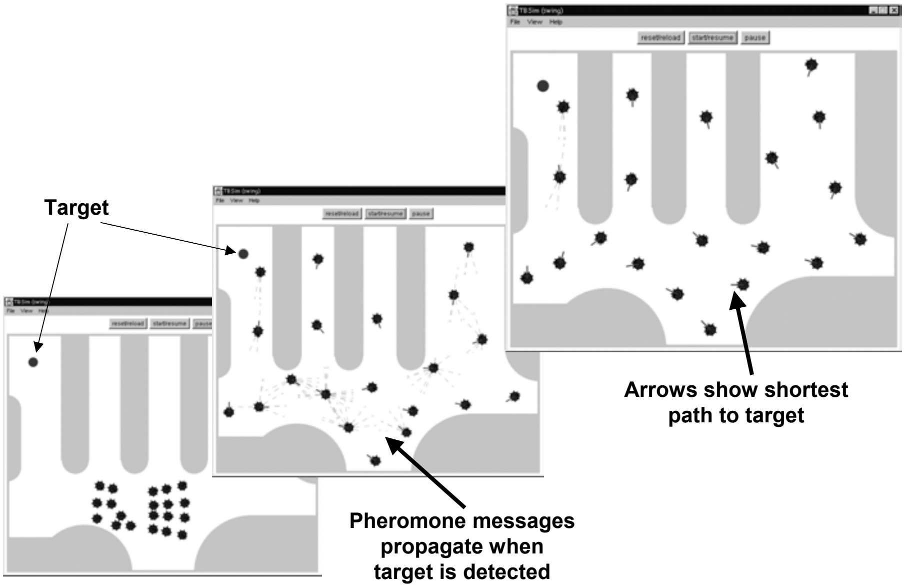
Pheromone Propagation

Path Visualization

Key Themes

- IR Virtual Pheromones
  - a message protocol that emulates pheromone diffusion
  - robot behaviors based on attraction & repulsion to gradients
- World Embedded Computation
  - the structure of a distributed computation is directly entailed by the environment
- World Embedded Display
  - robots distributed in the environment become pixels contributing to a larger overall picture

Dispersion With Pheromone Propagation

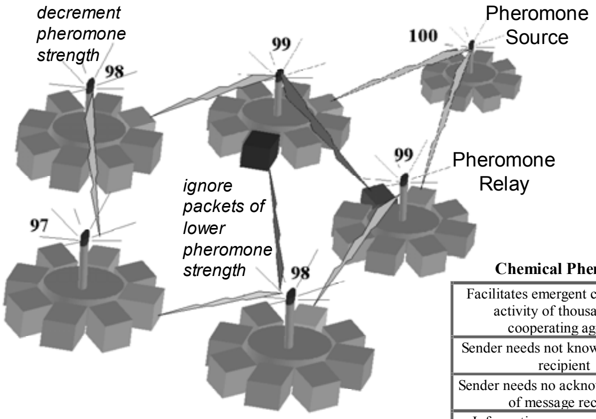


Target

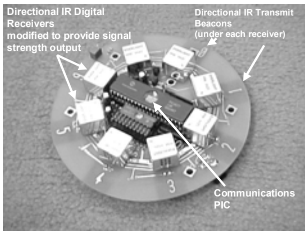
Pheromone messages propagate when target is detected

Arrows show shortest path to target

IR Virtual Pheromones


decrement pheromone strength
98
99
100
Pheromone Source
99
Pheromone Relay
ignore packets of lower pheromone strength
97
98

Transceiver Hardware

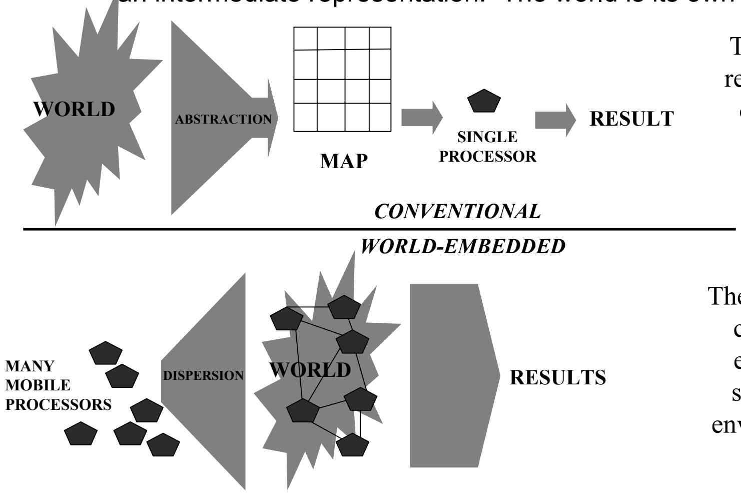

Directional IR Digital Receivers modified to provide signal strength output
Directional IR Transmit Beacons (under each receiver)
Communications PIC

Chemical Pheromones	Virtual Pheromones
Facilitates emergent coordinated activity of thousands of cooperating agents	yes
Sender needs not know identity of recipient	yes
Sender needs no acknowledgement of message receipt	yes
Information conveyed through intensity gradients	Yes, but by using intensity-tagged message packets
Propagation is sustained passively through the environment	Propagation is sustained by active relaying between robots
Different chemical types needed for different messages	Messages may be tagged with additional data

The virtual pheromone is inspired by properties of chemical pheromones used by social insects such as ants and termites.

World-Embedded Computation

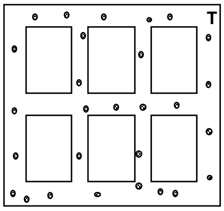
Many complex graph-theoretic algorithms can be performed without creating an intermediate representation. The world is its own map.



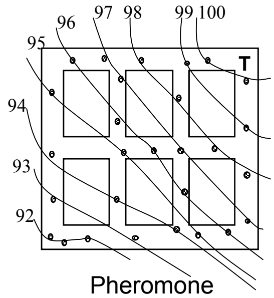
The computation remains the same, only the data is changed.

The structure of the computation is entailed by the structure of the environment itself.

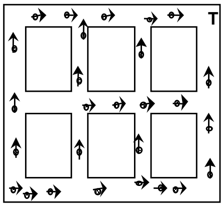
World-Embedded Computation of Dijkstra's Algorithm



Dispersed
Robots



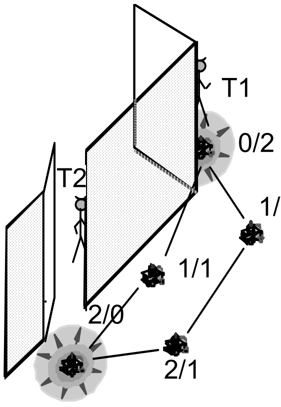
Pheromone
Wave Front

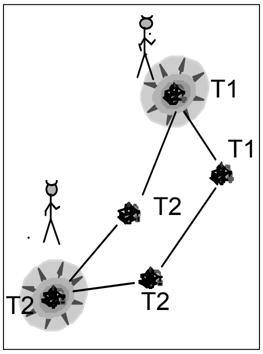


Local Gradient
Points Toward
Objective

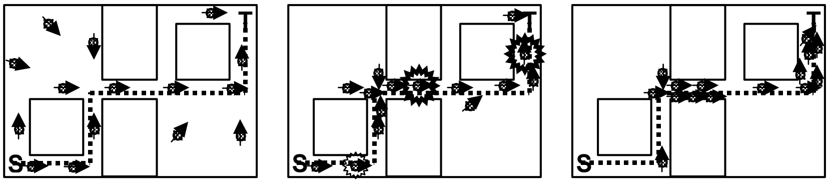
Using Multiple Pheromone Gradients

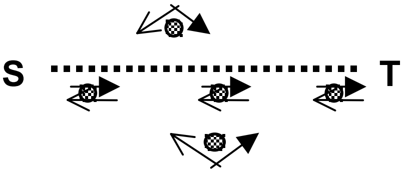
Overlapping pheromone wavefronts allow each robot to determine which target is nearest.





Multiple Gradients to Find Choke Points

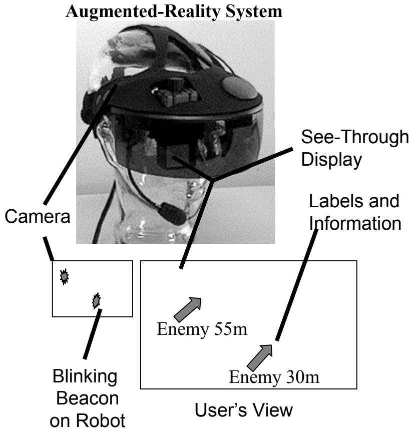




Opposing pheromone gradients drive robots toward median

World-Embedded Display

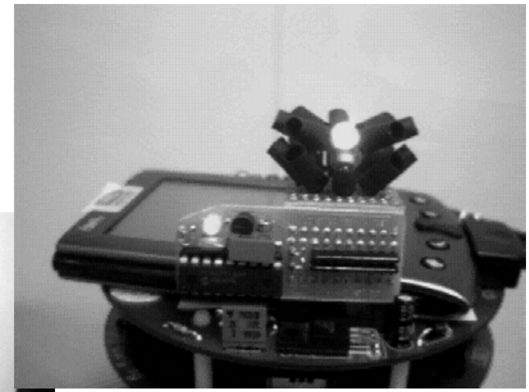
Information about the environment can be conveyed to the user even though robots have no explicit maps of their locale.



User's View Through Display

Augmented Reality Mast

Transmits 8 distinct codes in 8 directions



Codes convey a gradient arrow that retains constant orientation from any viewing angle

Augmented Reality HMD

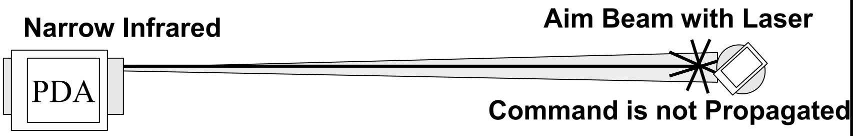
Virtual IO Display uses Pencil Cam For Tracking IR

User Sees Virtual Pheromone Vectors Overlaid Over Real World

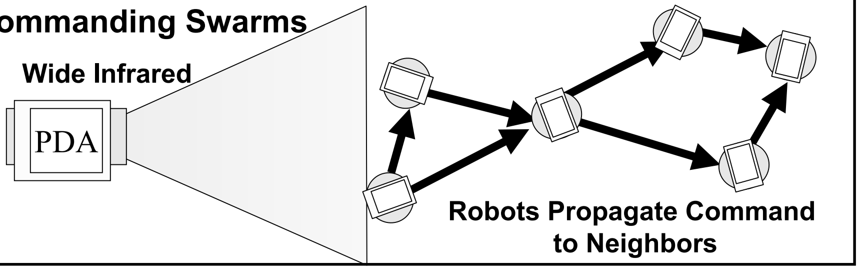
Commanding Robot Swarms

Command Pheromone
allows switching between different control systems

- **Commanding Individual Robots**

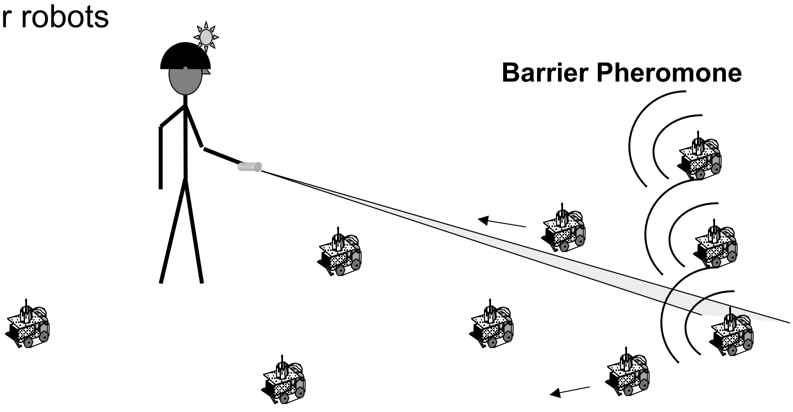


- **Commanding Swarms**



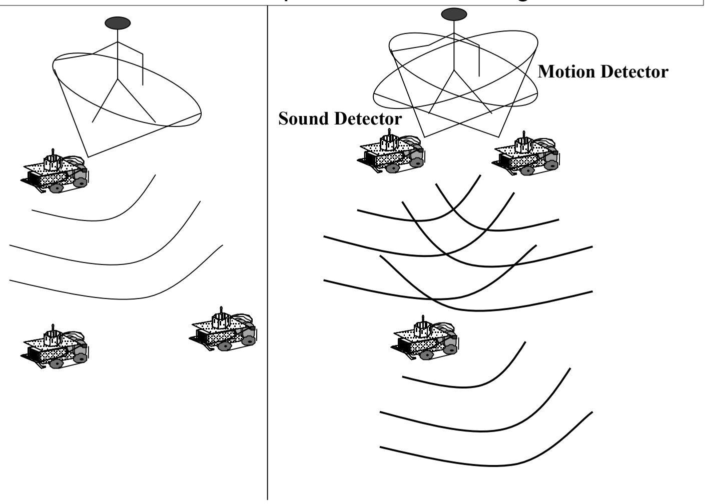
Controlling Dispersion

The handheld beacon triggers remote robots to halt
and emit a repelling pheromone, creating a barrier
to other robots



Multi-Sensor Detection From Combined Pheromones

Reinforcing pheromones reduce false positives in detecting an intruder.



Conclusion

- We introduce **Virtual Pheromones**:
  - frequent, small packets of data encoding a ‘distance’
  - propagated via an artificial medium
  - implement parallel, local, gradient-based algorithms
- Our systems demonstrate how VP can create coordinated global behavior in teams/swarms of robots
- The example systems are very different
  - Broadcast vs. line-of-sight comms
  - Ant algorithm vs. Dykstra’s algorithm
  - Abstract (localization) space vs. world-embedded
- **The VP technique:**
  - **Combines attractive features of Ant Algorithms with convenient communications technology**
  - **Extends to non-ant algorithms**

Andrew Russell
Monash University
Australia

1. Ant robot hardware

In order for a robot to be able to work with physical chemical trails it will require a sensor system to detect and accurately determine the location of chemical markings on the ground. The robot will also need the means to deposit volatile chemicals in a controllable way.

1.1.2 Tin oxide sensors

There are many sensors that respond to volatile chemicals. However, very few are suitable for use in robotic applications. Taking into account physical size, response time, sensitivity, cost, robustness, and power consumption, three technologies are commonly used in robotics. Probably the most convenient of these chemical sensors are those based on the changing conductivity of tin oxide.

1.1 Chemical sensors

Tin oxide sensors are available commercially. They are inexpensive, small in size, easy to interface and the latest miniature devices made by thick-film technology have relatively low power consumption. Some research groups have manufactured their own sensors. The tin oxide sensor element is heated to about 300˚C where the presence of reducing gasses causes a drop in sensor resistance. Sensitivities of the order of 1 ppm are claimed and sensors have been developed with responses optimized for solvents, halocarbons, and a range of combustible gasses. The relationship between sensor resistance and the concentration of detected gas is non-linear. Over a limited range of concentrations this relationship can be approximated by (Watson, 1984):

$$R \cong K R_o C^{\alpha} \qquad (1.1)$$

where

α = sensitivity
R = sensor resistance on exposure to target chemical
R_O = sensor resistance in clean air
K = scaling constant
C = concentration

As shown in Figure 1.1, a simple potential divider provides a signal that can be directly digitized by an analogue to digital converter.

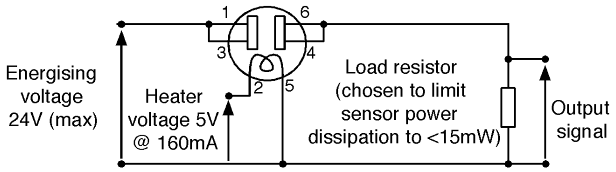

1
6
3
4
2
5
Energising voltage 24V (max)
Heater voltage 5V @ 160mA
Load resistor (chosen to limit sensor power dissipation to <15mW)
Output signal

Figure 1.1 Circuit to energize the Figaro TGS800 air quality sensor (redrawn from Figaro advertising material).

1.1.2 Quartz crystal microbalance

Quartz crystal microbalance sensors use a quartz crystal as a sensitive balance to weigh odor molecules. A chemical coating on the crystal is chosen to have a specific affinity for the target odorant molecules. When air containing molecules of this odor is drawn over the crystal some of the molecules become temporarily attached to the coating. This increases the effective crystal mass and lowers its resonant frequency. A simple model proposed by Sauerbrey predicts the effect of small amounts of added mass (Lu, 1984):

$$\Delta f = -\frac{2f^2\Delta m}{\rho v} \qquad (1.2)$$

where:

Δf	= change in crystal resonant frequency
f	= crystal resonant frequency
Δm	= increase in mass of the crystal per unit area,
ρ	= density of crystal material
v	= velocity of sound waves in the crystal material

A crystal coated with silicone OV-17 and camphor as target odorant provide a good combination for robotics experiments (Deveza, *et al.* 1994). However, a wide range of different coatings have proved useful. Here is a list of coating materials given by Moriizumi, Nakamoto, and Sakuraba (Moriizumi, *et al*. 1992):

- dioleyl phosphatidylserin,
- sphingomyelin (egg),
- lecithin (egg),
- cholesterol,
- polyethylene glycol,

- ethyl cellulose, and
- acetyl cellulose.

For the quartz crystal microbalance frequency shift is reported to be proportional to target odorant concentration. The circuits shown in Figure 1.2 produces a digital output at the difference frequency between a reference 10MHz oscillator (SG351P) and the sensor oscillator (HA7210IP). The low-pass filter on the output of the SN74HC74 removes spurious high frequency transitions caused by metastable failure.

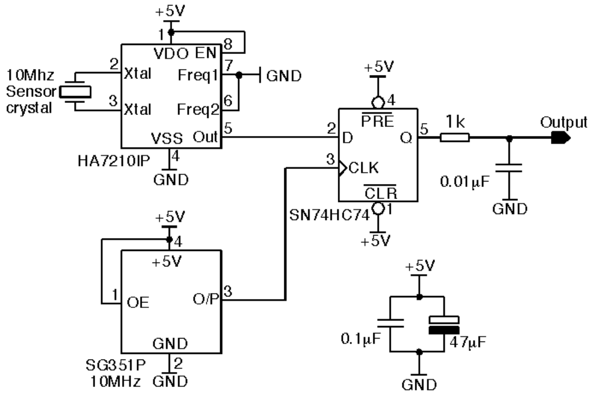


Figure 1.2 Quartz crystal microbalance sensor circuit.

1.1.3 Conducting polymer sensors

Conductive polymers have also been used to make chemical sensors for use in robotics. A cheap and safe method of making polypyrrole has been developed using electrochemical polymerisation of pyrrole between a pair of electrodes. The materials and processing techniques used to make conductive polymer sensors are cheap and readily available. Conductive polymer sensors are also easy to use for robotics applications being very small in size and exhibiting a substantial resistance change on exposure to volatile chemicals. Typical device exposed to methanol produced a 14% change in resistance with a response time to go from 0 to 95% of the final value of 24 seconds. However, there are problems of consistency between sensors and of stability and aging. The circuit shown in Figure 1.3 converts a change is sensor resistance into a varying frequency output. There is no d.c. component in the voltage across the polymer sensor.

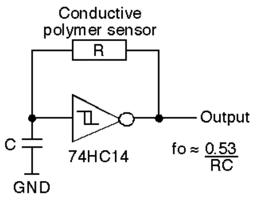


Figure 1.3 Conducting polymer sensor oscillator.

1.2 The control of airflow

Ants detect pheromone markings by tapping their antennae on the ground. At the interface between a solid surface and a fluid there is a thin layer of fluid that is essentially immobile. Volatile chemicals can only pass through this layer by a process of diffusion. The relatively slow rate of diffusion means that the chemicals within this layer of stagnant air do not move very far parallel to the surface and remain close to the source. When the ant's tapping antenna penetrates the stagnation layer it can gather reliable concentration readings for localizing the chemical markings. Larger scale robots measure chemical concentration at a much greater distance from the surface where local airflow can affect the measurements.

The fluid flow simulation shown in Figure 1.4 demonstrates that air drawn over a sensor crystal can originate some distance from the sensor inlet. In this case the sensor system is very poor at localizing chemical markings on the ground.

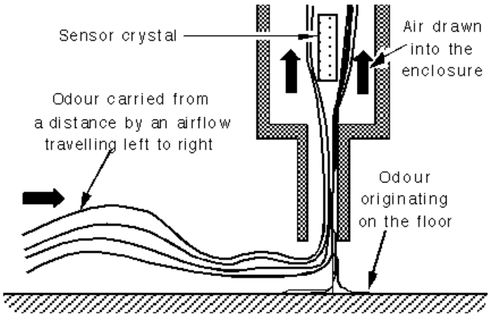


Figure 1.4 A simulation of airflow around an aspirated sensor. Four chemical particles originate on the ground below the sensor and four others are carried towards the sensor by a current of air travelling left to right.

When a chemical sensor moves up-wind towards a chemical marking on the ground erroneous results are produced as the chemical is detected well before the sensor arrives at the trail (Figure 1.6). A solution to this problem has been developed which involves providing an air curtain surrounding the sensor inlet. This establishes an outward current of air that prevents external chemical getting to the sensor inlet. In Figure 1.5 a fluid flow simulation shows that the air curtain deflects chemical carried towards the sensor by an external airflow.

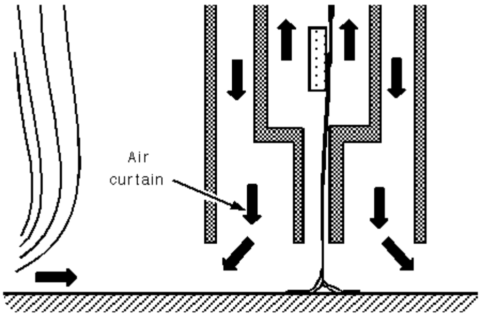


Figure 1.5 Simulation of the effect of an air curtain around the sensor inlet. Chemical particles introduced on the left-hand side are excluded from the sensor.

The improved localization capability of the air-curtain sensor is demonstrated in Figure 1.6. Once again the sensor approaches a chemical marking in an up-wind direction. In this case the sensor does not respond to the chemical until it is close to the marking.

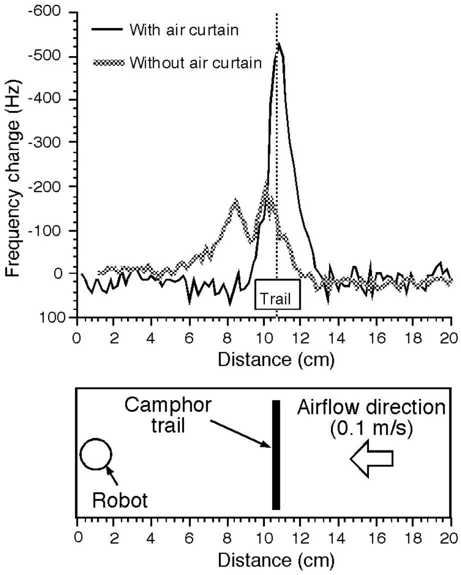


Figure 1.6 Response of a quartz crystal microbalance sensor with and without air curtain approaching in an up-wind direction and then passing over a camphor trail at 0.2 cm/sec.

1.3 Laying chemical trails

A simple solution for depositing chemical markings on the ground is based on a felt-tipped pen (Russell, 1999b). As shown in Figure 1.7, a plug of felt is soaked in the marking chemical (camphor dissolved in ethanol in this case) held in a reservoir. A motorized cam lowers the pen to the floor when a trail is to be deposited.

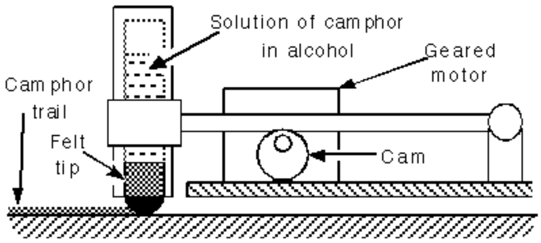


Figure 1.7 A simple chemical applicator.

The felt-tipped pen applicator is simple but has the drawback that there is no control over the quantity of chemical deposited. By using the metered pump arrangement shown in

Figure 1.8 there is much more control of the deposition of chemical and the supply of chemical can be completely cut off which is not possible with the simple applicator.

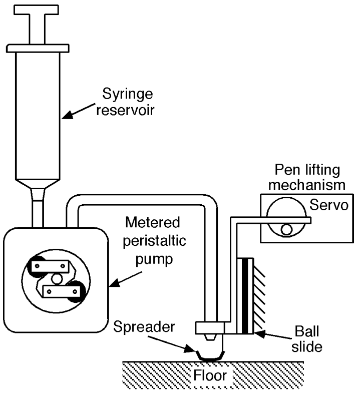


Figure 1.8 Metered camphor applicator.

References

Deveza, R., Thiel, D., Russell, R.A. and Mackay-Sim, A. 'Odor sensing for robot guidance', The *International Journal of Robotics Research,* Vol. 13, No. 3, June 1994, pp. 232-239.

Lu, C., 'Theory and practice of the quartz crystal microbalance', in *Applications of Piezoelectric Quartz Crystal Microbalances*, (C. Lu and A.W. Czanderna eds.), Elsevier, 1984, pp. 19-61.

Moriizumi, T., Nakamoto, T. and Sakuraba, Y., 'Pattern recognition in electronic noses by artificial neural network models', in *Sensors and Sensory Systems for an Electronic Nose*, J.W. Gardner and P.N. Bartlett (eds.), Kluwer Academic Publishers, 1992, pp. 217-236.

Russell, R.A., 'Laying and sensing odor markings as a strategy for assisting mobile robot navigation tasks', *IEEE Robotics and Automation Magazine*, September 1995, pp. 3-9.

Russell, R.A. *Odour Sensing for Mobile Robots*, World Scientific, ISBN 981023791X, 217pp. 1999b.

Watson, J., 'The tin oxide gas sensor and its applications', *Sensors and Actuators*, Vol. 5, 1984, pp. 29-42.

2 Ant robot software

Mobile robots are being employed in increasing numbers as a means of transporting material and partly finished assemblies within factories. Such robots are known as automated guided vehicles (AGVs) and most find their way around by following some sort of permanent guide path such as a reflective tape or buried wire. Chemical trails can serve a similar purpose by acting as a short-lived navigational marker. However, the characteristics of the available chemical sensors means that care must be taken in developing the method of trail following. A number of research groups have formulated algorithms to allow a robot to track a chemical trail.

2.1 The robot vehicle Sauro

At the Instituto Elaborazione Segnali ed Imamagini in Bari, Italy, Ettore Stella and co-workers have built a chemical trail following robot that uses alcohol as a chemical marker and conductive polymer sensors to detect the trail (Stella, *et al.* 1995). This vehicle, called Sauro, uses a pair of chemical sensors spaced 10 cm apart and located 1 cm above the ground to follow a 15 cm wide felt strip soaked in alcohol. The control algorithm consists of starting the vehicle with both sensors over the felt strip. As the robot proceeds it might deviate sufficiently that one sensor loses contact with the strip. The resulting fall in response from the sensor causes the robot to turn to bring the sensor back onto the tape (Figure 2.1). Using this algorithm Sauro has been able to follow the tape with a tracking speed of 60 mm/sec.

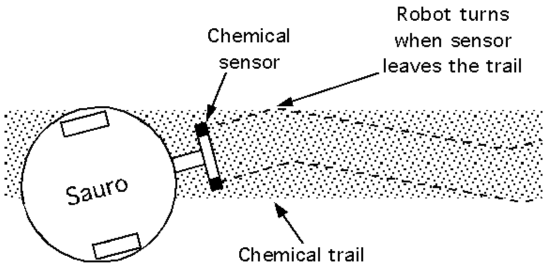


Figure 2.1 The robot Sauro follows a chemical trail using two sensors.

2.2 Robot control by neural network

Barbara Webb from the University of Nottingham investigates neural control in animals. By implementing the control schemes using mobile robots the capabilities of the schemes can be tested. In the area of chemical sensing Webb has developed a neuron model of ant osmotropotaxis (pheromone gradient following) and tested it using a small mobile robot built from Lego bricks (Webb, 1998). On this robot two Figaro chemical sensors played the part of the ant's antennae. The output from each chemical sensor was processed through seven neurons to produce a temporal difference signal. Difference values from left and right sensors were then combined with a speed signal to drive the left and right wheel motors. The result of this was to turn the robot towards higher chemical concentrations and away from lower concentrations. Unreliable chemical readings caused by air movements were stated to be a problem in these experiments.

2.3 Ant-like trail following

The trail-following algorithm employed by the ant *Lasius fuliginousus* seems to be very simple (Hangartner 1967). The normal mode of trail following for the ant is to curve to the left until the right antenna detects the trail and then to curve to the right until the left antenna contacts the trail. In this way the ant maintains the trail between its two antennae. An important feature of the ant's tracking algorithm involves turning back towards the trail after an extended period even if the appropriate antennae does not detect the pheromone (see Figure 2.2). This prevents the ant walking in circles when it loses the trail. Instead the ant moves forward in the general direction of the trail sweeping widely to left and right which gives a good chance of picking up the trail again. The following algorithm was derived from this and implemented on a six-legged robot having two chemical sensors mounted ahead of the robot simulating antennae (Russell, 1999a).

- Take 8 paces bearing to the right until completed or until trail detected by the left antenna.
- Take 8 paces bearing to the left until completed or until trail detected by the right antenna.
- Repeat

(If the left sensor detects the chemical trail then the sequence of 8 paces to the left is started or restarted and stimulation of the right sensor starts or restarts the sequence of 8 paces to the right)

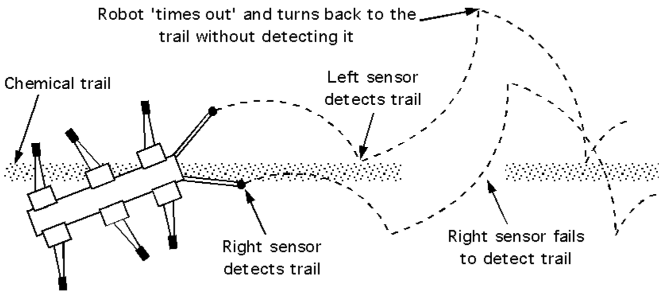

Robot 'times out' and turns back to the trail without detecting it
Chemical trail
Left sensor detects trail
Right sensor detects trail
Right sensor fails to detect trail

Figure 2.2 Robot trail following based on the ant *Lasius fuliginousus.*

2.4 Trail following with a single sensor

If the ant *Lasius fuliginousus* has a damaged antennae then the algorithm described above will still allow it to follow a pheromone trail with its remaining good antennae. A modification of this 'one antennae' algorithm is effective in guiding a mobile robot along a chemical trail (Russell, *et al.* 1997). In order to find the trail the robot moves forwards in a straight line ('u' in Figure 2.3). Upon detecting the trail the robot continues moving straight forwards 'v' to cut across the trail. The slow recovery of the chemical sensor means that edge of the trail is not detected until the robot is 'w' past the true edge. The robot then follows the left-hand edge of the trail using the following algorithm:

- Starting away from the odor trail, rotate 'x' until the edge of the trail is detected . Once again the slow response of the sensor means that the robot moves past the edge.
- Rotate a fixed distance 'y' away from the edge of the trail.
- Move forward a fixed distance 'z' .
- Repeat

This algorithm has been developed to ensure that the chemical sensor is given the maximum time to recover after being exposed to the chemical trail. After detecting the chemical the sensor is moved clear of the trail by quickly rotating distance 'y'. The robot then waits at this point until the sensor reading has returned close to its clean air reading before continuing.

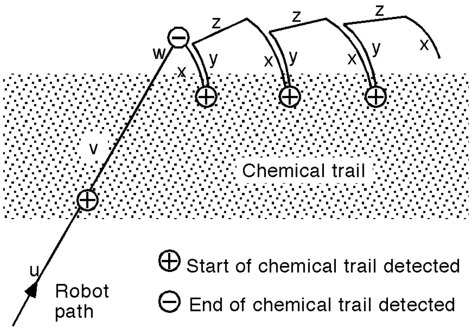

z
z
z
w
x
y
x
y
x
y
x
v
Chemical trail
u
Robot path
⊕ Start of chemical trail detected
⊖ End of chemical trail detected

Figure 2.3 Following a chemical trail with a single sensor.

If a robot has the ability to lay and to detect chemical markings it is possible to formulate a number of generic applications for this technology (Russell, 1999b).

2.5 Virtual umbilical

A mobile robot may be required to explore a partially known or unknown environment. This could be the site of an accident where no accurate maps are available or where an explosion has damaged the building. It would usually be essential that the robot finds its way back to the starting point at the end of its mission. If the robot paid out an umbilical cable on the outward journey it could follow the cable to return to its starting point. However, umbilical cables tend to snag and the equipment to pay out and retrieve an umbilical is bulky and cumbersome . The navigational effect of the cable can be gained by laying a trail of volatile chemical as illustrated in Figure 2.4. To return to its starting point the robot simply follows the trail it has laid. Following the trail will be made more difficult if the trail crosses itself or if the trail is crossed by the trail laid by another robot.

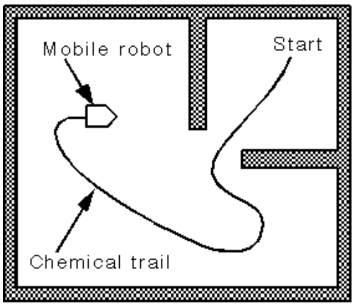


Figure 2.4 Finding the way back to the starting point - the virtual umbilical

2.6 Repellent marking

Many useful applications of mobile robots involve sterilizing, cleaning or searching large areas of floor. Floor area which has already been covered can be marked with a volatile chemical. The robot will avoid the marked areas and concentrate on new areas which have not been covered. Thus large areas can be cleaned, sterilized or searched without the need for accurate localization of the robot. A spiral area coverage technique using a repellent marking is illustrated in Figure 2.5. To start with the robot tracks around the room keeping its left side to the wall. When the robot meets the chemical trail it changes to keeping the trail on its left side. In this way the robot spirals in towards the center of the room covering all of the floor area. For complicated or cluttered rooms this basic algorithm will require further development.

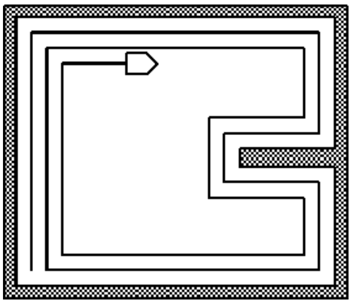


Figure 2.5 Efficient area coverage - the repellent marking

2.7 Pathfinder

One robot may lay a trail of volatile chemical in order to pass information to other robots. If a large quantity of material is to be transported a pathfinder robot could establish a clear path from the start to the goal. The path finder robot would be equipped with a range of sensors and the 'intelligence' to find a clear path through a cluttered and changing environment. This robot lays a trail of volatile chemical to mark its path. The bulk of the material is then transported by simple slave robots which only have sensors to follow the chemical trail and a proximity sensor to avoid bumping into the robot in front if it slows down. In Figure 2.6 the pathfinder robot is labeled 'A' and the slave load carrying robots 'B' and 'C'.

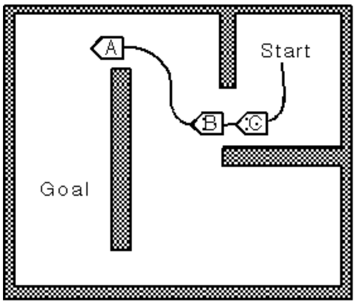


Figure 2.6 Marking a trail for others to follow - the pathfinder.

Is also envisaged that chemical markings could be used to warn of nearby hazards or to recruit individual robots from a circulating swarm of robots so that they can provide assistance with some task

References

Hangartner, W. 'Spezifität und inaktivierung des spurpheromons von *Lasius fuliginosus* Latr. und orientierung der arbeiterinnen im duftfeld', *Zeitschrift für vergleichende Physiologie,* Vol. 57, 1967, pp. 103-136.

Russell, R.A., 'Ant trails - an example for robots to follow?' *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, Michigan, May 1999a, pp. 2698-2703.

Russell, R.A. *Odour Sensing for Mobile Robots,* World Scientific, ISBN 981023791X, 217pp., 1999b.

Russell, R.A., Thiel, D. and Mackay-Sim, A., 'Recruiting swarm robots using coded odour trails', *Proceeding of the International Conference on Field & service Robotics*, Canberra, 8-10 December 1997, pp. 472-476.

Stella, E., Musio, F., Vasanelli, L. and Distante, A., 'Goal-oriented mobile robot navigation using an odour sensor', *Proc. 1995 Intelligent Vehicles Symposium*, pp. 147-151.

Webb, B., ‘Robots, crickets and ants: models of neural control of chemotaxis and phonotaxis’, *Neural Networks*, Vol. 11, No. 7-8, October/November 1998, pp.1479-1496.

3 Actual implementations

3.1 Implementing the *Lasius fuliginousus* algorithm (Russell, 1999a)

To provide an approximation to the form of an ant a six-legged robot vehicle has been constructed. This robot is based on Genghis, a walking robot developed at MIT (Brooks, 1989). Each of its six legs has two degrees of freedom and is actuated by radio control servo's. Total length of the robot is 22 cm. For the implementation of the *Lasius fuliginousus* algorithm two chemical sensors positioned 10 cm ahead of the robot and 13 cm apart simulate the pheromone sensing capabilities of the ant's antennae. A 68HC11 microcontroller generates the walking action of the robot, monitors the two chemical sensors and implements the *Lasius fuliginousus* algorithm. Figure 3.1 shows a photograph of the robot.

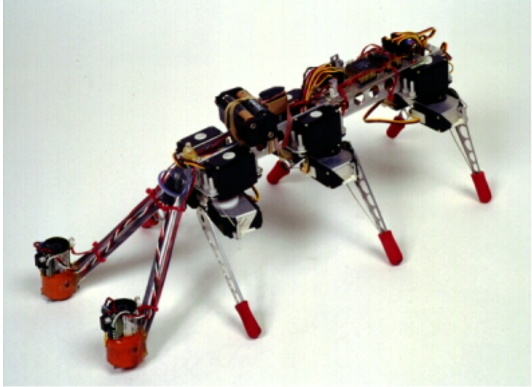


Figure 3.1 The ant robot showing its two chemical sensing antennae.

One of the important advantages of the *Lasius fuliginousus* algorithm is the ability to accommodate large gaps in the chemical trail. This is not surprising because the pheromone trails produced by biological ants are made up of many discrete odor patches and probably have regular gaps. To test the capabilities of this algorithm a camphor trail was laid on the floor consisting of a 55cm straight section, a gap of 40 cm and then a further straight section finishing with a curve. The ant robot successfully followed the camphor trail in both right to left and left to right directions. Figure 3.2 shows the trajectory of the robot.

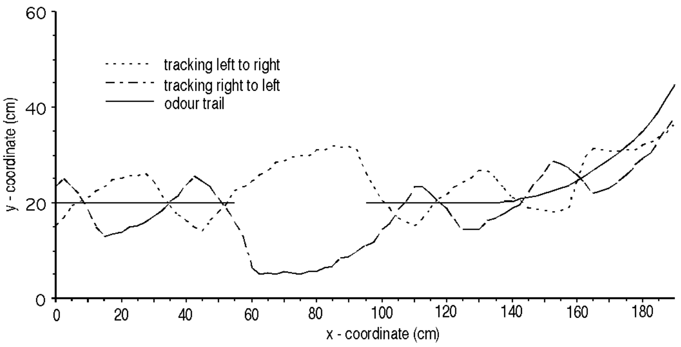


Figure 3.2 The ant robot track as it follows a chemical trail from right to left and left to right.

3.2 Following chemical trails with a single sensor (Russell, *et al.*, 1997)

For trail following experiments using a single chemical sensor it is possible to use a robot that is much smaller than the 6-legged ant robot. A chemical sensing robot was designed around the 10cm diameter RAT robot. The Reactive Autonomous Testbed (RAT) robot was designed to be sufficiently inexpensive that a new robot could be built for each experiment. The entire robot costs less than the price of one small precision electric motor.   An air-curtain chemical sensor incorporating a circular printed circuit board fan is mounted on the front of this robot (Figure 3.3). A single 68HC11 microcontroller controls the movement of the robot, monitors the chemical sensor and implements the 'single antennae' algorithm described in Section 2.4.

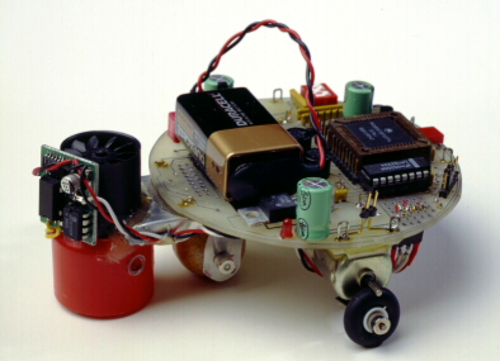


Figure 3.3 The RAT robot carrying a single chemical marking sensor.

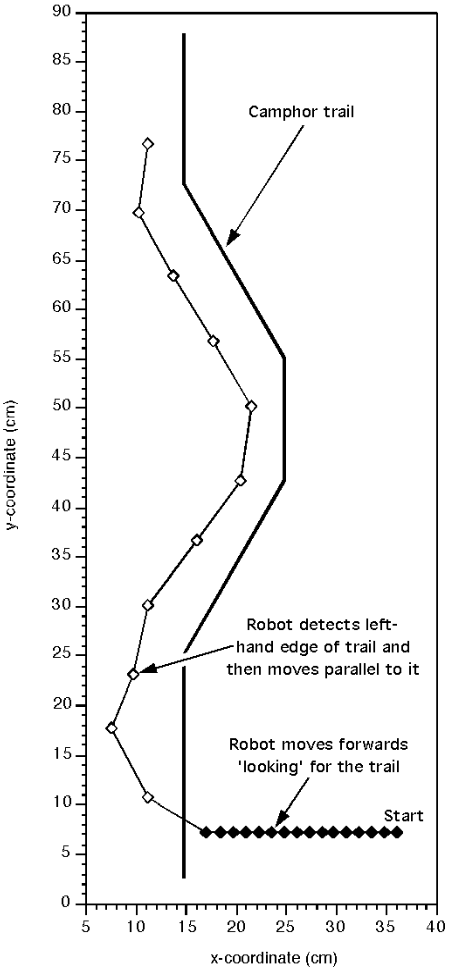


Figure 3.4 The path of RAT robot acquiring and then following a chemical trail.

The diagram shown in Figure 3.4 plots the trajectory of the RAT robot as it searches for the chemical trail. In this figure the diamonds show successive positions of the center of the circular robot. Upon detecting the camphor trail the robot turns and follows the left-hand edge of the trail.

References

Brooks, R.A. 'A robust layered control system for a mobile robot', *IEEE Journal of Robotics and Automation*, Vol. RA-2, No.1, March 1986, pp.14-23.

Russell, R.A., 'Ant trails - an example for robots to follow?' *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, Michigan, May 1999a, pp. 2698-2703.

Russell, R.A. *Odour Sensing for Mobile Robots,* World Scientific, ISBN 981023791X, 217pp., 1999b.

Russell, R.A., Thiel, D. and Mackay-Sim, A., 'Recruiting swarm robots using coded odour trails', *Proceeding of the International Conference on Field & service Robotics*, Canberra, 8-10 December 1997, pp. 472-476.