SANDIA REPORT

SAND97–1354 • UC–705
Unlimited Release
Printed March 1998

A Guidance and Control Algorithm for Scent Tracking Micro-Robotic Vehicle Swarms

Jeffrey L. Dohner

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND97-1354
Unlimited Release
Printed March 1998

Distribution
Category UC-705

A Guidance and Control Algorithm for Scent Tracking Micro-Robotic Vehicle Swarms

Jeffrey L. Dohner
Structural Dynamics Department
Sandia National Laboratories
P.O.Box 5800
Albuquerque, NM 87185-0439

Abstract

Cooperative micro-robotic scent tracking vehicles are designed to collectively “sniff out” locations of high scent concentrations in unknown, geometrically complex environments. These vehicles are programed with guidance and control algorithms that allow inter cooperation amongst vehicles. In this paper a cooperative guidance and control algorithm for scent tracking micro-robotic vehicles is presented. This algorithm is comprised of a sensory compensation sub-algorithm using point source cancellation, a guidance sub-algorithm using gradient descent tracking, and a control sub-algorithm using proportional feedback. The concepts of social rank and point source cancellation are new concepts introduced within. Simulation results for cooperative vehicles swarms are given. Limitations are discussed.

Intentionally Left Blank

Acknowledgment

Special thanks goes to Dr. Rush Robinett for funding this research. Furthermore, much appreciated, thanks goes to Dr. John Feddema for his insight and guidance.

Intentionally Left Blank

Contents

Acknowledgment v
Notation viii

A Guidance and Control Algorithm for Scent Tracking Micro-Robotic Vehicle Swarms

1 Introduction 1
2 Bug/scent environment 3
2.1 Robotic vehicle, "Bug",dynamics 3
2.2 A scent environment model 5
3 The guidance and control algorithm 8
3.1 Sensory compensation sub-algorithm 8
3.2 Guidance sub-algorithm and social rank 9
3.3 Control sub-algorithm 13
4 Simulation results 15
5 Conclusions 18
6. References 19

Figures

Fig. 1 Information transfer required for the guidance and control of a cooperative vehicle 2
Fig. 2 Side, top, and graphical view of a robotic vehicle ("bug") 4
Fig. 3 Bug environment with scent sources and obstacles 6
Fig. 4 Two bugs in proximity, the bug of lower rank allows the bug of higher rank to pass 12
Fig. 5 A bug near an obstacles, a left footed bug will turn left and a right footed bug will turn right 13
Fig. 6 A bug's actual trajectory location and its desired reference trajectory location, the control sub-algorithm forces the actual trajectory to follow the desired reference trajectory 14
Fig. 7 Two bugs "fight" with each other to arrive at their destinations 16
Fig. 8 Two bugs cooperate with each other to arrive at their destinations 16
Fig. 9 Trajectory of bugs at various instants of time, 6 bugs attempt to discover 5 source locations, a bug which finds a source remains at the source while other bugs search elsewhere for sources 17
Fig. 10 Trajectory of bugs at various instants of time, 5 bugs attempt to discover the location 3 sources hidden by obstacles, left footed bugs move left around walls and right footed bugs move right, bugs can switch footedness 18

Tables

Table 1: Simulation parameters 15

Notation

R_w -radius of a drive wheel

R -distance between bug's center line and a drive wheel

ω_{l_i} -angular rotation rate of bugs's left wheel

ω_{r_i} -angular rotation rate of bugs's right wheel

(x_i, y_i) -bug's location

θ_i -bug's orientation

ϕ -scent distribution

Q_j -source strength

$\left(x_{s_j}, y_{s_j}, z_{s_j}\right)$ -source location

Ω -scent domain

$\delta\Omega$ -domain boundaries

$\vec{v}$ -scent velocity

v_n -normal scent velocity

$\vec{n}$ -boundary surface normal

$G(\vec{r}_o, \vec{r})$ -Green's function

X_i -bug dynamic state vector

X_{r_i} -reference model state vector

$\left(x_{r_i}, y_{r_i}, \theta_{r_i}\right)$ - desired bug location

b -fictitious damping

ω_{max} -maximum angular rotation

V_{max} -maximum velocity

d_c -distance between desired and actual trajectory location

χ_{ij} -angle between desired and actual trajectory location

d_{max} -desired minimum distance between bugs

v_r -reference velocity

d_w -distance from an obstacle

$\phi_c(x_o, y_o)$ -direct field scent distribution

n_c -the number of bugs in proximity to the i^{th} bug

n_w -the number of wall points in proximity to the i^{th} bug.

$R(i)$ -the rank of the i^{th} bug

1. Introduction

In recent years the demand for covert methods of mobile search and surveillance has become paramount. Covert mobile search and surveillance could be achieved by using swarms of inconspicuous micro-robotic vehicles which, by tracking scent, could discover scent sources. To optimize search and surveillance, these micro-robotic vehicles would be required to work in a cooperative fashion. In this paper a cooperative guidance and control algorithm for these types of vehicles is presented. The objective of this algorithm is to create the minimal vehicle intelligence required to “sniff out” and monitor scent sources in a unknown environment. This algorithm is unique in that it uses the novel concepts of social rank and point source cancellation to produce cooperation.

Methods of cooperative guidance and control for a micro-robotic vehicle operating in a swarm of vehicles are still immature. Salido et al. (Salido et al., 1997) reiterated the deliberative and reactive approaches to cooperative vehicle guidance and control. The deliberative approach produces vehicles with higher levels of learning and interpretation; however, these vehicles also require significant compute capability. The reactive approach produces vehicles which are primitive and instinctual but require only minimal compute capability. In this paper it is assumed that, due to size, a micro-robotic vehicle is compute limited; therefore, a reactive guidance and control approach is used.

Hanspeter and Junkins (Hanspeter and Junkins, 1997) used an attractive/repulsive potential method to produce a cooperative swam solution. The vehicles in their solution acted in a reactive fashion. Artificial potentials were used to guide a vehicle around obstacles and limit its distance from other vehicles. To calculate their solution, a priori knowledge of the environment was required. In this paper a potential solution will also be used; however, it is assumed that the potential solution exist physically due to measurable scent diffusion through the environment. The potential solution and the scent diffusion solution are the same; therefore, no a priori environmental information is required.

Computationally non-taxing guidance and control algorithms have been developed for robotic vehicles using layered or behavioral based methods (Brooks, 1986; Feddema et al., 1997); however, these algorithms have not yet been modified to operate in cooperation with other vehicles. Pryor (Pryor 1998) used a genetic programing methodology to design a guidance and control algorithm for a vehicle operating in such a cooperative swarm environment. The development of this algorithm was based on the concept of evolutionary calculations. As in Pryor’s paper, in this paper a non-compute intensive, guidance and control algorithm which can be integrated into a swam of cooperating robotic vehicles is presented; however, this guidance and control algorithm will not be based on the concept of evolutionary calculations. This guidance and control algorithm is based on the concept of social rank and order.

The destination of a vehicle is a scent source. By tracking scent the location of sources can be discovered (Masoud, 1994). Once a vehicle determines that it has discovered a source, a declaration of source discovery is communicated to all vehicles in its proximity; the discovering vehicle remains fixed in location to the location of the source; and all vehicles in proximity to the source perform sensory compensation to cancel the effects of this source on their senses so as to be able to track other undiscovered scent sources. Sensory compensation is performed using point source cancellation. This cancellation is an innovation which uses an approximation based upon a simple model of the local scent environment around a source.

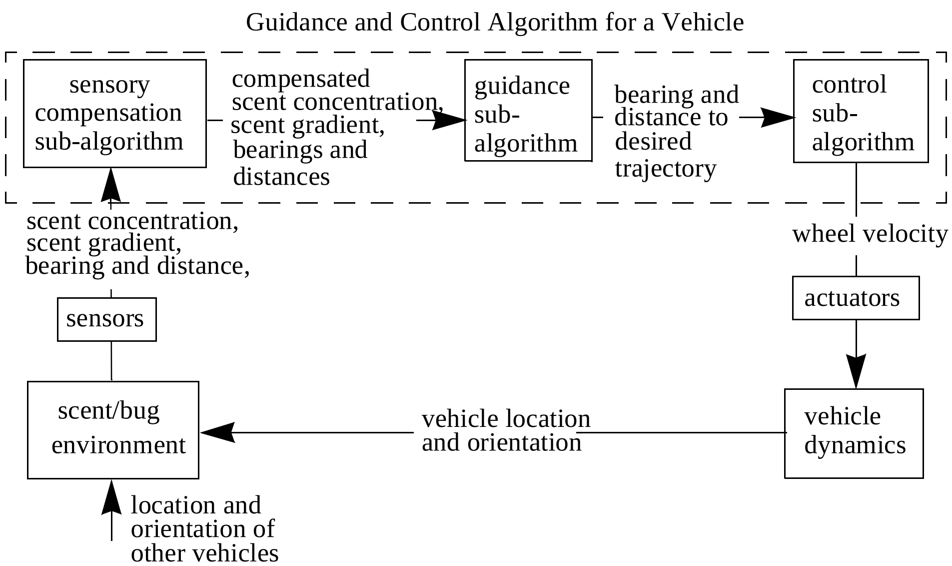

Fig. 1 Information transfer required for the guidance and control of a cooperative vehicle

The output of the sensory compensation sub-algorithm is input into the guidance sub-algorithm. The guidance sub-algorithm determines desired vehicle trajectory. If vehicle environment is known a priori, this trajectory can be computed prior to vehicle motion. However, if vehicle environment is not known a priori, this trajectory must be computed iteratively while moving to a source (Bemporad, et al., 1996; Oriolo, et al. 1996). In this paper desired vehicle trajectory is computed iteratively while moving to a source.

While moving to a source, vehicles must avoid one another and obstacles. Once a vehicle is in the proximity of another vehicle or an obstacle, the guidance sub-algorithm is used to alter desired vehicle trajectory such that avoidance is maintained. Many authors have used potential or artificial force models to produce this type of avoidance (Bemporad, et al., 1996; Khatib, 1986; Hong, et al., 1996); however, their focus has been on stationary obstacles. The focus in this paper will not be on the avoidance of only stationary obstacles but will be on the avoidance of stationary obstacles as well as all other *moving* vehicles. To do this the concept of social rank will be introduced. The output of the guidance sub-algorithm is a desired vehicle trajectory which directs a vehicle to scent sources while avoiding obstacles and other vehicles.

The control sub-algorithm is used to drive vehicle inputs in such a way that the actual vehicle trajectory is driven to the desired vehicle trajectory. In this paper, this sub-algorithm is a simple, robust, proportional control algorithm.

The guidance and control algorithm presented in this paper consist of a sensory compensation sub-algorithm, a guidance sub-algorithm, and a control sub-algorithm. The purpose of this algorithm is to create the minimal vehicle intelligence required to “sniff out” and monitor scent sources in a unknown environment. This algorithm is based upon concepts in the literature as well as the new concepts of social order and point source cancellation.

2. Bug/scent environment

To verify the performance of the guidance and control algorithm, simulations were performed using a model of the dynamics of a vehicle directed by the guidance and control algorithm in a model of a scent environment. In the following sub-sections, this vehicle and scent environment model are outlined.

2.1 Robotic vehicle, “Bug”, dynamics

Figure 2 is an illustration of a robotic wheeled vehicle. For the rest of this paper, this vehicle will be referred to as a “bug”. A bug is constructed from two drive wheels on either side of a chassis with stabilizing wheels fore and aft. The radius of the drive wheels is R_w, and the distance between the bug’s center line and a drive wheel is R.

Bugs exist in swarms. The dynamics of the i^{th} bug in a swam of bugs can be approximated (Feddema et al., 1997) by

$$\dot{X}_i = B(X_i)u_i \tag{1}$$

$$\text{where } X_i = [x_i, y_i, \theta_i]^T,\ u_i = \left[\omega_{r_i}, \omega_{l_i}\right]^T, \tag{2,3}$$

$$B(X_i) = \begin{bmatrix} \frac{R_w}{2}\cos(\theta_i) & \frac{R_w}{2}\cos(\theta_i) \\ \frac{R_w}{2}\sin(\theta_i) & \frac{R_w}{2}\sin(\theta_i) \\ \frac{R_w}{2R} & -\frac{R_w}{2R} \end{bmatrix}, \tag{4}$$

ω_{r_i} is the angular velocity of the bug’s right drive wheel, ω_{l_i} is the angular velocity of the bug’s left wheel, (x_i, y_i) is the location of the bug, and θ_i is the orientation of the bug. The constraint $-\omega_{max} \le \omega_{r_i} \le \omega_{max}$, and $-\omega_{max} \le \omega_{l_i} \le \omega_{max}$ are imposed upon bug dynamics by clipping the angular velocity of drive wheels. Bugs can move in only the $z = 0$ plane.

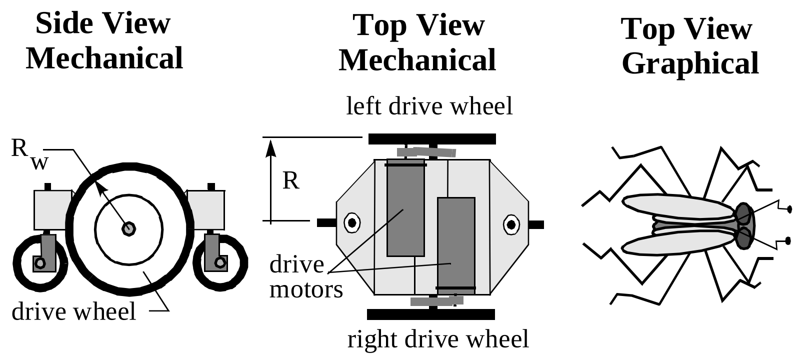

Fig. 2 Side, top and graphical view of a robotic vehicle (“bug”)

A number of assumptions are made about the capability of bugs:

- Bugs can locally communicate whether they are or are not at a source location;
- Bugs can locally sense the distance and bearing of obstacles;
- Bugs can locally sense the distance and bearing of other bugs;
- Bugs can estimate their relative distance from a discovered source;
- Bugs can estimate their relative location and orientation in space;
- Bugs can locally measure scent and scent gradient.

A number of assumptions are also made about the limitations of bugs:

- Bugs have limited memory;
- Bugs have limited compute capability;
- Bugs know nothing about their environment;
- Bugs can sense and communicate only locally.

In short, bugs have low intelligence, poor senses, and primitive communication capacity.

2.2 A scent environment model

Bugs track scent distributions. In this section a model of these distributions will be presented. This model will be used in simulation to numerically verify the guidance and control algorithm.

If scent distributions are diffusive in the space, $\phi(\vec{r})$ is a scent distribution, Q_j is the source strength of the j^{th} of n_s sources, and $\vec{r}_{s_j}$ is the location of the j^{th} source, then

$$\nabla^2\phi(\vec{r}) = \sum_i^{n_s} Q_j\delta\left(\vec{r}-\vec{r}_{s_j}\right) \tag{5}$$

for $\vec{r} \in \Omega$ where $\vec{r}$ points to (x, y, z). Equation 5 is a potential problem where the potential function is the scent concentration, $\phi(\vec{r})$. Normalized scent velocity, $\vec{v}$, is given by

$$\vec{v} = \nabla\phi(\vec{r}). \tag{6}$$

For hard boundaries the velocity of scent normal to any obstacle boundary is assumed to be zero. Therefore, if $\vec{n}$ is a boundary surface normal, then, from equation 6,

$$v_n = \vec{n}\cdot\nabla\phi(\vec{r}) = \frac{\partial}{\partial n}\phi(\vec{r}) = 0 \tag{7}$$

for $\vec{r} \in \delta\Omega$ where v_n is normal normalized scent velocity. Equations 5, 6 and 7 describe the scent environment of a domain, Ω, with non-diffusive boundaries, $\delta\Omega$.

Equation 5 can be transformed into integral equation form (Anderson et al. 1984) as

$$c_o\phi(\vec{r}_o) = \int_{\delta\Omega}\left[G(\vec{r}_o, \vec{r})\frac{\partial}{\partial n}\phi(\vec{r}) - \frac{\partial}{\partial n}G(\vec{r}_o, \vec{r})\phi(\vec{r})\right]d\delta\Omega + \sum_{j=1}^{n_s} Q_jG\left(\vec{r}_o, \vec{r}_{s_j}\right) \tag{8}$$

where $\left\{\begin{bmatrix} c_o = 1/2 \;\; if \;\; (x_o, y_o, z_o) \in \delta\Omega \\ c_o = 1 \;\;\; if \;\; (x_o, y_o, z_o) \in \Omega \end{bmatrix}\right\}$, $\delta\Omega$ represents smooth spatial boundaries,

$\frac{\partial}{\partial n} = \vec{n}\cdot\nabla$, and

$$G(\vec{r}_o, \vec{r}) = \frac{1}{4\pi\left|\vec{r}_o - \vec{r}\right|}. \tag{9}$$

As shown in Figure 3, $\delta\Omega$ can be discretized using n_p uniform, single node elements. Using this discretization equation 8 can be approximated as

$$c_o\phi(\vec{r}_o) \approx \sum_{k=1}^{n_p}\left[G(\vec{r}_o, \vec{r}_k)\frac{\partial}{\partial n}\phi(\vec{r}_k) - \frac{\partial}{\partial n}G(\vec{r}_o, \vec{r}_k)\phi(\vec{r}_k)\right]\Delta\delta\Omega_k + \sum_{j=1}^{n_s} Q_jG\left(\vec{r}_o, \vec{r}_{s_j}\right) \tag{10}$$

where $\Delta\delta\Omega_k$ is the area of an element.

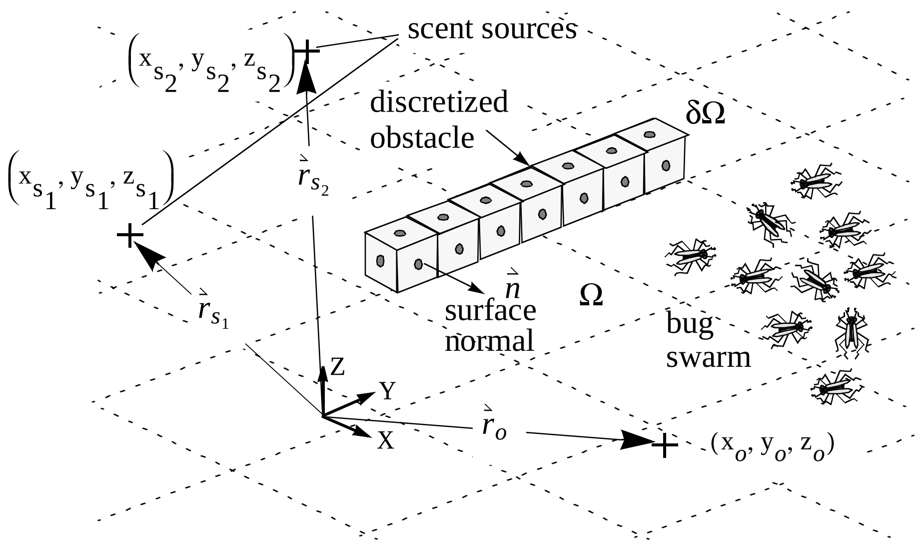

Fig. 3 Bug environment with scent sources and obstacles

When $\vec{r}_o \in \delta\Omega$ equation 10 can be written in matrix form as

$$\frac{1}{2}\Phi \approx \mathbf{G}\frac{\partial \Phi}{\partial n} - \mathbf{H}\Phi + \mathbf{Q} \tag{11}$$

where $\Phi = [\phi_1, \phi_2, \dots]^T$, $\frac{\partial \Phi}{\partial n} = \left[\frac{\partial \phi_1}{\partial n}, \frac{\partial \phi_2}{\partial n}, \dots\right]^T$,

$$\mathbf{Q} = \left[\sum_{i=1}^{n_s} Q_i G\left(\vec{r}_1, \vec{r}_{s_i}\right), \sum_{i=1}^{n_s} Q_i G\left(\vec{r}_2, \vec{r}_{s_i}\right), \dots\right]^T,$$

$$\mathbf{G} = \begin{bmatrix} 0 & G(\vec{r}_1, \vec{r}_2)\Delta\delta\Omega_2 & \dots \\ G(\vec{r}_2, \vec{r}_1)\Delta\delta\Omega_1 & 0 & \dots \\ \vdots & \vdots & \ddots \\ G\left(\vec{r}_{n_p}, \vec{r}_1\right)\Delta\delta\Omega_1 & G\left(\vec{r}_{n_p}, \vec{r}_2\right)\Delta\delta\Omega_2 & \dots \end{bmatrix},$$

$$
\mathbf{H} = \begin{bmatrix} 0 & \frac{\partial}{\partial n}G(\vec{r}_1, \vec{r}_2)\Delta\delta\Omega_2 & \cdots \\ \frac{\partial}{\partial n}G(\vec{r}_2, \vec{r}_1)\Delta\delta\Omega_2 & 0 & \cdots \\ \circ & \circ & \circ \\ \circ & \circ & \circ \\ \circ & \circ & \circ \\ \frac{\partial}{\partial n}G\left(\vec{r}_{n_p}, \vec{r}_1\right)\Delta\delta\Omega_2 & \frac{\partial}{\partial n}G\left(\vec{r}_{n_p}, \vec{r}_2\right)\Delta\delta\Omega_2 & \cdots \end{bmatrix}.
$$

When $\vec{r}_o \notin \delta\Omega$ equation 10 can be written as

$$
\phi(\vec{r}_o) \approx \mathbf{G}_o \frac{\partial}{\partial n}(\Phi) - \mathbf{H}_o\Phi + \sum_{i=1}^{n_s} Q_i G\left(\vec{r}_o, \vec{r}_{s_j}\right) \tag{12}
$$

where

$$
\mathbf{G}_o = \left[G(\vec{r}_o, \vec{r}_1)\Delta\delta\Omega_1 \; G(\vec{r}_o, \vec{r}_2)\Delta\delta\Omega_2 \; \ldots\right],
$$

$$
\mathbf{H}_o = \left[\frac{\partial}{\partial n}G(\vec{r}_o, \vec{r}_1)\Delta\delta\Omega_1 \; \frac{\partial}{\partial n}G(\vec{r}_o, \vec{r}_2)\Delta\delta\Omega_2 \; \ldots\right].
$$

From equations 7, 11, and 12, the scent concentration at any location $\vec{r}_o \in \Omega$ can be found as

$$
\phi(\vec{r}_o) \approx -\mathbf{H}_o\left(\frac{1}{2}\mathbf{I} + \mathbf{H}\right)^{-1}\mathbf{Q} + \sum_{i=1}^{n_s} Q_i G\left(\vec{r}_o, \vec{r}_{s_i}\right). \tag{13}
$$

Normalized scent velocity is calculated using equation 6 and the gradient of $\phi(\vec{r})$. The gradient of $\phi(\vec{r})$ is determined by using central differences (Gerald, 1980) and equation 13.

The half space $z = 0$ boundary is the only surface on which bugs can move. The scent boundary condition on this surface is $\frac{\partial\phi}{\partial z} = 0$. This boundary condition can be imposed by using the method of images (Junger and Feit, 1986).

3 The guidance and control algorithm

The guidance and control algorithm is the reactive intelligence of a bug. This algorithm is comprised a sensory compensation sub-algorithm, a guidance sub-algorithm, and a control sub-algorithm. The interaction of these three sub-algorithms is shown in Figure 1. Sensors measure scent, bearings, and distances in the scent/bug environment. These measurements are modified in the sensory compensation sub-algorithm where the effects of discovered scent sources on measurements is cancelled. The output of the sensory compensation sub-algorithm is the input to the guidance sub-algorithm. In the guidance sub-algorithm, compensated scent inputs drive the dynamics of a reference model which defines the behavior of the bug. Bearing and distance measurements are used in higher levels of guidance to produce avoidance with walls and other bugs. The output of the guidance sub-algorithm is a desired trajectory. A bug wishes to move along this trajectory. The control sub-algorithm attempts to force the bug's actual trajectory to this desired reference trajectory by commanding the angular velocity of the bug's wheels. Details of each of these three sub-algorithms are given in the following sub-sections.

3.1 Sensory compensation sub-algorithm

The objective of a bug is to find a scent source. The location of a scent source is determined by monitoring the gradient of scent in four perpendicular directions. If the gradient of scent in all four directions is negative, the bug is at a source. It is assumed that all sources are point sources in the $z = 0$ plane.

Once a bug has discovered a source, it remains at the source and communicates to other bugs in proximity that it has located that source. Other bugs then attempt to compensate for the effects of this source on their sensory perception. This enables a swarm of bug to track compensated scent to remaining, undiscovered sources. Sensory scent compensation is performed by analytically manipulating measured scent distributions.

Scent distributions, $\phi(\vec{r}_o)$, were modeled using equation 13. Equation 13 is comprised of a direct diffusion term, $\sum_{i=1}^{n_s} Q_i G\left(\vec{r}_o, \vec{r}_{s_i}\right)$, and an indirect diffusion term, $-\mathbf{H}_o\left(\frac{1}{2}\mathbf{I} + \mathbf{H}\right)^{-1}\mathbf{Q}$. The indirect diffusion term is a function of environmental geometry; therefore, to cancel the effects of indirect diffusion on measured scent, environmental geometry must be know. However, it was assumed that bugs have no knowledge of their environment; therefore, indirect diffusion effects cannot be canceled. Nevertheless, direct diffusion effects are not a function of environmental geometry and can be canceled. In this paper only direct diffusion effects will be cancelled.

If $\phi(\vec{r}_o)$ is measured scent, then compensated scent,

$$\phi_c(\vec{r}_o) \equiv \phi(\vec{r}_o) - Q_j G\left(\vec{r}_o, \vec{r}_{s_j}\right) , \tag{14}$$

is measured scent with direct scent diffusion effects from the j^{th} source cancelled. In this paper this type of cancellation will be called point source cancellation.

In equation 14 source strength Q_j and location $\vec{r}_{s_j}$ must be estimated before $\phi_c(\vec{r}_o)$ can be calculated. If $\vec{r}_x$ is a bug's location and $\vec{r}_x$ is in the neighborhood of $\vec{r}_{s_j}$, then diffusion from the j^{th} point source dominates the scent distribution, $\phi_c(\vec{r}_x)/\phi(\overrightarrow{r_x}) \approx 0$, and Q_j can be approximated as $Q_j \approx 4\pi R_j \phi(\vec{r}_x)$ where $R_j = \left|\vec{r}_x - \vec{r}_{s_j}\right|$. Therefore, bugs in proximity to a discovered source can approximate the strength of that source if they know the distance they are from the source. Distance from a source can be determined from bearing and distance measurements. Moreover, source location can also be determined from these same measurements. Using equation 14 and scent, bearing, and distance measurements, bugs can approximate compensated scent, $\phi_c(\vec{r}_o)$, and track compensated scent to other sources.

3.2 Guidance sub-algorithm and social rank

Using compensated scent, bearing, and distance measurements, the guidance sub-algorithm produces a desired trajectory which directs bugs towards scent sources while avoiding obstacles and other bugs. If every bug had the same guidance sub-algorithm, then every bug would have equal opportunity to arrive at a source. This would result in bugs "fighting" with each other to arrive at a source. Moreover, once a bug had arrived at a source, it would have to struggle with other bugs to maintain its position. This conflict among bugs reduces the effectiveness of the swam to locate and monitor scent sources. To mitigate this conflict, bugs are ranked in a social order. Within this social order each bug has a *unique* social rank.

The social rank of the i^{th} bug, $R(i)$, is used to determine how it will move relative to every other bug. If two bugs run into each other, the bug with the lower rank must move to allow the bug with the higher rank to pass; therefore, conflict is avoided and cooperation results. Rank is initially determined randomly such that no two bugs have the same rank. In code this initial rank is represented as a random number greater than 0 and less than 1.

A bug's rank can change. If a bug finds a scent source, then that bug is given a rank higher than the rank of any bug which has not found a source. This allows a bug which finds a source to remain at a source location while other bugs look for other sources elsewhere. In code this elevation of rank is performed by simply adding 1 to a bug's present rank.

To randomize the path that bugs take around obstacles, bugs are also left or right footed. A left footed bug will always turn left when it runs into a obstacle, whereas a right footed bug will always turn right. Bugs can change their footedness. When two bugs with the same footedness run into each other, the one of lower rank will change its footedness to the opposite foot. Therefore, bugs tend to move in different directions around obstacles.

The guidance sub-algorithm is the solution to a set of simple mathematical equations which represent the dynamics of a conveniently defined yet desirable reference system (Ogata, 1970). These equations are a function of bug rank and footedness and are not compute intensive. For the i^{th} bug, these equations are

$$\dot{X}_{r_i} = GX_{r_i} + M(X_i) + \sum_{\substack{j=1 \\ j \neq i}}^{n_c} N_{ij}\left(X_{r_i}, X_j\right) + \sum_{k=1}^{n_w} W_{ik}\left(X_{r_i}, X_{w_k}\right) \qquad (15)$$

where

$$X_{r_i} = \begin{bmatrix} x_{r_i} \\ \dot{x}_{r_i} \\ y_{r_i} \\ \dot{y}_{r_i} \end{bmatrix},\ G = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -b & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -b \end{bmatrix},\ M(X_i) = \begin{bmatrix} 0 \\ V_{max} b \cdot \cos(\eta_i) \\ 0 \\ V_{max} b \cdot \sin(\eta_i) \end{bmatrix}, \qquad (16,17,18)$$

$\left(x_{r_i}, y_{r_i}\right)$ is the bug's desired trajectory location, b is a fictitious damping factor, η_i is the negative of the direction of scent velocity, $V_{max} = R_w \omega_{max}$, $N_{ij}\left(X_{r_i}, X_j\right)$ and $W_{ik}\left(X_{r_i}, X_{w_k}\right)$ are additional terms used to account for interactions with other bugs and obstacles, n_c is the number of bugs in proximity to the i^{th} bug, and n_w are the number of wall points in proximity to the i^{th} bug. The values of n_c and n_w are a function of the location of all bugs in the environment. In simulation these values are determined by checking the distance between bugs and obsticles.

Without wall and bug interactions, equation 15 reduces to a conveniently defined reference systems of equations representing the motion of a unit mass driven along a frictional surface. From these equations it can be seen that the maximum velocity of this unit mass is V_{max}; the rate at which this velocity can approach its maximum is dependent upon the damping factor, b; and the terms $N_{ij}\left(X_{r_i}, X_j\right)$ and $W_{ik}\left(X_{r_i}, X_{w_k}\right)$ are forces which alter the direction of this unit mass for the purpose of avoiding other bugs and obstacles. The trajectory of this unit mass without obstacle and bug avoidance force interactions will be referred to as the desired reference trajectory. The trajectory of this unit mass with avoidance force wall and bug interaction is the desired trajectory. The desired trajectory is the output of the guidance sub-algorithm.

The terms $N_{ij}\left(X_{r_i}, X_j\right)$ are used to modify the desired reference trajectory of the i^{th} bug for the purpose of avoiding collisions with other bugs. A kinetic model is used to produce this avoidance. Figure 4 shows two neighboring bugs. The location of the j^{th} bug and the desired reference trajectory location of the i^{th} bug are a distance $d_c = \sqrt{\left(x_j - x_{r_i}\right)^2 + \left(y_j - y_{r_i}\right)^2}$ apart and are misaligned by a bearing angle $\chi_{ij} = \operatorname{atan}\left\{\left(y_j - y_{r_i}\right)/\left(x_j - x_{r_i}\right)\right\}$. If d_{min} is the minimal distance that these two locations are allowed to be apart, then the following logic can be used to produce separation.

$$\begin{aligned} & if\ d_c < d_{min}\ and\ R(i) < R(j) \\ & then\ N_{ij}\left(X_{r_i}, X_j\right) = K(d_c - d_{min}) \cdot \left[0\ \ \cos(\chi_{ij})\ \ 0\ \ \sin(\chi_{ij})\right]^T \\ & else\qquad N_{ij}\left(X_{r_i}, X_j\right) = \left[0\ 0\ 0\ 0\right]^T \end{aligned} \tag{19}$$

Equation 19 is a kinetic model for the force between two colliding balls with elastic constants K. A kinetic model is used since, for large K, it closely approximates the desired inequality constraint $d_c \geq d_{min}$. Since this kinetic model contains no losses, momentum exchanges could produce bug speeds, $v_r = \sqrt{\dot{x}_{r_i}^2 + \dot{y}_{r_i}^2}$, which exceed maximum allowable vehicle speed, V_{max}. Therefore, the additional constraint

$$v_r = \sqrt{\dot{x}_{r_i}^2 + \dot{y}_{r_i}^2} \leq V_{max} \tag{20}$$

should be imposed. This constraint is imposed by clipping the velocity of the reference model

.

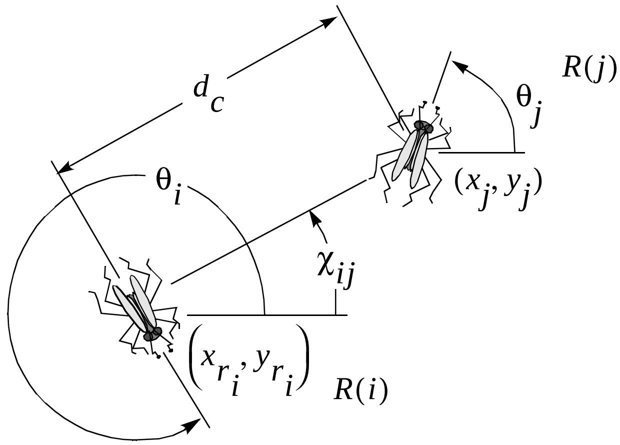

Fig. 4 Two bugs in proximity, the bug of lower rank allows the bug of higher rank to pass

The terms $W_{ik}\left(X_{r_i}, X_{w_k}\right)$ are used to modify the desired reference trajectory of the i^{th} bug for the purpose of avoiding collisions with obstacles. As shown in Figure 5, obstacles are represented as a distribution of fixed points. It is assumed that a bug can locally sense the distance from these fixed points to its present, desired reference trajectory location. If $\left(x_{w_{k_{min}}}, y_{w_{k_{min}}}\right)$ is the obstacle location closest to the desired reference trajectory location, then the minimal distance from the i^{th} bug's desired reference trajectory location to this obstacle is $d_w = \sqrt{\left(x_{r_i} - x_{w_{k_{min}}}\right)^2 + \left(y_{r_i} - y_{w_{k_{min}}}\right)^2}$, and the bearing to this point is $\eta_{ik_{min}} = \operatorname{atan}\left\{\left(y_{r_i} - y_{w_{k_{min}}}\right) / \left(x_{r_i} - x_{w_{k_{min}}}\right)\right\}$. Using distance and bearing information, the following logic is use to produce obstacle avoidance.

$if\ d_w < d_{min}$

$$\begin{aligned} &then\ W_{ik}\left(X_{r_i}, X_{w_k}\right) = V_{max} b \cdot [0, -\sin\eta_{ik_{min}}, 0, \cos\eta_{ik_{min}}]^T \\ &else\qquad W_{ik}\left(X_{r_i}, X_{w_k}\right) = [0, 0, 0, 0]^T \end{aligned} \tag{21}$$

for a left footed bug,

$$if\ d_w < d_{min}$$

$$\begin{aligned} &then\ W_{ik}\left(X_{r_i}, X_{w_k}\right) = V_{max} b \cdot [0, \sin\eta_{ik_{min}}, 0, -\cos\eta_{ik_{min}}]^T \\ &else\qquad W_{ik}\left(X_{r_i}, X_{w_k}\right) = [0, 0, 0, 0]^T \end{aligned} \tag{22}$$

for a right footed bug.

Notice that equations 15 through 22 can be implemented using relative not absolute bearings and distances. Therefore, bugs do not have to know their absolute locations.

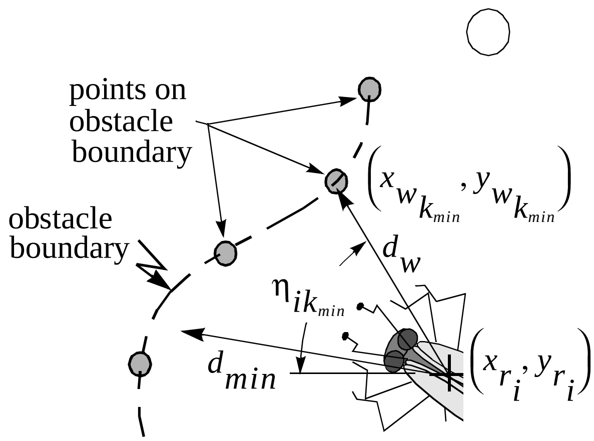

Fig. 5 A bug near an obstacle, a left footed bug will turn left and a right footed bug will turn right

3.3 Control sub-algorithm

Equations 15 through 22 comprise the guidance sub-algorithm. This sub-algorithm determines the desired trajectory of a bug. The control sub-algorithm drives a bug's actual trajectory to this desired trajectory.

Figure 6 shows a bug at the actual trajectory location (x_i, y_i). It is desired force this trajectory location to be as close as possible to the desired trajectory location $\left(x_{r_i}, y_{r_i}\right)$. If

$\vec{n}_i = \cos\theta_i \hat{\imath} + \sin\theta_i \hat{\jmath}$, $\vec{m}_i = \cos\beta_i \hat{\imath} + \sin\beta_i \hat{\jmath}$, and $\varepsilon_i = |\beta_i - \theta_i| = \mathrm{acos}\left(\vec{n}_{r_i} \cdot \vec{m}_{r_i}\right)$

where ε_i is the bearing from (x_i, y_i) to $\left(x_{r_i}, y_{r_i}\right)$, and if d is the distance between (x_i, y_i) and $\left(x_{r_i}, y_{r_i}\right)$, then a control sub-algorithm which will drive (x_i, y_i) to $\left(x_{r_i}, y_{r_i}\right)$ is given below.

$$\text{if } \left|\vec{n}_{r_i} \times \vec{m}_{r_i}\right| \geq 0, \quad \text{then } \omega_{r_i} = \frac{v_c}{R_w} + \frac{R}{R_w} K_p \varepsilon_i \ , \quad \omega_{l_i} = \frac{v_c}{R_w} - \frac{R}{R_w} K_p \varepsilon_i$$

$$\text{else } \omega_{r_i} = \frac{v_c}{R_w} - \frac{R}{R_w} K_p \varepsilon_i \ , \quad \omega_{l_i} = \frac{v_c}{R_w} + \frac{R}{R_w} K_p \varepsilon_i \tag{23}$$

where

$$\text{for } d \geq d_{min}/10 \,,\ v_c = V_{max} \text{ else } v_c = V_{max} \cdot \frac{d}{d_{min}/10} \,, \tag{24}$$

and K_p is a positive constant. Equation 24 is used to orient a bug towards the desired trajectory location and equation 24 is used to limit bug velocity such that actual trajectory does not over run desired trajectory. Only relative, not absolution, angles and distance are required for control.

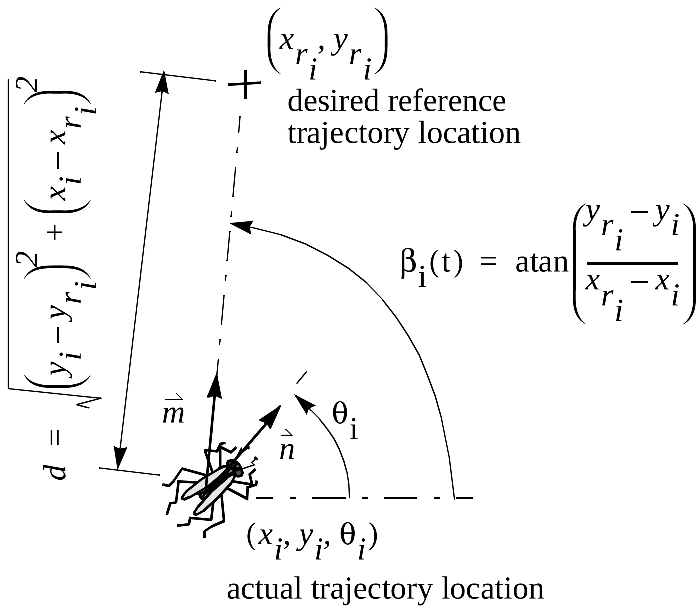

Fig. 6 A bug's actual trajectory location and its desired reference trajectory location, the control sub-algorithm forces the actual trajectory to follow the desired reference trajectory

The effect of K_p on closed loop dynamics will be examined for a simple case. From equation 1 to 4,

$$\dot{\theta}_i = \frac{R_w}{2R}\omega_{r_i} - \frac{R_w}{2R}\omega_{l_i} \,. \tag{25}$$

Substituting equation 23 into 25, taking the Laplace transform, and solving for bug orientation gives

$$\frac{\Theta}{B}(s) = \frac{K_p}{s + K_p} \tag{26}$$

where $\Theta(s) = L(\theta(t))$, $B(s) = L(\beta(t))$, L is the Laplace transform operator, and s is the Laplace transform parameter. This is the response of a simple, first order system with proportional feedback. As K_p increases, the tracking error is driven to zero; therefore, for low values of K_p, the bug's coordination is very poor and it staggers about the desired trajectory; whereas, for large values of K_p, the bug is very coordinated and its actual trajectory and the desired trajectory are very close.

4 Simulation Results

In this section three numerical examples are presented. The parameters used in these examples are given in Table 1.

Table 1: Simulation parameters

ω_{max} (r/s)	R_w (m)	R (m)	d_{min} (m)
0.8858	0.125	0.508	5.0

In the **first example**, two bugs cross paths in an attempt to arrive at different sources. Two scenarios are considered. The first scenario is shown in Figure 6. In this scenario the bugs do not cooperate with each other. Each bug is programed with an identical guidance and control algorithm, and as a result, they "fight" each other to arrive at a source. In the second scenario, shown in Figure 7, the two bugs cooperate. The bug with the lower rank, moves out of the way to allow the bug with the higher rank the opportunity to arrive at its source first. As a result it takes about one third less time for the cooperating bugs to arrive at their respective sources than it takes for the non-cooperating bugs to arrive at their sources.

In the **second example**, the ability for a swarm of bugs to find a distribution of sources is demonstrated. The scenario is shown in Figure 8. The locations of five equal strength sources ($Q_j = 1.0$) are determined by a swarm of six bugs. Once a bug finds a source it remains at that source and tells other bugs what it has found. Bug locations at various instances of time are shown. Not all bugs find a source.

In the **third example**, five bugs are used to determines the location of three equal strength sources ($Q_j = 1.0$) hidden by a set of obstacles. This scenario is shown in Figure 9. When bugs run into a obstacle, left footed bugs go left and right footed bugs go right. When bugs run into each other, they check their footedness. If they have the same footedness, the bug with the lower social rank changes his footedness. Therefore, bugs tend to more around obstacles in opposite directions.

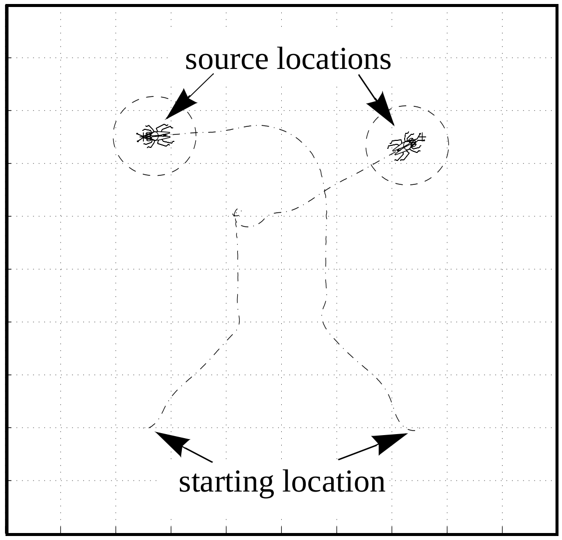

Fig. 7 Two bugs “fight” with each other to arrive at their destinations

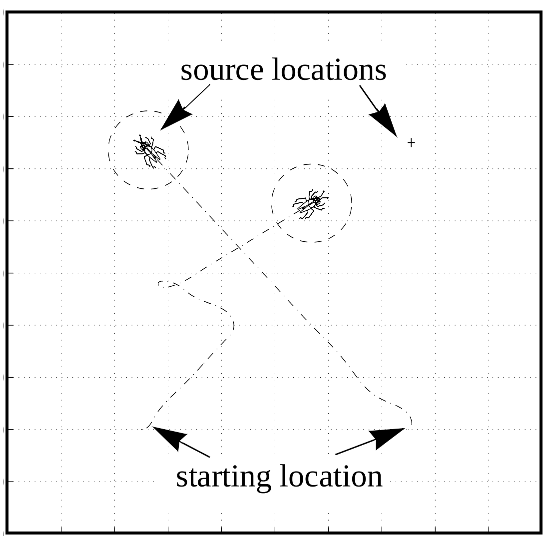

Fig. 8 Two bugs cooperate with each other to arrive at their destinations

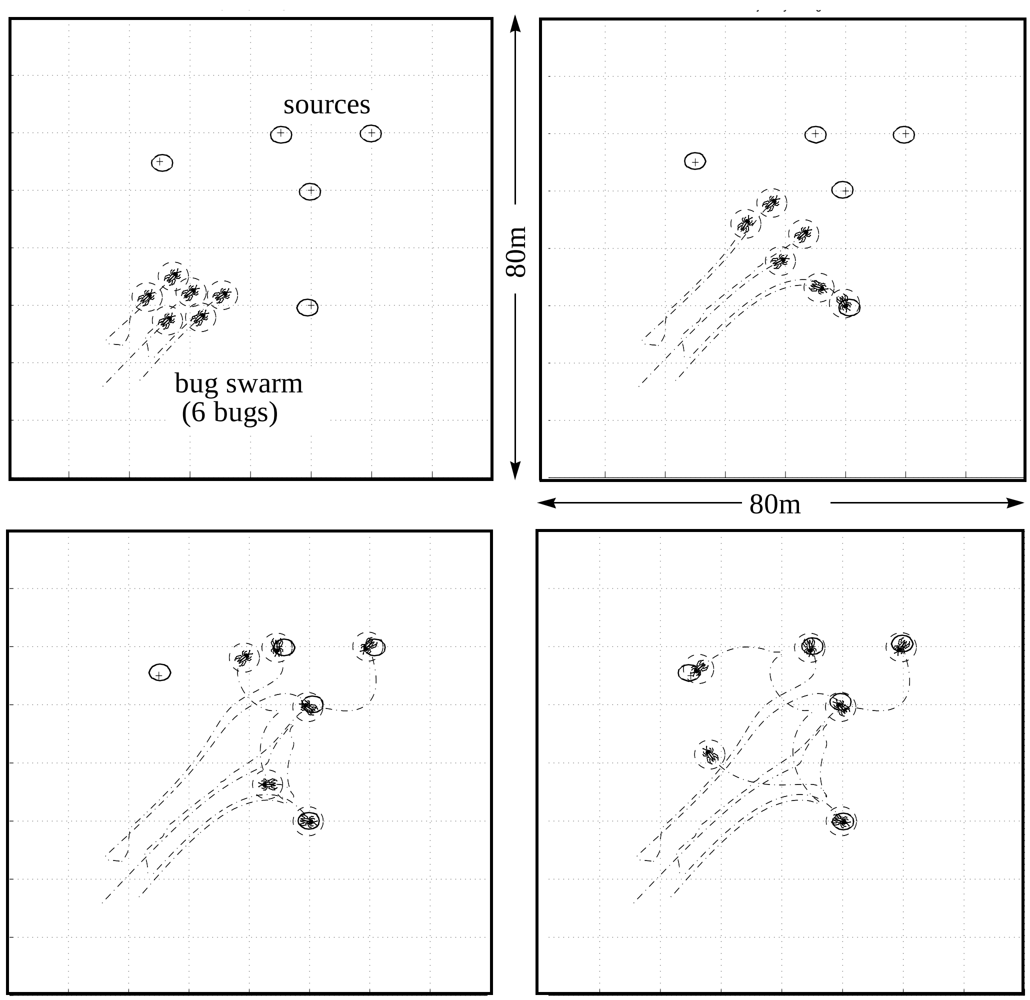

Fig. 9 Trajectory of bugs at various instants of time, 6 bugs attempt to discover 5 source locations, a bug which finds a source remains at the source while other bugs search elsewhere for sources

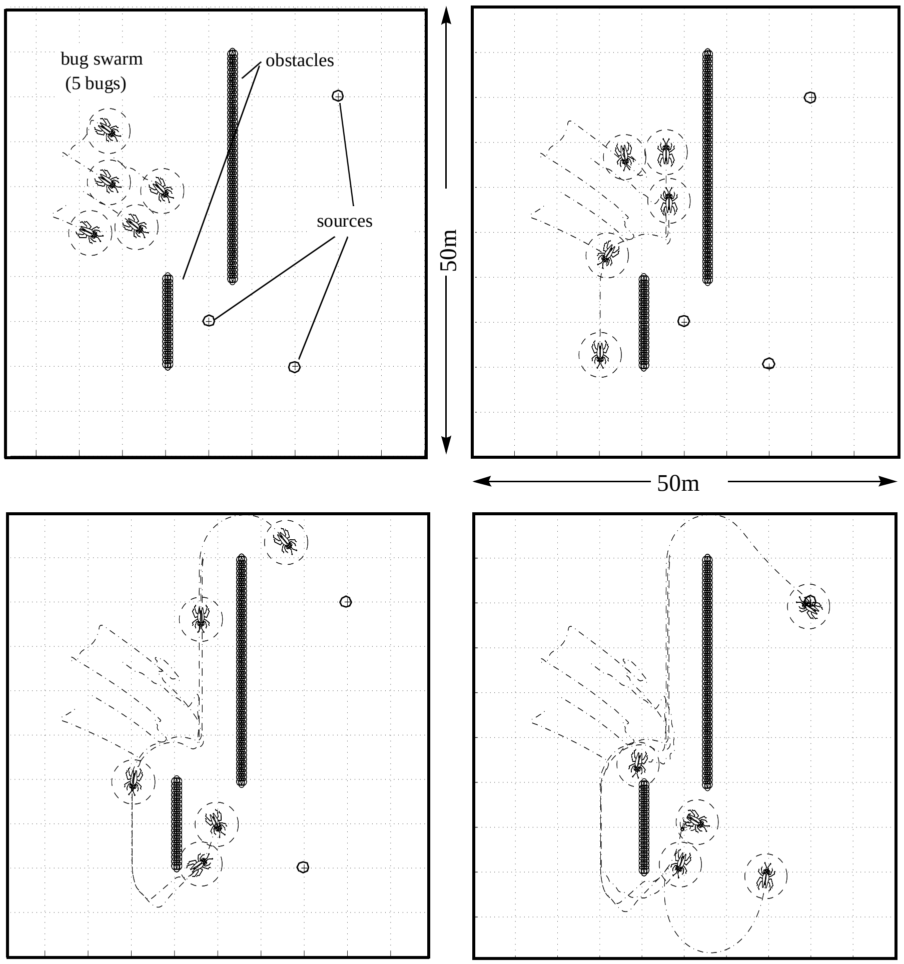

Fig. 10 Trajectory of bugs at various instants of time, 5 bugs attempt to discover the location of 3 source hidden by obstacles, left foot bugs move left around walls and right footed bugs move right, bugs can switch footedness

5 Conclusions

In this paper a guidance and control algorithm for scent tracking cooperative vehicle swarms was presented. The objective of this algorithm was to create the minimal intelligence in a vehicle required to cooperatively “sniff out” and monitor scent sources in a unknown environment. This guidance and control algorithm was based on the new concepts of social rank and point source cancellation. Social rank allows vehicles to move to scent sources in a cooperative non-combative fashion, and point source cancellation allows vehicle to compensate their sensory perceptions for the purpose of tracking scent to undiscovered sources.

The vehicles guidance and control algorithm outlined in this paper consisted of a guidance sub-algorithm, a control sub-algorithm, and sensory cancellation sub-algorithm. The guidance sub-algorithm used a reference model of simple yet desirable dynamics to determine desired trajectories. The control sub-algorithm used simple, robust proportional feedback control, and the sensory scent cancellation sub-algorithm used source point cancellation.

Simulations were performed to verify the guidance and control algorithm. It was found that the algorithm could be used to move vehicles to scent sources in a cooperative fashion. In general the algorithm work very well; however, it was discovered that vehicles, using this algorithm, did have some limitations:

- Depending on the complexity of the environment, *m* vehicles could not always find *m* sources. To find all source, many vehicles were required. As in many optimization problems, as the complexity of the domain increased the number of local minimum in the solution increased and greater search capability was needed to find a global minimum.

- It was difficult for a vehicle to find a source which was surrounded by a cluster of other sources. As sources became closely spaced, they tended to look like a single compact source. A compact source is a point source which diffuses more scent in one direction than another. A simple source diffuses equal scent in all directions. Bugs had no ability to distinguish between simple and compact sources.

There were a number of aspects of this algorithm which were not addressed in this paper and need to be studied in greater detail:

- The effect of sensor noise needs to be further examined. Sensor noise could slowed the progression of vehicles to sources and add to the difficulty of source identification.

- The effect of delays needs to be examined. Time lags in sensors and processing could have a profound effect on a vehicles ability to find a source.

- The effect of environmental noise needs to be understood. Environmental noise such as wind and wheel slippage could hamper guidance and perception.

References

Anderson, D.A., Tannehill, C.J., and Pletcher, H.R., 1984, *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corp., New York, NY.

Bemporad, A., De Luca, A., and Giuseppe, O., 1996, “Local Incremental Planning for a Car-Like Robot Navigating among Obstacles”, *IEEE International Conference on Robotics and Automation,* No. 2, pp. 1205-1211.

Brooks, R.A., March 1986, "A Robust Layered Control Systems for a Mobile Robot", *IEEE Journal of Robotics and Automation*, Vol. RA-2, No.1, pp. 14-23.

Feddema, J.T., Kwan S.K., Driessen, B.J., Spletzer, B.L., and Weber, T.M., 1997, "Miniature Autonomous Robotic Vehicle (MARV)", *unpublished Sandia National Laboratories document.*

Gerald, C.F., May 1980, *Applied Numerical Analysis, 2nd edition*, Addison-Wesly Publishing Company., Reading, MA.

Hanspeter S., and Junkins J.L., 1997, "Bug Control", Technical report, Aerospace Engineering Department, Texas A&M University.

Hong, S., Kim, S., Park, K., and Lee, J., 1996, "Local Motion Planner for Nonholonomic Mobile Robots in the Presence of the Unknown Obstacles", *IEEE International Conference on Robotics and Automation*, No. 2, pp. 1212-1217.

Junger, M.C., and Feit, D., 1986, *Sound Structures, and Their Interaction, 2nd edition*, MIT Press, Cambridge, MA.

Khatib, O., Spring 1986, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *International Journal of Robotics Research*, Vol. 5, No. 1, pp. 90-98.

Masoud, A.A., Masoud A.S., and Bayoumi, M.M., 1994, "Robot Navigation Using a Pressure Generated Mechanical Stress Field 'The Biharmonic Potential Approach' ", *IEEE International Conference on Robotics and Automation*, pp. 124-129.

Ogata, K., 1970, *Modern Control Engineering*, Prentice-Hall, Inc., Englewood Cliffs, N.J.

Oriolo, B., Panzieri, S., and Ulivi, G., April 1996, "An Iterative Learning controller for Nonholonomic Robots", *IEEE International Conference on Robotic and Automations*,No.3, pp. 2676-2681.

Pryor, R.J., 1998, *Developing Robotic Behavior using a Genetic Programming Model*, SAND98-0074, Sandia National Laboratories Internal Report

Salido, J., Dolan, J.M., Hampshire, J., and Khosla, P., 1997, " A Modified Reactive Control Framework for Cooperative Mobile Robots", *Proceedings of the SPIE - The International Society of Optical Engineering, Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, vol. 3209, pp.90-100.

Intensionally Left Blank

Distribution List

MS1413 1114 T.A.Michalske

MS0501 2334 S.M.Kohler
MS0501 2334 JD.Jordon

MS0503 2338 S.L.Trauth
MS0503 2338 T.G. Smith
MS0503 2338 S.D.Tucker

MS0841 9100 P.J.Hommert
MS0833 9103 J.H.Biffle
MS0828 9104 R.Thomas
MS0826 9111 W.Hermina
MS0834 9112 A.C.Ratzel
MS0835 9113 T.C.Bickel
MS0827 9114 R.Griffith
MS0825 9115 W.H.Rutledge
MS0833 9103 J.H.Biffle
MS0836 9116 C.W.Peterson
MS0443 9117/9118 H.S.Morgan
MS0437 9117/9118 C.Adams
MS0443 9117/9118 S.N.Burchett
MS0443 9117/9118 A.F.Fossum
MS0443 9117/9118 S.W.Key

MS0321 9200 W. Camp
MS9202 9202 R.J.Pryor
MS1111 9221 S.S.Dosanjh
MS1111 9221 C.C.Ober
MS1110 9222 R.C.Allen
MS1110 9222 D.E.Womble
MS1110 9223 D.Greenberg
MS1109 9224 A.L.Hale
MS1111 9225 G.Heffelfinger
MS0441 9226 R.Leland
MS0819 9231 J.M.McGlaun
MS0820 9232 P.Yarrington

MS0439 9234 D.R.Martinez
MS0439 9234 J.L.Dohner (10)
MS0439 9234 C.R.Dohrmann
MS0439 9234 T.Hinnerichs
MS0439 9234 B. Driessen
MS0439 9234 R.V. Field Jr.

MS0439 9234 J.M.Redmond

MS0557 9741 T.J.Baca
MS0557 9741 T.L.Paez

MS1003 9611 T.D. Robinett III
MS1003 9611 J.T. Feddema
MS1003 9611 B.L.Spletzer

MS9018 8940-2 Central Tech. Files
MS0899 4916 Tech. Library (2)
MS0619 12690 Review & Aproval Desk (2)
For DOE/OST I