SANDIA REPORT

SAND98–2572
Unlimited Release
Printed November 1998

Optical Communication System for Remote Monitoring and Adaptive Control of Distributed Ground Sensors Exhibiting Collective Intelligence

Keith M. Stantz, Stewart M. Cameron, Michael W. Trahan, John S. Wagner

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.

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: A05
Microfiche copy: A01

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND98-2572
Unlimited Release
Printed November 1998

Optical Communication System for Remote Monitoring and Adaptive Control of Distributed Ground Sensors Exhibiting Collective Intelligence

Keith M. Stantz, Stewart M. Cameron, Michael W. Trahan, John S. Wagner
Pulsed Power and Laser Initiatives Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 8715-1188

Abstract

Comprehensive management of the battle-space has created new requirements in information management, communication, and interoperability as they effect surveillance and situational awareness. The objective of this proposal is to expand intelligent controls theory to produce a uniquely powerful implementation of distributed ground-based measurement incorporating both local collective behavior, and interoperative global optimization for sensor fusion and mission oversight. By using a layered hierarchal control architecture to orchestrate adaptive reconfiguration of autonomous robotic agents, we can improve overall robustness and functionality in dynamic tactical environments without information bottlenecks. In this concept, each sensor is equipped with a miniaturized optical reflectance modulator which is interactively monitored as a remote transponder using a covert laser communication protocol from a remote mothership or operative. Robot data-sharing at the ground level can be leveraged with global evaluation criteria, including terrain overlays and remote imaging data. Information sharing and distributed intelligence opens up a new class of remote-sensing applications in which small single-function autonomous observers at the local level can collectively optimize and measure large scale ground-level signals. AS the need for coverage and the number of agents grows to improve spatial resolution, cooperative behavior orchestrated by a global situational awareness umbrella will be an essential ingredient to offset increasing bandwidth requirements within the net. A system of the type described in this proposal will be capable of sensitively detecting, tracking, and mapping spatial distributions of measurement signatures which are non-stationary or obscured by clutter and interfering obstacles by virtue of adaptive reconfiguration. This methodology could be used, for example, to field an adaptive ground-penetrating radar for detection of underground structures in urban environments and to detect chemical species concentrations in migrating plumes. Given is our research in these areas and a status report of our progress.

Acknowledgment

This work was supported by 9000 discretionary LDRD funds, proposal 99-0837, within the Department of Energy under Contract No. DE-AC04-94AL85000.

Contents

Figures

Tables

1 Introduction

The report fulfills the requirements for the discretionary LDRD proposal 99-0837 entitled "Optical Communication System for Remote Monitoring and Adaptive Control of Distributed Ground Sensors Exhibiting Collective Intelligence".

The ability of the United States to control emerging threats and to verify nonproliferation treaty activities is diminishing. This report addresses aspects of this problem, devises a technical solution, and ultimately implements this solution to counter or thwart these threats and enforce treaty resolutions.

The Problem. United States strategic doctrine is predicated on a technologically superior intelligence apparatus to protect global interests and aggressively preempt emerging challenges to national security. To date, reconnaissance has been dominated by stand-off (long-distance) remote-sensing technologies, satellites and radar systems (e.g., UWAC). Typically, protocol dictates first assess global activity and then initiate some level of human intervention. This methodology is wrought with weaknesses in which leaders of hostile countries and international as well as national terrorists are finding ways to circumvent. As an example, selective concealment of illicit NBC weapons capability is increasingly employed by adversaries in the aftermath of the Gulf War. By constructing underground facilities and detaining weapon inspectors, direct observation by satellite or airborne reconnaissance overflight assets and scientific scrutiny are thwarted. As a result, the above mentioned reconnaissance tools are increasingly less useful because of their limited ability to sense ("smell") chemical and biological agents, to measure ("feel") what sort of activity is occurring, to image ("see") the subtleties of its environment, to assess ("think") the situation, and to adapt ("react") to such an environment. To anticipate and preempt emerging threats and comply with US laws and international treaties necessitates close-up information, collective intelligence, autonomous un-manned covert methodologies, quick response-times, and immunity to countermeasures. Strong motivation exists to deploy semi-autonomous, ground- and air-based, sensing networks in combination with secure theatre communication nodes.

A Solution. In the scenario proposed here, mesoscopic-scale mobile robotic warfighters carrying specialized sensor packages equipped with miniaturized optical reflectance modulators could be infiltrated to suspected proliferation sites or trip-wire locations, interdiction chokepoints or cease-fire boundaries. Data acquisition between a distant mothership or human operative consists of positionally registered and interactively monitored remote transponders using a wide-bandwidth, atmospherically compensated laser communication protocol. To improve robustness of the intelligence gathering process, the robotic agents themselves could be endowed with rudimentary learning ability for collaborative and organized collective behaviors including local remote-sensing coordination (search, evasion, navigation), self-compensation, and adaptive reconfiguration for global optimization of a mapping signature (e.g., signal-to-noise ratio for anthrax, TNT, etc.). Comprehensive representation of the tactical environment in denied areas using an inter-netted architecture of modular war-fighters which possess redundancy overlapping mission capabilities and a degree of decision-making autonomy will significantly enhance operational functionality in dynamic scenarios subject to hazard uncertainties, imprecise information of competing constraints. By creating fault tolerance and improving response-time to active or passive threats

without information bottlenecking, and intelligent control system which tasks low-level behaviors in coordinated cooperative fashion can improve the performance of distributed surveillance systems, particularly in complex, information-dense clutter backgrounds such as hostile urban battlespace. When combined with next-generation enabling technologies for ground-based measurement and a communication protocol to support multi-robot connectivity and sensor data-sharing, this approach will form the basis for a new paradigm to remotely exfiltrate critical data with high spacial resolution and situational awareness from previously undetectable targets such as underground structures and migrating effluents or CBW plumes. Areas of specific concern to be addressed by this approach would likely include suspected NBC proliferation sites, underground WMD storage/production facilities in violation of arms-control agreements, terrorist staging and training centers, and clandestine deployment sites for impending military incursions.

Solution Implementation. Implementation of the above multi-agent system combines physics-based many-body particle models, artificial intelligent algorithms and global optimization techniques, and sensor R&D (not discussed here).

The interactions and collective effects of large particle ensembles is the study of physics-based models. Physical solutions with strong theoretical underpinnings integrate particle-particle interactions over the entire particle ensemble. Particle (or robot) attributes, such as charge, mass, or spin, can be linked to macroscopic physical states through known force laws, such as electromagnetism and gravity. The introduction of additional attributes represented by various neural network topologies expands and enhances the autonomy of each robot and the swarm's collective response optimized by genetic algorithms to adapt to unexpected threats or opportunities. As a result, the actual physical motion and decision-making attributes can be studied and generated from a superposition of real and fictitious forces generated by system decision-making neural processes.

Artificial intelligent algorithms realize particle (robotic) attributes. The brains of the robots consist of genetically-trained neural networks developed using Sandia developed (and subsequently enhanced) software package, the neural network designer (NND). Neural networks analyze and interpret the robot's sensor values giving it a certain level of consciousness by sensing, measuring, imaging, assessing, and reacting to its surroundings or a threat. The superposition of these "smart-" or "pseudo-" forces and the collective forces of the physics-based models give the swarm the flexibility and the ability to respond rapidly and to optimally readjust in order to fulfill its mission statement (i.e., swarm theory).

Lattice gas models and plasma dynamics (physics models) are non-linear. Neural networks are non-linear. Environmental conditions are non-linear. Therefore, a swarm of small, smart, ground-based, sensor-driven robots is a large, nonlinear-coupled, complex system. Such a system could as easily destroy itself rather than characterize an unknown effluent. The ensemble's state must remain within a certain defined phase-space of its variables (e.g., particle attributes, functional forms of forces, image processing techniques). Using evolutionary techniques such as genetic algorithms (GAs), the swarm learns how to behave in an optimal fashion. Learning is inherently nonlinear and discontinuous. GAs implement non-derivative, stochastic methods, which is better at overcoming local maximums and finding optimal solutions for a larger set of fitness functions. Given a population of swarms, GA operators evolve the swarm's attributes (behaviors) to optimally accomplish a given task.

The swarm must be taught how to interact locally in order to maintain desirable global states. Instead of studying the macroscopic state of well defined particles, can we determine par-

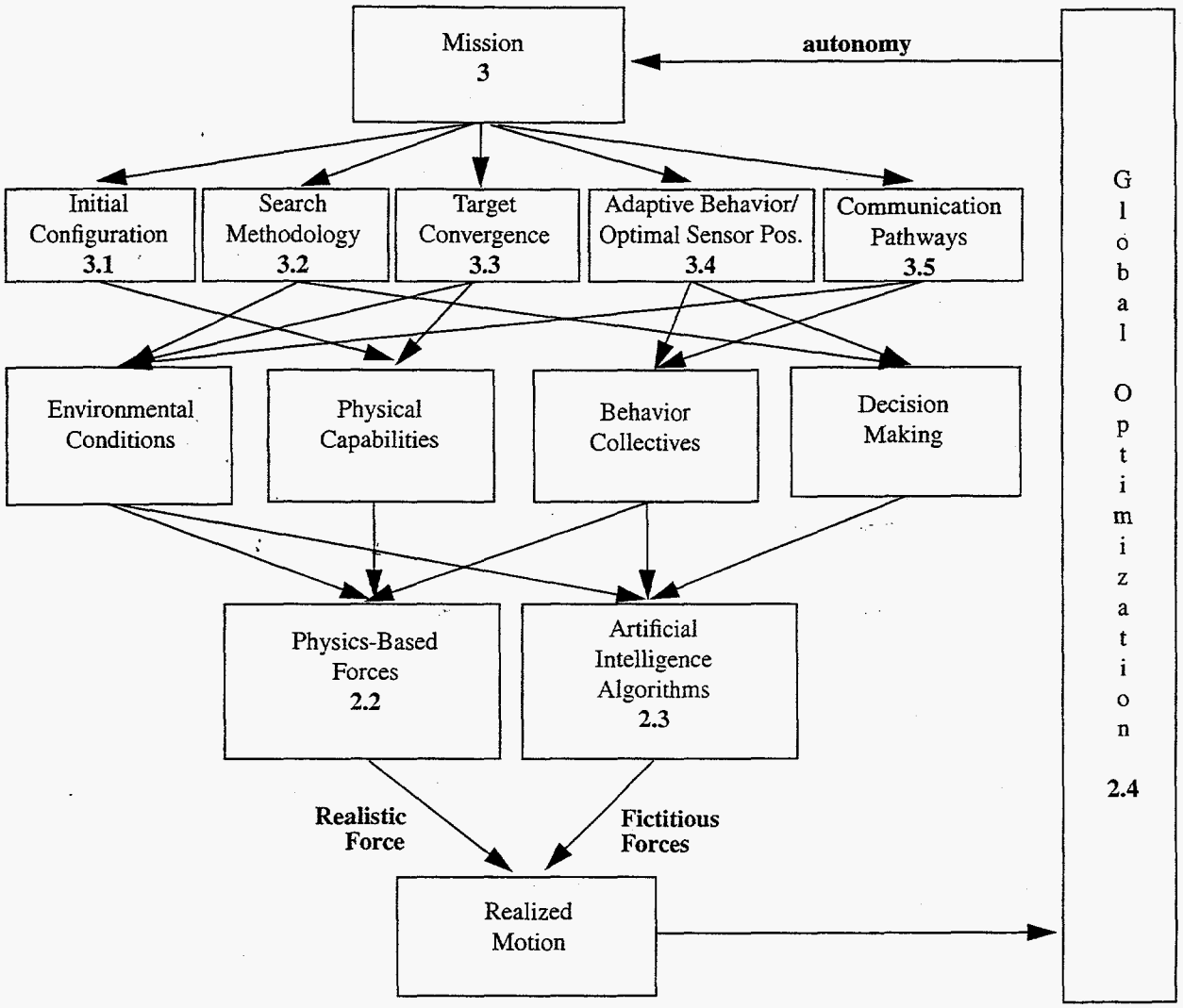


Figure 1. Global Optimization of a Robotic Swarm for a Specific Mission Using Physics-Based Models.

ticle attributes that fulfills an overall mission statement. One may coin the phrase "reverse-physics" in analogy to reverse-engineering the BIOS of a PC, cloning. The factors inserted into the fitness functions when determining the networks and swarm configurations include the derived physics equations limiting the swarm's state within a stable phase-space, each robot's collection of attributes or neural networks, the communication network topology between robots, and data image processing or signal enhancement.

The latter fitness function objectives deal with the swarm's collective intelligence capabilities. Each robot within the swarm represents a neurode. The interconnections between robots realizes the synaptic connections of a brain, a neural network, creating a collective intelligence to the entire swarm. From this architecture, the ultimate goal of remote-sensing is achieved. Differ-

ent sensors and local data transformations allow for data fusion and complex imaging techniques. In addition, this global intelligence also provides the swarm with the ability to sense its own state, providing self-diagnostics (i.e., self-awareness) and furthering robustness.

To recapitulate, our working models have a somewhat hierarchal approach to adaptive behavior and remote-sensing. The potential fields modeled by the LG or PIC simulations provide a fast, trip-wire response to an immediate target or event. Local, nearest-neighbor neural network models study this behavior. Local analysis transforms point sensor information for further processing and initiates additional pseudo-potential fields to adapt to local disturbances. Collective global neural networks provide imaging-like capability to determine the swarm's entire state. Evolutionary algorithms teach the swarm which physical attributes are needed to provide desired macroscopic dynamics when attaining mission objectives. These mission scenarios attempt to solve real problems in the theatre of emerging threats. Two targeted threats include hazardous bio-chemical plume remediation and ground-penetrating radar of underground structures. Our goal is to develop and apply these ideas both theoretically (and through simulation) and experimentally leveraging state-of-the art Sandia technologies in the area of emerging threats.

The block diagram in figure 1 interprets the global optimization of a robotic swarm for a mission using physics-based models. Once a mission statement is decided upon, different required activities are linked to environmental conditions, physical abilities (e.g., robotic platforms), collective behavior (e.g., obstacle avoidance or remote-sensing), and decision-making processes. Accomplishing or dealing with and other factors is realized by modifying and controlling the physical ensemble (dynamics) of the agents. Response, avoidance, and sensing all require a level of intelligence (conscienceness) at both a local (effecting subset of swarm) and a global (same correlation length as the collective) level. When and at what level one effect dominates (in time) over others (series of tasks) can be determined through global optimization algorithms. The end result is the most efficient and successful method to accomplish the mission.

This paper reports on our physics-based model of an intelligent, autonomous, multi-agent system (section 2), implementation of this model during various states of a possible remote-sensing mission (section 3), and our work applied to three project areas (section 4). This report concludes with a brief status (section 5) of our R&D effort.

2 Model

2.1 Overview/Perspective

As stated in the introduction, our model should include technologies that enhance remote-sensing signals and images while behaving in an autonomous manner, adapting to and overcoming changes within its environment. A simple example would be to follow a plume across a field (plume mitigation). A more challenging scenario attempts to reconfigure the transmitting and receiving antennas mounted on each robotic platform to best resolve an underground facility (ground penetrating radar application).

Autonomous, adaptive behavior refers to techniques that allow a swarm to either move with a target, surround a hazardous source, recognize multiple sources, avoid obstacles, and other similar skills. The ability to perform these complex behaviors -- plume remediation, ground penetrating radar imagery, and ballistic missile interception -- build on the previously mentioned testbed of intelligent algorithmic capabilities. Our approach is to provide a solid foundation on multi-agent (swarm and flocking) ideas by developing a model that can be tested both in theory and through simulations, before and during hardware development. This section conceptually outlines how physics-based models and artificial intelligence can be combined and globally optimized to develop an intelligent swarm methodology. The flow diagram in figure 2 forms a picture of such a structure and provides additional insight into this methodology.

Overall, the logic is as follows : (1) sample the environment, (2) recognize local and global interactions, and (3) adaptively modify the physical state of the swarm by applying adiabatic changes to the potentials.

The macroscopic state of an N-robot swarm can be determined using a physics-based model, such as a lattice gas or a plasma. The advantage of these models is that their collective properties depend (almost) entirely on nearest-neighbor interactions, keeping the complexity of the robots to an absolute minimum. The swarm's state is defined by three types of forces (1) a repulsive force, defining a minimum distance between robots; (2) an attractive force, providing the cohesive "glue" binding the robots into a swarm; and (3) a source potential (in response to a robot's sensor readings). Therefore, the strength of these forces, represented by α, β, and γ, define the state of the swarm. In addition to these forces, local neural networks provide additional potential terms modifying the state represented by the parameters $\delta_1, ..., \delta_p$. In order for the swarm to act autonomously, such as overcoming environmental constraints (terrain) and searching out targets by adaptively configuring, a feedback mechanism must be in place that adiabatically evolves the swarm's state equations, $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$ and $\Delta\delta$. This mechanism implements a second neural network or a genetic algorithm, or a combination of both, depending on the mission statement assigned to the swarm. Processed information from each robot is used as input into the global net which understands the macroscopic concerns of the swarm's environment during its mission, such as obstacle avoidance and source identification (imagery). Eventually, information and possibly additional processing must be communicated. As with ultra-wideband, short-pulsed radar, the robots should transfer its findings to the user for verification and mission modification. A bidirectional communication link provides this capability and additional control over the swarm's behavior if and when necessary. This potentially can enhance the adaptability, changing mission objectives and/or eliminating dead-lock states, thus minimizing physical and

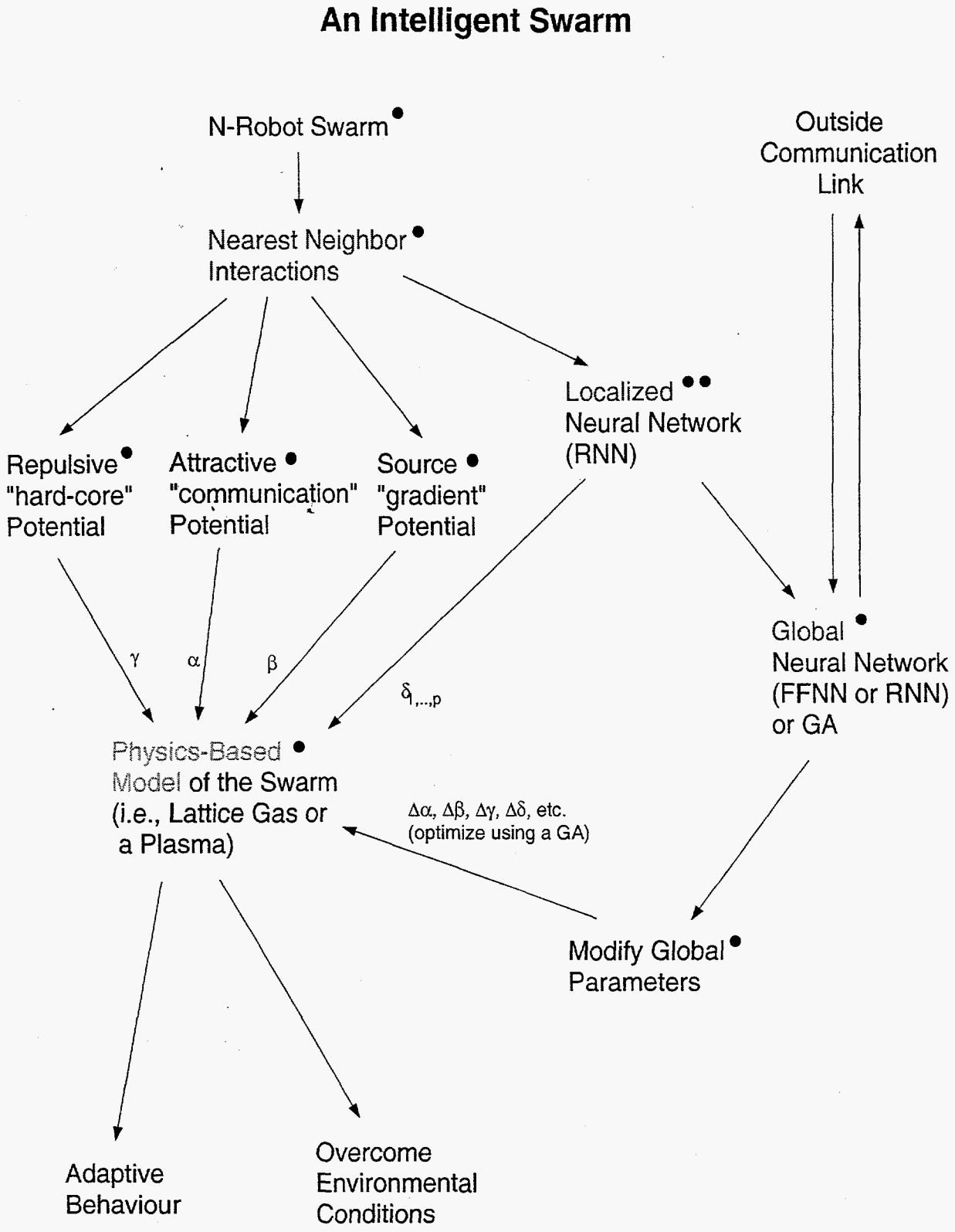


Figure 2. Block or Flow Diagram Representing An Autonomous, Intelligent Swarm Based on a Physics Models.

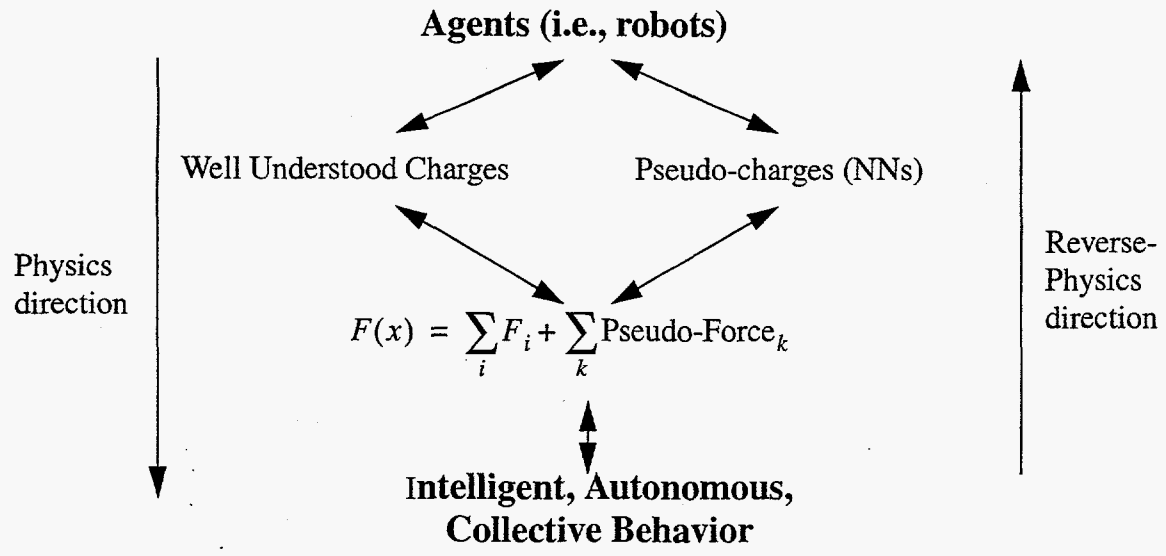


Figure 3. Block Diagram on "Reverse-Physics" Concept.

communication bottlenecks.

Evolutionary techniques such as genetic algorithms globally optimize the behavior of the swarm, efficiently fulfilling its mission objectives. Evolutionary programming techniques have the advantage of being stochastic, avoiding local minimums or maximums, and not relying on a function being smooth or continuous (much like ourselves). Incorporating theoretical physics-based calculations that describe the collective (macroscopic) state of an N-body system, a population of swarms evolve (or learn) an optimal form for its nearest-neighbor (microscopic) interaction potentials. Particle attributes lead to forces which, in turn, lead to predictable collective behavior. Our goal is to reverse such a process. To summarize, through the use of artificially intelligent algorithms and evolutionary techniques, the robot's attributes (nearest-neighbor interactions) which recreate not only the desired (macroscopic) state of the swarm but prevent instabilities and promote robustness can be determined. This will be referred to as "reverse-physics".

The rest of section 2 discusses physics-based models (section 2.2) and artificial intelligent algorithms (section 2.3). Global optimization through evolutionary programming (section 2.4) determines the robots' and swarm's attributes necessary to perform autonomously (section 2.5).

2.2 Physics-Based Models

2.2.1 Introduction

Nature appears to be governed by only four fundamental forces : gravitational, electromagnetic, weak, and strong. Associated with each of these forces is a coupling constant (which can be complicated, SU(2) weak interaction) and a "functional form" (mediated) dependent on the attributes of the particles involved (and the gauge symmetry (or spontaneously broken symmetry) of the force). A short list of particle attributes includes mass, spin, electric charge, hyper (or weak) charge, and color charge. A combination of these forces bind the atom and its nucleus

together, form planets and galaxies, and determine plasmas and gases. By measuring particle interaction cross-sections, the form of the interaction potential related to the particles' attributes can be determined. It is through various statistical means (i.e., partitions functions, mean field theories, etc.) that the macroscopic and dynamical state of an many-body system is determined.

Unfortunately, particles alone cannot infiltrate a hostile country and remote-sense underground testing facilities. Something with alittle more intelligence is needed : robots. Forces, formed by attractive, repulsive, and gradient potentials, require a minimal amount of hardware but would be no smarter than a gas or plasma. Additional forces are needed to allow the swarm to adapt according to sensor information. These "pseudo-forces" are local, nearest-neighbor interactions and are associated with a robot's attribute, a "pseudo-charge". Realizing the brains of each robot by a set of genetically-trained neural network topologies (called attributes), a trained can decipher the conditions within its environmental. Evolutionary programs determine the form and strength of the pseudo-forces in response to these local neural nets. The combination of these forces influences the state of this multi-agent system and allows it to perform complex behavior patterns. The latter intelligent and evolutionary techniques will be discussed in sections 2.3 and 2.4; but first, two many-body physics models best suited to realize swarm behavior (and incorporate additional pseudo-potentials) -- lattice gases (LG and LG automata) and Particle-In-Cell (PIC) codes representing plasmas -- are presented.

2.1.2 Lattice Gas Methods

A lattice gas[1,2] is a collection of particles whose positions take on the interconnecting points of a square, triangular, hexagonal (or other topological) lattice. Any one lattice site can allow only a single particle to reside on it. If the kinetic energy of the particles are negligible and a simple, constant nearest-neighbor potential is introduced, the partition function for such an ensemble follow that of an Ising model[1]. Such a partition function is theoretically well understood. Therefore, the thermodynamic states of the gas can be derived, such as the volume, pressure, temperature or internal energy, and density, and linked to those of the swarm, such as communication length, sensor range, cohesiveness, and speed. Current research extends lattice gas models to include kinematics and the effects of heterogeneous mixtures, thus determining phase transitions and flows (e.g., hydrodynamics). Our attempt is to modify these models that realize gaseous systems by endowing them with intelligence (attributes) and train the robots to follow the well-controlled states predicted by these models.

At every time step within the algorithm, the particle system evolves by applying propagation and collision operators. The propagation operator moves each robot along a single lattice edge dependent on the applied vector force. The collision operator (collision rules), defined by the interaction potentials, determines which edge the particle prefers. Three potentials control the state of the swarm include a repulsive, an attractive, and a gradient (source) field. The functional form of these fields can vary, resulting in different state equations. In this paper, the functional forms of the potentials are $F_{repul}(r) = \frac{1}{r^2}$ for the repulsive force, $F_{attract}(r) = r$ for the attractive force, and $F_{src}(r) = \frac{\partial}{\partial r} Signal(r) \approx \frac{\Delta signal}{\Delta r}$ for the source force. Each of these forces interact with their line-of-sight (LOS) nearest-neighbors. Such an interaction range can be realized and measured passively by applying IR or sonar detector responses, or the fusion of both.

Each robot determines the forces acting on it according to

$$F(r) = \alpha \cdot F_{attract} + \beta \cdot F_{src} + \gamma \cdot F_{repul} \cdot \qquad (1)$$

Optimizing the coupling strength parameters α, β, and γ (and eventually the functional forms) allows the swarm to reconfigure, adapting to the environment it senses.

Lattice gas computational techniques with the correct choice for the lattice and its collision rules lead to the macroscopic Navier-Stokes equations, both 2- and 3-dimensional forms[3]. Leveraging off this methodology and the current research studying various interaction rules, the swarm's state can be taught to remain within a stable phase-space (away from turbulent flows) by introducing the theoretical physical equations (e.g., the Navier-Stokes equations) into the fitness function. As a result, the swarm's behavior remains robust and stable. In addition, self-diagnostics, such as global neural networks sampling the swarm's own state, further reduce divergent behaviors. Ultimately, a genetic algorithm determined the best form of potential functions that allow the swarm to react autonomously to its environment, such as avoiding obstacles and analyzing targets.

This model targets applications dealing with adaptive reconfiguration of (ground-based) sensors in the field of remote-sensing. Most detectors in the research community consist of the same elements (scintillators, solid-state radiation detectors, calorimeters, etc.) but of different sizes and in different configurations in order to optimize the phase space of a targeted signal. By assigning different sensors (sensor fusion) on the robots and invoking intelligent algorithms, it can adaptively form an optimal configuration for a sensitive signal within a changing environment. Two such applications include characterization and mitigation of gaseous plumes and ground penetrating radar of underground structures.

In order to match simulations of swarms to a lattice gas models additional features are being explored, such as expanding the nearest-neighbor model to that of a line-of-sight interactions for the repulsive and cohesive forces, adding pseudo-potentials as the result of local artificial intelligence algorithms (neural networks), deriving phase transitions from gaseous to crystalline states to better control macroscopic structure of the swarm (i.e., condensing onto a target), and predicting critical phenomena to prevent local instabilities from becoming global instabilities. All of these results scale relative to the size the swarm, such as the cohesiveness of the swarm. Such theoretical work and verification through simulation will establish our ability to control the macroscopic state of a swarm through simple, local interactions.

2.1.3 Plasma Physics Methods

Plasma physics[4] deals with the collective effects of a system of electromagnetic fields and of particles, such as conducting liquid or gas. Particle dynamics, conduction, occurs when the charged particles move under the action of the applied fields. Due to this dynamical system, the mass motion of the particles couples with the system and the fields. Biological systems also contain such effects. A flock[5] of interacting individuals or birds must not only communicate with on another but deal with the wake of the entire group. Simulating self-consistent aerodynamic forces is a strength of realized by numerical techniques developed through plasma physics models.

To compute the long-range correlations in a plasma requires a summation of the forces

between all particles, which is of $O(n^2)$. Particle-in-cell (PIC) codes[6-11] break up the physical space into a mesh or cells. The particles in each cell determine the local charge density, from which the potential fields at the mesh corners are approximated implementing Poisson's equation. In effect, an average smeared charge simulates the overall (long range) particle field of the plasma. Simulations can occur with speeds of $O(n)$, allowing many simulations to run in real-time. The stability of these algorithms are well documented, using statistical mechanical methods to determine how mesh size, time scale, particle density, and energy conservation influence performance (stability and consistency).

Direct realization of the PIC codes onto a robotic platform is not inherent due to its averaging effect at the mesh points which simulates the "flocking" force. To alleviate some of these problems, nearest-neighbor interactions within each cell are included. Two examples of which are the repulsive and attractive forces that prevents particle collisions and clusters elements around center-of-mass regions, giving the swarm the ability to divide. Because these forces are of the form r^{-1} and r^{-2} (e.g., electrostatic and gravitational interaction types), incorporating these functional forms into the theoretical structure of multi-particle systems is well understood. As a result, stable and robust behavior is maintained by training the robot's intelligence relation to the derived macroscopic plasma equations.

Two applications in the area of aerodynamics include multiple target condensation and ballistic missile intercept simulations. In each case, an external field created by the target is applied to the mesh. By modifying physical parameters, such as center-of-mass, friction, dissipation, aerodynamic factors, and thrust, the swarm's response can be optimized.

2.1.4 Summary

The advantages of modeling swarm dynamics with particle simulation codes include (1) these codes have been benchmarked, (2) they provide an accurate dynamical description of N-body system, (3) the theoretical equations describing the particle ensemble are well developed and tested, (4) the link between short- and long-range behavior is established, (5) resolves the n^2 computational bottleneck, (6) a methodology exists to include and determine swarm behavior when adding of other features or attributes, and (7) robust training and performance can be built into the framework of the swarm's behavior and tested in real-time.

Even though lattice gas automata and PIC codes contain many common and desirable features, they can complement one another when realized onto a robotic platform. Direct realization of the PIC codes onto a robotic platform is somewhat incomplete due to its averaging effect at the mesh points. However, its ability to efficiently simulate many real-world features, such as inertial forces, platform constraints, friction and drag effects, gravitational forces, and boundary conditions, in an efficient manner is a large advantage. Our current work applies these codes in the area of many-body (flock) flight dynamics. Lattice gases also has the inherent problem of being restricted to a lattice structure (which may be accounted for as an inefficiency), but all of its dynamics are self-contained in the definitions of its nearest-neighbor interactions. Such a system can be directly realized by current robotic sensor technology. Similar to plasma physics techniques, LG ensembles can link physical (thermodynamical) parameters with robotic variables such as maneuverability and reconfigurability. Applications include the ability to adapt and reconfigure in order to optimize its remote-sensing capabilities. It is our opinion that these two concepts will eventually merge together into a single technique resolving any direct realization

defects to a robotic platform.

2.3 Artificial Intelligence

2.3.1 Introduction

Artificial intelligent algorithms realize additional particle (robotic) attributes. Therefore, the robot's "brains" consist of genetically-trained neural networks. Neural networks[12,13] analyze and interpret the robot's sensor values giving it a certain level of consciousness by sensing, measuring, imaging, assessing, and reacting to its surroundings or a threat. The superposition of these "smart-" or "pseudo-" forces (strengths δ_i) and collective forces (strengths α, β, and γ) of the physics-based models give the swarm the flexibility and the ability to respond rapidly and to optimally readjust in order to fulfill its mission statement (i.e., swarm theory).

Inspired by the field of neuroscience, artificial neural network models brain and/or nervous system activity which realizes intelligence. The basic processing unit is the neuron. A neuron consists of a stoma or base, called a neurode, and many interconnecting synapses, or connections, between other neurons (via its dentrites and axon). Associated with each neurode of a neural network is an activation function. The form of which is modeled by a squashing function, a nonlinear function which remains at a constant activation once beyond a threshold value, remains unactivated below a threshold, and has some function form linking these two activation states ("on" and "off"). The input into a neurode is the superposition of many other neurode output activation values weighted by individual synaptic connections. Thus, a neurode's output activation depends on the weighted sum of previous connected neurode activity and the form of its activation function. The architecture or structure of the neural network is defined by the inter-connectivity of its neurodes, the form of the activation function, and the values of its weights. Architectures which only interconnect between successive layers in the same direction are called feedforward neural networks (FFNNs). If each neurode in one layer is connected to each neurode in the following layer, it is called a fully-connected FFNN. Architectures with connections that feedback to previous layers, called recurrences, are called recurrent neural networks (RNNs). Different feedback or recurrent topologies form different types of RNN (e.g., Elman and Jordan).

The process by which one determines the weights of a neural network is called training[1]. There are two types of training methodologies : supervised and unsupervised learning. The latter has not been used in these studies; therefore, it will not be discussed. Supervised training compares calculated neural network outputs to the values associated with the inputs, called the expected outputs. Determination of the weights depends on the output residuals, the differences between the neural network outputs and the expected outputs. Therefore, training a neural network is an iterative process, pairs of input-expected output pairs are "shown" to a neural network and the weights are optimized to give the smallest residuals. The network learns to correlate (recognize) an input pattern by activating certain outputs. Traditional learning algorithms implement back-propagation or gradient search methods to find an optimal set of weights. However, if the net's objective contains discontinuities or if the initial conditions are far from the best solution, training can get caught within a local minima. Implementation of evolutionary programs, such as genetic algorithms (GAs)[14], resolves many of the above mentioned problems. GAs are stochas-

1. in the strictest sens this is not completely accurate, but this is how it will be defined in the context of this report.

tic and based on non-derivative techniques, thus they are less likely to get caught in a local minima. They also span a larger variable phase-space without requiring a smooth, continuous objective, thus more likely to find a global optimal solution for a larger set of objectives.

Successful implementation of FFNN and RNN architectures has been employed. Recurrent neural networks trained to decipher nearest-neighbor information interprets directional (vector) motion from sensor readings by "remembering" past events. Feedforward networks trained to recognize bitmap patterns help determine the extent of an obstacle, its edge, or other features that will be needed to overcome it through adiabatic ensemble transitions.

2.3.2 Immediate Sensor Response

An immediate sensor or trip-wire response is an involuntary reaction to the environment. As an example, lets consider the situation where a wind-blown plume of toxic gases (biological agents) propagates toward a community as a result of an overturned truck. Sensors on the robots measure the gradient of the emerging threat. Through nearest-neighbor interactions, the strength of source potential increases, causing the swarm's (dynamical) state to follow the plume or to track-down the truck. By determining the optimal set of parameters for the potentials, $[\alpha, \beta, \gamma, \delta]_{target}$, the swarm condenses onto (or beneath) the gaseous effluent. The behavior of the swarm could also be trained to send a fraction of the robots to follow and the rest to find the truck. After the swarm reconfigures, it has more time to think, to process sensor information, and optimize signature signals.

Applying the source potential alone is not a solution. Depending on the relative strengths of the parameters α, β, and γ, various and important physical parameters are controlled. Cohesiveness is maintained by keeping α relatively large, but not too large so the swarm can no longer follow. The relative ratio of α to β has an effect on the radial variance and density of the swarm. A lower density improves mobility (speed) and maneuverability. Specific ratios may provide patterns advantageous to remote-sensing applications. All three forces combined allows the robots to adapt to according to the plume's shape, size, and dissipation rate.

Future work will focus on further development of the state equations for lattice gases implementing statistical mechanical methods. These derivations will allow for a better understanding how density and temperature relate to communication and sensor ranges and how nearest-neighbor interactions relate to swarm cohesion, condensation patterns, and obstacle avoidance.

2.3.3 Local Intelligence

Referring back to the overturned truck example, determining the wind (or plume) direction would be helpful. By sharing nearest-neighbor sensor data, a robot determines propagation direction implementing its local RNN trained to recognize time-dependent sensor information to determine. This directional or "inertia" pseudo-potential creates a "flocking"-force toward the source by modifying the swarm's state. This section begins to develop the tools, a neural network, trained to determine the direction of a time-dependent source (TDS). After some introductory remarks, the neural network design, results from simulations, and conclusions are discussed.

Introduction. An advantage of a swarm is its use of many sensors to spread-out and sample different types of data. A neural network processes the data. In turn, these results allow the swarm to

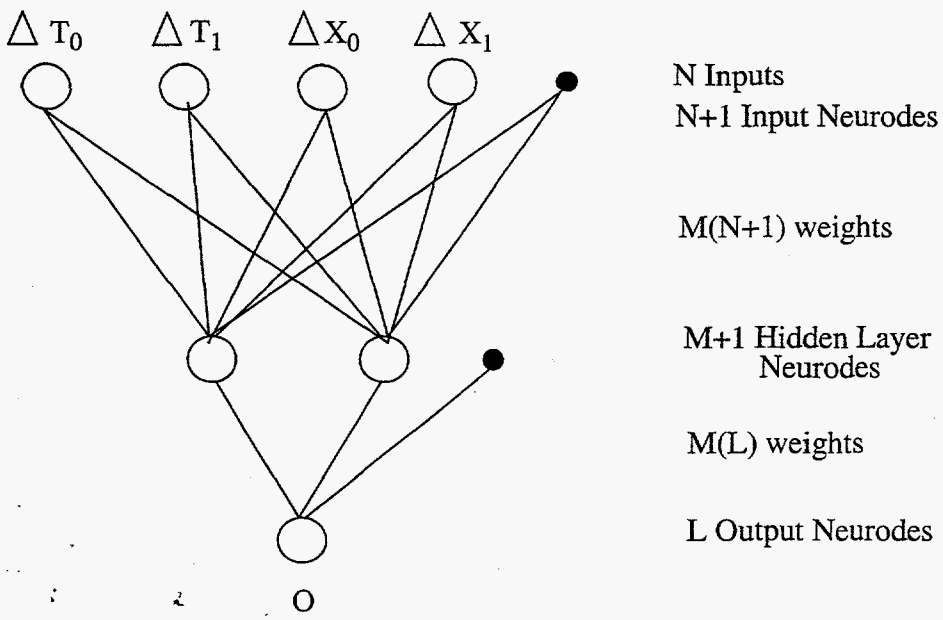


Figure 4. A feedforward neural network (FFNN) trained to determine the propagation direction of a gaussian signal. Dark neurodes represent threshold units and open circles represent neurodes applying sigmoidal activation function.

adaptively reconfigure relative to the source, optimizing data quality. The input data can have three different time dependencies due to (1) the motion of the robots, (2) the motion of the source, and (3) the motion of both the swarm and the source. Type two will be considered. These results will help in the development of remote-sensing applications such as using bio-chem sensors recognizing gaseous plumes and seismic sensors to detect geophysical events.

Each robot processes relative information, sensor data communicated by a neighbor, that determines the path of a simulated plume. Because of the limitations or inappropriateness of the global delay-line feedforward neural network (see section 2.3.3), a gradual progression in the analysis is taken. Test case one simulates a 1-dimensional (1D) TDS along a line, test case two simulates a 1D TDS in a 2D plane, and test case three simulates a 2D TDS in a 2D plane. The latter is the objective, but such an analysis can eventually pursue 1-, 2- and 3D TDSs in 3D space. In each case, the base function representing the form of the source (e.g., a plume) is a gaussian function. The ability of the chosen neural network architecture to generalize, training a NN with a limited representation of the entire phase-space yet perform flawlessly over the full phase-space, is important. The result is a genetically trained recurrent neural network (RNN).

Neural Network Design. Some possible design changes under study include the form of the inputs, the NN architecture, and the extent at which to communicate information.

Optimally, each robot would like to communicate what it believes to be the path of the source. Many iterative techniques determine the form of this function (in this case the path of the TDS) by implementing a Taylor series expansion in the phase-space around a given point (a given

robot).

$$f(x,t) = f(x_0,t_0) + (t-t_0)\cdot\frac{\partial f}{\partial t} + (x-x_0)\cdot\frac{\partial f}{\partial x} + \ldots \qquad (2)$$

For a robot in the swarm, the first term, $f(x_0,t_0)$, represents a (noise) threshold. If the robots' sensors exceed a threshold of 2%, relative to the TDS's maximum amplitude, it evokes its NN to process the information. The second term represents the time interval (or the sample time-step), which is defined as $\tau = t - t_0$ and held constant during the sequence of an event. In similar fashion, the spatial distance between samples, the third term, is held constant and defined by $\delta = x - x_0$, the nearest-neighbor distance along a line (1D) or on a grid (2D). The time and spatial derivatives, second and third terms respectively, are represented by the quantities $\Delta T_0 = f(t,x) - f(t_0,x)$, $\Delta T_1 = f(t,x_0) - f(t_0,x_0)$, $\Delta X_0 = f(t_0,x_0) - f(t_0,x)$, and $\Delta X_1 = f(t,x_0) - f(t,x)$. After expanding equation (1) to first-order, a robot need only interrupt its nearest-neighbor, who communicates its sensor values $f(t_0,x_j)$ and $f(t_0-\tau,x_j)$. It is the differences in these values that are used as the inputs to first train the NN and later to determine the direction of the source.

The 1D case implements a single, hidden-layer FFNN architecture (see figure 4) with the four previously described inputs. In addition, a digital realization of the 1D case is presented to evaluate the NN's performance. In the 2D plane, the robots require knowledge of its previous state (i.e., memory) in order to resolve the direction of a TDS. By incorporating recurrency into the FFNN architecture, a RNN (see figure 6) can be trained to recall the necessary information (that of the hidden layer) about the previous sequence of states, thus resolving the direction of the TDS.

Typically, the input neurode activation function, IAF(x), differs from the hidden and output neurodes. IAF(x) usually is equal to x. However, only directional information is needed. Robust performance is preferred over detailed analysis. Digitizing the inputs using a step function

$$\mathrm{IAF(x)} = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \qquad (3)$$

forms a bipolar, digital input signal. The activation function representing the hidden and output neurodes is the hyperbolic sigmoid squashing function

$$AF(x) = \tanh(a\cdot x - b) + c. \qquad (4)$$

The parameters a, b, and c are set to 5 (initially), 0, and 0, respectively.

Increasing the level of communications includes extending the range or spatial distribution and the history of the swarm's sensor readings. In each case, equation (1) can be expanded to include higher-order terms improving the network's ability to measure environmental parameters more precisely or to improve signal-to-noise ratios.

Simulations. Most time dependent sources can be broken-down into a single representative function. In 1D space, a 1D gaussian pulse with a constant amplitude and width (sigma) propagates in the positive or negative x-direction (to the right or left). In a 2D plane, a radially propagating

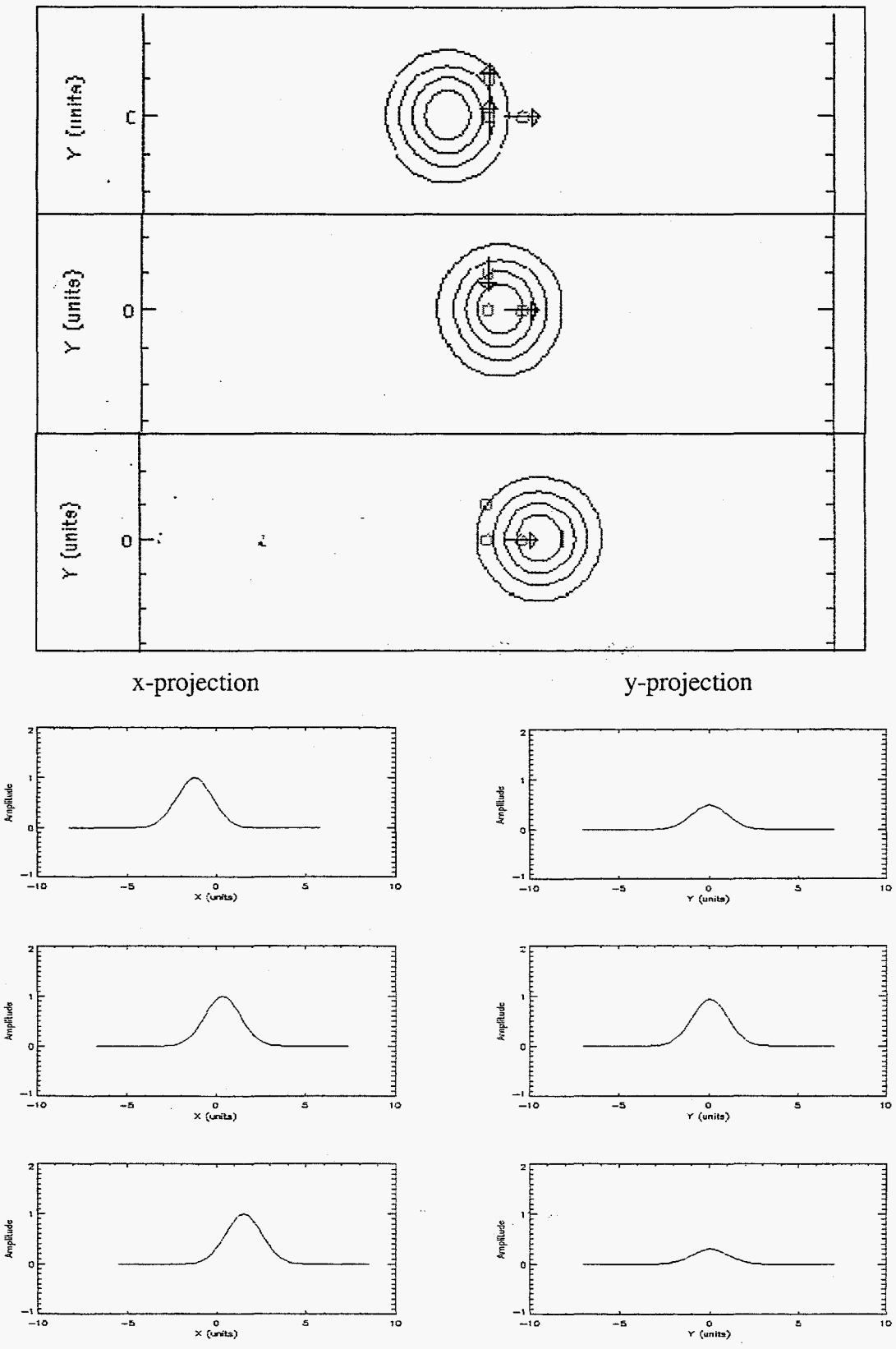


Figure 5. Top, a three robot swarm sensing a 2D gaussian pulse. Bottom, the x- and y-projections of the 2D gaussian pulse as seen by one of the robots.

gaussian pulse (1D representation) and a 2D gaussian function translated in the x-y plane can be represented. (From which, future experiments understanding 1-, 2-, and 3-dimensional time-dependent pulses can be understood). Additional functional forms can be considered as abstractions or superpositions of different chemicals within a plume having the form

$$f(x,t) = \sum_k a_k \cdot g(\sigma_k, \mu_k)$$

1D Space, 1D Gaussian TDS. The FFNN in figure 4 and the activation functions in equations 3 and 4 define the architecture. Sixteen possible input states, 24, exist, but only twelve samples are viable for a right- and a left-propagating pulse. A genetic algorithm optimizes the weights to recognize these states. This training session maximizes the fitness function

$$fitness = \frac{1}{0.1 + \sum_j^{nCases} \left(\sum_i^{nOutputs} (\zeta_i - O_i)^2 \right)_j}, \tag{5}$$

where ζ is the expected output, O is the neural network's output, and nCases is the number of randomly sequenced input/output pairs seen by the net for each generation. The results from the trained neural network are compared to those from a digital realization (i.e., boolean algebra). Using a left and right propagating gaussian pulse as the validation test case, an ambiguous state occurs within the digital model. This error is sensitive to the relative values of δ and τ, the spatial and time sampling rates. The trained FFNN resolves this situation.

A second simulation introduces the gaussian pulse

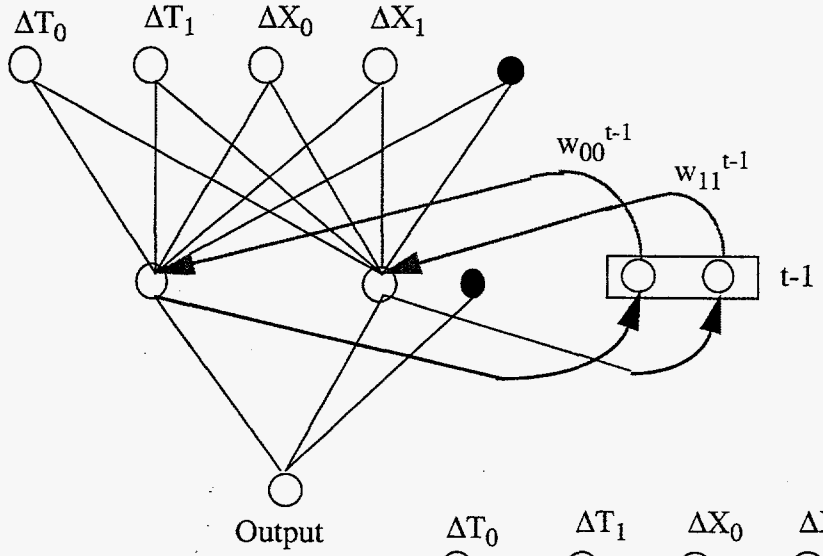
$$g(x,t) = A(t) \cdot e^{-0.5\left[\frac{(x-\mu(t))}{\sigma(t)}\right]^2}, \tag{6}$$

where $A(t) = A_0 e^{-\alpha t}$ and $\sigma(t) = \sigma_0 e^{\beta t}$, which simulates a TDS with a decaying amplitude and a spatially spreading width. Setting α and β in the [0, 0.1] range attempts to replicate the characteristics of a wind-blown, gaseous plume into the validation set. The successful validation results indicate that this architecture (the inputs) provides a fairly general solution with minimal training.

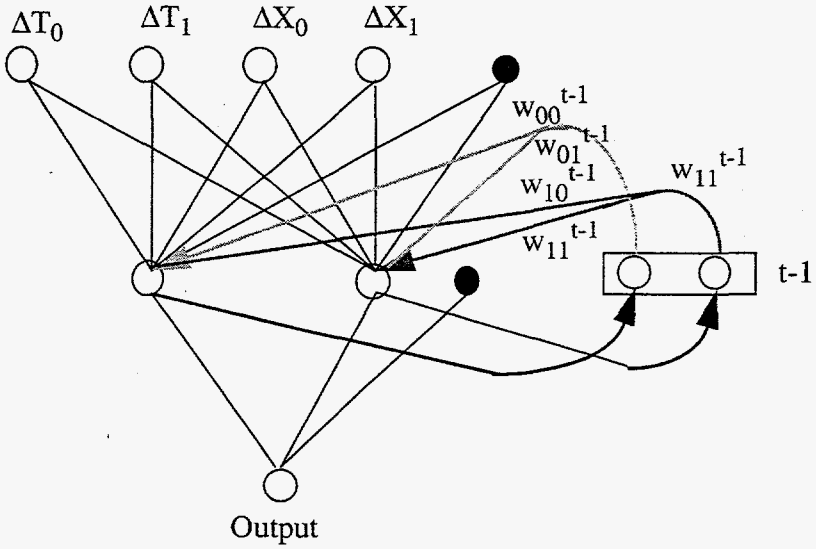
2D Space, 2D Gaussian TDS. In figure 5, a 2D gaussian pulse propagates in the +x-direction (of the xy-plane). The swarm consists of three robots at the center of this plane, top graphs. Each robot samples the pulse, every τ, and communicates its sensor values to their nearest-neighbors, a distance δ apart. The plots in the bottom right (bottom left) project the pulse as seen by a robot onto the x-axis (y-axis). Two orthogonal modes are evident. The propagation angle θ depends on the ratio of these two modes : (1) a 1D gaussian pulse with a time-dependent mean, and (2) a 1D gaussian pulse with a time-dependent amplitude (see equation 7).

$$signal(\theta) = c_{\perp}(\theta) \cdot A(t) e^{g(\mu)} + c_{\parallel}(\theta) \cdot A e^{g(\mu(t))}. \tag{7}$$

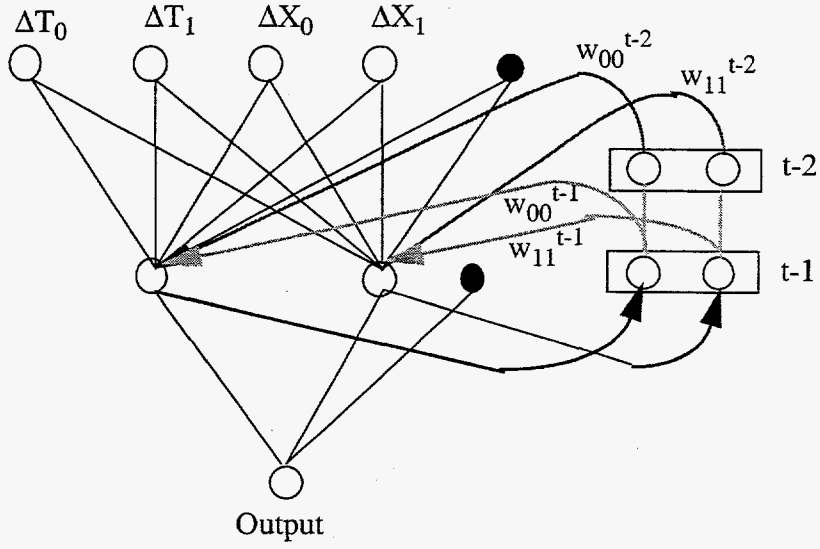
A FFNN cannot be trained to distinguish between these two modes. In some instances, the results show that all but one state can be learned. Like the 1D simulations, it is the propagation direction



Top graph, one level of self-recurrent hidden layer nodes.



Center graph, one level of fully-connected recurrent hidden layer nodes.

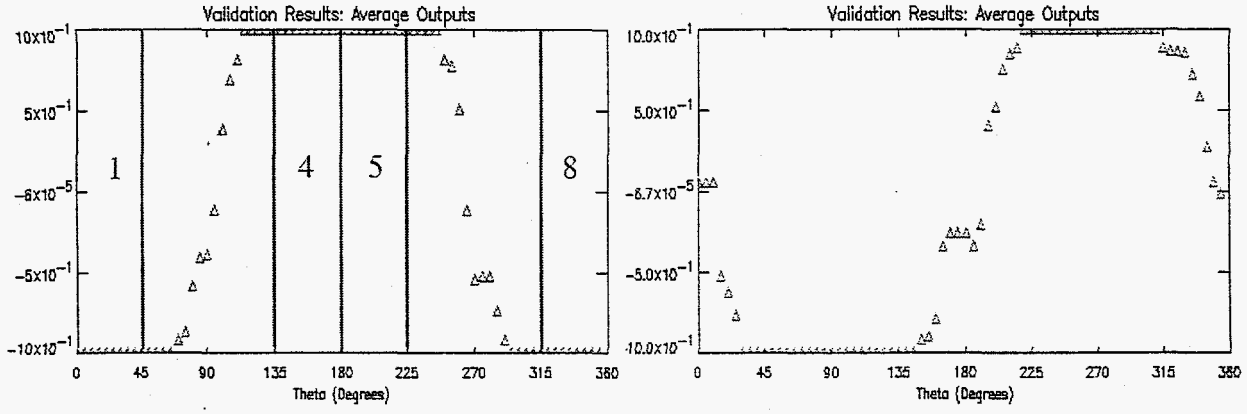


Bottom graph, two levels of self-recurrent hidden layer nodes.

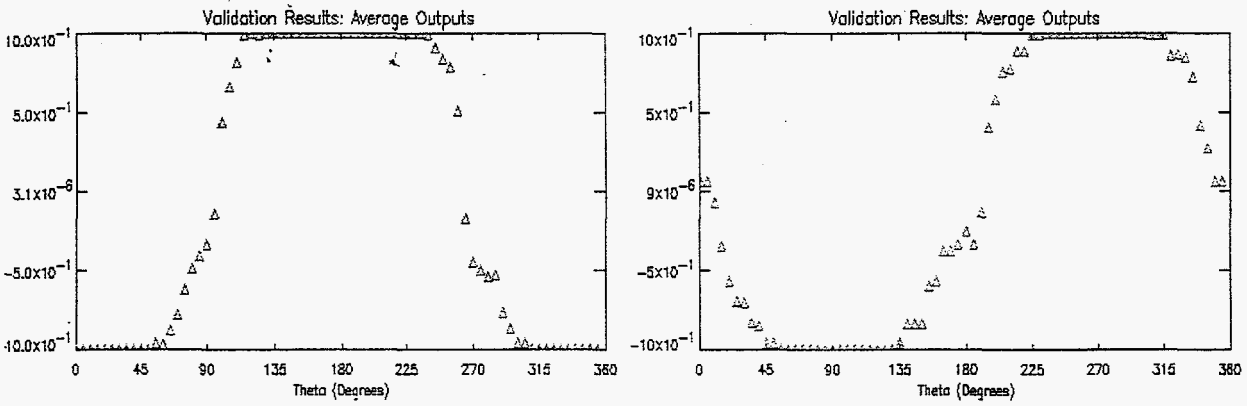
Figure 6. Recurrent Neural Network Architectures.

Output in x-Direction

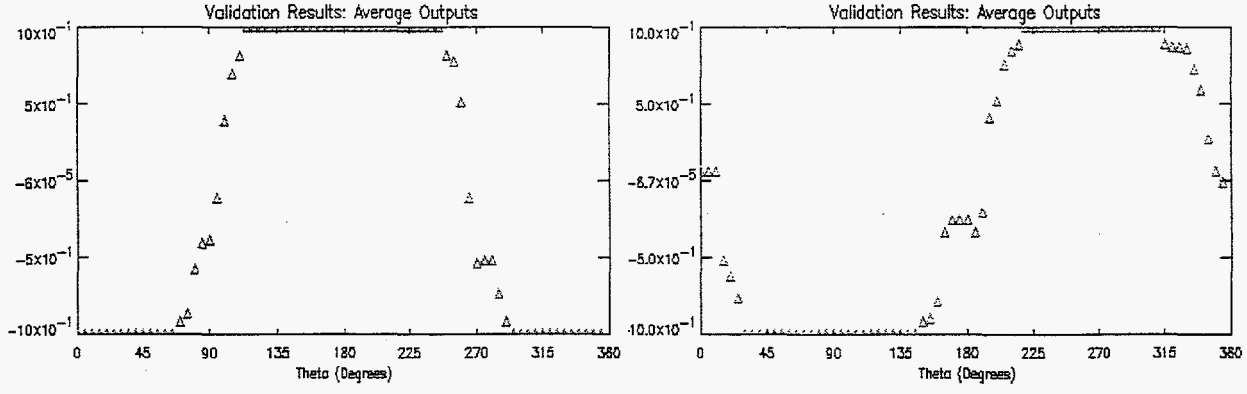
Output in y-Direction



RNN with one level of self-recurrent hidden layer nodes.



RNN with one level of fully-connected recurrent hidden layer nodes.



RNN with two levels of self-recurrent hidden layer nodes.

Figure 7. Validation results for three different RNNs. The x-axis represents the pulse's propagation angle, θ, and the y-axis is the average output value. (Note : inputs into these networks only use nearest-neighbor sensor data.)

of the pulse that causes difficulty. As an example, pulses with a θ of 45^0 (the -x and +y directions) and -45^0 (the +x and -y directions) show at least one input state belonging to both states in equation 7; therefore, the FFNN can never train correctly. Even when eliminating this state from the training set, the validation results remain poor. Thus, a FFNN of any configuration, such as changing the number hidden layers and hidden nodes and forming different activation functions, cannot resolve such a superposition. By recalling past patterns, it may be possible to resolve directional motion.

Recurrency is equivalent to feedback loops within the network and effectively stores the knowledge of the previous states in its weights over time. Having the ability to recall past information allows the robot to separate out the two modes in equation 7. Thus, genetically-trained, recurrent neural-networks (RNNs), shown in figure 6, stores the time-dependent history of the 2D gaussian pulse. The three RNNs in figure 6 contains a weighted feedback loop of its own hidden layer output neurode to itself (top graph), a fully-connected, weighted feedback of its output hidden neurodes to its inputs (center graph), and a weighted feedback loop of its hidden layer neurodes for two time-steps to itself (bottom graph). Each robot determines the direction of the TDS along the grid (right or left, up or down), and it is the sum of these determinations for the entire swarm that resolves the pulse's direction. Each RNN architecture (see figures 6a, 6b, and 6c) is trained the same way and their performances are compared.

The training set for the RNNs consists of the set of 2D gaussian pulses propagating at an angle θ lying between 0 to 45^0, 135^0 to 180^0, 180^0 to 225^0, and 315^0 to 360^0 (regions 1, 4, 5, and 8 in the top left plot of figure 7), with an initial radial distance between 3 and 8 units from the center of the swarm. For these tests, the pulse's width and amplitude remain constant. To optimize convergence, each generation has as its training set four sequences of states, where each sequence is generated from the four regions listed above. The activation functions IAF(x) and AF(x) are defined by equations 3 and 4.

Comparing the validation results provides a measure of the RNN performance for those regions excluding the training set. The validation results are plotted in figure 6 for all three RNN configurations, where the validation set samples the entire phase space ($0 < \theta < 360^0$) in 5^0 increments. The direction of the pulse is determined by taking the average directions,

$$\text{Average Output} = \left(\frac{1}{N}\right) \cdot \sum_{i}^{N} O_i \ , \qquad (8)$$

where O_i is +1 or -1, i is the state related to the time increment, $t = i \cdot \tau$, and N is the total number of sensor readings taken by the swarm over the duration of the pulse. The graphs on the right plot the average output in the x-direction as a function of θ, and the plots on the left represent the y-direction. There is very little difference between validation plots for the three RNN architectures. Because the training of the network is successful, the results in regions 1, 4, 5, and 8 are expected to be good. Outside these regions, the network generalizes by producing a smooth transition from a left-propagating, average output of -1, to a right-propagating, average output of +1, pulse, and back again. In addition, the shape of the curves in the y-direction (see figure 7) are the same and include the anticipated 90^0 phase shift. Therefore, a limited training set is capable of generalizing to the full phase-space for the time sequence of events.

Conclusions. Using only nearest-neighbor information, a swarm of robots with recurrent neural networks genetically-trained to recognize sequential events can determine the direction of a pulse in two dimensions. The lack of feedback loops in the networks architecture (i.e., FFNN) cannot decipher two-dimensional motion. Thus, local NNs has the potential to transform raw data into more useful form such as direction. Such a transformation can also enhance global pattern recognition (e.g., gradient maps). Multiple transformation in combination with sensor fusion techniques can provide the swarm with a level on consciousness not yet explored.

How well does this system handle noise? Because nearest neighbor interactions are close in space and time, the noise response in both the sensors will be approximately the same; therefore, the differential inputs into the neural network are minimized (excluding noise due to communication errors). However, if this is not true, noise could affect the RNN results. To simulate these results, the input values are no longer discrete but follow a sigmoidal function, like that of equation 4. Letting the parameter a become large (a=10), the activation function approaches a step function, thus the inputs are nearly discrete. However, as the slope, AF(x=0), decreases, the network has difficulty distinguishing between the motion of the two modes described by equation 7, within the region of the peak from that within the tails. A consequence is that a small dip in the training and validation results occurs at 0^0, 90^0, 180^0, and 270^0 degrees. As the slope decreases, these peaks become pronounced. Reverting back to equation 2, the first term requires refinement. The addition of a second threshold, which is set to a value indicating the strong presence of a signal (i.e., a sensor value greater than $A_0/2$), would provide the NN the knowledge to resolve this discrepancy and improve the swarms tolerance to noise. In addition, increasing the number of nearest-neighbor communications also provides for an improved signal to noise ratio (SNR). Such ideas can be linked to physical models such as the Ising model providing a measure to ambient noise levels represented by the thermodynamic quantity T, temperature.

The next step is to add motion to the robots. While the robots are in motion, an additional input into the RNN is their previous motion (left or right and/or up or down). Alot of flocking behavior recognizes their nearest neighbors, the motion of the flock (macroscopic inertia), and their own previous motion. Ideally, these values can be inserted into a RNN (with short-term memory capabilities) to recall past motion of themselves, the flock, and a time-dependent disturbance.

2.3.4 Global Intelligence

Global or collective intelligence provides two key initiatives : mission oversight (e.g., determining the swarm's state) and remote-sensing (e.g., image processing, sensor fusion). Referring back to the hypothetical incident of an overturned truck, a global neural network interconnects all the robots (as if they were neurodes) and determines whether the swarm has moved ahead or fallen behind the plume and how fast the plume is dissipating. Depending on the mission, the swarm may wish to modify its entire state, speeding ahead and reconfiguring to optimize the swarm's ability to remote-sense chemical concentrations or to break apart to remote-sense and find the position of the truck.

Two levels of pattern recognition have been performed. The first implements a delay-line feedforward neural network to determine plume characteristics. The second classifies bitmap images.

2.3.4.1 Characterizing a Plume Implementing a Feedforward Neural Network

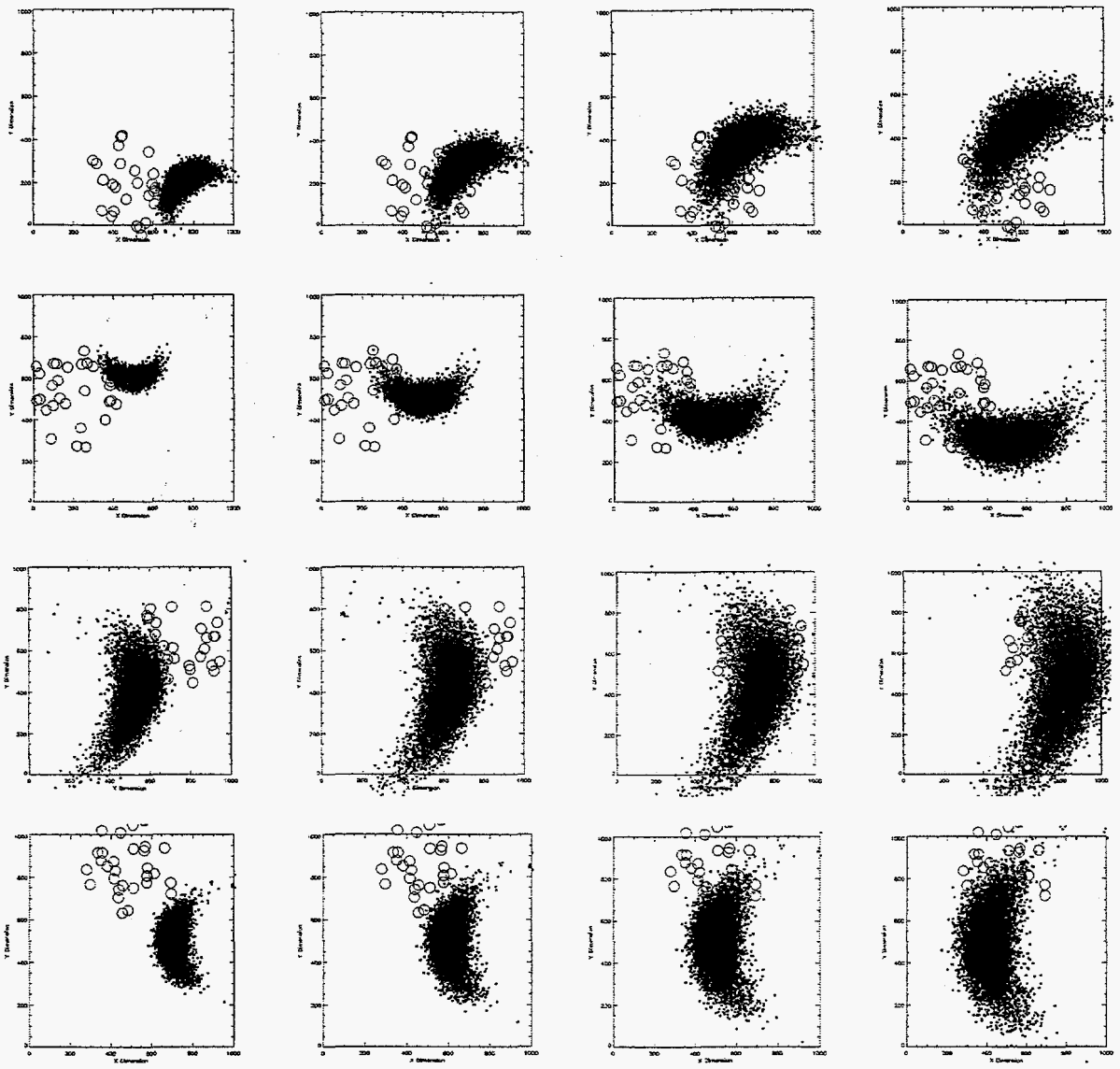


Figure 8. Simulated training sets for a wind-blown gaseous plume propagating at 120^0 (first row), 270^0 (second row), 10^0 (third row), and 180^0 (fourth row).

A swarm of robots individually registers and stores sensor information over time. By communicating their knowledge, a delay-line feedforward neural network attempts to characterize a wind-blown plume. This analysis implements a stationary swarm, in other words an intelligent remote-sensing array. Results are incomplete mainly due to insufficient training set and limited CPU power. However, these tests develop the form of the fitness function, resolving convergence issues that influence this, previous and subsequent genetically-trained models and neural network designs.

This implementation begins by describing the plume simulations followed by a description of the neural network architecture. The next section outlines the genetically trained process, concentrating on the form of the fitness function, and the validation process. Finally, results are tabu-

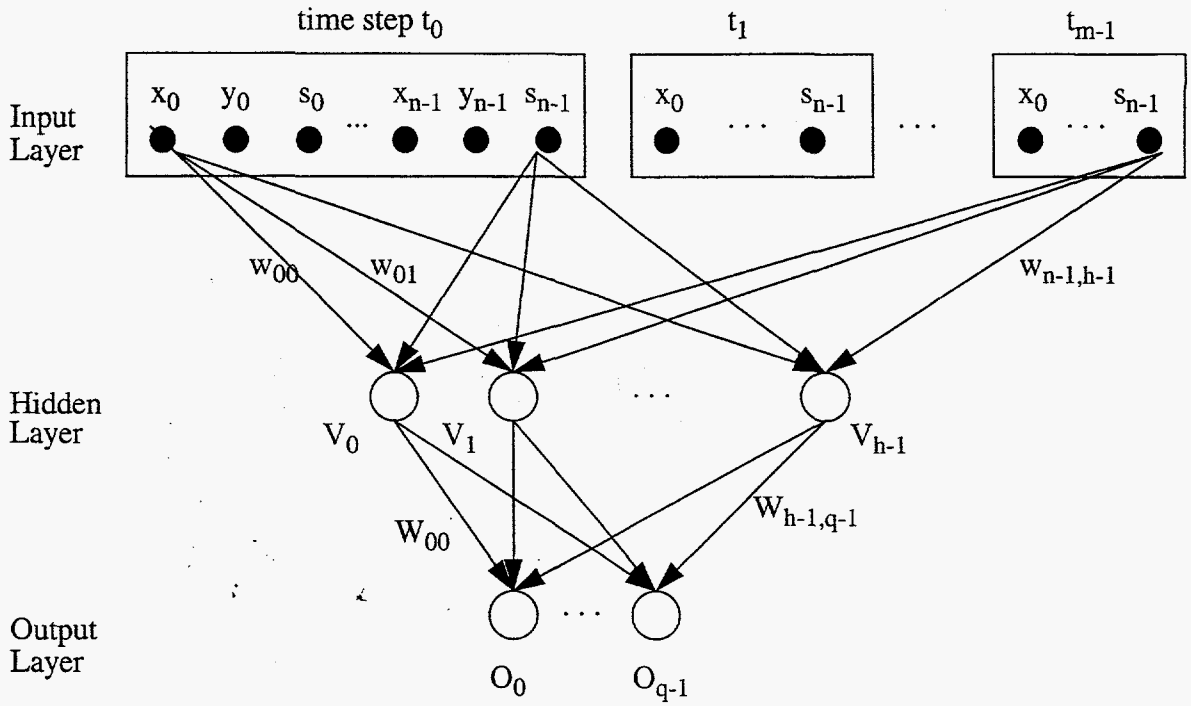


Figure 9. A schematic of a delay-line feedforward neural network with a single hidden layer.

lated and conclusions drawn.

Simulation of a Gaseous Plume. The simulated plume consists of 10^3 particles of equal mass. Its momentum distribution is gaussian. The wind blows the centroid of the plume with momentum p_{plume} in the direction θ. Each robot senses those elements comprising the gaseous plume and is capable of measuring its own position. A sample event shown in the top row of figure 8 depicts the kinematics of the plume as it propagates, spreads, and passes over a swarm. The origin of this plume is at (x_p, y_p) of (900,10), the magnitude of its momentum is p_{plume} which is set to 50 units, its direction of motion is defined by the angle θ which is set to 120^0, and it spreads and dissipates according to σ_p which is set to 0.1 p_{plume} Perpendicular to the wind direction the plume moves with a momentum $p_\perp$ which is set to zero and spreads with standard deviation of $\sigma_{p\perp}$ which is set to 4.0 p_{plume}. The kinematics remain independent of time. The center of the swarm is (x_c,y_c) which is set to (500,250). The robots' distribution includes the swarm's center, (x_c,y_c), which is set to (500,250), and its radius of extent, r_{ex}, which is set to 250. A robot's sensor range, r_s, is set to 10 (shown as circles in figure 8). Training and validation processes vary (x_c,y_c), θ, and (x_p, y_p).

The simulation procedure follow the following steps : determine the momentum distribution of the particles, calculate its path according to its kinematics, iterate with a time step equivalent to a 50 unit displacement of the plumes centroid, measure the number of particles falling within a robot's sensor range, and finally input these sensor values, the robots' positions, and information

from the two previous time steps into the neural network, which is used during the training process.

Once trained, the optimal GA-designed NN must be validated. The validation routine reads in simulated plume data, feeds this information to the NN, and calculates the characteristics of the plume -- x_p, y_p, θ, p, $\sigma_{p\perp}$. Further analysis determines whether the network has learned to recognize the above characteristics.

The structure of the NN. This analysis implements a FFNN with a single hidden layer. Other test runs measure the performance of a delay-line FFNN with 2 hidden layers. Depicted in figure 9 is the architecture of an m-step delay-line FFNN used in these tests. The net's inputs consist of each robot's position, (x_i, y_i), and sensor value, s_i. In addition, a time-dependent history of these values are also inputted. Therefore, if there are n robots and m time steps, the number of inputs is $3 \cdot m \cdot n$. The number of weights also depends on the number of hidden and output nodes, h and q; therefore, the total number of weights is $(3 \cdot m \cdot n + q) \cdot h$. These weights represent the genes or variables of the GA. Most of the tests implement a 32-robot sensor array (n), 3 time-delayed inputs (m), 5 to 17 hidden nodes (h), and somewhere between 1 and 5 outputs (q).

Training Process. Training data in a two dimensional plane consists of four nearly orthogonal wind-blown directions -- east (10^o), west (180^o), north (120^o), and south (270^o) -- the parameters of which are listed in table 1 and shown in figure 7. In addition, the swarm's central positions relative to the plumes and its direction also vary in distance (radial distance) and angle (φ). Also, a certain number of sensors must register non-zero measurements (5), labeled as "Time Steps Implemented".

Table 1: The training data set for a wind-blown plume.

Training Set	(x_p,y_p)	θ (degrees)	φ (degrees)	Radial Distance	Time Steps Implemented
1	900,10	120	45	442	5-9
2	500,800	270	53	390	4-7
3	50,300	10	13	762	10-15
4	1000,0	180	32	610	9-11

Development of the fitness function serves two important purposes. One, it provides a measure of the net's performance during training. Two, it should provide consistent results, i.e., steady-state. The rest of this section outlines the process deriving a fitness function these qualities.

The fitness function depends on the errors in the network's output values. The form of the reduced-chisquare is used :

$$fitness = \sum_{j=0}^{p-1} \frac{1}{(0.1 + \chi_j^2)} , \qquad (9)$$

where p is the number of patterns. A pattern is defined as the set of NN inputs over three consecutive time steps, called test cases. These test cases are used to train the NN. The reduced-chisquare is defined as

$$\chi^2 = \sum_{k=0}^{q-1} \left(\frac{(ExpectedOutputs_k - O_k)}{\sigma_k} \right)^2 , \qquad (10)$$

where ExpectedOutputs is the assigned outputs for the given input pattern. The procedure to genetically train this NN consists of the following steps : (1) randomly choose a training set listed in table 1, (2) randomly choose three consecutive time steps within this training set, (3) calculate its fitness, and (4) execute the GA operators of selection, crossover, and mutation. Successful training occurs when the GA converges toward a maximum fitness of 10.

Upon closer inspection, the outputs contain five spurious (discrete, unique) states even while the population's fitness remains at 10. These states are independent of the network's structure but dependent on the GA's training methodology. As an example, the groups of time steps {3,4,5}, {4,5,6}, and {5,6,7} produce consistent results, but the addition of {6,7,8} causes the GA to fluctuate between three distinct but closely spaced states. After modifying the fitness function to

$$fitness = \frac{1}{\left(0.1 + \sum_{j=0}^{p-1} \chi_j^{\,2} \right)} , \qquad (11)$$

the GA converges to a single state.

A second modification improves the GA's convergence rate. Training set 3 converges relatively slowly compared to the others. Tests show that by tuning the sigmas improved the concurrent convergence of all four parameters. The ratio of the sigmas used is

$$\sigma_1 = \sigma_2 = \sigma_4 = \frac{\sigma_3}{10} , \qquad (12)$$

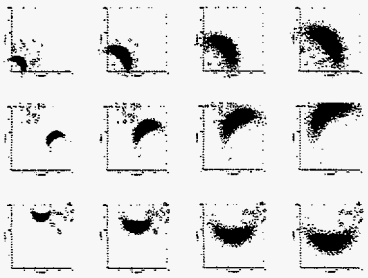
where the magnitude of each sigma is set to 0.1. This factor appears to better define or enhance the fitness's phase space near its solution, O_3. However, attempts to anneal each sigma with its own schedule proved inconsequential.

It most runs, the fitness initially converges toward a local minima defined by the residuals calculated for training sets 1, 2, and 4. Increasing the mutation rate and the addition of equation 4 improves global convergence of the GA for all four training sets. Consistent training performance occurs after setting all sigmas to 0.1 or 0.05 and inserting an additional penalty term into the fitness function,,

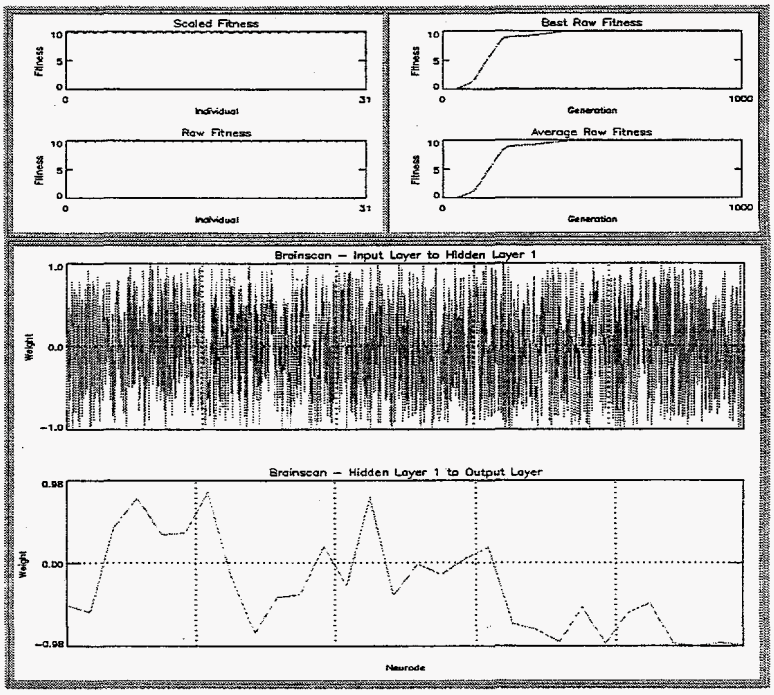
$$fitness_{new} = fitness \times \left[1 - \frac{G}{10} \right] , \qquad (13)$$

where

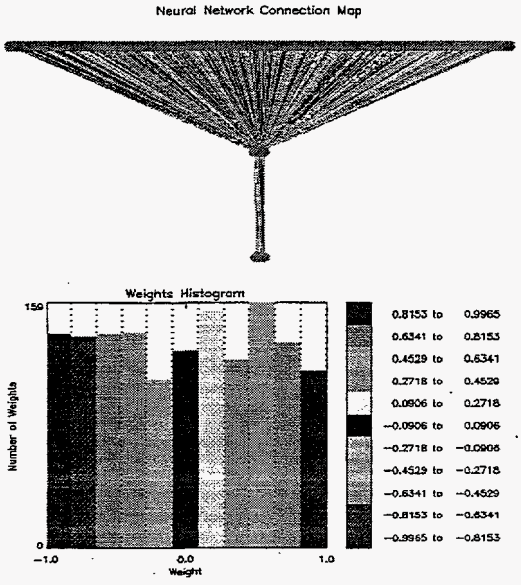
$$G = \frac{1}{\left(0.1 + \sum_{k}^{(p-1)\cdot(q-1)} (\sigma_k - \bar{\sigma})^2 \right)} . \qquad (14)$$



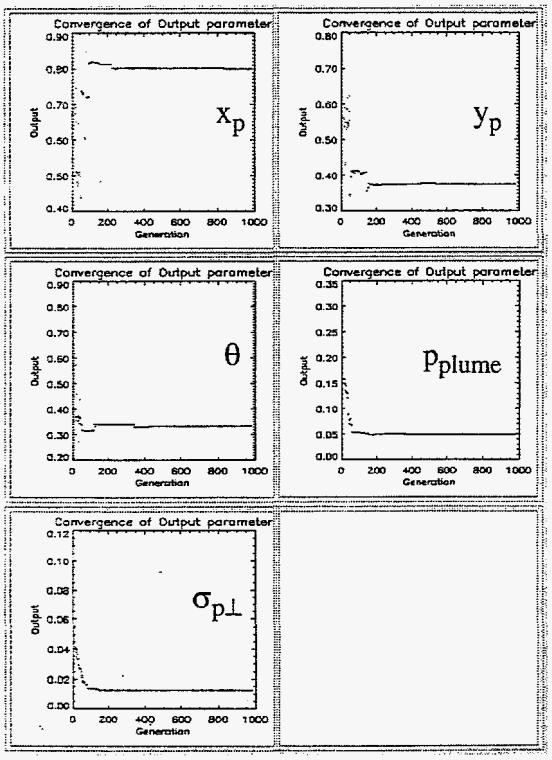
Training Set



32 neural networks compete over 1000 generation



Simple Collective 3-layer net; each robot shares a piece of the input layer and provides its location and signal strength.



Performance evaluation based on multiple fitness criteria.

Figure 10. A flow chart showing how a population of delay-line FFNN are genetically-trained to recognize a migrating plume.

In equation 6, σ_k is the sigma of the residual distribution when plotting differences between NN

outputs and ExpectedOutputs and σ-bar is the average of all these deviations. One may consider such a term as a renormalization of the errors or as a pseudo-annealing schedule. With the addition of this penalty term, modest convergence-rate improvements indicate that this form of the fitness function may help the GA better avoid local minima. All subsequent analysis implements equations 3 through 6.

Validation data. Table 2 lists additional simulations with a variety of plume dynamics and swarm distributions. The data from these simulations are used to test the optimal genetically-trained neural network design.

Table 2: The Plume Validation Data Set

Validate Set	(x_p,y_p)	theta (degrees)	phi (degrees)	Radial Distance	Time Steps Implemented
1	(10,10)	45	0	339	2-11
2	(10,10)	45	0	339	2-11
3	(800,400)	120	0	707	9-17
4	(500,1000)	270	71	403	5-9
5	(10,900)	345	40	914	12-17
6	(800,800)	225	0	424	4-10

Results. Displayed in figure 10 is a flow diagram of an example training process. The training set of robot positions and sensor values and the expected plume's characteristics are realized onto a delay-line FFNN. The genetic algorithm operators evolve the weights of the networks over 1000 generations. Convergence and stability of the outputs are verified. The optimal design, the weight distribution, characterizes simulations with plume dynamics. Tests include measuring NN performance for angles the network is trained to recognize, for angles not in the training set, for different neural network architectures, for different numbers of time-sequenced training sets, and for noisy training sets.

Validation results for the optimal net are listed in table 3. The first four rows (Training Data

Table 3: Neural-Network Performance (input data consists of three time steps, i.e., m=3)

	Training Data Set				Validation Data Set					
	1	2	3	4	1	2	3	4	5	6
mean	120	249	9.5	176	106	84	203	36	12	74
std. dev.	17	2	1.7	5	46	12	6	1	0	31

Set) indicate the network learned to recognize those trained angles. However, angles and robot

distributions not presented in the training set are incorrect.

The net's performance is also tested relative to the number of hidden neurodes and layers and the form of its activation function. Increasing the number of hidden layer neurodes or the number of hidden layers did not improve performance. An N-5-1 architecture recognizes a 225^0 angle as 74^0. An N-17-1 architecture recognizes 225^0 angle as 103^0. An N-47-1 recognizes a 2250 as 122^0. And, an N-17-5-1 recognizes a 225^0 angle as 102^0. Changing the activation function from a sigmoidal function to a linear function improved the averaged outputs to 225^0 but with large deviations, a standard deviation near 100^0, indicating inconsistency over time. A linear squashing function gives similar results as the sigmoidal function, output angle of 124^0.

The most revealing test compares the delay-line FFNN performance as a function of the number of input time steps, trained with a sequence of 1, 2, and 3 time steps (m) of sensor data. Both training and validation results are very similar to those in table 3, indicating that the net did not take the time sequenced data into account

Past experience indicates that noise minimizes memorization and improves generalization. Two types of noise are added to the sensor readings -- random amplitude and placement and smearing s_i according to a gaussian distribution. Because the results for both techniques are similar, angle recognition results for type 1 noise has been displayed, see table 4.

Table 4: The effects of noise during training.

	0%	2.5%	5%	7.5%	10%
train1	0.505	0.506	0.485	0.5	0.48
traun2	0.723	0.709	0.747	0.73	0.72
train3	0.348	0.342	0.334	0.34	0.33
train4	0.035	0.031	0.033	0.033	0.035
valid225	75	75	60	68	104
sigma	29	22	24	31	31

As before, the net successfully recognizes trained angles, but generalization to other angles remains difficult due to the small training set.

Conclusions. This analysis is fairly complicated. The cause of the validation failures is due to an insufficient training set. Four angles and swarm positions covers a very small sample of the full phase-space for plume motions and swarm distributions. Because of this situation, all other results are somewhat limited in interpretation except modifying the time steps. As m is changed from 1 to 3, the results remain very similar. This result alone indicates that m may be too small. If this is the case, an inordinate amount of memory and communications may be necessary to do pattern recognition of time-series data at a global level. Either some preprocessing of the data should be performed or pattern recognition should be static.

The successes of this analysis is the development of a reliable form of the fitness function and the technique used to train the net. Both of which has been used successfully in other applica-

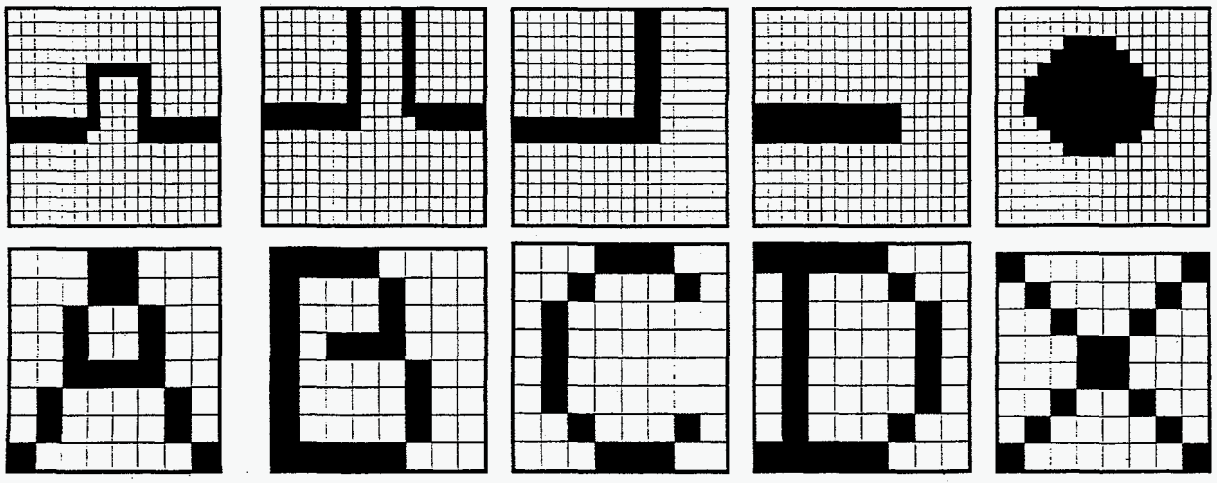


Figure 11. Top row, typical patterns needed to be recognized when a swarm avoids obstacles. Bottom row, images implemented in tests.

tions. Training a net to learn multiple outputs has also shown to be possible.

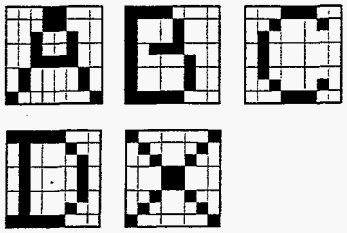
Providing a more complete training set should provide good results. This task is feasible if additional compute power or smarter algorithms can be acquired. Both of which has been demonstrated in other areas (see section 2.4).

2.3.4.2 Character Recognition Using Genetically Trained Neural Networks

Introduction. The neural network model constructed for pattern recognition uses a three layer feedforward architecture. To facilitate the input of patterns or characters, a graphic user interface (GUI) has been developed to convert the traditional representation of each character or pattern to a bitmap. The 8 x 8 bitmap representations used for these tests are mapped onto the input nodes of the FFNN in a one-to-one correspondence. The input nodes feed forward into a hidden layer, and the hidden layer feeds into five output nodes correlated to possible character outcomes. During the training period the GA optimizes the weights of the NN until it can successfully recognize distinct characters. Systematic deviations from the base design test the network's range of applicability. Increasing capacity, the number of letters to be recognized, requires a nonlinear increase in the number of hidden layer neurodes. Optimal character recognition performance necessitates a minimum threshold for the number of cases when genetically training the network. And, the amount of noise significantly degrades character recognition efficiency, some of which can be overcome by adding noise during training and optimizing the form of the network's activation function.

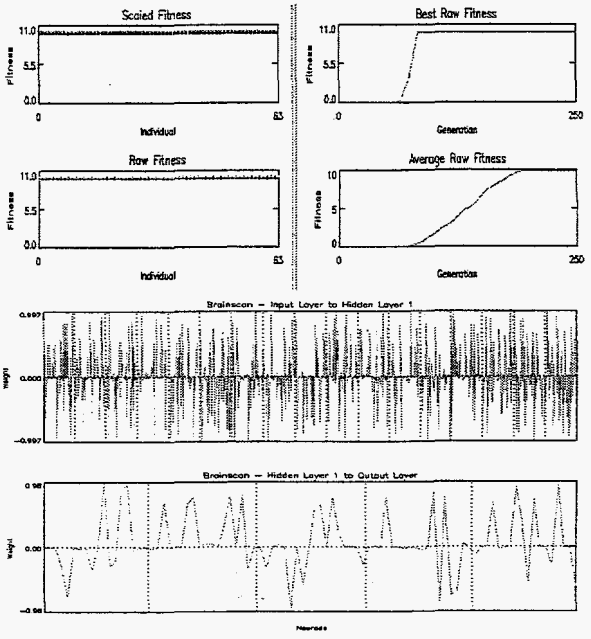
This section focuses on the development of a feedforward neural network which addresses the initial stage of its pattern recognition capabilities. The following topics will be discussed within this section : a description of the NN training set, the NN architecture, the training process, the validation process, systematic performance analysis, and conclusions. A more complete analysis and discussion can be found in the following reference [16].

Training Set. A character bitmap is an 8 x 8 array of bits, where a +1.0 indicates an "on" bit and a -1 an "off" bit. The database contains five capital Roman letters -- A, B, C, D, and X -- displayed in bottom row of figure 11. By extracting the two-dimensional string of bits one row at a time, a

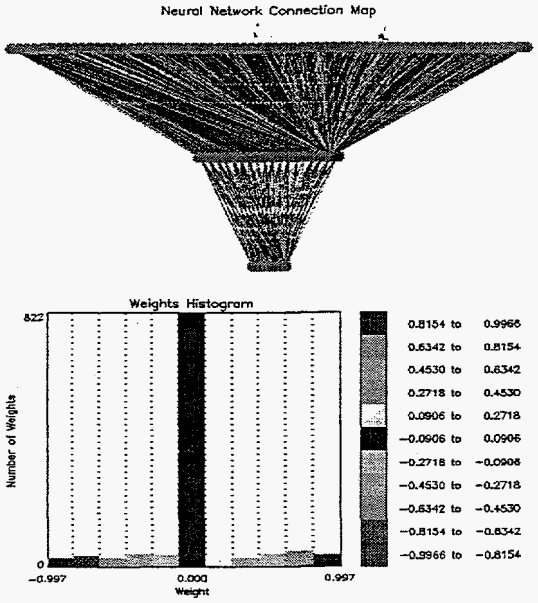


Training set

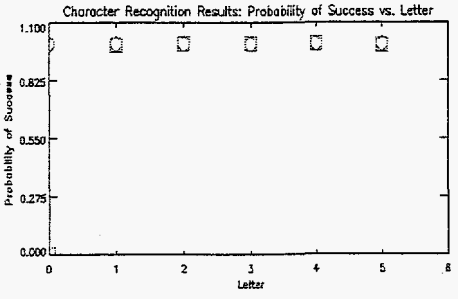
25 randomly sequenced bitmap images are seen by the population of nets per generation



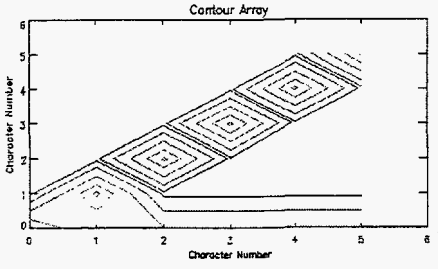
A population of 64 FFNN is being trained implementing a GA. Top right, fitness versus generation. Convergence occurs when average fitness of population approaches 10.



Optimal neural network design.



Validation process analyses optimal neural network design.



Validation process measures perfect character recognition efficiencies.

Figure 12. Character recognition flow chart (training and validation results of base design).

64 bit vector, representing the input pattern, defines the NN inputs. The total number of outputs is 5, one for each pattern or character. Assigned to each input pattern is the expected outputs vector which contains a +1 for the that character inputted into the network and a -1 for the other outputs. As an example, if the input vector is the image data for the letter B, the expected output vector is set to [-1, +1, -1, -1, -1], where each output represents a letter.

Neural Network Architecture. The base design [16] is a single hidden layered feedforward neural network with 64 inputs nodes, 17 hidden nodes, and 5 output nodes. The form of the activation function is the hyperbolic tangent sigmoidal squashing function of equation 4 with parameter a equal to 5 and parameters b and c set to zero.

Training Process. The GA implements (1) a roulette-wheel selection method, (2) a one-point crossover technique keeping 8 out of the 64 (population size) networks as elitists and allowing 90% of the population to crossover, and (3) a flat or random mutation rate effecting 10% of the population's genes (i.e., weights). After each generation, the results from 25 randomly sequenced characters (i.e., the number of cases, nCases) determine a network's fitness. The GA searches for and converges toward the global maximum of the fitness function,

$$fitness = \frac{1}{0.1 + \left(\sum_{nCases} \sum_{nOutputs} |Outputs - ExpectedOutputs| \right)^4} \qquad (15)$$

This form of the fitness function is similar to that of equation 11, proportional to the inverse of the output residuals summed over the case numbers within a generation.

Each net within the population of 64 is shown a random sequence of 25 character images (the number of cases, nCases) and their corresponding outputs. The NN outputs are compared to those that are expected, ExpectedOutputs (defined as supervised learning). The network under the guidance of the GA "learns" how to recognize these patterns and categorize them as letters. When the entire population of 64 nets converges toward a fitness of 10.0, an optimal solution has been found. This optimal NN design can then be tested or validated. A flow diagram of this training as well as the validation processes and their results implementing the base design are shown in figure 12.

Validation Process. The validation process randomly chooses a sequence of characters, represents these characters as one-dimensional bit-vectors, measures how well the optimal neural network categorizes these characters, and plots the results. First, 10^5 randomly sequenced characters are analyzed. Each character plus a blank is sampled approximately 15,000 times, producing a statistical error less than 1%. (The blank pattern is defined as an input vector of -1s and an output sequence of all -1s.) Next, the optimal NN successfully categorizes a character when the output activation representing the input letter exceeds a +0.9 threshold while all other outputs remain below +0.9. Using this definition, character recognition efficiency is defined as the fraction of times an input character exceeds the threshold while keeping all other characters outputs below this threshold. Both these efficiencies and the raw output activations ($-1.0 < Outputs < +1.0$) of all the character outputs for each character input are tabulated, averaged, and then plotted. Correlations (or NN confusions) between pairs of letters can be seen by plotting the average activation values of each character output as a function of the character input. This data can be represented

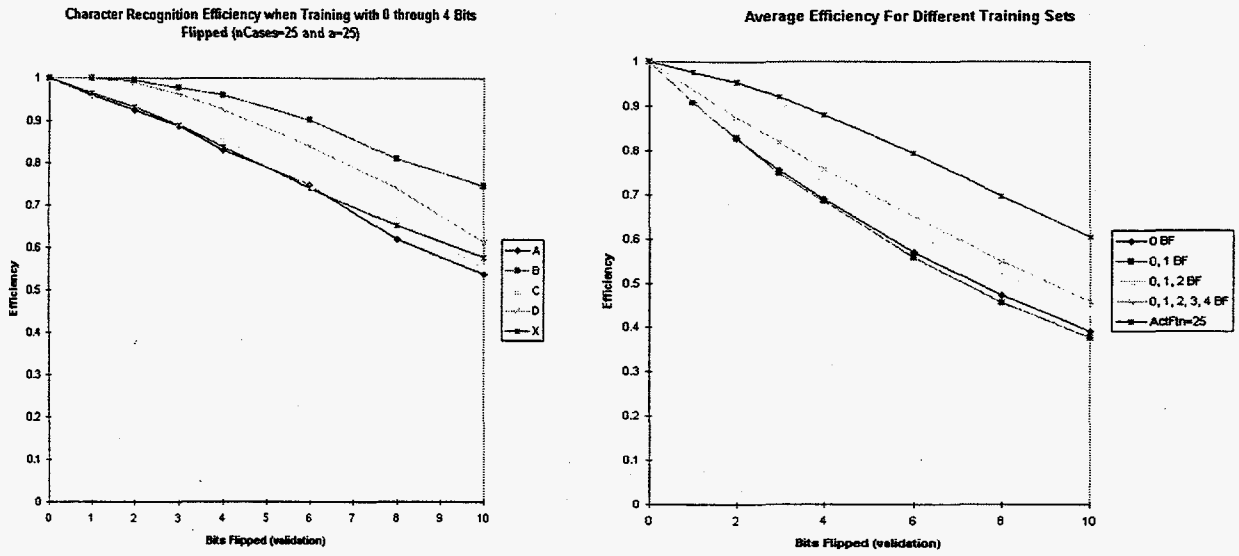


Figure 13. The left graph plots validation character recognition efficiencies for noisy images with the number of bits flipped (BF) ranging from 0 to 10, when the NN is trained with noisy images (0 through 4 bits flipped). Also plotted, and labeled as ActFtn, is when the activation function approaches a step function. The graph on the right is a plot of the average efficiency results for different training methods.

by a contour plot where only the y=x line is expected to be populated when validation is optimal. Both plot types are shown in the bottom right of figure 11.

Systematics. To understand the range of applicability of the current or base design, various key parameters influencing the networks ability to recognize characters are systematically modified. Three such variables include the number of hidden layer neurodes, the number of cases in the training set per generation, and noise level.

The Number of Hidden Layer Neurodes. Using the definition for NN performance in reference [16], results indicate an increasingly nonlinear response of the hidden layer neurodes relative to the network's capacity.

The Number of Cases. The average character recognition efficiency over three convergent runs for nCases equal to 1, 5, 7, 10, 17, and 25 reveals a threshold value somewhere near 25, where the networks training and performance is optimal. In other words, increasing the number of times each character is seen by the genetically-trained neural network during each generation appears to better map the fitness function's phase space, thus increasing the NN's pattern recognition capabilities.

Noise. Noise is defined as randomly flipped bits within a character's bitmap image. For a multi-agent system, a flipped bit translates into a false measurement or decision from a single robot. The question becomes how does noise effect the pattern recognition capabilities of the collective. Two sources of noise are implemented : (1) flipping bits during the training process and (2) flipping bits during the validation process.

Tests include training the neural network with character images that have (i) no bits flipped (BF), (ii) half the images with 0 BF and the other half with 1 BF, (iii) a third of the images with 0 BF, a third with 1 BF, and a third with 2 BF, and (iv) a fifth of the images with 0 BF, a fifth with 1 BF, and so on up to 4 BF. The second source of noise is during the validation process. Validation results measure eight different character recognition efficiencies for those images with BF set only to 0, 1, 2, 3, 4, 6, 8, and 10, respectively. Results from this analysis are displayed in figures 13.

Noise within the validation set, but not within the training set, can significantly effect recognition efficiency, by nearly 30% when 4 bits are flipped. Training with images with up to 4 bits flipped improves efficiencies by about 6 to 7 percent. Efficiencies for individual characters can improve up to 17 percent.

The Form of the Activation Function. Finally, a net trained using 250 cases per generation, an activation function with the parameter set to 25 which approaches the form of a step function, and training images containing 0 through 4 randomly flipped bits. Results are shown in figure 13. Looking at the left graph, the character recognition efficiency is plotted as a function of noise. The symbols B and D are much improved and begin to drop off precipitously beyond 4 BF, the BF training set limit. Other letters are still much improved but fall off more gradually with noise. Looking at the right graph, the average character recognition efficiency is plotted over all five characters as a function of BF. The same curves produced from the other four training techniques are also plotted for comparison. Thus, tuning the activation function and increasing the number of cases show greatly improves the net's ability to overcome a noisy environment or inefficiencies.

Conclusions. Genetically training a 3 layer FFNN to recognize capital Roman letters of limited resolution has been shown to be successful. Basic rule-of-thumb calculations have been used to determine the architecture of the network and the parameters of the GA. Tests indicate increasing the capacity of the network in order to recall all 26 letters of the full alphabet would appear difficult unless additional character recognition techniques are incorporated. The number of cases plays a role in convergence rates but more importantly performance (i.e., character recognition efficiency). Attempts to quantify the minimum number of cases as a function of the neural networks architecture is instructive but somewhat limited. The major problem appears to be the ability of the network to overcome noise. Flipping bits within the characters bitmap image degrades recognition efficiency. Additional training with noisy characters regains some efficiency, but only a modest percentage 4 bits flipped has a 30% loss while training with noise only gains 6 to 7% back. However, the greatest improvement comes from an activation function that approaches the form of a step function. On average, almost 70% percent of the base design losses are regained with a modified activation function. One can conclude that speed (a parallel version of the code) and improved techniques (architecture, training, character resolution) need to be explored.

As a final note, realization onto a multi-agent system appears promising. Because a large percentage of weights are close to zero (approximately 80%), communication requirements can be relaxed or optimized. In other words, communication network theories can be added to the fitness function, thus both interconnections between nodes (communication topology) and the NN's pattern recognition capabilities can be concurrently optimized into a single, flexible design.

2.4 Global Optimization

The ability to produce desirable macroscopic, collective swarm behavior from local, nearest-neighbor interactions is a huge multivariate analysis problem. Genetic algorithms[14,15] are stochastic and based on non-derivative techniques, thus they are less likely to get caught in a local minimum. They also span a larger variable phase-space without requiring a smooth, continuous objective, thus more likely to find a global optimal solution for a larger set of objectives.

A genetic algorithm performs three operations onto the elements of a population, e.g., a population of neural networks or swarms. Three operations form the basis of a GA. The first operation is selection. Here, the fitness of each element in the population is calculated according to its fitness function, which represents how well the neural network recognizes the inputs by determining its output residuals. The smaller the output residuals, the higher the fitness for that element in the population. Therefore, selection preferentially chooses high fitness elements. The second operation is crossover (e.g., mating). Here, selected elements share sequences of genetic material. In single-point crossover, a common point is chosen along the genetic code (e.g., weights of neural network) of two selected elements, called the parents. Sequences of genes are exchanged about this point. The result is a population of offspring which contains pieces of its parent's solution space. The third operation is mutation. To prevent the population from evolving toward and remaining within a local minima, mutation continually adds new random values (i.e., diversity) into the population. As a result, a larger parameter (weight) phase-space is continually spanned for a global optimum. Like the back-propagation method, a GA is iterative, where each iteration is called a generation. By showing the neural network many input-output pairs over many generations, the population converges to an optimal solution and is considered to have been successfully trained.

Implementation of a genetic algorithm (GA) on top of a collective behavior model to optimize the swarm's performance requires compute power. Our new parallel master-slave GA, optimizing our LG model, is one step toward reducing simulation time. In order to make parallel the GA, the CPU intensive lines must be determined, extracted, and distributed among many processors. This piece of the code is the calculation of the fitness function which requires a full simulation of the physics-based models. Implementing a Master/Slave architecture allows the Master to distribute the population of swarms, LG simulations, among N processors or Slaves (N is equal to 20 in most applications). After a slave finishes a simulation implementing the genetics representative of that particular swarm, it returns a fitness value to the Master. The Master performs the selection, crossover, and mutation operators and redistributes the population of swarms to the Slaves. The language used is called MPI, message passing interface. Thus, the Master sends the genetics and a execute command to each Slave and continuously polls the Slaves to determine whether they have calculated a fitness value. Both information and transfer rates are minimal. GA execution on the Master is much faster than LG simulations. Speedup is essentially linear.

Two initial applications implementing these codes are obstacle avoidance (section 3.3.1) and ground-penetrating radar (section 4.2.2). First, simulations implementing nearest-neighbor interactions reveal that rather mundane conditions can cause a swarm to become unstable (ie., break apart) when an obstacle is introduced. By allowing a robot's potential strength parameters to depend on time and/or space, a genetic algorithm evolves a population of competing swarms determining which set of spatial regions (or time steps) should change its potential fields. The result is enhanced performance. Two, a swarm of robots fitted with antennas has been simulated implementing a GA to control the relative position of the robots and the phase of the emitted high-frequency radar pulses, optimizing the signal's intensity on an underground facility probing a two dielectric sub-terrain profile. Both the signal's phase and robots' positions are varied, deriving

and simulating sensitivities to wave coherence and signal amplitude. Also, various physical parameters, such as the robot-robot attractive and repulsive forces, the target's relative potential strength, friction and drag, and others, are modified to determine swarm behavioral changes for the effluent plume (LG) and ballistic missile (PIC) scenarios. From which the swarm's response -- -speed of pursuit, cohesive properties (i.e., when to break into two swarms following two targets), and obstacle avoidance -- are controlled.

2.5 Autonomy

Autonomy is the ability of the swarm to achieve a goal without the necessity of a leader, an "alpha-" robot, if you will. Tasks can be as simple as following a target or as complicated as imaging an underground structure. Some jobs require only the basic set of instinctive forces in order to respond autonomously, such as following a wind-blown gaseous plume; others require an increase in the level of intelligence necessary, such as obstacle avoidance, multiple target identification, and triangulation; while a group requires the highest levels of intelligence and coordination, such as remediation, GPR, and BMD (Ballistic Missile Defense). At this time, a conceptual behavior for three different behaviors are discussed within the above mentioned framework. Later, simulation results refine these concepts providing additional insight into the most challenging tasks.

2.5.1 Following a Target

A wind-blown plume of toxic gases propagates toward a community due to an overturned truck. Sensors on the robots measure the gradient of the emerging threat. Through nearest-neighbor interactions, a pseudo-potential is applied causing the swarm's state to follow the plume or find the truck (the origin). By determining the optimal set of parameters for the potentials, $[\alpha, \beta, \gamma, \delta]_{target}$, the swarm follows and condenses onto the target, or a fraction of the robots follow while other robots move toward the truck. The direction of the plume is determined when each robot shares nearest-neighbor sensor data and implements its RNN trained to recognize time-dependent sensor information. An additional "inertia" potential is introduced, modifying the swarm's state. A global neural network interconnects all the robots and determines whether the swarm has moved ahead or fallen behind the plume. Depending on the mission, the swarm may wish to modify its state so it can speed ahead of the plume, reconfigure, and determine the origin of the plume, or break into two swarms and do both. Depending on the relative parameters α, β, and γ, the number of robots following the swarm (swarm cohesiveness) can be controlled. Maximum cohesiveness is accomplished by keeping α relatively large. β has an effect on the radial variance (i.e., density) of the swarm, decreasing depending on the pulse shape, size, and dissipation rate. Knowing the above parameters determines the state equations through statistical mechanical methods for lattice gases. Understanding how the density and temperature changes as a function of communication and sensor ranges and the number of nearest-neighbor interactions can feedback into the design.

2.5.2 Obstacle Avoidance

Two simple realizations of autonomous adaptive behavior include obstacle avoidance and

multiple source identification. When avoiding an obstacle, the swarm first recognizes that a stationary (non-source) object falls along its path. It modifies its form (e.g., state) in order to move around the obstacle while remaining cohesive and preventing instabilities. Once overcoming the object, it returns to an optimal state. Referring to the flow diagram in figure 1, the initial state of the swarm, $[\alpha, \beta, \gamma, \delta]_0$, follows a plume, possibly optimized for speed along the plume's direction of motion. It encounters an obstacle. Each robot's local NN recognizes that the object is in its path. The global NN combines information from all robots and determines the extent of the image (object)., forming a new state -- $[\alpha, \beta, \gamma, \delta]_{obstacle}$. This new state reduces the swarm's density and elongates in the direction perpendicular to the obstacle by modifying its pseudo-potential field. Therefore, the swarm is capable of determining a path around the object and providing additional space to maneuver around this obstacle. Such a transition -- $[\alpha, \beta, \gamma, \delta]_0 \rightarrow [\alpha, \beta, \gamma, \delta]_{obstacle} \rightarrow [\alpha, \beta, \gamma, \delta]_0'$ -- must be adiabatic, meaning the swarm's state must remain in a steady-state, thereby preventing instabilities. To achieve this transition requires additional "adiabatic" terms in the GA's fitness function. However, the return or final state need not be the initial state due to the perturbation (in both space and time) of the swarm relative to the source.

2.5.3 Multiple Sources

A second example of autonomous adaptive behavior is a swarm reacting to multiple sources (or targets). The swarm's initial state reacts to a time-dependent event, seismic activity. Each robot communicates asynchronously with its nearest-neighbor, processes their information using a neural network, modifies the swarms's state through δ, and begins to move in the direction where it thinks the origin of the activity occurred. While gathering additional information, a second source, a discharge of gases, propagates nearby. Processing the swarm's vector image, the global network recognizes two types of sources and computes their directions. The state of the swarm changes -- α/γ decreases as does the range of the forces -- allowing the swarm to expand and split into two. As the population of the swarm decreases, the state must also change in order to maintain the cohesiveness of each swarm separately.

2.5.4 GPR

Phased-array, short-pulsed, ground-penetrating radar (GPR) and plume remediation require sophisticated imagery techniques beyond the design criteria of the robots. Probing for subterrain structures can be thought of as an inverse scattering problem. The goal is to develop a technique that can identify structural geometries, such a triangular or conical shapes, in an unknown dielectric medium. Initially, surveillance or the autonomous activity of the swarm (sensing the pulsed seismic source and triangulating to find the epicenter) positions the swarm near a target. By controlling the position and phase of the antenna-flavored robots, an iterative process is employed to map out the sub-terrain and optimize imaging of the underground structure. First, each robot transmits a pulse and receives a signal. From the time-of-flight and strength of the measured signal, depth and attenuation (dielectric) is determined. Second, an image of the subterrain just beneath the swarm is developed from which the beginnings of a potential field is formed. Third, image processing (time-frequency spectra) by a trained neural network searches for possible tar-

gets. Fourth, a GA incorporates the potential field into its fitness function and determines an optimal configuration by changing its state variables. Fifth, after repositioning, the phases of the antenna array are optimized, probing deeper into the ground. Sixth, the optical potential is updated and inserted into the GA, further improving the position of the swarm and its robots. Finally, as the swarm is grouped into fewer and larger clusters of phased-array antennas, the swarm can image structures deep within the sub-terrain, scanning horizontally by modifying its phase or angle. Once a target image is suspected, the GA reconfigures the swarm providing the highest quality data for image extraction and target verification.

2.5.5 Remediation

Plume remediation "burn" biological agents within a wind-blown, gaseous plume. A three step processes is envisioned. One, biochemical sensors indicate the presence of anthrax (trip-wire configuration). Two, the swarm follows the plume and adaptively reconfigures to surround the plume. The global neural network uplinks the information where the user analyses the images and verifies the threat. An ignitor, such as magnesium or titanium dust, is inter-dispersed within the plume. Third, an external signal (e.g., optical communications) triggers a pulsed solid-state laser from each robot igniting the edges of the plume. After remediation, the swarm enters a search (or validation) mode, providing further remediation and verification.

3 Mission Strategy

In its simpliest form, a multi-robot system on a data-gathering mission can be broken-down into five steps : (1) initial configuration, (2) search methodology, (3) target convergence, (4) adaptive/optimal sensor position, and (5) communication. This section explores each topic.

3.1 Initial Configuration

Initial configuration of a swarm considers the swarm's makeup (e.g.., sensor heterogeneity), the distribution pattern (e.g., the number of swarms and its geometrical positioning), the intilligent algorithms necessary to complete a mission (e.g., neural network architectures), and other similar parameters deemed important to accomplish a mission. Because the success of a mission depends on how efficiently the swarm performs these mission steps, initial configuration is determined last.

Evolutionary programming (i.e.,genetic algorithms) can determine the number of swarms and their heterogeneity, how many robots of each sensor type are needed to efficiently carry out a mission. Currently, artificial "performance" or "mission" functions are being considered to simulate the steps within a mission. From this, a (parallel) GA determines the number of swarms, the number of robot types, and parameters describing the physical state of the swarm or swarms. An example mission implementing many of the above mentioned qualities is bio/chem plume remediation (see section 4.1). First, an initial trip-wire position requires fast response-times. Tracking and characterization of the plume occurs. Two different robot types reconfigure. And, adaptive behavior after the remediation process occurs. Many of these tasks are currently being simulated. Once the complete mission has been simulated, global optimization of the entire process implementing evolutionary techniques will determine the number of each type of robot and the distribution of robots as it evolves over time in order to ensure success.

3.2 Search Methodology

Search methodology or strategy is defined as the optimal technique a swarm uses to find a source or target. How a swarm scans a hostile region depends on its ability to maneuvor quietly and efficiently. Two methods or test cases are employed : a random-walk and a GA determined path.

Test case one simulates a multi-agent system consisting of 32 randomly positioned and randomly walking robots searching a target. Analysis compares target seeking and condensation performance as a function of the swarm's initial geometry, the number of swarms deployed, the robot's ability to communication information, and the collective motion of the swarm through the introduction of a pseudo robot-to-robot interaction potential. These tests indicate that the random motion of the robots overshadow many of the above parameters, dominating the performance of the swarm to search out a target. As a result, only the distance between the swarm and the target is correlated to performance, leaving many of the above parameters statistically ineffective. What is missing is an optimal collective search pattern.

Test case two begins to address the shortcomings of a random-walk search strategy. The simplicity of random walkers is an attractive attribute but can be inefficient. Providing an open communications channel between robots alleviates some problems, but the overhead can make it

more complex. The aim is to determine a robot's optimal trajectory within a pre-defined search area. Here, optimal means energy conservation, i.e., covering that maximum area in the fewest moves. Obstacles, such as boundary conditions, features in the terrain, and other robots, perturb this trajectory. Implementation of a GA determines the path type and the resources, memory and sensor parameters, necessary to create the path.

First, the introduction outlines the two test cases to be simulated : a random walk and GA determined path. Second, results are compared. Third, conclusions are discussed.

3.2.1 Introduction

Test Case 1. A multi-robot system on a data gathering mission can be broken down into five steps. The initial random-walk algorithm handles each step (except the last) by (1) randomly distributing 32 robots over an area that is 200 x 200 units^2, (2) initiating a random-walk procedure for each individual robot, searching this area for a source (or signal), (3) communicating the coordinates of the robot that first contacts the target, and (4) equally distributing all the robots at the center of the target. The only constraint is that a robot cannot step outside the search area (i.e., a boundary condition).

The diagnostics used to measure the performance of the robotic system consist of three graphs : (i) the number of robots reaching the target as a function of step number, (ii) the number of robots entering the target region as a function of CPU usage, and (iii) the single-step CPU usage as a function of communication step number. In most cases, the number of steps taken to find a target (first contact) and the CPU usage for each step are used to measure performance, where the step number is iterated only after all 32 robots have moved one step. For this test case, performance is measured relative the step number of first contact. Realistic scenerios include a robot's microscopic specifications (description), for example sensor range, motion detection, communication range, communication efficiency, noise suppression, and duration of communictions, and its collective macroscopic characteristics, for example deployment, multiple swarms, coverage, and collective motion.

Genetic Sequence	Definition
g_0	No Obstacles
g_1	Obstacle to the Left (L)
g_2	Obstacle to the Right (R)
g_3	Left and Right (L+R)
g_4	Obstacle Above (U)
g_5	Left and Above (L+U)
g_6	Right and Above (R+U)
g_7	Left, Right, and Above (L+R+U)
g_8	Obstacle Below (D)
g_9	Left and Below (L+D)
g_{10}	Right and Below (R+D)
g_{11}	Left, Right, and Below (L+R+D)
g_{12}	Above and Below (U+D)
g_{13}	Left, Above, and Below (L+U+D)
g_{14}	Right, Above, and Below (R+U+D)
g_{15}	Right, Left, Above, and Below (R+L+U+D)

Figure 14. The genetic sequence for a robot.

Test Case 2. Here, the objective is to determine a robot's trajectory which searches the largest area in the fewest moves, i.e., energy conservation. Additional constraints such as remaining covert (i.e., no communications) and controlling the swarm's collective state (e.g., flocking dynamics) are and will eventually be implemented if not already through the use of PIC codes (see section 2.1.3).

A robot's search mode is defined by a simple micro-instruction set. These instructions (or commands) represent the genes of the genetic algorithm. Therefore, the values assigned within the robot's genetic sequence as shown in figure 14 defines a robot's path, a GA optimized micro-program. This path depends on the robot's ability to conserve energy while avoiding obstacles, such as other robots, boundary conditions, terrain, countermeasures, and other constraints. Confining a robot to roam on a square lattice and restricting the robot's sensing capabilities to its nearest-neighbors (left, right, up, down (L, R, U, D)), its programmed genetic sequence accounts for any immediate contingency. As an example, if the situation (the environment) fulfills the criterion of the fourth gene -- an obstacle to the left and the right of the robot's current position -- the solution is the value of g_3. The g_3 response allows the robot to move left, right, up, down, or stand-still (the set {L, R, U, D, SS} which has a cardinality of 5 defined by the lattice structure). As a result, 16 possible genes each acquiring one of 5 values exist.

The simulation allows a population of 64 (and 128) single robot swarms to move within a 10 x 10 unit^2 square lattice. The GA evolves the population until the robot's genetic sequence (i.e., trajectory) corresponds to the global optimum of the GA's fitness function, equation 16. The simulation steps are as follows : (1) chosen are a random set of genetic values and of initial robot positions; (2) the robot moves 100 steps guided by its genetic code; (3) a robot's fitness is calculated, .

$$fitness = \frac{1}{\left(0.1 + \left(A_{Total} - \sum_{k}^{nMovesnRobots} \sum_{i} Pts_{k,i}\right)^2\right)}, \qquad (16)$$

where A_{Total} is the total grid area (e.g., 100 units^2), nMoves is the total number of moves of a robot (for a single robot, nMoves is equal to 100), nRobots is the size of the swarm, and $Pts_{k,i}$ keeps count of the number of grid points each robot traverses; (4) a roulette wheel technique selects two robots at a time to mate; (5) single-point crossover method produces genetically-mixed offspring; (6) mutation occurs randomly over the entire population; and, (7) each offspring randomly chooses its initial position before steps 2 through 6 are repeated. In these tests, an obstacle is defined as a boarder, a neighboring robot, or the previous move (all three are indistinguishable to a robot). After a few hundred generations, the genetic sequence evolves into an optimal search pattern.

3.2.2 Results and Discussion

Test Case 1. The performance of the random motion search algorithm is evaluated as a function of various characterics of the swarm : (1) the swarm's initial distribution, (2) the number of swarms, (3) the communication between robots, and (4) the influence of a pseudo-potential governing the robot's movements.

First, simulations study the swarm's performance in relation to its shape. The initialization state for each run consists of 32 randomly positioned robots covering ~10% of the total search area. First contact measurements are taken for five target positions and three swarm center positions. The two shapes are labeled as circular representing an ellipse with a major to minor ratio of ~2:1 and as cigar representing an ellipse with a major to minor ratio of ~7:1. After averaging first constact times over different target positions while holding the swarm's distribution constant and over different swarm positions while holding the target position constant, the results do not reveal any statistical correlation in performance between swarm shapes.

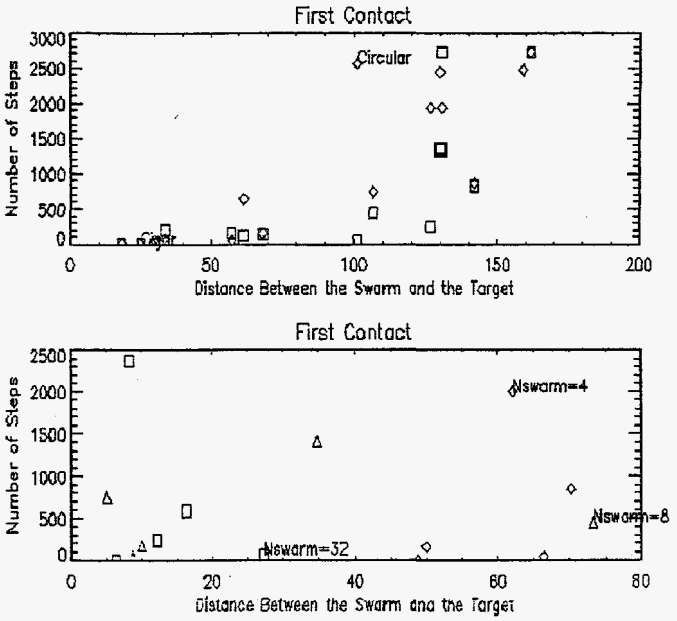


Figure 15. Top graph is a plot of the distance between the source and nearest swarm versus the number of steps needed to make first contact. The bottom graph is the same plot for different swarm sizes.

Second, robot clustering, i.e., the number of swarms, is tested. 32 robots covering 10% of the total search area are grouped differently. Again, results reveal no obvious correlation. Even though one may argue that 32 randomly spaced robots is slightly better than 4 swarms of 8 robots, the underlying cause does not appear to be related to the number of swarms. To see this, the minimum distance from the center of a swarm to the target is plotted as a function of step number (see figure 15). This plot indicates that the distance to the nearest swarm is correlated to the number of steps necessary to make first contact, but only up to 100 units, where the distribution becomes random. As the number of swarms increases, this correlation breakdown-distance decreases (for 4 swarms down to 60 units and for 8 swarms down to 30) because fewer robots comprising a swarm means fewer directions eminating from the center of that swarm are being probed.

Third, a communication range is enforced. Once a robot locates a target, it broadcasts a signal to all robots within its communication range (e.g., 12.5, 25, or 50 units). This signal lasts for a brief time -- 1, 5, or 25 steps. Therefore, once a robot senses a target, it broadcasts for N steps to those robots within its communication range, who in turn, communicates this same information for N steps to all its neighbors lying within its communication range. Eventually, all communication stops and M (< 32) robots converge upon the target. In table 5, results are tabulated. As the communication range increases, the time (i.e., the number of steps) needed to broadcast the signal decreases in order to attract the same number of robots. This effect can also depend on the size and distribution of the swarm. Such parameters are important if the objective of the swarm is to remain covert.

Fourth, the influence of a nearest-neighbor potential field on the search dynamics is explored. The robot-robot interaction consists of (1) a repulsive hard-core contribution so if the distance between two robots is less than the maximum allowable step size, the chosen direction for the robots is in opposite directions (OD), (2) a "turn-off" piece so if the robot moves outside

Communication time interval	16 Swarms			1 Swarm		
	Comm. Range 12.5 units	Comm. Range 25 units	Comm. Range 50 units	Comm. Range 12.5 units	Comm. Range 25 units	Comm. Range 50 units
1 step		1	<20>			
5 steps		1	<22>	3	10	32
10 steps		5	<23>			32
25 steps			<23>		<21>	32
50 steps					<23>	32
100 steps		5				32

Table 5. The number of robots reaching the target after 1200 steps (steady-state condition) as a function of communication range (12.5, 25, 50 units) and the number of swarms (1 and 16).

the communication range of its nearest neighbor, the robot remains dormant (STOP), and (3) a random-walk section inbetween these two extemes causes the robots to move in a random direction (RD). In essence, this logic is similar to a fuzzy set for the motion parameter, expanding the robot's kernal of motion. However, the end result is somewhat uninteresting. The swarm of robots diffuse outward and essentially stop (reaching a somewhat steady-state condition, lattice). Additional intelligence is necessary to cause the swarm to propagate in an efficient manner. One can imagine that the shape of the swarm will depend on the average number of robots a single robot must remain in communication with. However, this parameter can also effect the ability of the swarm to overcome obstacles.

Test case 2. The genetic micro-instructions of figure 14 does not provide the robots with knowledge of past events, i.e., no memory. As a result, the optimal genetic sequence traces out a box, a path following the circumference (the boarder) of the 10 x 10 unit^2 square lattice. A maximum area between 43 and 49 units^2 depending on the robot's initial position is covered. At the opposite extreme, a map representing the lattice sites stores the full history of the robot's trajectory, perfect recall. This is not unprecedented, in biological systems passive communication ("memory") occurs by modifying the environment. Ants deposit chemicals (pheromones) indicating the presence of food, danger, or another ant (had passed this way). Implementing the same GA, the genetics recreate a square-wave trajectory, covering an area from 90 to 100 units^2 depending on its initial conditions (figure 17C). An interesting feature of this analysis is the last gene, g_{15}, an obstacle in all four directions. In this situation, the robot can SS or it must move in a direction that minimizes any overlap with other regions. A property that may need to be exploited in future work.

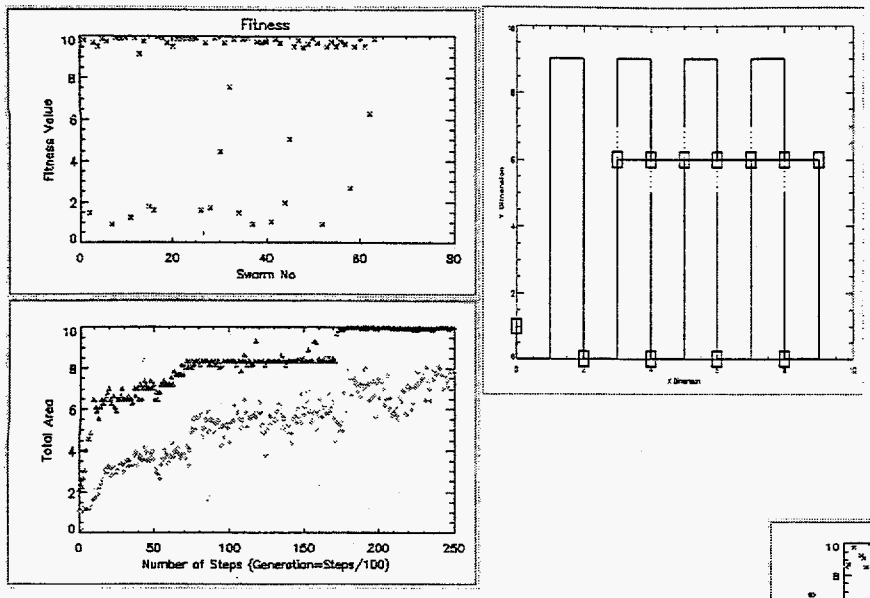


Figure 17A. The robot recalls its last previous move. Plotted are the evolution of the fitness function and the robot's trajectory when started at a random initial position.

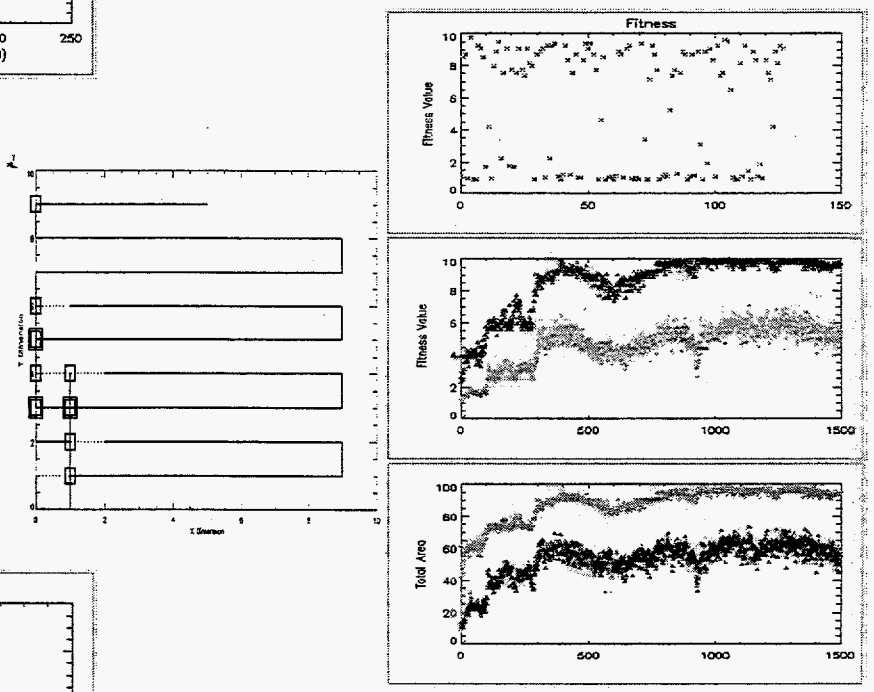


Figure 17B. The robot recalls its last two previous moves. The plots consist of the evolution of the fitness function and area covered as well as the final path of a randomly positioned robot.

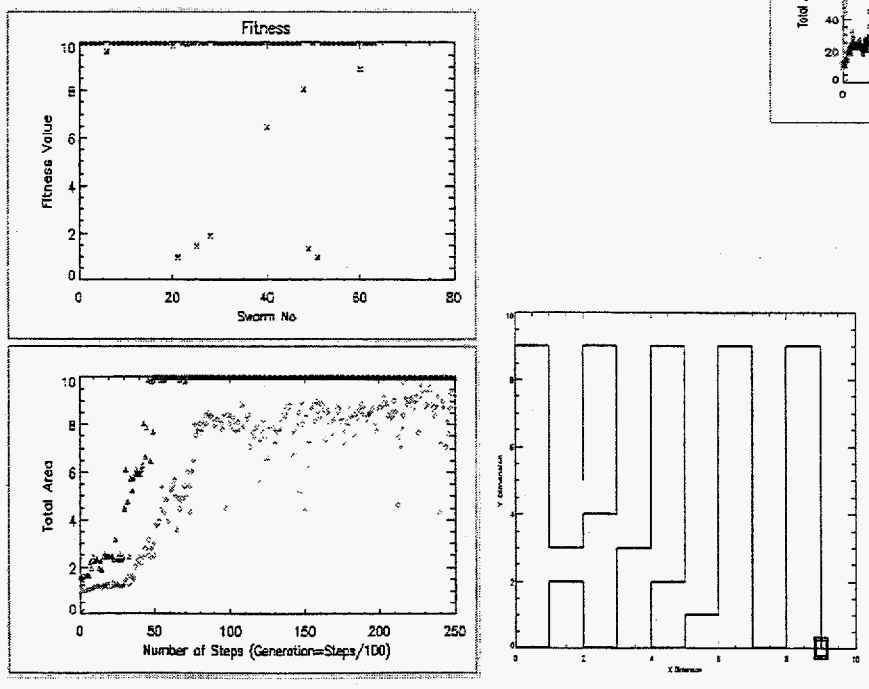
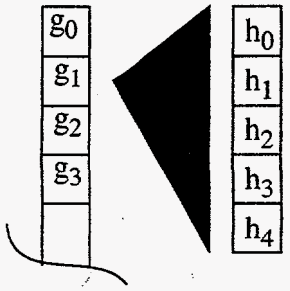


Figure 17C. The robot has full memory, it recalls up to 99 previous steps. Plotted are the evolution of the fitness function and the robot's trajectory when started at a random initial position.



IF (obstacle to Left) AND (previous move SS) THEN move h_0
IF (obstacle to Left) AND (previous move Left) THEN move h_1
IF (obstacle to Left) AND (previous move Right) THEN move h_2
IF (obstacle to Left) AND (previous move Up) THEN move h_3
IF (obstacle to Left) AND (previous move Down) THEN move h_4

Figure 16. Expanding a gene from figure 14 to account for the previous move (history) traversed by a robot.

The decision making processes of the robots must include past events. The next two simulations incorporate genes that depend on the robot's previous move (figure 16) and two previous moves. The optimal solution resuls in a square wave trajectory covering 87 to 98 units^2 (figure 17A), differing from the full memory case in how the track begins to move. Going from zero to one previous move increases the genetics by a factor of five, going from one to two previous moves creates a robot with 400 genes. Simulations also reach a total search area from 87 to 98 units^2 (figure 17B), but the GA converges more slowly due to the size of the genetic sequence. These results indicate that the robots require only short-term memory, thus keeping the genetic code simple.

The addition of other robots (2, 4, and 8) perturbs the basic pattern. Qualitatively, each robot attempts to follow a similar square-wave path but in a restricted region near its initial position. Overlapping paths does depend on memory.

3.2.3 Conclusions

The random-walk technique dominates the performance of the search algorithm overshadowing other parameters such as deployment strategy. Implementing a GA enhances efficiency until the density of robots within the search region becomes significant. At this point, differences between the methods are minimal.

A low density of robots with short-term memory, knowledge of its previous step and recognition of obstacles (boundary, other robots), can efficiently search an area for a signal. Complex systems, which includes obstacle recognition and a higher density of robots, perturbs this path. Overcoming such complexity requires aditional memory and the means to access it. Obstacles and low densities of robots need only short-term memories, but efficient, global swarm movements require a game plan, the influence of a long-term memory. One technique that efficiently realizes short-term memory is a recurrent neural network architecture. Long-term memory may need to be genetically determined and programmed into the robot's actions or "being".

Future directions may include (1) enhancing the GA performance and assuring results are position independent, (2) recognizing and responding to different obstacles such as terrain, boundary, and other robots, (3) determining the effect of swarm size and its state variables, (4) accounting for sensor range dependencies, (5) studying time-dependent genetic sequences, and (6) combining long- and short-term memory. Implementation of global optimization techniques

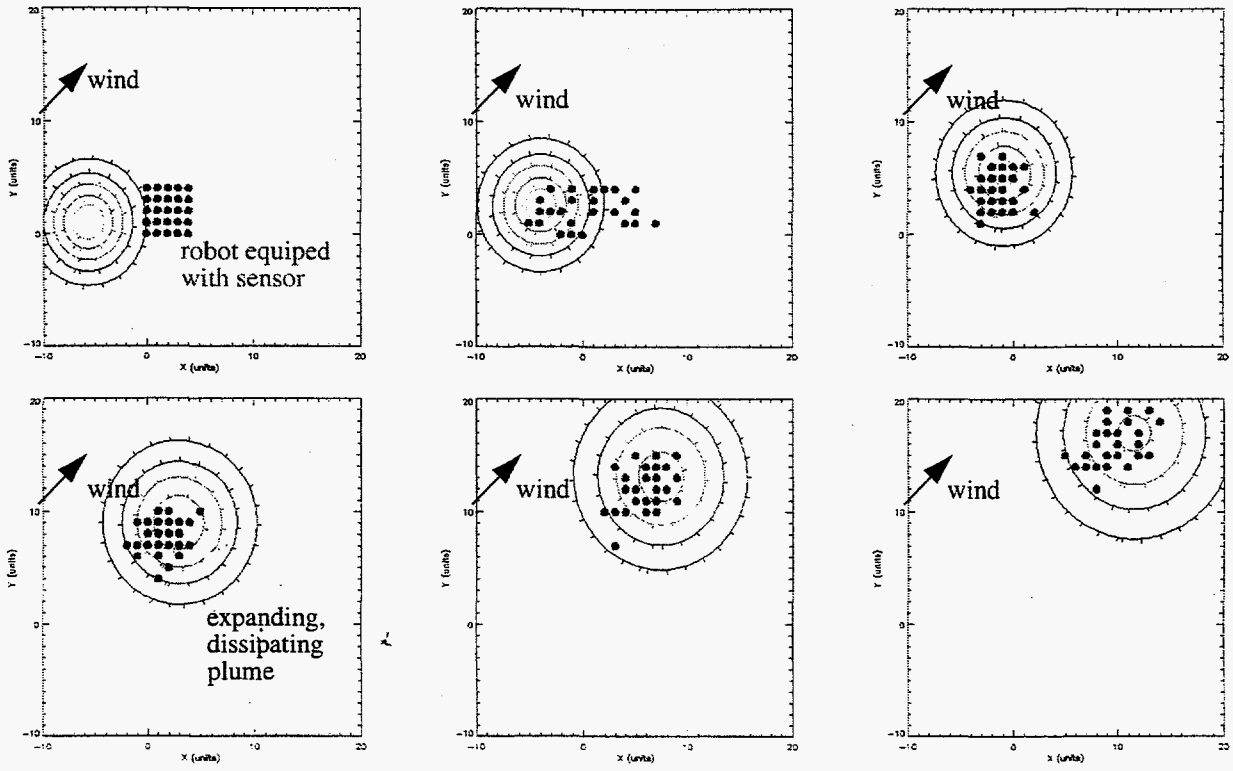


Figure 18A. A swarm of robots controlled by the potentials fields collectively follow a dissipating, wind-blown gaseous plume modeled as a gaussian pulse.

can not only determine the genetic micro-program within each robot, but develop individual neural network structures for each robot designed to handle environmental conditions by accessing various levels of memory.

3.3 Target Convergence

Physics-based models -- lattice gases (LG) and plasmas (section 2.2) -- quantify the macroscopic state of an N-body system through its microscopic, nearest-neighbor interaction potentials. By defining the swarm's collective state by its partition function, thermodynamic properties can be calculated which would allow density to be related to speed or maneuverability and parameters like pressure, tension, temperature, and internal energy to be related to cohesiveness. Current research in statistical phys-

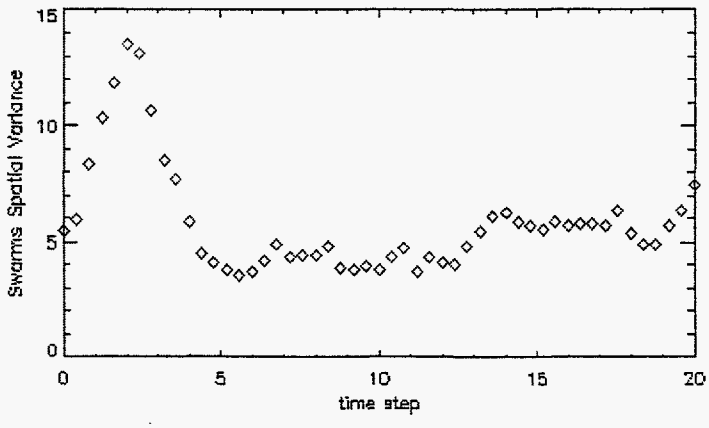


Figure 18B. The radial variance of the swarm as function of time as it follows a target.

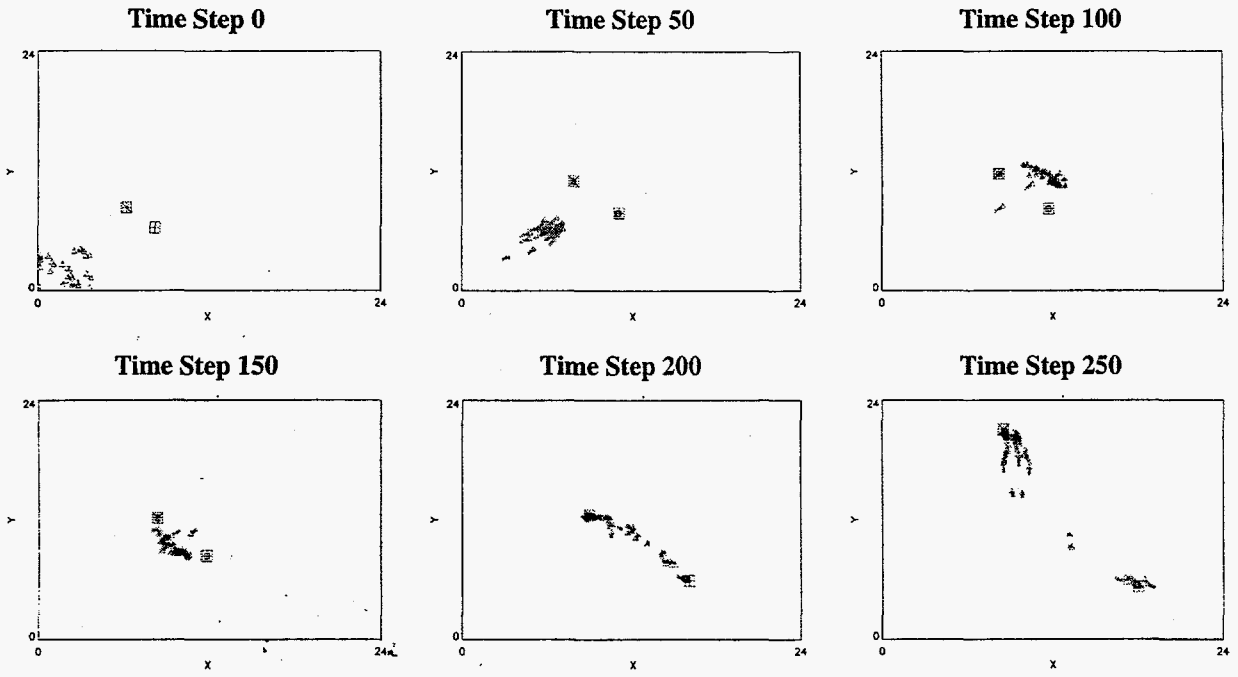


Figure 19. A swarm can split.

ics extends LG models by incorporating additional particle species, linked to heterogeneity, and outside potentials, called pseudo-potentials. By incorporating measured sensor and robotic transfer functions into these models realizing molecular, gaseous, and plasma systems, well-controlled and understood states for the swarm can be derived.

In these next two examples, simulations of a swarm of ground-based sensors using a LG model and an airborne UAVs implementing PIC codes track various moving targets. The goal is to determine whether cohesive, macroscopic states of the swarm can be maintained implementing only nearest-neighbor forces.

3.3.1 Simulation of a Swarm Tracking a Gaseous Plume

Three potentials control the state of the swarm -- a repulsive, an attractive, and a gradient (or source) potential. Displayed in figure 18A is a LG simulated swarm following a source, a 2-dimensional gaussian pulse expanding and dissipating. The potential strengths -- α, β, and γ -- are tuned so that the swarm remains intact and follows the gradient of the signal's field. The functional form of the forces follows from equation 1, but eventually the GA will determine an optimal form. For this simple example, density and possibly internal energy (or temperature) are important thermodynamic parameters controlling the size and speed of the swarm. Plotted in figure 18B is the time dependent radial distribution of the swarm. The initial peak signifies the swarm's ability to spread-out, amoeba-like, toward the pulse. The average radial variance plateaus to a minimum after condensing onto the source, and the swarm's density slowly decreases as a result of the pulse dissipating over time. Quantifying such macroscopic states relative to swarm behavior will verify the physics equaiton which will allow for autonomous behavior during the training of the swarm. With the ability of the global neural networks to monitor the environment

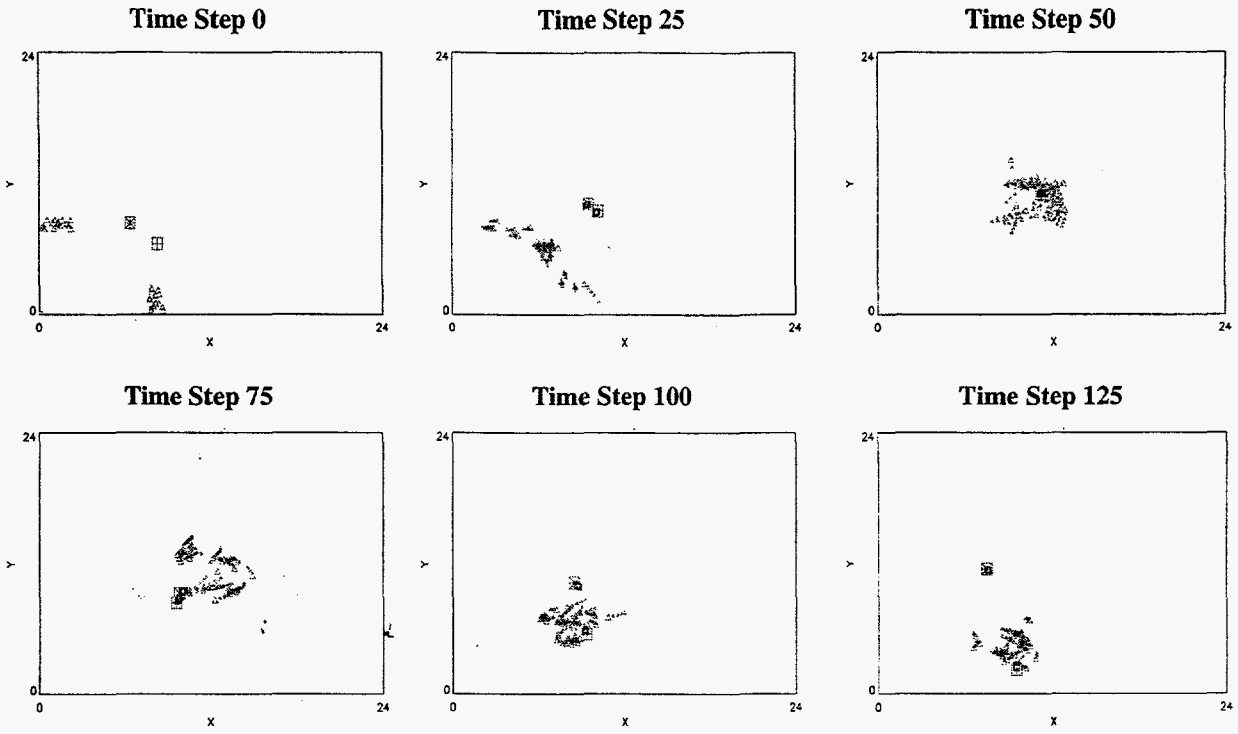


Figure 20. Swarms can join.

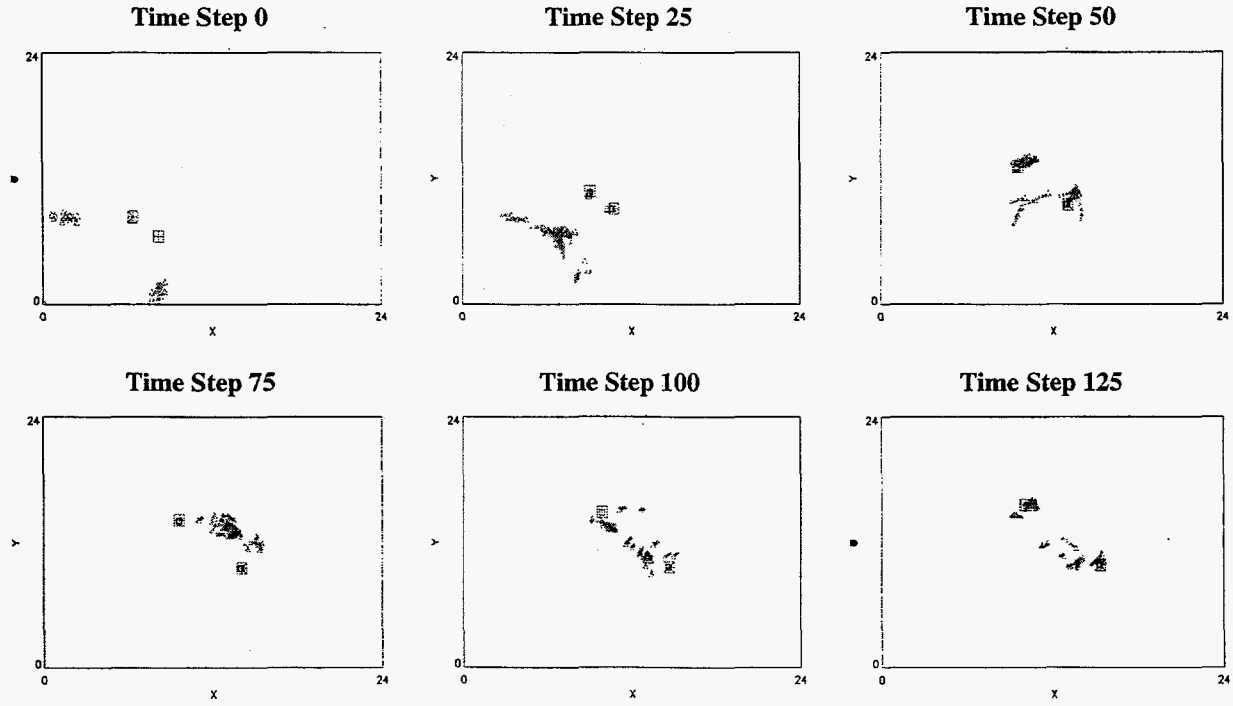


Figure 21. Swarms can cross.

(through pattern recognition) and its own state, the swarm will be able to autonomously determine, control, and change its macroscopic state by modifying its local interactions.

In conclusion, the connection between performance or behavioral patterns can be linked to physics-based models. Simulations such as in figure 18 begin to match density (radial variances) with such quantities as cohesiveness and maneuvorability, two important swarm state parameters.

3.3.2 Simulation of an Airborne Swarm Tracking Multiple Targets

Introduction. Swarm dynamics can be modeled using particle simulation concepts. The interactions between individuals can be modeled through the standard force equations and equations of motion. This allows localized, collective behavior and collision avoidance to be handled.

Additional, problem-specific forces can be handled by using vector addition to combine them to the individual interaction forces. This permits the modeling of a variety of physical effects such as the macroscopic center-of-mass or swarming force, friction, dissipation, and aerodynamic forces, gravitation, thrust (currently implemented as an attraction to or repulsion from another individual), and other environmental forces such as obstacles, boundaries, terrain, and weather (only periodic and reflective boundaries are currently implemented).

As shown in figure 19, a plasma molecular dynamics code has been modified to simulate flight dynamics and collective behavior[19]. Force laws for friction, drag, and inertia, a pursuer's swarming and target seeking forces, and a target's swarm avoidance force are added. A target's or pursuer's charge and mass are specified to control their relative attraction/repulsion and agility. For efficiency and maximum flexibility, the swarm model implements grid-optional particle simulation concepts. The main window and configuration menu of our simulator. In the main window, the frames in figures 18, 19, and 20, targets are shown as squares and their pursuers are shown as triangles. Although the simulator is merely a prototype at this point, it is robust and easily extensible. The code runs on both a Unix (Sun Solaris) workstation and a Windows NT workstation.

Simulations. Our first example (figure 19) starts with a single swarm tracking two targets. The targets are close to one another but have divergent headings. At time step 50, the targets diverge and the swarm begins to rapidly overtake them. At time step 100, the swarm overshoots the targets and begins to reverse course. At time step 150, the swarm spreads out between the two diverging targets. At time step 200, the swarm separates tracking both targets. By time step 250, the splitting of the swarm is well underway. This example shows a swarm's ability to split and track multiple targets.

Our second example (figure 20) starts with two swarms tracking two targets. The targets are close to one another and on a collision course heading away from the swarms. At time step 25, the targets converge and the swarms are in the process of merging. At time step 50, the combined swarm overtakes both targets and envelops them. At time step 75, the targets reverse direction and the swarm remains in pursuit. At time step 100, one of the targets changes direction. So, the targets are now moving in opposite directions. The swarm concentrates on only one target. By time step 125, the two targets are well separated and the swarm is pursuing only one of them. This example shows that targets can work together to "spoof" their pursuers, by forcing swarms to join, allowing one of the targets to escape.

Our final example (figure 21) starts with two, closely-spaced targets on intersecting paths. Each target is pursued by a small swarm. At time step 25, the targets' paths cross and the swarms are in the process of colliding. At time step 50, the targets begin to separate and the combined

swarm overshoots them. At time step 75, the two targets continue to separate and the swarm spreads out between them. At time step 100, the swarm splits apart following both targets. By time step 125, the two, well-separated targets are now being pursued by two separate swarms. This example shows that, with a slight decrease of the pursuers' swarming force, swarms can cross without joining ensuring that a target cannot escape.

To summarize, the ability to model swarms of UAVs and/or fighter aircraft using grid-optional particle simulation concepts has been successfully demonstrated. Three examples of a target/interceptor scenario show that the model is capable of very complex behaviors. The model is easily (and has been) extended to 3-dimensions and can be used for land-, air-, and sea-based scenarios.

3.4 Adaptive Behavior/Optimal Sensor Position

As mentioned in section 2.5, autonomous behavior means the swarm must be capable of adapting to its environment while tracking, following, or searching for a source. Therefore, the ability to overcome obstacles due to objects or variations in the terrain must be considered. Simulations implementing nearest-neighbor interactions reveal that rather mundane conditions can cause the swarm to break apart. Obstacles, such as a wall, causes the swarm to spread itself thin until a critical number of connections or connection length (which can be related to communication and sensor ranges) causes the swarm to divide. The objective of the GA is not only to globally optimize neural network designs but to optimize the potential fields. In other words, including physics parameters into the fitness function has the advantage of training the robots to perform pattern recognition (at both the individual and collective level) and assert additional pseudo-potentials. One type of behavior presented next is obstacle avoidance in the form of a wall.

3.4.1 Obstacle Avoidance

Introduction. The setup begins with a swarm configured in a square with its lower left robot occupying the coordinate (0,0), see top left picture in figure 24. A plume, represented by a 2-dimensional gaussian pulse initially at the coordinates (-5.6, +1.0), propagates according to the wind's velocity vector (+0.43, +0.40), dissipates exponentially in amplitude with a decay constant 0.015, and spreads exponentially in width with x and y constants of (0.015, 0.015). A simulation of such a signal at nine different time steps is shown in figure 24.

By tuning the three potential strength constants for the attractive, gradient, and repulsive forces (α, β, and γ), the swarm not only remains in contact with the source but remains intact. The addition of a 1 unit thick wall placed at a y of 20 and spanning a length from -10 to 13 in x prevents the swarm from following the target. Using the values of α, β, and γ described in section 3.3.1, simulations show the swarm stopped at the wall with only an occasional robot or two splitting apart remaining in contact with the plume. Instead, the (parallel) genetic algorithm evolves a population of swarms optimizing nearest-neighbor potential strengths (e.g., realized by amplifier gains). These variables can change as a function of time (time-dependent genes) or their position relative to the obstacle (spatially-dependent genes). The former technique modifies the state of the swarm in time, while the latter modifies the potential of individual robots as they enter different regions of space. In the end the results are similar; therefore, only the space-dependent behavior will be shown.

Region No.	(x_{min},y_{min})	(x_{max},y_{max})	α	β	γ
1	(28,15)	(28,28)	94.8	5.8	63.1
2	(3,7)	(15,37)	5.7	82.7	5.7
3	(12,9)	(22,18)	15.1	92.2	14.2
4	(10,5)	(19,40)	62.3	89.1	86.1
5	(6,6)	(30,30)	10.4	87.3	82.4

Table 6. The GA results for the potential field strengths within the 5 different regions displayed in the top left picture of figure 22.

Simulations. Without any intelligence, i.e., NN induced pseudo-potentials, can a swarm of robots interacting with only its nearest-neighbors according to equation 1 overcome an obstacle. A parallel GA (section 2.3.3) optimizes potential strengths within five regions relative to the obstacle. During each generation, every swarm within the population simulates the LG algorithm nine times, defined as the number of cases. The fitness function represents a measure of the average number of robots finishing in the region where the gaussian exits the field of view beyond the barrier -- x>10 and y>17 -- at time step 70.

$$fitness = \frac{1}{\left(0.1 + \frac{\sum_{nCases} (nBots - nFollow_j)}{nCases}\right)}, \qquad (17)$$

where nBots is the size of the swarm (25), nFollow is the number of robots finishing in the above mentioned region, and nCases is set to 9. Equation 17 produces a reasonable convergence rate; however, it is more instructive to plot the average number of robots over the nine cases for the optimal swarm solution in the population as a function of the generation number. This plot is shown in figure 22. Due to the low statistics, i.e., a swarm of 25 robots, the density of values (fitnesses) near the maximum is small and the form of the fitness func-

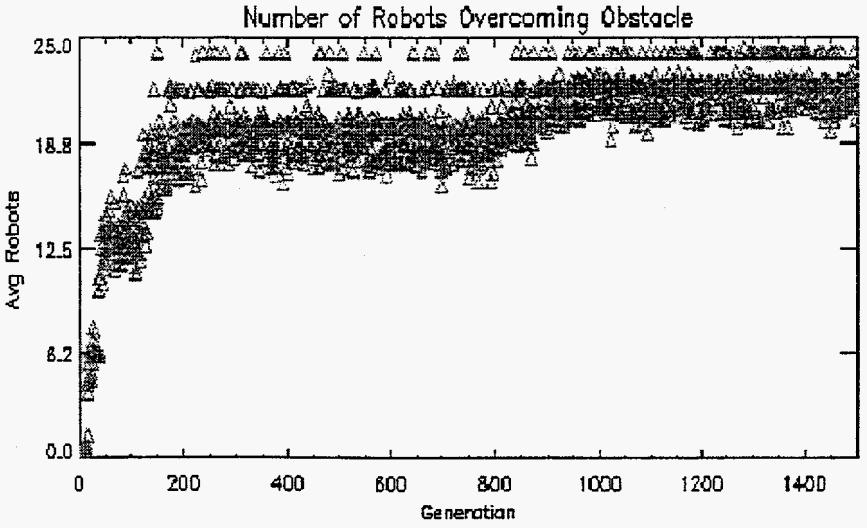


Figure 22. The GA converges toward an optimal set of potential field strengths that allows the swarm to overcome an obstacle.

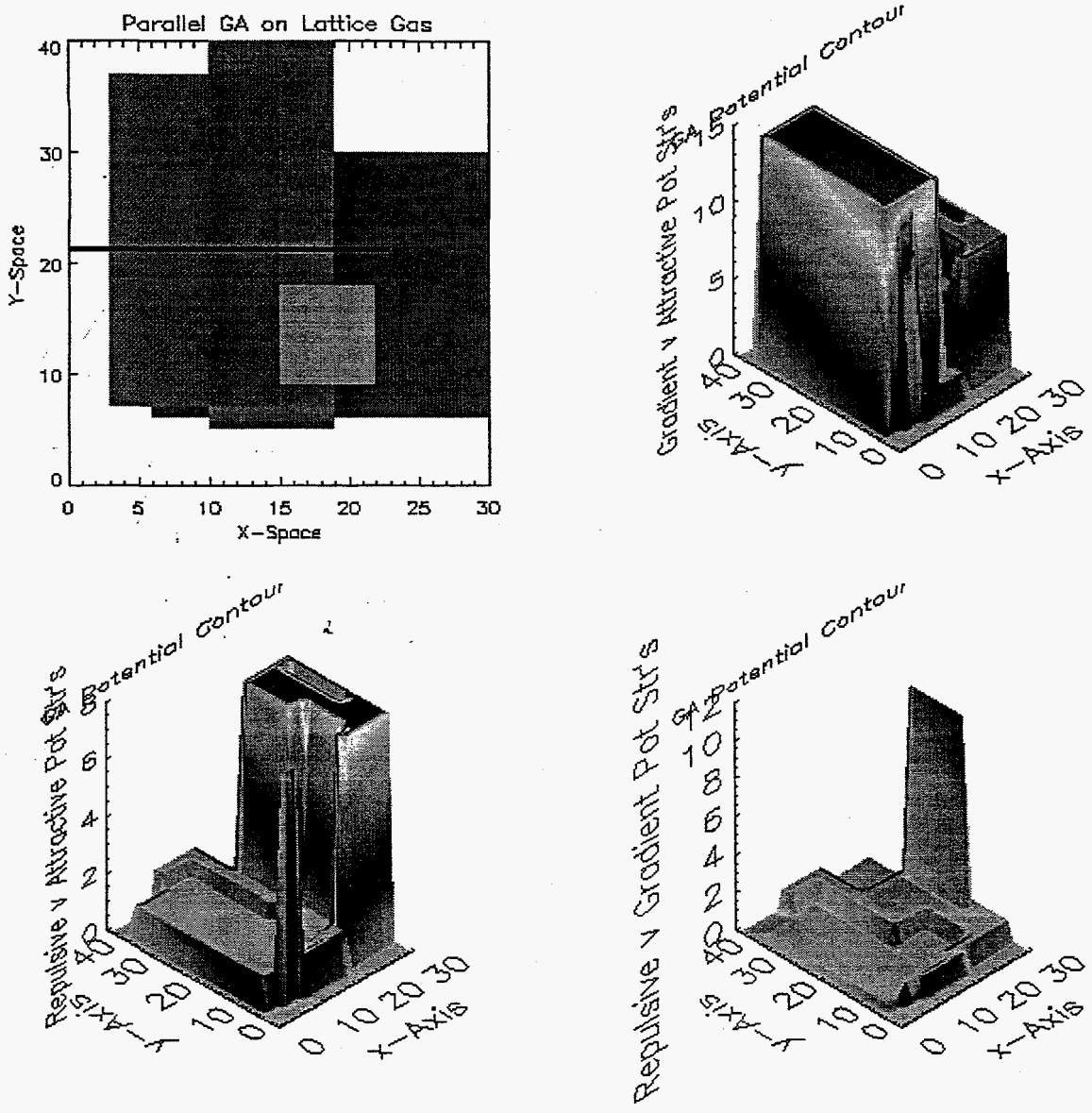


Figure 23. Top right, the potential field regions relative to the wall. Top left, bottom left, bottom right: 3D surface plots of the gradient versus attractive, repulsive versus attractive, and repulsive versus gradient potential field strengths.

tion, the fitness appears discrete. The variation in the plot is created by the randomness in the robots' movements from one LG simulated run to the next. By increasing nCases, these deviations decrease (proportional to the square root of nCases), allowing the GA to converge faster. The choice of nine for the number of cases is a trade-off between compute time and convergence rate.

The results of the GA are tabulated in table 6 graphically displayed in figure 23, where the top right plot shows the placement of the five weighted regions and the upper right, lower left, and lower right surfaces compare the relative magnitudes for α, β, and γ, the gradient versus attractive, repulsive versus attractive, and repulsive versus gradient fields, respectively. Three observations

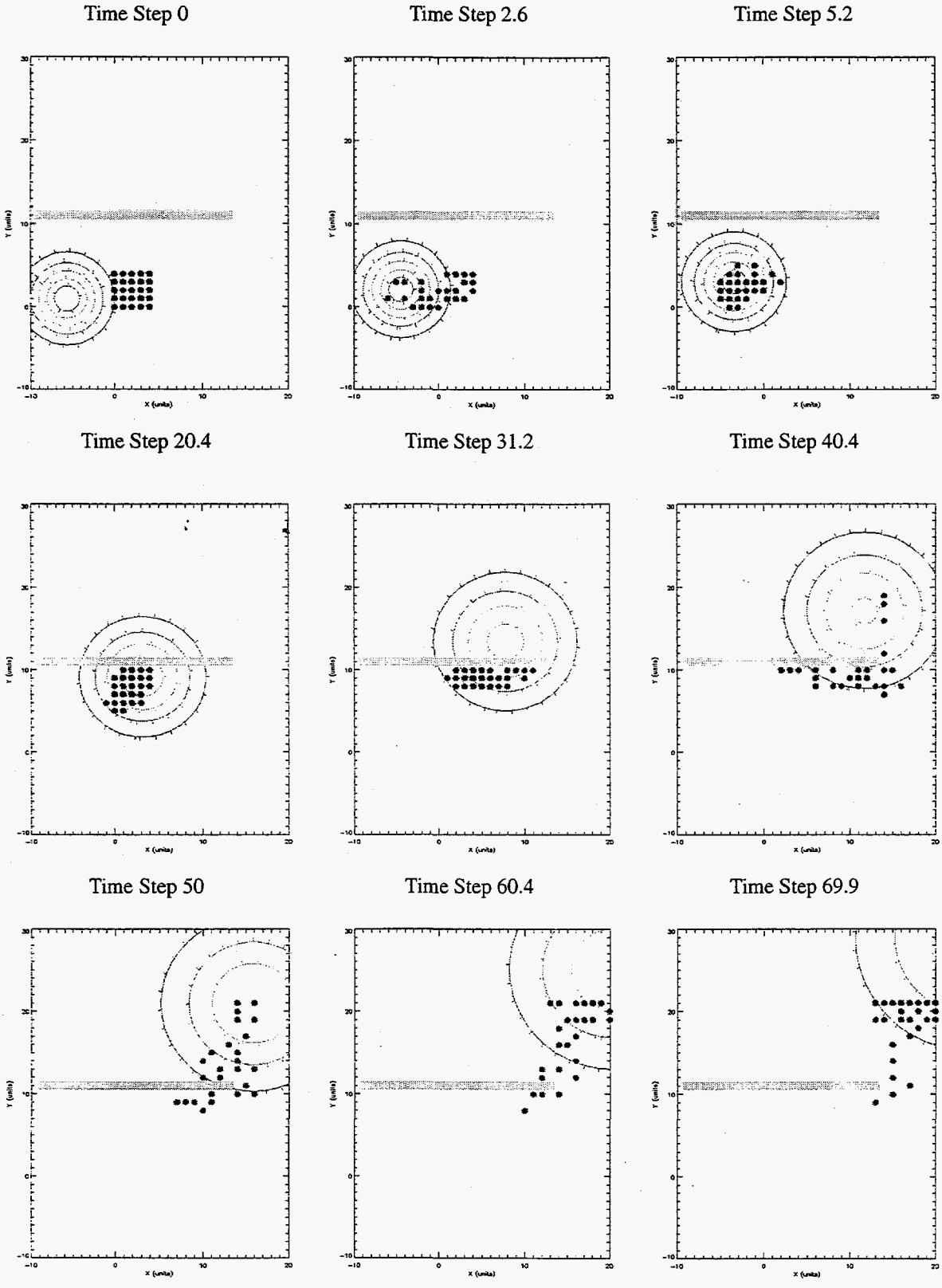


Figure 24. Lattice Gas simulations of a swarm overcoming an obstacle. Potential field regions are defined by table 6.

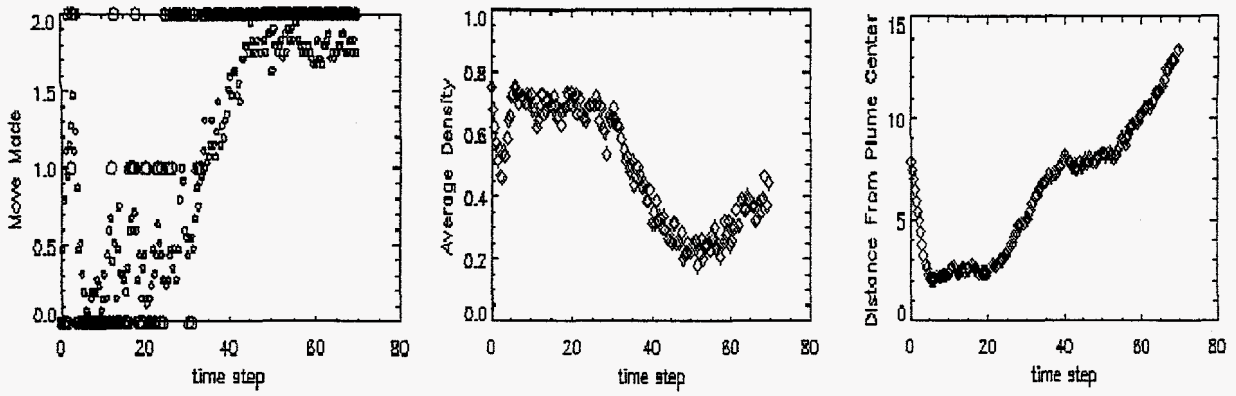


Figure 25. The type of move made by all robots, the average density, and the distance from the plume plotted as a function of time step for the simulation shown in figure 23.

can be made. One, initially the swarm is guided by the source potential which remains high relative to both repulsive and attractive forces. Two, as the swarm is obstructed by the wall, the repulsive force increases, especially at the corner and on the other side. Three, during this whole sequence the source potential remains relatively constant.

The potential field distributions determined by the GA and graphed in figure 23 are inserted into the LG simulation code. Pictured in figure 24 are snap-shots in time of the swarm's behavior as it follows and attempts to overcome the wall. Initially, the swarm quickly condenses onto and tracks the simulated plume. As the wall impedes the motion of the swarm, the robots collect at and begin to spread out along this barrier. Closer to the edge, the repulsive forces dominate. As a result, the decrease in density improves the swarm's maneuverability and increases its speed, but at the cost of a cohesiveness, which the GA compensates for. The plots in figure 25 track the swarm's movements, its average density, and its distance from the center of the plume. As the plume begins to separate from the swarm at time step 25, the right-most plot in figure 25, the repulsive forces increase, thus lowering the average density, the center plot, and increasing its maneuverability, left-most plot. The swarm quickly turns the corner and catches up to the target during time-steps 40 to 50 (right-most plot of figure 25 and frames 6 through 8 in figure 24). Due to the somewhat ambiguous fitness criteria requiring all robots to finish within a specified region, the swarm is prevented from condensing back onto the plume for time steps greater than 50. This situation can be easily resolved by upgrading equation 17.

To summarize, the GA modifies the swarm's state (and its behavior) in order to overcome an obstacle while remaining intact, i.e., a collective unit. Simulations measure speed, average density, and separation distances between the plume and swarm. Matching the theoretical equations to these measurements is progressing. By leveraging the pattern recognition techniques (section 2.3.3) to determine boundaries and edges of obstacles, the swarm will be capable of overcoming many obstacles autonomously.

3.5 Communication Pathways

Interconnectivity requires communication techniques internal to the swarm and external to a host. The former realizes the interconnectivity or networking of the swarm, which can be represented by the synapses of a neuron; the latter provides a means for the swarm to communicate its

findings to others (e.g., battalion commander).

3.5.2 Internal Communications

The elements of a swarm share information with their nearest-neighbors. A simple implementation would be a laser communication system. A directed communication link to a neighboring robot can provide local, line-of-sight communication channels. This eliminates electromagnetic broadcast systems. Even though a viable technique, it prevents multiple users talking at the same time or many frequency bands, complicating designs. Directed laser pathways are covert and controllable.

Information must also be available to the entire swarm, global intelligence. One scenario would be for a single element in the swarm to require a broadcast of information. If such polling is limited (which is realistic), broadcasting is viable. Over time different robots within the swarm calls upon the global intelligence of the swarm to decide whether to change state. This information is gradually passed onto other robots. Thus global intelligence constantly upgrades the swarm's state and does so in an adiabatic fashion by separate robots within the swarm implementing its global intelligence algorithms and passing this information to the swarm. Some work has evolved in networking methods and their protocols to replace broadcast methods; however, nothing has been simulated to date.

3.5.3 External Communications

A major weakness limiting the operational utility of unguided optical communication channels in atmosphere has been sensitivity to dispersive scattering and turbulence noise sources (scintillation) which cumulatively degrade signal fidelity, gain aperture, and reception distance. We have developed a frequency-agile adaptive receiver system based on optical parametric amplification capable of overcoming these adverse effects without compromising transmission rates or minimum BER threshold levels. Undistorted data streams from an E/O reflectance modulator can be detected through scattering layers (clouds) a time-gated optical amplifier (mixer) properly compensated for wavefront distortion with high bandwidth and gain. For a nonideal detector characterized by thermal noise, optical preamplification can improve resulting electrical SNR despite the 3 dB ASE floor. By working in a type II phase matching geometry at the degenerate point, both direct incoherent (classical amplifier) and enhance local oscillator homodyne coherent detection (phase sensitive quantum amplifier) modes of operation are possible by adjusting input polarization. Common mode rejection techniques can be incorporated to mitigate pulse stack-up and improve the noise We have constructed a baseline laboratory prototype based on the OPA receiver concept for proof-of-principle experiments under variable visibility conditions to detect known reference signals. The transmitter is a Ti:sapphire femtosecond laser operating at a KHz pulse repetition rate which is wavelength shifted to 1.31 microns to match the properties of a Sandia-built miniaturized surface normal reflectance modulator.

4 Applications

The application domain for an adaptive array of sensors or an intelligent multi-agent system is large and permeates throughout our technology base. Three currently relevant areas of emerging threats include bio-chemical terrorism or warfare, nuclear and biological weapons research and its concealment, and ballistic missile defense against these weapons. Such a study determines how to overcome an emerging threat before, during, and after initiation of the threat. Three proposals -- remediation, ground-penetrating radar (GPR), and ballistic missile defense (BMD) -- are presented, implementing our swarm models in order to destroy, prevent, and intercept weapons of mass destruction.

4.1 Remediation

Remediation means the ability to kill or inactivate dispersed biological/chemical agents and/or toxins (Vx, anthrax, etc.) by a controlled thermodynamic combustion process. Such a process can be performed autonomously implementing an intelligent multi-robotic system.

4.1.1 Introduction

Detection, identification, and remediation of biological toxins in dispersed plumes generated by hostile military forces, terrorists, or illicit proliferation activities pose a challenging security problem. Characteristic spectral signatures are transient in nature due to prevailing wind currents, exhibit significant wavelength diversity, and can be disguised by natural obscurants or interferences which make quantitative chemical recognition by conventional optical measurements problematic. The limited field-of-view of laser sources makes stand-off detection and hazard-interdiction over realistic operational areas impractical for countermeasures. An alternative approach is to spatially map the migrating plume using an intelligent constellation of robots deployed with sensitive agent-specific point sensors.

Envisioned in figure 26 is a distributed sensing architecture in which forward-deployed autonomous elements act initially as an advanced warning trip-wire and then act collectively under a coordinating situational awareness umbrella to optimize interrogation and remediation of the sampling space. First, a robotic swarm positions to secure an area. The robots' platforms would be equipped with a sophisticated optical spectrometer capable of measuring anthrax concentration levels at 1 cfu. Second, using advanced intelligent algorithms, the swarm characterizes the plume and coordinates those robots with laser ignition devices. The same laser providing optical detection of the bio-hazard locally stimulates optical ignition of a munition of UAV-deployable chemical energy source (e.g., magnesium dust) which is remotely targeted into the plume. Third, the robots tailor their collective behavior to insure destruction of the hazard by controlled combustion from the plume boundary, indiscriminate dispersal by convection and radial shock waves can be avoided. By manipulating the control state vector, it may be possible to actively orchestrate convective flow mechanics and local combustion kinetics so as to permit segregation of the hazardous agent into a prescribed spatial region or to define optimal placement for fuel-air explosives.

An outline in section 4.1.2 realizes the remediation process, discussing technical issues and providing simulations allowing such a process to occur in real-time. This application concludes

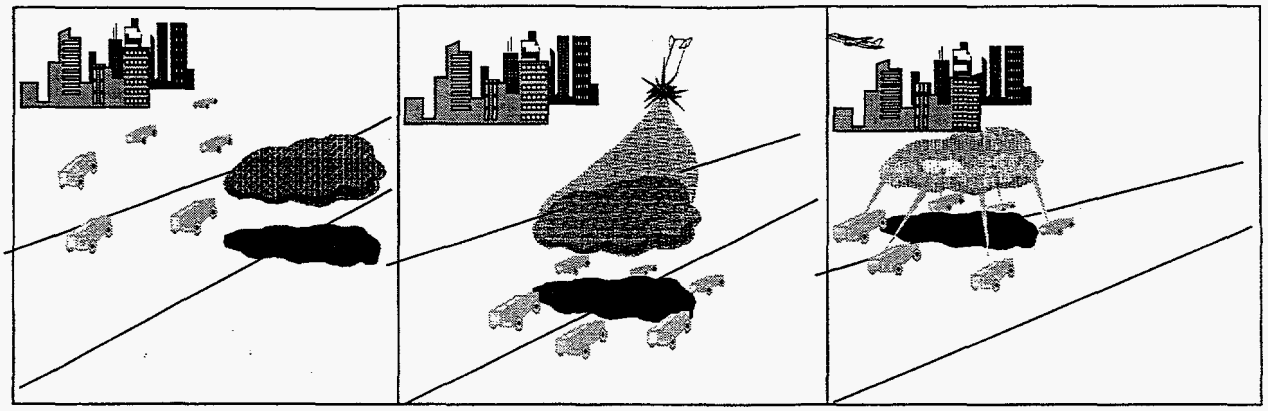


Figure 26. The remediation process. Left picture indicates swarm initially configured in a trip-wire distribution. Middle picture shows swarm adaptively reconfiguring around the perimeter of hazardous effluent and the inter-dispersal of combustible dust via a mortar shell. Right picture has coherent ignition about the circumference, an inward explosion.

with a summary.

4.1.2 Realization of the Remediation Process

The outline of this section consists of the following topics : the spectrometer or bio/chem sensor, the deployment and optical ignition of the munitions dust, the simulations of a heterogeneous swarm adaptively reconfiguring as realized by our physics-based model of a LG, and the discussion of the Monte Carlo simulation process.

Spectrometer. Mounted on each robot is a biological/chemical sensor that is based on surface-enhanced Raman spectroscopy (SERS). This technique leverages previously developed optical-receptor tags which integrate specific molecular recognition sensors using engineered enzymatic or monoclonal antibody transduction. The high-gain resonant Raman process interrogates the immobilized receptor site which acts as a concentrator, and generates a unique fingerprint of the agent complex with modest laser requirements. Therefore, such a design is amenable to miniaturization and allows it to be realized by on-board microchip lasers. Sandia proprietary multivariate chemical analysis tools implementing genetic algorithms determine optimal spectral regions sensitive to various hazardous biologicals and train the neural networks to distinguish between various agents. To date, prototypes have measured sensitivities at the 1 cfu level.

Ignitor. The remediation process requires the dispersement of a UAV or munition-deployable chemical energy source (e.g., magnesium or titanium dust) and the optical ignition of this chemical.

(1) Ignitor Material. The properties of the chemical energy source should be essentially inert, easily dispersed, and releasing an amount of energy necessary to destroy hazardous toxins or biologicals. Magnesium and/or titanium dust fulfills these criteria.

(2) Optical Ignition. On-board microchip lasers or PCSS-enabled diode arrays are used to locally stimulate optical ignition/implosion of the magnesium or titanium dust (right picture in figure 26). Because ignition and combustion methods are expected to be relatively slow compared to plume dynamics, we do not expect timing issues to be critical compared to current tech-

nology. Laboratory tests will be performed calibrating pulse power and duration as a function of ignitor material and dispersal fraction. Due to the chain reaction mechanism, the major dominant variable will be ignition threshold, rendering most parameters insensitive relative to combustion.

Deployment of Ignitor Dust. Different mission scenarios may require specialized ignitor application techniques. Three techniques are envisioned. The first discharges a mortar shell from a plane just above the plume (center picture in figure 26). The combination of instrumenting a GPS devise on a robot and allowing the swarm to emit a beacon effectively draws a "bulls-eye" on the plume for the fighter jet pilot. The second slightly less sophisticated method utilizes mortar shells targeted from nearby ground artillery. The third model realizes a swarm of airborne robots deploying its titanium dust (ignitor) while being guided by the ground-based robots. Such a system would provide an autonomous intercept system. To date, PIC code fighter aircraft simulation dynamics exists, and these codes can be modified to anticipate mission specific scenarios such as remediation.

Autonomous, Intelligent, Multi-Agent Swarm of Robots. Leveraging physics-based models, such as lattice gas (LG) statistical models, quantifies both the robot's collective state and its dynamics but in a novel way achieving multiple objectives. Trained intelligent algorithms, such as artificial neural networks (NNs), assess the environment and autonomously reconfigures the swarm's state (adaptive behavior).

(1) A Trip-Wire Mode and Follow Response. The robot's initial configuration should secure sensitive or highly-populated areas, a trip-wire mode (left picture in figure 26). Activating the sensors allows the swarm to coalesce and follow beneath the effluent which is subject to the prevailing wind patterns. By minimizing the communication over-head and only requiring obstacle detection sensors to maintain the swarm's cohesiveness and orchestrate its movements, the swarm's response time is maximal.

(2) Perimeter and Sensor-Array Distribution. Two "flavors" (types) of robots -- ignitor robots and remote-sensing robots -- must be positioned to monitor and initiate combustion. Realizing the physics-based model for a swarm, the robot positions (i.e., density of states) are controlled by applying two inter-connected pseudo-potential fields. One field is highly-attractive near the edges of the plume, thus attracting the ignitor-flavored robots (center picture in figure 26); and, the second combines a flat and inverse-radial field, thus distributing remote-sensing-flavored robots evenly beneath or inside the plume and sparsely (but not zero) beyond the plume's perimeter. We have shown that certain desired swarm behavioral patterns can be achieved by genetically training (evolutionary algorithms) potential field parameters. As an example, modifying the strength of the local nearest-neighbor forces keeps the swarm cohesive and provides control over macroscopic parameters such as density and speed (e.g., maneuverability) while heterogeneous potentials adaptively redistribute the sensors and ignitors.

Remote-sensing robots image the spectral dynamics of combustion within the plume's region and beyond its perimeter. After the controlled inward-directed combustion, the thermal imaging sensors mounted onto the robot's platform, the swarm measures the thermodynamic spectral images during remediation. Trained neural networks (using Monte Carlo simulation data) determine anomalous (time-series) patterns such as jets, mis-ignited regions, and anticipated motion of plume due to forces of implosion. We expect our image recognition database (various neural network architectures) to correctly identify suspicious patterns and autonomously react to them.

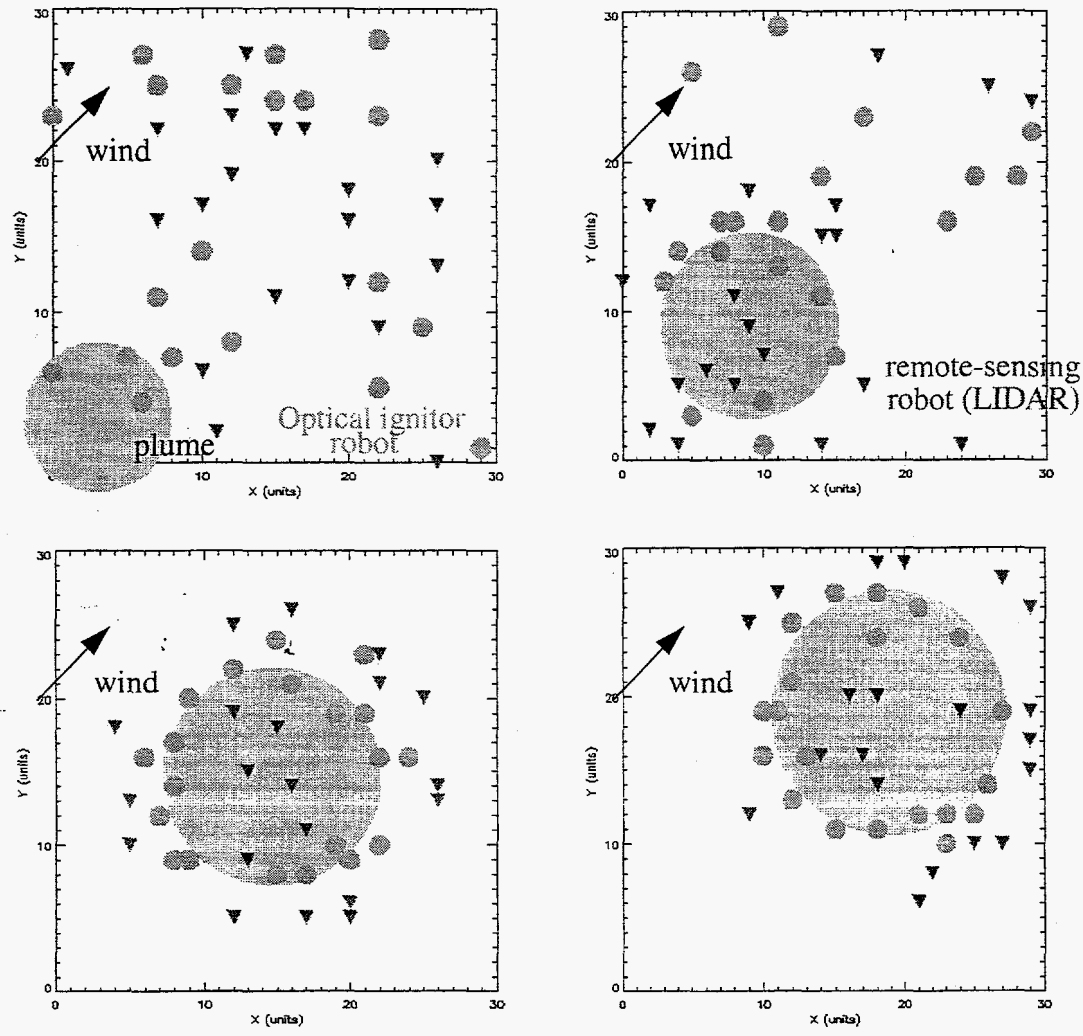


Figure 27. Simulation of a heterogeneous swarm of robots performing the remediation process.

The swarm must reconcile between different forms of sensor imagery types : the plume's chemical and biological makeup and its physical and dynamical properties. Spectrometer results determine hazardous concentration levels. Back-lighting LIDAR techniques measure the physical and dynamical macroscopic properties of the plume, characterizing its direction of motion, speed, and dissipation or expansion. We propose to enhance and expand our intelligent algorithm test-bed to better control the swarm's physical state. The robots will measure and encircle the plume's perimeter by sampling and communicating information at the local level and then perform pattern recognition utilizing the entire sensor array and applying neural networks for global (macroscopic) information. To control such behavior, successful local implementation of recurrent neural networks to decipher a sequence of events (images or patterns) will be expanded to extract global information.

In figure 27 is a 2D graphical display of a simulated swarm following a gaseous effluent that is propagating at a constant speed and angle (45^0) and exponentially dissipating (with a constant of 5%). The time evolution of the ignitor (circles) and remote-sensing (inverted triangles) robots reveal the effects of the two inter-connecting pseudo-potentials. Starting out with a random distri-

bution of robots (top left picture in figure 27), the swarm coalesces onto and propagates with the source (top right, bottom left and right pictures). After time 30, nearly all the robots have condensed onto the target (see figure 28 and between the top right and bottom left pictures in figure 27). Just before this time, the pseudo-potentials are turned-on. The full complement of both perimeter and sensor force-fields allow the heterogeneous swarm to reconfigure and adapt to the plume -- ignitor robots at the edges and remote-sensing robots interior and exterior to the plume. The final plot indicates the steady-state behavior of this distribution as the plume continually expands and propagates.

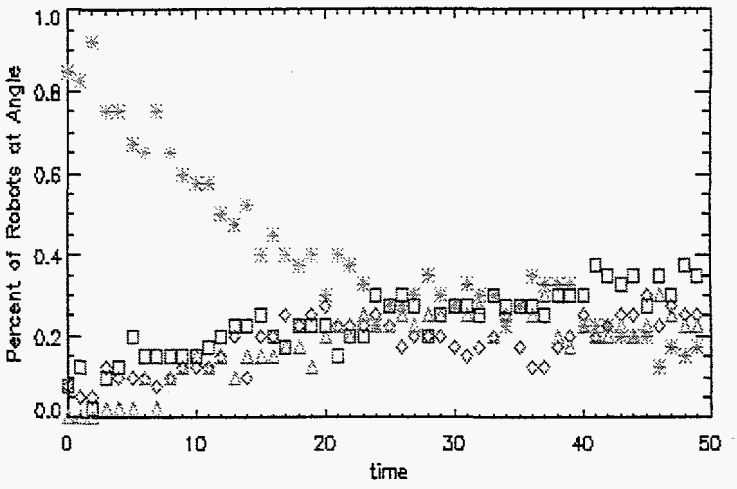


Figure 28. Angular distribution of robots about plume. Asterisk, first quadrant, 0^0 to 90^0. Box, second quadrant, 90^0 to 180^0. Triangle, third quadrant, 180^0 to 270^0. Diamond, fourth quadrant, 270^0 to 360^0. Swarm recognizes when condensation ends and redistribution begins.

(3) Autonomously Adapting to Environmental Conditions and Remediation Verification. The robots correlate and adapt their actions relative to the environment (i.e., terrain) and its influence on the plume (i.e., wind direction). Once collected beneath the plume, reconfiguring the number of robots beyond the plume's perimeter and directly beneath it while it is propagating and expanding can be correlated to the swarm's density state. If the wind direction changes or a large obstacle must be overcome, the swarm adapts remaining in contact, redistributing, and measuring an acceptance level for relative positioning. This is important to account for environmental influences while maintaining optimal signature measurements and final verification.

Environment Factors. Wind-shifts, perimeter expansion, terrain, obstacles are all examples of environmental factors that can hinder or abort remediation attempts can be accounted for through modified force constants (follow), application of pseudo-potentials (pictured to the left), and changing genetically optimized potential and pseudo-potential field strengths (obstacle avoidance). Thus, these real-time adaptive techniques improve the swarm's autonomy and behavior allowing it to perform within a variety of environments.

Verification. Each stage of remediation must be verified as well as the final result. Error or validity-functions determine whether the constituents of the effluent are threatening, when the robots are in place, when the ignitor dust has been applied, whether combustion was self-contained, if environmental conditions changed before, during, or after remediation, and how successful. The latter has not yet been considered. After remediation a search strategy can be employed to scan a region for contaminants. We propose to employ a set of robot instructions that searches the fall-out area and "scrubs" (laser eradication or isolated remediated events) pockets of surviving biological agents.

Monte Carlo Simulations. Existing Monte Carlo atmospheric physics-based analysis codes simu-

late combustion (and convective flow) dynamics, such as turbulence, flares, and shock-waves. These codes will determine three technical issues. First, the technical difficulty would be to inwardly-direct the combustion, confining both the extent of the implosion and its fall-out. GAs in combination with MC codes can determine the phase (coherence), duration, and position of the optical ignitors. Second, it is the images reproduced from these simulations that help train the neural networks to recognize suspicious combustion patterns which the swarm keys during verification. The final technical problem resolved by MC analysis is the correlation response times between plume dynamics and ignition. The swarm's autonomous behavior should produce a fast response times compared to expected plume dynamics. However, in situations where this should become a factor, the sequence recognition capability of the RNNs can be implemented to determine when the dynamics is acceptable for ignition procedure.

4.1.3 Summary

Both state-of-the-art sensor technology and autonomous, adaptive, remote-sensing techniques analyze and assess the emerging threat. Our autonomous, adaptive, physics-based models of a swarm allows added control, stability, and robustness (to counter-measures) by including dynamical physical parameters in the fitness function (to optimize swarm behavior). Also, by training the robot's behavior via intelligent algorithms to overcome natural obscurants and artificial interferences in its pursuit of the emerging threat, its sensor contact with the plume's toxic elements maximizes, increasing its sensitivity to remote-sense trace chemical elements. This proposed, novel technique overcomes many of the difficulties subject to conventional remote-sensing optical stand-off measurements, such as line-of-sight interferences and transient responses. Next, by miniaturizing the spectrometer (Biopraxis CRADA) and combining Sandia patented spectral analysis algorithms, mobile sensors capable of unprecedented sensitivity to detecting minute levels (1 cfu) of chemical and biological warfare molecules can be strategically positioned. Global optimization implementing GA incorporates physics Monte Carlo analysis teach the swarm how to behave in order to control combustion dynamics. Finally, a methodology to simulate asynchronous communication networks and protocol realized by a swarm of intelligent robots realizes real-time neural network imaging capabilities through the use of parallel processing systems validating hazardous extinction.

4.2 Ground-Penetrating Radar (GPR)

Ultra-wideband, short-pulsed phased-array antenna array provides image data in both time and frequency domains enhancing structure identification. Combining these techniques with the adaptability of a reconfigurable, mobile, N-element antenna structure incorporated with intelligent algorithms adds positional information to the (hyperspectral) imagery. Such a system has the ability to find those optimal pathways and to leverage these pathways so as to image man-made structures deployed deep within the subterrain.

4.2.1 Introduction

Detection of deeply buried underground structures in cluttered terrain or urban environments with unknown geological strata is very difficult to accomplish surreptitiously without recourse to indirect detection methods (e.g., remote vibrometry, secondary signatures) prone to

misinterpretation or simple countermeasure. More direct sensing methods such as ground-penetrating backscatter or synthetic aperture radars have offered limited utility for covert applications against subsurface targets because of conflicting practical requirements concerning antenna size scaling versus broadcast frequency content to offset attenuation and dispersion of the electromagnetic pulse in penetrating layered soil media. Because of range losses to the ground surface and impedance mismatch in air, large waveguide antenna geometries with easily detectable microwave emission signatures are required from airborne platforms.

An alternative approach known as a fully adaptive phased array radar, despite many theoretical advantages, has been plagued by the added complexity and longer convergence times that accompany a control loop involving many degrees of freedom. As an example of the new type of cooperative remote-sensing and synchronization problems which can be addressed with a distributed hierarchical architecture of mobile robots operating cooperatively at the local level subject to global optimization criteria, we propose the ad hoc construction of a phased impulse radar aperture in denied territory using an intelligent, self-organizing robotic collective of remotely activated rf pulsers. Each basic radiating element of the distributed grid will consist of an omnidirectional dipole antenna (50-100MHz) combined with a compact optically triggered photoconductive semiconductor switch (PCSS), transmission line, and time-delay circuit or phase-shifter module. Detection will be accomplished using the same receivers in a time-gated cooperative listening mode following the initial impulse to eliminate multi-path dispersion and clutter echoes. Using intelligent algorithms operating synergistically at both the local (repositions, internal timing) and expert level (image quality, resolution), the assembled array can be iteratively optimized in individual phase (synchronization with broadcast laser pulse), amplitude, and spatial orientation ($\lambda/2$ spacing to avoid grating sidelobes) relative to other nearby sensors to create by electromagnetic superposition a directive illumination pattern with modifiable aperture robustly adaptable to local environmental factors such as soil type, dispersion, and anomalous reflections. Since antenna characteristics are determined primarily by the geometric position of the radiators and their relative amplitude and phase excitation, the outgoing radiation pattern would exhibit controllable sidelobe reduction for minimizing clutter and jamming by external noise sources. Active beam steering and the creation of shaped transfer functions as part of an adaptive control loop to maximize signal-to-noise ratio (SNR) could be used to correct for sensing errors, crosstalk, and accommodate building shadowing or constrained search modes. Therefore, beam manipulation and signal management would be orchestrated locally by sub-arrays. This design concept has the advantage of being amenable to covert infiltration and compact packaging, and can be overlaid with other autonomous ground sensor measurements including co-registered range-resolved optical imagery and passive extoreceptive signatures. Laser vibrometry when combined with change algorithms, novelty filters, and frame-to-frame difference processing could detect vibrations of natural specular reflections as corroborating evidence. Ongoing Sandia research has developed optically triggered high-gain GaAs semiconductor switch technology as impulse sources for UWB ground-penetrating radar transmitters[18,19].

In the following sections, a GA controlled simulated phased-array antenna structure probes a various subterrain profiles. The GPR section concludes with a summary or our work.

4.2.2 Simulations

The setup used in these simulations is shown in figure 29. The subterrain's cross-sectional area covers 1500 cm by 1500 cm. Each of the swarm's 8 robots transmits a single 15 cm wave-

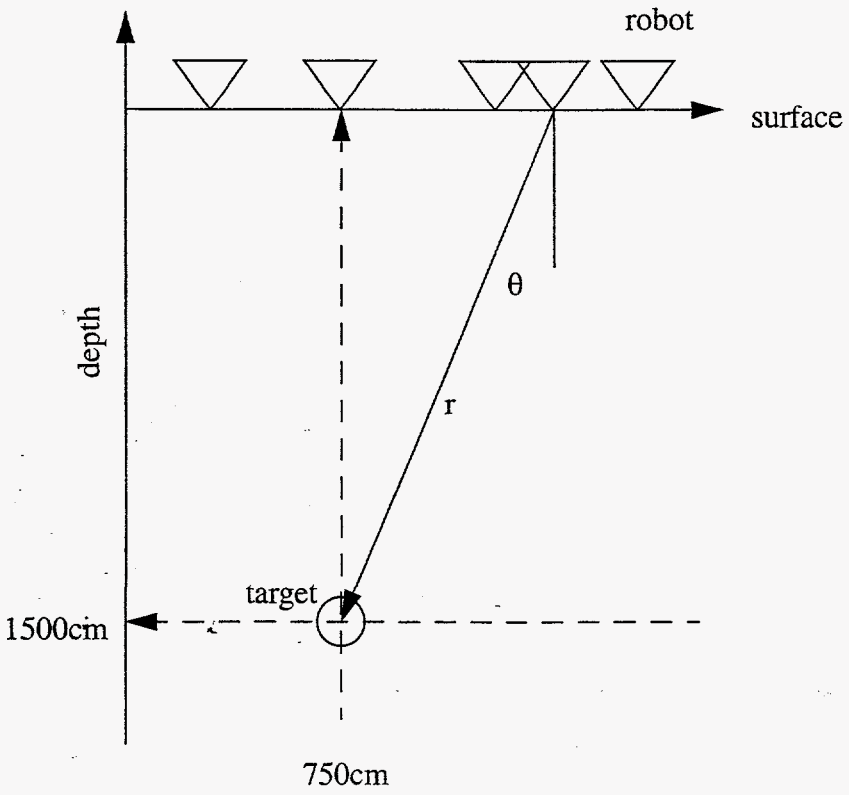


Figure 29. Schematic diagram of the adaptive phased-array structure for simulation tests.

length monocycle cosine pulse. This simulates an end-fire antenna radiation profile similar to the Yagi-Uda antenna in experimental section. The target resides at a depth of 15.0 m and an x-position of 7.5 m and measures the total transmitter signal amplitude reaching it. Measured relative to the depth (or y) axis, the angle, θ, indicates the wave's direction, which is related to the time delay or phase of the transmitted pulse.

The signal's path (or ray) depends on the robot's position relative to the source and the relative phase (to those of the swarm) of its transmission. These parameters depend on the subterrain's dielectric cross-sectional profile. Dielectric materials affect the signal amplitude and pathlength or angle of transmission, θ. Ignoring dispersion, Snell's Law, the subterrain's profile defined by their index's of refraction, and the robot's positions form a transcendental equation in θ. Numerical techniques (Newton's method) solve for each robots' transmission angle defining the shortest path through the subterrain to the target. Therefore, each subterrain profile requires a different transcendental equation to solve. Except for the simple case of a single dielectric material with an index of refraction, n, of 1.0, three other subterrains (or test cases) are simulated : a horizontal dielectric layer (figure 30), a vertical dielectric layer (figure 33), and a horizontal layer with an opening (figure 32). A genetic algorithm determines the swarm's optimal phased-array structure.

Optimizing the adaptive phased-array structure provided by the swarm has the GA maximizing the total signal reflecting off the target. Scattering and signal reception will be done later, only the signal at the target is considered here. The parameters controlling the phased-array are a

robot's position and time-delay (tDelay) or phase of the transmitted pulse. These are the GA's genes, 16 in total since there are eight robots. After a population of 32 swarms with random phases and positions are produced, the three GA operations are performed on this population : (1) selection, (2) cross-over, and (3) mutation. The tests performed in this section implement a roulette-wheel selection technique (with one or two elitists), one-point crossover method (with 90% probability), and a flat, random mutation profile (set at 5% of all genes).

The fitness function used in the GA depends on the transmitted pulse's amplitude and phase. The amplitude is determined by the cosine function,

$$residAmp = 2 \cdot Nbots - \sum_{i}^{Nbots} \cos\left(\frac{(2\pi)}{\lambda} \cdot (r + c \cdot tDelay)\right), \qquad (18)$$

and the relative phase of the swarm is defined as

$$residPhase = \sum_{i}^{Nbots} phase(robot(1)) - phase(robot(i)), \qquad (19)$$

where

$$phase(r, tDelay) = r + c \cdot tDelay \,. \qquad (20)$$

As before, the fitness function is proportional to the inverse of both the amplitude and phase residuals,

$$fitness = \frac{1}{(0.1 + residTotal)} \,, \qquad (21)$$

where

$$residTotal = \alpha \cdot residAmp^{p} + \beta \cdot residPhase^{q} \qquad (22)$$

In the above equation, α and β are set to 1.0, and p and q are set to 4. This normalizes the fitness function so the GA converges toward a value of 10. To prevent calculating fitnesses from small values, a final normalization scheme of the signal's amplitude at the target is implemented :

$$Amplitude \rightarrow \frac{Amplitude}{\left(\frac{r}{dTgt}\right)} \,, \qquad (23)$$

where the target's depth, dTgt is 15.0 m.

The results of the GA produce a time delay with an arbitrary offset since residPhase of equation 19 depends on the phase of the first robot (by definition). To determine the minimal time delay, the term

$$residtDelay = \sum_{i}^{Nbots} tDelay_i \qquad (24)$$

has been added to equation 22 with a strength of 1 and an exponent of 4 (as well as 1 and 2). For two different runs, the phases of the robots converge to the values 1815 cm and 1680 cm, an integer number of wavelengths. In addition, convergence is longer, from approximately 100 to 600 generations. It appears the basin of the fitness function representing the optimal solution is somewhat flat and would require annealing techniques in order for the GA to converge toward the global minimum. A real-time application would benefit from such an improvement. Incorporating additional physics-based equations and imaging analysis tools into the fitness function will probably dominate convergence. At this time, it does did not make sense to continue using the additional term of equation 23 to remove the timing offset; therefore, this has been dropped.

Test Case One. In figure 30, the subterrain consists of a horizontal layer of dielectric material (n=2) ranging in depth from 5 to 10 m (shaded region). The signal amplitude within this material degrades according the measured attenuation factor for igneous rock (~1 dB/m). For the above mentioned physical cross-section, the amplitude degrades by a factor ranging from 0.15 to 0.35 depending on the path length. By renormalizing the amplitude in equation 24 to account for these additional losses, the fitness maintains an optimal convergence value of 10.

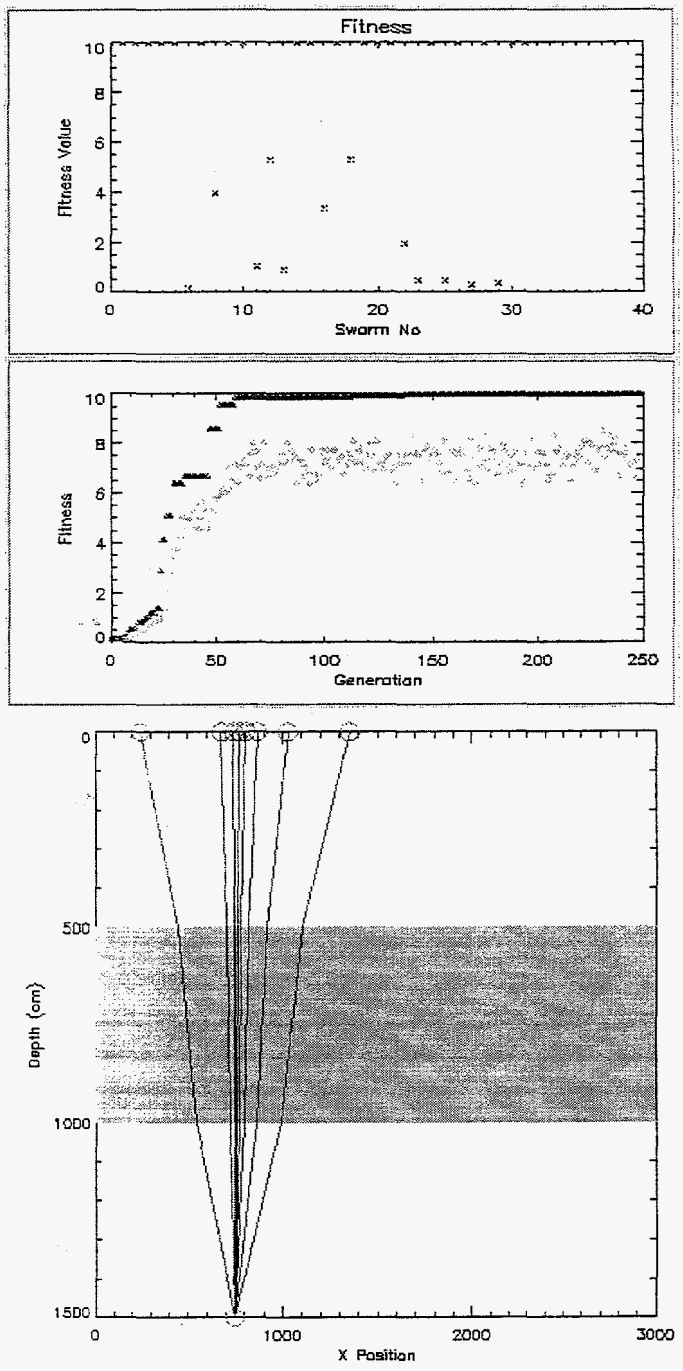


Figure 30. GA phase-array results for a subterrain consisting of a horizontal layer of dielectric material. Top plot, fitness for each swarm in the population and the maximum and average fitness for each generation. Bottom plot, final array configuration and signal path.

Initially, the swarm positions itself in the range from 2750 to 3000 cm (far right in diagram). Their positions and phases are concurrently optimized by the GA, the results from which are shown in figure 30. Plotted in the top two graphs are the final fitness distribution for the population of swarms and the population's maximum and average fitnesses during the evolution process. Because these plots are produced after mutation is applied, the effect of this 5 percent degrades the average fitness keeping it from plateauing at 10. The bottom display is a picture of the optimal

GA-determined solution.

Intuitively, one would expect the optimal solution having all the robots transmit their signal from a position directly above the target. This is essentially what happens (bottom picture in figure 30). Even though the magnitude of the signal at the target does depend on the total distance from each robot, it is a weak relationship. Once above the target's location and the extent of the swarm is small relative to depth, the superposition of the amplitudes changes by less than 1 percent. The contribution of the amplitude, residAmp, in the fitness function is no longer dominant, equation 18, leaving the phase term as the more sensitive parameter.

Adding the dielectric material focuses the electromagnetic waves, modifying the angle, θ, and therefore the phase, tDelay, and reduces the wave's amplitude. This focusing causes the path length through the dielectric is nearly independent of the angle, rays are nearly parallel (see figure 30). If one plots the path length, the angle, and the propagation time to the target for different robot positions relative to the x-position of the target (i.e., transcendental equation), it shows that for small distances, the curves are either flat (path length and propagation time) or linear (angle). This flattens and widens the fitness function's phase-space near the optimum solutions, explaining why the GA has some difficulty converging (slowly) to the absolute minimum. Thus, relatively speaking, the index of refraction has a small effect on the angle and path length curves, but a large effect on the propagation times (i.e., time delays). Relative timing is important.

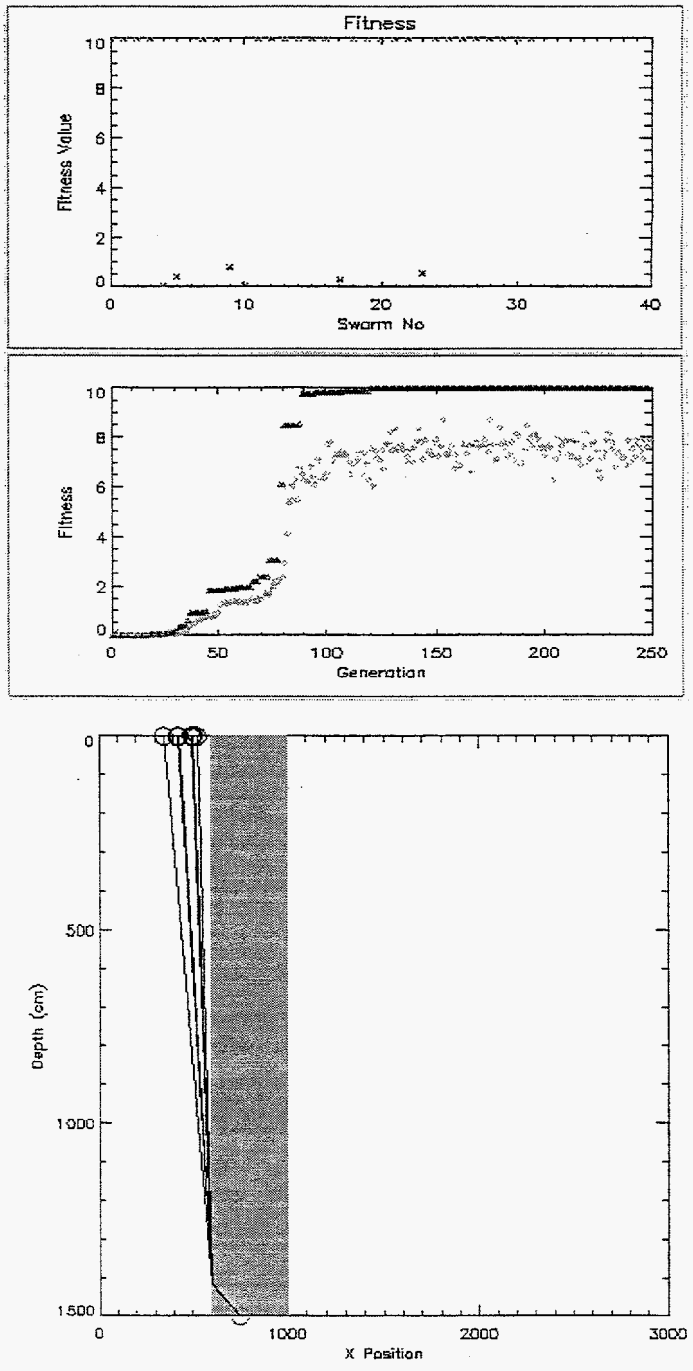


Figure 31. GA phase-array results for a subterrain consisting of a vertical layer of dielectric material.

Test Case Two. Instead of restricting the dielectric (n=2) in depth, it is allowed to occupy a band in x surrounding the target -- 600 cm to 1000 cm (see figure 31). Results are plotted in figure 31. The maximum path length through the dielectric material would degrade the signal's amplitude by 96 percent. The optimal position of the swarm is shown in figure 31, where the GA moves the

swarm to the side with the smaller dielectric barrier. As before, the dielectric profile appears to be the denominating factor. As for test case one, a series of plots relating the angle, the path length, and the propagation time to the robot's relative position from the target indicates an effect essentially opposite to that of a horizontal profile. The farther the robots are from the target, the smaller the difference between the path length and the angle (phase) for various indexes of refraction, excluding propagation time. Figure 31 reveals that focusing all the energy of the robots at a single point near the target may be sufficient to achieve an image of the target. However, these positions will change because the scattering energy must propagate through to the opposite side of the dielectric. To optimize target detection and image quality which depends on θ, the sensitivity of the transmitted and received signals will probably be sensitive to the slope of the curves produced by the transcendental equations which are linked to the subterrain profile. Incorporating such terms into the fitness function will help determine swarm behavior that would enhance resolution.

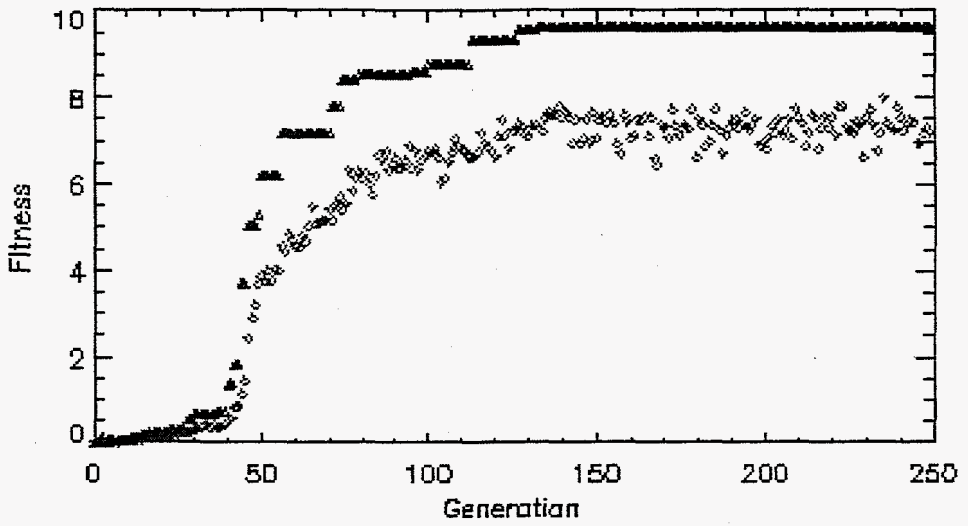


Figure 32. Plot of the fitness as a function of generation.

Test Case Three. The final simulation considers a hole, a minimal resistant pathway, into a horizontal strata of dielectric material (n=2). As pictured in figure 32, the hole spans in x from 5 to 8 m. The evolution of the population's maximum (triangles) and average (diamonds) fitnesses is shown in figure 32 as well as the optimal GA-produced ray-tracing solution after 5, 25, 50, 100, and 200 generations. The results indicate a dichonomy between positional and relative phase convergence, where the first 50 generations position the swarm above the hole and the next 100 generations optimize the relative phases of the transmitter pulses.

Conclusions. These three test cases provide insight into how a GA optimized adaptable phased-array antenna can determine an optimal path through the subsurface, avoiding or minimizing the effects of large dielectric barriers. One, a horizontal dielectric barrier focuses the electromagnetic waves and degrades the amplitude of all transmitters equally. Two, a vertical cross-section of dielectric material causes the swarm to behave quite differently, probing a target from the sides of the dielectric. Three, a hole in the horizontal strata indicates a two stepped process : amplitude measurement dominates the swarm's global position (and distribution) and then local or phase optimization improves and refines image processing of the target structure. The choice depends on the geometry of the subterrain. Enhancing the amplitude mode can be accomplished by pulsing the monocycle signal for extended periods of time. Here, a stochastic (a GA) method may

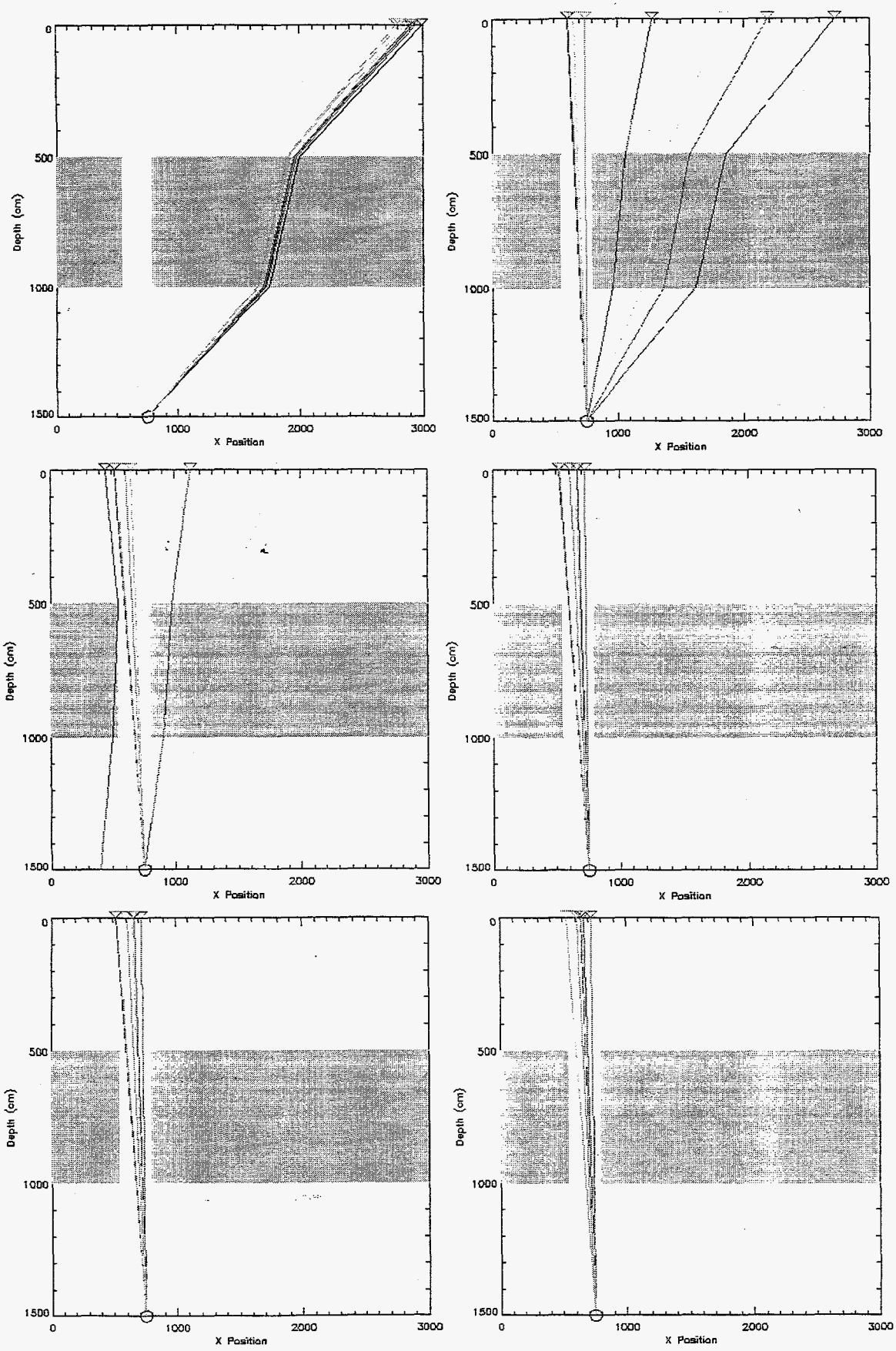


Figure 33. The solution space of the GA after 0, 5, 25, 50, 100, and 200 generations.

converge fastest. Local phase refinement (implementing monocycle pulses only) may converge more quickly implementing gradient search methods in conjunction with local substrata and target (structural) imaging based on ultra-wideband, short-pulsed EM techniques.

To recapitulate, initial analysis help better understand what abilities a swarm of robots require in order to adaptively reconfigure a phased-array antenna. By implementing intelligent algorithms, such as GAs and ANNs, optimal signal transmission and swarm distribution can be taught to quickly find (converge) optimal pathways which can enhance structural GPR imagery.

4.2.3 Summary

Implementing radar techniques to probe an unknown (possibly changing) dielectric medium is a challenging problem requiring novel techniques. By finding optimal imaging pathways and artificially intelligent analysis techniques, it may be possible to probe deeper and image clearer beneath the subterrain for structures. The technique discussed in this section implements an autonomous, adaptive phased-array antenna realized by a swarm of robots.

From the analysis thus far, an iterative approach is hypothesized. The robots fitted with antennas reconfigure into groups of different densities, their signals phased in such a way so as to probe a certain layer of depth over a certain area. Ultra-wideband, short-pulsed radar techniques implemented over the spatial distribution of the swarm provide 3D (hyperspectral) images (position-time-frequency) outlining the profile of the subterrain. Analyzing this imagery using the swarm's global neural network intelligence, the genetically trained swarm reconfigures along optical pathways increasing penetration depth. It is this basic loop where the GA optimizes the swarm's ability to reconfigure and adapt to various subterrains by interpreting the GPR images (as well as its ability to maneuver on the surface). Some of this work will occur on-line, requiring fast convergence rates. From the simulations, hybrid-optimization schemes improve convergence rates by matching subterrain profiles to algorithmic techniques (i.e., artificially intelligent swarm).

4.3 Ballistic Missile Defense

4.3.1 Introduction

To date, the United States Ballistic Missile Defense (e.g., the PATRIOT) is dominated by a single sophisticated rocket intercepting a launched missile (e.g., a SCUD) from a hostile country or terrorist organization. The difficulty in shooting down a missile has been compared to intercepting a shot bullet by shooting it at it. In contrast, a flock of simple airborne UAVs possessing collective intelligent behavior has a greater capacity to intercept a moving target. As a result, the flock's behavior reconfigures, developing an interconnected web that drastically increases the missile intercept cross-section. Realistic dynamics are possible through the implementation of physics-based particle-in-cell (PIC) codes, see sections 2.1.3 and 3.3.2. Measured or well-understood environmental conditions effecting flight dynamics are represented by potential fields. UAV aerodynamic parameters, their nearest-neighbor influences, and the entire flock are not only be simulated but optimized using evolutionary programming techniques when considering a robot's design and its collective state. Heterogeneity may be required if a variety of payloads and multiple strikes are needed to missile intercept and detonate different missiles or their launch position. But it is the swarm's ability to learn that assures high performance within an operational theatre where countermeasures must be overcome. All of these considerations can be simulated in real-

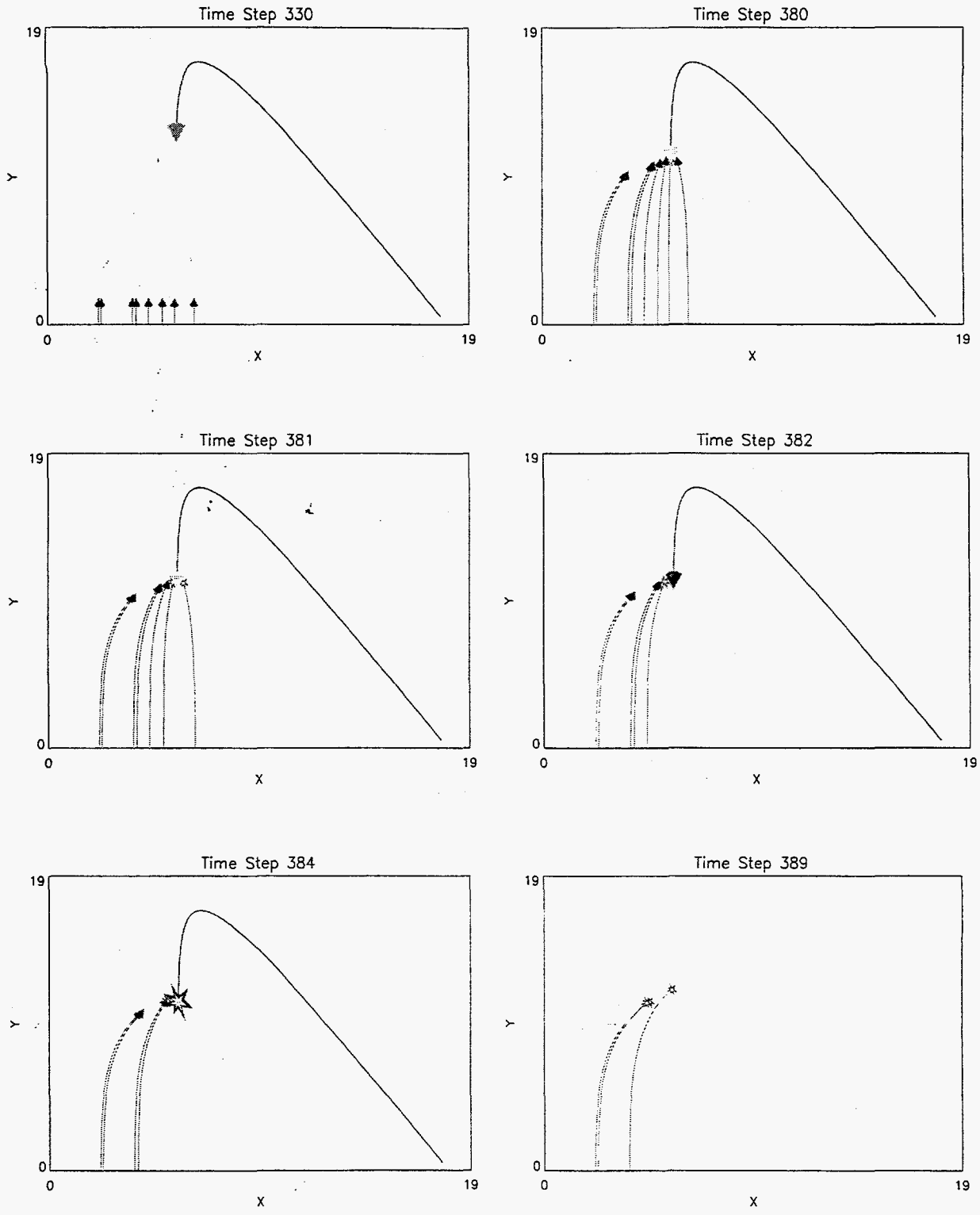


Figure 34. Flocking behavior of airborne UAVs when intercepting a missile.

time due to the reduced processing time afforded by PIC algorithms.

4.3.2 Simulations

The simulation results for the interception of a missile implementing a flock of eight airborne UAVs are shown in figure 34. At time step zero, a missile is launched. Eventually, all of its fuel is spent and it begins to descend, around time step 300. At time step 330, a swarm of airborne UAVs sense the missile's position, by radar, and begins to condense toward the target. At time step 380, segments of the swarm reach the missile. Time steps 381 through 389 show the swarm multiply striking the missile detonating its payload.

Gravitational forces dominate the missile's trajectory. The use of PIC codes have a long and successful history when simulating dynamics in gravitational and electrodynamical potential fields, i.e., astrophysics and plasmas. Simulations of the aerodynamics not only include the energy spent by the fuel to propel the missile but also the drag and friction forces due to the atmosphere. Also, the UAV's flight dynamics affect each element of flock onto one another (e.g., wake) which is also dependent on how fast, how close, and how different maneuvers effect the flock (e.g., pitch, thrust). All of these forces -- the flocking force, the flight dynamics, and the attractive potential due to the missile's presence -- are distributed onto the PIC code's mesh. Due to the computational technique of PIC codes, these additions add minimally to the overall calculations providing real-time simulation results.

4.3.3 Conclusions

The dynamics of BMD techniques implementing swarms of airborne UAVs can be simulated in real-time. Leveraging our intelligent algorithms, a network of these UAVs has the ability to increase missile intercept probabilities and overcome counter-measures.

4.4 Optical Communications

Laser optical communication potentially offers significant advantages for remote coordination and data exfiltration from covert distributed ground-sensing networks. The spatial coherence of laser transmitters offer favorable gain scaling with low probability of intercept, and the relative temporal coherence of the optical carrier can support tremendous information bandwidth without baseband interference or frequency allocation problems. A major weakness limiting the operational utility of nonideal optical communication channels propagating in atmosphere, however, has been adverse effects of extinction (loss), scattering (dispersion), turbulence (degraded coherency), and fade which degrade realizable transmission bandwidth and gain aperture for acceptable bit error rate. We have investigated the use of a previously developed active reflectance imaging technique based on an optical parametric amplifier (OPA) receiver[20] to enhance detector sensitivity and error rate performance for unguided digital communication links affected by cloud-like conditions. Using a kilohertz repetition rate femtosecond laser system operating at eyesafe wavelengths, we have evaluated the role of signal-spontaneous OPA beat noise (s-ASE) on amplified signal, noise figure, and channel sampling capacity for various binary modulation formats in both direct and coherent detection modes to establish fundamental response limitations as a function of turbidity.

The optical communication and laser vibrometry designs are synergistic requiring a single

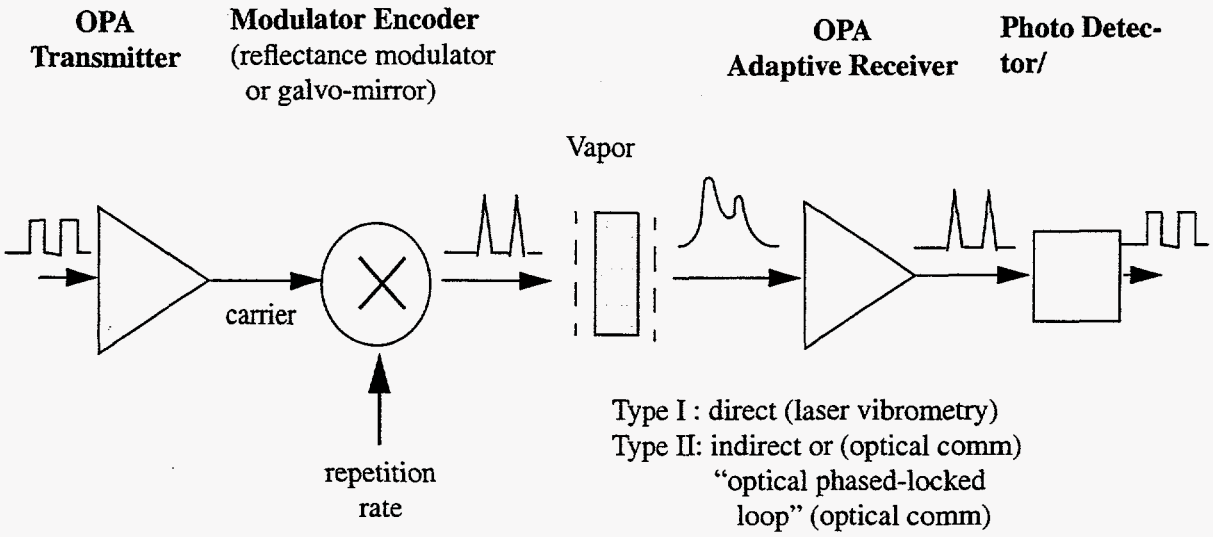


Figure 35. Test setup of an optical communications pathway.

experimental platform. In figure 35, an experimental block diagram shows signal amplification and modulation in preparation for transmission, scattering material, and an OPA adaptive receiver reconstructing the original data stream. Binary data is first amplified by an OPA (120 fs widths), modulated by a carrier, and transmitter through an attenuator (e.g., vapor). Variations of the OPA concept phase-matched near the degenerate point are expected to have cross-functional benefit to remote laser vibrometry (a precursor to sensor placement) and noiseless quantum amplification for robust communication/imaging links. Future work will incorporate multiplexing in time and frequency to access terahertz bandwidths. An increased bandwidth in combination with fast optical filter techniques allows for pseudo-random encryption methodologies securing data pathways.

5 Summary

The major theoretical achievements are the development of two physics-based collective behavior simulation codes, the development of parallelized genetic algorithm for optimization of behavioral characteristics, and new results focused on local/global coordination of collective behavior and adaptive reconfiguration of swarms looking at plumes and underground structures. Two new physics-based collective models -- a lattice gas (LG) and a PIC (plasma-based) codes -- model collective behavior, such a system has the advantage of being a well-benchmarked physics design tool. Each robot's potential fields modeled by the LG or PIC simulations provide fast, tripwire responses to an immediate target or event, point sensor information/transformation for further processing (target vector direction implementing recurrent neural networks), application of pseudo-potential fields (linked to genetically-trained neural network decisions) to adapt to local disturbances, and global imaging-like capabilities to study and respond to the environment. Development and implementation of a parallel GA on top of collective behavior models optimizes spatial- and time-dependent nearest-neighbor potential interactions that realized adaptability in the swarm's performance -- obstacle avoidance, ground-penetrating radar, and remediation. Behavioral performance has also been studied versus other physical parameters, such as friction and drag during flight dynamics, to determine swarm behavioral changes for the ballistic missile (PIC) defense. As a result, successful simulations implementing a heterogeneous swarm internally reconfigure while characterizing a wind-blown plume demonstrates autonomous execution of the remediation process and global positioning of a phased-array antenna is capable of finding the best imaging pathways through a layered dielectric subterrain to a target.

Laser optical communication potentially offers significant advantages for remote coordination and data exfiltration from covert distributed ground-sensing networks. We have investigated the use of a previously developed active reflectance imaging technique based on an optical parametric amplifier (OPA) receiver to enhance detector sensitivity and error rate performance for unguided digital communication links affected by cloud-like conditions. Using a kilohertz repetition rate femtosecond laser system operating at eyesafe wavelengths, we have evaluated the role of signal-spontaneous OPA beat noise (s-ASE) on amplified signal, noise figure, and channel sampling capacity for various binary modulation formats in both direct and coherent detection modes to establish fundamental response limitations as a function of turbidity. Incorporating phase II OPA techniques ("optical phase-locked loop") are expected to minimize error rates and provide realizable optical transmission capabilities.

6 References

[1] K. Huang, Statistical Mechanics, John Wiley & Sons, (1987)

[2] J. Hardy, O. de Pazzis, Y. Pomeau, Molecular dynamics of a classical lattice gas: Transport properties and time correlation, Physical Review A, Vol 13, No 5, (1976)

[3] U. Frisch, B. Hasslacher, Y. Pomeau, Lattice-Gas Automata for the Navier-Stokes Equation, Physical Review Letters, Vol 56, No 14, (1986)

[4] J. Jackson, Classical Electrodynamics, John Wiley & Sons, (1975)

[5] C. Reynolds, Flocks, Herds, and Schools: A Distributed Behavioral Model, Computer Graphics, Vol 21, No 4, (1987)

[6] C. Birdsall, A. Langdon, Plasma Physics Via Computer Simulation, McGraw-Hill, (1985)

[7] A. Appel, An Efficient Program for Many-Body Simulation, SIAM J. Sci. Stat. Comput., Vol 6, pp. 85-103, (1985)

[8] J. Wagner, J. Poukey, G. Leifeste, BUCKSHOT-MAGIC Comparison for RADLAC Conditioning Cell Parameters, SAND88-1119, (1988)

[9] J. Wagner, M. Mazarakis, BUCKSHOT Simulations of Beam Injection on the Recirculating LINAC, SAND89-0523, (1989)

[10] Hockney, R. W., and Eastwood, J. W., Computer Simulation Using Particles, McGraw-Hill, New York, 1981

[11] Tajima, T., Computational Plasma PHysics -- with Applications to Fusion and Astrophysics, Addison-Wesley (Benjamin Fontier Series, Reading, MA, 1989)

[12] J. Hertz, A. Krogh, R. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley Publishing Company, (1991)

[13] S. Haykin, Neural Networks: A Comprehensive Foundation, Maxwell MacMillan, (1994)

[14] D. Goldberg, Genetic Algorithms, Addison-Wesley, (1989)

[15] J.S. Wagner, M.W. Trahan, W.E. Nelson, G.C. Tisone, B.L. Preppernau, How Intelligent Chemical Recognition Benefits from Multivariate Analysis and Genetic Optimization, Computers in Physics, Vol. 10, No. 2, 1996

[16] K. M. Stantz, C. Diniz, M.W. Trahan, J.S. Wagner, Character Recognition Using Genetically Trained Neural Networks, SAND98-2145, (1998)

[17] G. Loubriel, J. Aurand, M. Buttram, F. Zutavern, W. Helgeson, M. O'Malley, High Gain GaAs Photoconductive Semiconductor Switches for Ground Penetrating Radar, Presented at the 1996 22nd International Power Modulator Symposium, Boca Raton, FL (June 1996)

[18] G. Loubriel, M. Buttram, J. Aurand, F. Zutavern, Ground Penetrating Radar Enabled by High Gain GaAs Photoconductive Semiconductor Switches, Ultra-Wideband Short Pulse Electromagnetics 3, Proceedings of the 3rd International Conference, Albuquerque, NM (May 1996)

[19] M.W. Trahan, J.S. Wagner, K.M. Stantz, P.C. Gray, R. Robinett, Swarms of UAVs and Fighter Aircraft, The International Conference on Nonlinear Problems in Aviation and Aerospace, submitted, (April 1998).

[20] E. Desurvire, Erbuim-Doped Fiber Amplifiers: Principles and Applications, Wiley-Interscience Publication, (1994)

Internal Distribution:

10	MS 1188	Keith M. Stantz, 9512
1	MS 1188	Stewart M. Cameron, 9512
1	MS 1188	Michael W. Trahan, 9512
1	MS 1188	John S. Wagner, 9512
1	MS 1188	Roy Hamil, 9512
1	MS 9018	Central Technical Files, 8940-2
2	MS 0899	Technical Library, 4916
2	MS 0619	Review & Approval Desk, 15102 For DOE/OSTI
1	MS 0188	LDRD office (Donna L. Chavez)