Extended Virtual Spring Mesh (EVSM): The Distributed Self-Organizing Mobile Ad Hoc Network for Area Exploration

Kurt Derr, *Member,* IEEE, Milos Manic, *Senior Member, IEEE*

Abstract–Mobile Ad hoc NETworks (MANETs) are distributed self-organizing networks that can change locations and configure themselves on the fly. This paper focuses on an algorithmic approach for the deployment of a MANET within an enclosed area, such as a building in a disaster scenario, which can provide a robust communication infrastructure for search and rescue operations.

While a virtual spring mesh (VSM) algorithm provides scalable, self-organizing and fault tolerant capabilities required by a MANET, the VSM lacks the MANET's capabilities of deployment mechanisms for blanket coverage of an area and does not provide an obstacle avoidance mechanism. This paper presents a new technique, an Extended Virtual Spring Mesh (EVSM) algorithm that provides the following novelties: 1) new control laws for exploration and expansion to provide blanket coverage, 2) virtual adaptive springs enabling the mesh to expand as necessary, 3) adapts to communications disturbances by varying the density and movement of mobile nodes, and 4) new metrics to assess the performance of the EVSM algorithm. Simulation results show that EVSM provides up to 16% more coverage and is 3.5 times faster than VSM in environments with 8 obstacles.

Index Terms—Mobile ad hoc network (MANET), robot, self adaptive, self organizing, swarm, unmanned autonomous vehicles, virtual spring mesh network, wireless sensor network

I. Introduction

Distributed, self-organizing networks MANETs (Mobile Ad hoc Networks) can be formed by various entities such as micro air vehicles [1], millibots [2], nanobots [3], micro-electromechanical systems, micro satellites, underwater vehicles [4], ground vehicles [5]-[6], or unmanned autonomous vehicles (UAVs). MANETs can be used in variety of ways, ranging from communications relaying to sensing purposes as intelligent, distributed Wireless Sensor Networks (WSNs). Some of these communication and sensing applications include critical infrastructure protection, search and rescue operations, military reconnaissance, minesweeping, communications infrastructure deployment on a battlefield in advance of military personnel, detecting and tracking both discrete and diffuse targets, surveillance, border monitoring, disaster recovery, and others. These applications are inherently dynamic and distributed and rely on maintenance of communications connectivity. UAV swarms are also dynamic and distributed and require properties such as scalability, distributed operation, self healing, and fault tolerance.

The traditional virtual spring mesh (VSM) algorithm [7]–[9] exhibits many of the properties necessary to address these dynamic problems. These properties are important in various contexts such as self adaptive security solutions, denial of service detection, eliminating single points of failure in wireless routing of static mesh networks [10], and artificial immune systems for computer security [11]–[13]. However, the traditional VSM algorithm does not address the following problems: 1) exploration of unknown areas; 2) blanket communications/sensing coverage of unknown areas; 3) obstacle avoidance. Further, current VSM literature does not provide metrics for effectiveness assessment of the algorithm, namely: 1) coverage effectiveness; 2) formation effectiveness, and 3) connectivity effectiveness.

The Extended Virtual Spring Mesh (EVSM) algorithmic approach presented in this manuscript extends the VSM algorithm with several enhancements. These enhancements involve new algorithmic techniques that enable a swarm of robots to explore unknown areas using wireless communications, avoid fixed obstacles of arbitrary shape, and provide blanket communications/sensing coverage. As demonstrated by examples in sections VI and VII, the approach is fault tolerant and scalable to any number of robots.

The rest of the paper is organized as follows. Section II presents related work, Section III discusses the theory behind the virtual spring mesh algorithm, Section IV describes the Extended VSM algorithm in wireless sensing environments, Section V presents an analysis of the VSM algorithm, Section VI discusses the results of EVSM test cases in a simulation environment, section VII presents the results of test cases in a noisy wireless simulated environment, and section VIII presents our conclusions and future work.

II. Related Work

Howard [14] has developed a greedy and incremental approach to the deployment of a mobile sensor network. Nodes

Manuscript received January 14, 2010. Accepted for publication February 18, 2011.

Kurt Derr is with the Idaho National Laboratory, Idaho Falls, ID 83415 USA (e-mail: kurt.derr@inl.gov, or kdiddm@yahoo.com).

Milos Manic is with the University of Idaho, Idaho Falls, ID 83402 USA (e-mail: misko@uidaho.edu).

are deployed one at a time and must retain line-of-sight with one another. A node determines optimal location by using the data from previously deployed nodes to calculate a location that will maximize sensor network coverage. EVSM, on the other hand, 1) uses range and bearing information from local neighbors only to determine the next position to move to, and 2) is adaptive and will reconfigure and change node position in the vicinity of a failed node.

The Virtual Force Algorithm (VFA) [15] is a grid-based approach (discrete coordination system) that uses a combination of attractive and repulsive forces for effective deployment of sensors. Sensors do not move during the execution of the VFA algorithm, rather a cluster head computes their positions and transmits the new locations to the nodes. EVSM on the other hand is a completely decentralized deployment approach. Each EVSM robot in the swarm calculates their position based on the position of their neighbors and obstacles. No cluster head or centralized resource computes robot positions. VFA deployment strategy is for sensor nodes to not have overlapping coverage, while the EVSM strategy is to provide overlap to ensure there are no gaps in coverage. Also, EVSM is based on a continuous coordination system, unlike VFA, where sensors can be located anywhere in the area of interest (AOI). Some shortcomings of VFA are 1) the sensor density cannot be guaranteed, 2) sensor nodes may not stop moving, and 3) sensor nodes may move out of the region of interest.

IVFA and EVFA algorithms [16] improve upon the shortcomings of the VFA. IVFA sets a maximum movement per iteration, prevents nodes from moving out of the region of interest, and incorporates an effective communications distance measure into the force equations to assist the wireless sensor network in reaching a steady state. EVFA provides additional improvement in achieving steady state by decreasing the virtual forces between nodes exponentially with distance.

Other approaches use triangular deployment strategies as does EVSM. In [17] nodes are initially deployed in the center of some area of interest at a compound area and spread out to provide disk shaped coverage of the area. EVSM does not require sensor nodes to be centrally deployed within some AOI. Sensor nodes may be located together within any compound area of some AOI and will spread out to provide coverage of the entire area.

III. Virtual Spring Mesh (VSM) Algorithm

Virtual spring mesh (VSM) algorithms are a control mechanism by which large numbers of robots can form a distributed robotic macro sensor [7]-[8]. Swarm based technologies, such as the VSM, have been applied in multiple application domains [18]-[19]. Our distributed mobile ad hoc network exploration algorithm with wireless communications is an extension of the VSM concept.

A spring mesh is an undirected graph where the nodes represent mobile entities and the edges are the spring connections [7]. This paper considers the mobile entities to be robots. A spring formation algorithm creates spring connections among self selected pairs of nodes. The VSM algorithm applies virtual forces to the spring connections. Virtual springs have similar properties to mechanical springs; i.e., a natural length, l_o, and a spring constant, or stiffness, k_s. If the robots are too close to one another or far from one another, the virtual spring will act to push them apart or pull them together, respectively.

The control law in (1) defines the virtual forces and rules of motion for the swarm robots [9]. Error is defined in (2) as the difference between the actual spring length and the natural spring length.

$$\ddot{x} = \left[\sum_{i \in S} k_S (l_i - l_0)\hat{u}\right] - k_d \dot{x} \quad (1)$$

$$E = l_i - l_0 \quad (2)$$

where $\dot{x}$ is the velocity of the robot, k_d is the spring damping coefficient, $\hat{u}_i$ is the unit vector between robots connected through a virtual spring, l_o is the natural spring length, l_i the current length of the i^{th} spring, S is the set of springs connected to this robot, and $\ddot{x}$ is the robot's acceleration or motion. (1) is a self organizing force, $\mathbf{F_{SOF}}$, that enables the robots to spread out and move to positions equidistant from one another through simple interactions between neighboring robots.

A. Spring Mesh Formation

A number of formation algorithms are possible for creating and maintaining the spring mesh, such as full connectivity, nearby neighbors, N-nearest neighbors, attachment sites, Delaunay graphs, and acute-angle test. The acute–angle test is used for all examples discussed in this paper to form the spring mesh. The acute-angle test [8] states that for a spring mesh connection to form between robots L and M there must be no robots contained within the diameter of the circle LM, as shown in the right side of Figure 1. In the left side of Fig. 1, robot N4 is within the circle LM and the acute-angle test $\angle$LN4M (angle between nodes L, N4, and M) fails. Therefore no spring connection is created between robots L and M.

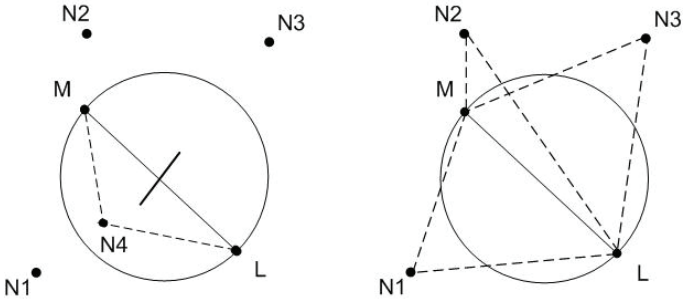


Fig. 1. Failed (left side) and Successful (right side) Acute Angle Test

The acute-angle formation algorithm minimizes the communications overhead between robots. However the robots must be capable of sensing the presence of other robots within some radius R (wireless sensing range) and determine their distance and bearing.

The application of the acute-angle test creates a hexagonal lattice formed by the robots as the mesh converges to a stationary state. The robots on the periphery of the mesh are *edge* robots and the remaining robots are *interior* robots. The virtual springs between robots approach their natural length, as

defined in (1), as the VSM stabilizes resulting in triangle formations between all vertices of the graph. Polygons with more than 3 sides may appear within the graph prior to stabilization. Avi files showing each of the steps from initial configuration to final configuration for some of the virtual spring mesh examples depicted in this paper may be viewed at http://mhrg.if.uidaho.edu/.

Collision avoidance of fixed objects or in dynamically changing environments is not addressed in the VSM algorithm. Typical approaches used by other researchers optimize a cost function such as the distance to a destination to determine the motion of a robot [20]-[21].

B. VSM Self Healing Properties

Robots are able to detect faulty behavior, initiate self repair and automatically reconfigure themselves with the VSM algorithm. The mesh degrades gracefully when robots fail and reforms as a result of the dynamics of the algorithm. This self healing property of the VSM algorithm is illustrated in [7],[9].

C. Enhancements to the VSM Algorithm

The enhancements to the VSM algorithm are as follows:

1. **Boundary Detection.** Keeping the robots within an area of interest or boundary area is not discussed in [7]-[9]. Any practical deployment of robots for search and rescue or establishing a mesh network would require restricting the area of coverage.
2. **Collision Avoidance.** Robots avoid collisions with one another due to the repelling and attractive forces in (1). However obstacle or collision avoidance with fixed or moving obstacles is not built into the VSM algorithm.
3. **Adaptive Springs.** The VSM algorithm naturally spreads robots out as the virtual springs approach their NSL. The area of interest may not be fully covered at this point due to robot failures or an insufficient number of robots. The natural spring lengths for robots need to be adaptive to allow the mesh to react to failures and expand, if necessary, to maximize coverage of the area within wireless range limits.

IV. Extended VSM (EVSM) with Control Laws Using Wireless Communications

The Extended Virtual Spring Mesh (EVSM) algorithm is meant to overcome the deficiencies of VSM with several new novel features. An overview of the new EVSM algorithm is presented first followed by the algorithm's novelties: exploration control law, expansion control law, metrics, self-organizing force with wireless, and adaptive springs. A summary of the terms that are used to present the EVSM algorithm are listed in Table I.

The EVSM algorithm uses wireless communications to detect other robots within range. One approach to detect fixed non-robotic objects, or obstacles, is to use ultrasound; however other remote sensing technologies such as LIDAR would also provide the same capability. Each robot employs wireless ad hoc communications available for the entire 360° around the robot to detect the signal strength of other robots with wireless range. Range sensing communications, also available for the entire 360° around the robot, is used to detect fixed objects for collision avoidance and aid in navigation.

TABLE I
EVSM Terms

Term	Definition
l_i	Current length of the *ith* spring
E	Error
m	Force multiplier constant
w_i	Normalized distance to wall *i* weight
F_{EXPL}	Exploratory force
F_{EXPN}	Expansion force
F_D	Driving force
F_{SOF}	Self organizing force
F	Force applied to each robot
k_s	Spring constant
k_d	Damping constant
R	Wireless sensor range
NSL, l_0	Natural spring length
d_{min}	Minimum robot distance to wall
v_{max}	Maximum robot movement per time step

EVSM introduces spring mesh, exploratory, and expansion forces with a collision avoidance feature. The exploration force allows the robots to move about the area of interest and into the areas surrounding fixed objects. The expansion force enables the robots to spread out to provide uniform communications coverage.

The pseudo code for implementing the EVSM algorithm on each robot is shown in Table II. Prior to executing the VSM algorithm in Table II, the simulation is initialized with one of the environments depicted in Figure 8. The simulation is then run until the average F_E is at a theoretical maximum value or 1000 iterations. In practice this code would run until movement falls below some threshold for a period of time. Next we describe each line of pseudo code in the EVSM algorithm outlined in Table II.

1. The acute angle test noted in Figure 1 is used to calculate the virtual spring mesh connections. By applying the acute angle test for robots within R distance of one another, the algorithm determines which robots are neighbors of one another.
2. For each robot the distance and bearing to each of their neighbors (from step 1) is calculated. Bearing is measured counter clockwise from the positive x axis. The distance to each neighbor is calculated based on (22) through (25).
3. A robot may change roles throughout the deployment of the mesh. The status of the robot is determined to be either interior or edge. A robot at position p_i must satisfy the following conditions to be classified as an edge robot: 1) Two neighbors at positions p_j and p_k that are at an angle $\theta_i \geq 90^\circ$ apart from p_i, and 2) no neighbors within a polygonal area formed from positions p_i, p_j, p_k, and a point p_m projected out from p_i at an angle $\theta_i / 2$ and distance $d_{i,m}$.

4. Edge robots are at the frontier of the mesh where other forces apply to spread the mesh, like water, throughout an area.
5. Apply the driving force $\mathbf{F_D}$, which is either the exploration, $\mathbf{F_{EXPL}}$, or expansion, $\mathbf{F_{EXPN}}$, force to the robot. The exploration force will move an edge robot towards the most distant visible walls. The expansion force will move an edge robot outward from the mesh at an angle equidistant to the robots two neighbors. The exploration and expansion forces are explained in section IV.A and IV.B, respectively.
6. End of edge robot calculations.
7. Calculate the self organizing force from (1) which applies to both edge and interior robots. $\mathbf{F_{SOF}}$ is a vector that provides both direction and magnitude for movement of the robot. The self organizing force is explained in detail is section IV.F.
8. The total force **F** applied to a robot is a combination of $\mathbf{F_{SOF}}$ and the driving force $\mathbf{F_D}$ ($\mathbf{F_{EXPN}}$ or $\mathbf{F_{EXPL}}$).
9. The distance to the closest obstacle, d_{obs}, sensed by the robot via ultrasound or LIDAR is calculated.
10. The minimum allowable robot distance to an obstacle, d_{min}, is compared to d_{obs}.
11. **F** is a vector specifying both magnitude and direction. The force F is applied to the robot if the new robot position will be > d_{min}.
12. Otherwise
13. The force **-F** is applied to the robot. The magnitude of the movement of the robot is the same as **F** but 180 degrees opposite in direction.
14. End of algorithm for one time step of one robot

The running time of EVSM is the sum of the running times for each function identified in Table II. The code within each function runs no more than n times, where n is the number of robots neighbors. $\mathbf{F_{EXPL}}$ is quadratic in the number of walls because each wall visible to an edge robot must be compared to the other visible walls to determine whether a wall is blocked from the robot's sensors by another wall. Therefore the running time of the algorithm is $O(n+p^2)$, where p is the number of walls visible to an edge robot.

TABLE II
EXTENDED VIRTUAL SPRING MESH ALGORITHM

Line #	Definition
1	Calculate spring mesh connections
2	Calculate distance and bearing to neighbors
3	Calculate if robot is interior or edge robot
4	IF edge robot THEN
5	Apply $\mathbf{F_{EXPN}}$ or $\mathbf{F_{EXPL}}$ as the driving force $\mathbf{F_D}$
6	ENDIF
7	Calculate $\mathbf{F_{SOF}}$
8	Calculate total force: $\mathbf{F} = \mathbf{F_{SOF}} + \mathbf{F_D}$
9	Calculate distance to closest obstacle, d_{obs}
10	IF $d_{obs} > d_{min}$ THEN
11	Move robot based on **F**
12	ELSE
13	Move robot based on **-F**
14	ENDIF

Collision avoidance is built into the EVSM algorithm. Any obstacle within d_{min} distance from a robot will be avoided.

The number of simulation time steps is 1000 or 2000 for all test cases in this paper. The force, **F**, applied to each robot consists of a self-organizing force, $\mathbf{F_{SOF}}$, and a driving force, $\mathbf{F_D}$. $\mathbf{F_{SOF}}$ will evolve the swarm into a disk shaped pattern in the absence of obstacles and boundaries. $\mathbf{F_D}$ will move edge robots of the swarm in a specific direction.

$$\mathbf{F} = \mathbf{F_{SOF}} + \mathbf{F_D} \quad (3)$$

$\mathbf{F_{SOF}}$ is the basic control law, explained in detail in Section IV, with wireless sensing. $\mathbf{F_D}$ is an expansion force, $\mathbf{F_{EXPN}}$, or exploration force, $\mathbf{F_{EXPL}}$.

$$\mathbf{F_D} = \mathbf{F_{EXPL}} \textit{ or } \mathbf{F_{EXPN}} \quad (4)$$

The collision avoidance feature applies to the edge robots to: 1) avoid collisions with fixed non-robotic objects and 2) to maintain a minimum distance, d_{min}, from those objects. Collision avoidance prevents the robots from colliding with fixed objects using remote sensing technology.

The following subsections discuss the EVSM exploration and expansion control laws with wireless communications and metrics. The metrics are used to evaluate the effectiveness of the EVSM algorithm for the test cases discussed in Section VI.

A. Exploration Control Law

Traditional movement or exploration algorithms for robots include random walk, follow wall, seek open and fiducial [22]. However these algorithms do not address the movement of a mesh. Other distributed dispersion algorithms use gradient based communications to spread information throughout the network and to guide robots [23]. The EVSM algorithm uses only local information acquired from a robot's neighbors to disperse the robots throughout some area.

An adjustment to the spring lengths alone is not enough to create an exploratory force for the mesh. An additional robot control law is defined to enable the robots to explore areas using remote sensing technology where there are a number of fixed objects such as walls or moving obstacles. The remote sensing technology has a sensory range sufficient for the entire search area. The exploration control law defines an additional vector force, $\mathbf{F_{EXPL}}$, which moves the edge robots in the direction of distant objects pulling the interior robots along.

$$F_{EXPL} = \sum_{i=1}^{n} (m \times w_i) k_s (l_i - l_0) \hat{u}_i - k_d \dot{x} \quad (5)$$

where m is a force multiplier constant, w_i is a normalized weight, n is the number of walls visible to this robot, and $\hat{u}_i$ is the unit vector previously described.

Visible walls, or obstacles, that are within the sweep angle of an edge robot are assigned weights. The sweep angle is the largest angle of an edge robot in which there is no visible neighbor robots. The weights are proportional normalized distances to the walls visible to each robot. The weights are used to determine the bearing and movement of each robot. A weight is assigned as follows:

$$w_i = d_i \times \frac{1}{\sqrt{\sum_{i=1}^{k} d_i^2}} \quad , i=1,\ldots,k \quad (6)$$

where d_i is a distance to i^{th} wall, and k is the number of visible walls. Distance metrics may be used to calculate d_i.

This approach to calculating weights ensures that the furthest visible walls/obstacles are weighted more heavily than the closest walls, causing the robots to explore distant areas. Each wall within the sweep angle, θ, of an edge robot exerts an exploratory force on that robot causing the mesh to move towards the walled areas. As the robots move through the area the walls visible to the robot will change, further influencing the movement of the robot. At some point the wall forces will equalize and the robot movement will stop or oscillate in a small area.

$\mathbf{F_{EXPL}}$ will draw robots to distant objects, or repel robots from close objects (fixed or moving). Dynamically changing environments may contain both fixed and moving obstacles. The moving speed of the mesh, the speed and direction of the moving obstacle relative to the mesh, and the size of the mesh are all factors in collision avoidance. When moving obstacles are traveling towards the swarm at a higher velocity than the swarm, the swarm may not move fast enough to avoid a collision with the obstacle.

Theorem 1 The EVSM robotic mesh will avoid collisions with objects in dynamically changing environments.

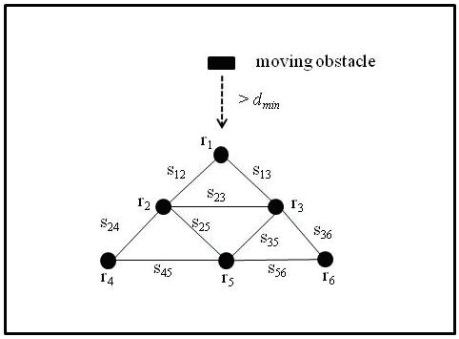
Proof: The EVSM algorithm treats obstacles as virtual nodes, which addresses enhancement numbers 1 and 2 (section III.C) with the VSM algorithm. When the robots within the mesh come within d_{min} distance of a new static obstacle, the robots are repelled from the object by (1) as l_i-d_{min} becomes negative, where l_0 is replace by d_{min}. The robots then move around the static obstacle with the movement defined as:

$$\ddot{x} = \left[\sum_{i \in S} k_s (l_i - d_{\min})\hat{u}\right] - k_d \dot{x} \tag{7}$$

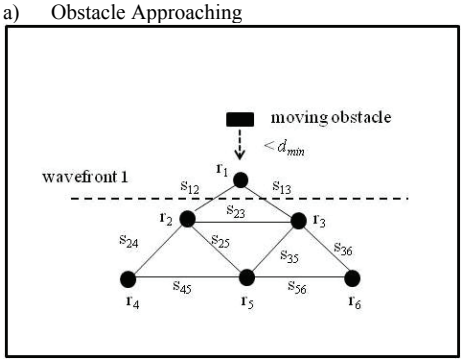
The same proof applies for moving obstacles. However the size of the mesh, the degree of expansion of the mesh, and the speed and direction of a moving obstacle will determine whether the moving obstacle will collide with the mesh. A moving obstacle approaching robot r_1 in a mesh is shown in Figure 2a. As the obstacle moves within d_{min} distance of r_1, r_1 will repel, compressing springs s_{12} and s_{13} in Figure 2b. l_{12}-d_{min} and l_{13}-d_{min} both become negative exerting a negative force on r_2 and r_3. r_2 and r_3 next move towards r_4, r_5, and r_6. The same pattern repeats creating a series of wave like movements as the robot mesh repels away from the obstacle. At each time step a wavefront, m, of robots moves away from the obstacle.

$$W = \sum_{m=1}^{M} \sum_{i=1}^{N} \ddot{x}_i \tag{8}$$

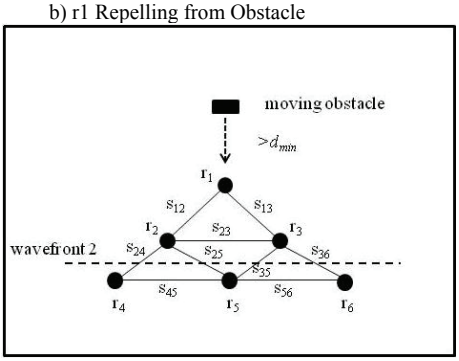
Where W = total movement by all robots in all wave fronts, M = number of wave fronts in the mesh, N = number of robots in the wavefront m, and $\ddot{x}_i$ is the acceleration of robot i on wavefront m. A mesh with 10 wave fronts moving at 1 meter per robot would take 10 movements for the entire mesh to move 1 meter. An obstacle moving at 1 meter per second head on towards r_1 would quickly collide with the mesh.



a) Obstacle Approaching



b) r1 Repelling from Obstacle



c) r2 and r3 Repelling from r1

Fig. 2. Spring Mesh Repelling from Moving Obstacle

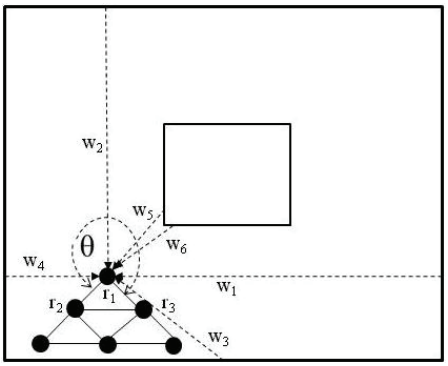


Fig. 3. Example Depiction of Exploration Force

Figure 3 depicts an example of $\mathbf{F_{EXPL}}$ where n is 6. Robot r_1 is an edge robot with robot neighbors' r_2 and r_3 in an enclosed area with one office/obstacle. A weight is assigned to each of the walls within the sweep angle of the edge robot r_1 based on (6). The sweep angle θ for the edge robot r_1 extends from the virtual spring connection between robots r_1 and r_2 to the virtual spring connection between robots r_1 and r_3. Each

visible wall in (5) for an edge robot exerts an attractive or repulsive force on robot r_1. The complete exploration force is a summation of the individual forces exerted by walls within the sweep angle θ.

The collision avoidance feature of EVSM ensures that whenever the distance between the new robots position R_i and the wall coordinate W is less than or equal then the predefined distance d_{min}, the force F_i exerted by a wall object F_w is repulsive (4).

$$\forall((R_i(x_i, y_i) - W(x, y)) \le d_{\min}), F_i = -F_w \quad (9)$$

Otherwise the force is attractive, moving the robot towards the wall. For the test cases in this paper d_{min} =10 meters.

B. Expansion Control Law

The expansion force moves the current robot in the direction of the bisector of the sweep angle of the neighbors of an edge robot. This force moves the edge robots outward from the interior office areas pulling the interior robots along. Like the exploration force, $\mathbf{F_{EXPN}}$ is attractive or repulsive if the new robot position is greater than or less than d_{min} from a wall/obstacle, respectively.

Figure 4 depicts an example of $\mathbf{F_{EXPN}}$ which is similar to Figure 3 where robot r_1 is an edge robot with robot neighbors' r_2 and r_3 in an enclosed area with one office/obstacle. θ again represents the sweep angle for the edge robot r_1. The robot r_1 moves in the direction of θ/2.

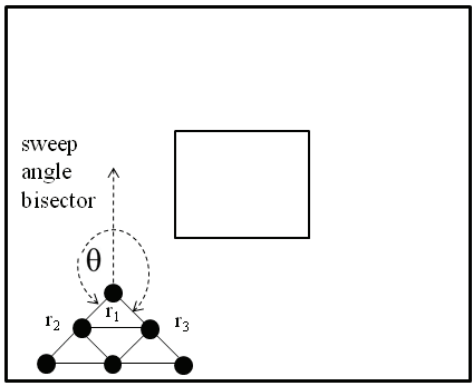


Fig. 4. Expansion Force Example

C. Effectiveness Metrics Introduced

Measures of effectiveness are necessary to assess the performance of the deployed MANET for different environments and algorithmic operating parameters. Several metrics are defined to evaluate the overall behavior of the swarm of robots: 1) coverage effectiveness, 2) formation effectiveness, and 3) connectivity effectiveness.

Coverage is an emergent part of the algorithm based on the number of robots and natural spring length. A robot and two neighbors typically form a triangular area, T_A. The coverage effectiveness, C_E, is the ratio of the area covered by all robot triangular areas divided by the total area of interest. Shapes other than triangular areas, such as 4 sided polygons, do not provide wireless connectivity between all elements and are therefore excluded from the C_E measurement.

$$C_E = \left(\sum_{i=1}^{t} T_{A_i}\right) / A,\ 0 \le C_E \le 1 \quad (10)$$

where t = total number of robot triangular areas

T_{A_i} = robot triangular area i

A = size of the total area of interest minus fixed obstacle size within the area minus area within d_{min} of walls and obstacles

Coverage effectiveness provides a measure of how much of a given area will be covered by the mesh network. The greater the coverage the higher the probability that if a node breaks down, the network will still heal and operate and provide communications coverage or sensing of the area. The spatial coverage of the robotic deployment increases as C_E approaches 1.

Formation effectiveness, F_E, is a measure of the ideal spacing arrangement between robots. The dynamics of the VSM algorithm moves all spring lengths towards their natural length over time if there is sufficient space available. Exploration, $\mathbf{F_{EXPL}}$, and expansion, $\mathbf{F_{EXPN}}$, forces and obstacle avoidance may cause deviation from that natural movement to achieve maximal coverage of an area.

$$F_E = \frac{\sum_{i=1}^{n} A_i}{n * I},\ 0 \le F_E \le 1 \quad (11)$$

A_i = triangle size in mesh

I = ideal triangle area (sides of natural spring length)

n = number of triangles in mesh

The spring mesh moves towards all spring lengths equal to their natural length in an acute angle formation as F_E approaches 1. The number of robots with one or no neighbors and/or the error (difference between actual spring length versus natural spring length) for each virtual spring increases as F_E approaches zero.

V_E is the connectivity effectiveness metric which represents the fraction of the total number of robots that are connected to one another. R_V is the total number of robots connected to one another by virtual springs and within wireless range of one another. R_T is the total number of robots.

$$V_E = R_V / R_T \quad (12)$$

Any two robots that are mobile mesh neighbors can communicate with each another. The remaining robots function as wireless relays forwarding communications traffic between source and destination. A V_E of 1 indicates that all robots in the mesh can communicate with one another.

D. Upper Bounds of Coverage, Formation, and Connectivity Effectiveness

The upper boundary values for C_E, F_E, and V_E will be presented. The theoretical maximum coverage for a mesh algorithm will be explained using geometrical relationships. Ma [17] has proven that equilateral triangle formations provide the most efficient coverage and least gaps over other approaches such as squares or circles. The size of an equilateral triangle is:

$$T_A = (S^2\sqrt{3})/4 \quad (13)$$

where S=length of a side of the triangle=NSL, and T_A = area of an equilateral triangle with sides of length S. The maximum number of triangles $|T|$ that may fit into the available area A in the area of interest is:

$$|T| = \left\lfloor \frac{A}{T_A} \right\rfloor \quad (14)$$

The maximum possible coverage effectiveness, C_{Emax}, for an available area A is then:

$$C_{E\max} = \frac{T * T_A}{A} \quad (15)$$

For a stabilized spring mesh not under compression, S equals NSL, the natural spring length. Equation 15 represents an upper limit of the number of robots needed to deploy in an area of size T_A to achieve a specified level of coverage.

The relationship between the number of triangles and vertices in a triangulation [24] is:

$$|V| - 2 \le |T| \le 2|V| - 5 \quad (16)$$

where |V| = number of vertices/robots and $|T|$ = number of triangles. Therefore for a given mesh with T number of triangles, the minimum number of vertices, V_{min}, and maximum number of vertices, V_{max}, are:

$$V_{min} = (T+5)/2 \quad (17)$$

$$V_{max} = T + 2 \quad (18)$$

where each vertex represents a robot.

This establishes lower and upper bounds on the number of robots necessary to achieve a triangulation of a given shape.

The theoretical maximum formation effectiveness, F_E, is reached when all of the triangles have adopted their ideal size. From (11) this means that each triangle is an equilateral with each side equal to NSL.

Connectivity effectiveness, V_E, is simply a measure of the number of robots that are connected to one another in the mesh. V_E should always be 1 unless there is some imbalance of forces applied to the robots or wireless signal degradation reaches a threshold where neighboring robots can no longer sense each other.

E. Convergence Analysis

Convergence will be explained in terms of the gradient of potential energy using the analogy of virtual springs. A virtual spring has potential energy when the spring is compressed or stretched. Convergence is achieved when all virtual springs in the mesh are at their natural spring length l_0 and the interactive potential energy of the mesh is zero; i.e., the robots have stopped moving [8], [16], [17], [25]. No potential energy exists between robots separated by a distance greater than the wireless sensing range, R; i.e., no virtual spring exists. A hyperbolic function representing the potential energy [26] between robots r_j and r_k, Figure 5a, connected by a virtual spring is:

$$V_{jk}(r_j, r_k) = K_{jk} \log(\cosh(E_{jk})) \quad (19)$$

where E is defined in (2) and K_{jk} is the gain which regulates both the magnitude of the potential energy and the force generated from the potential energy. K_{jk} is the ratio of the maximum output torque of the robot's driving motor divided by the distance between the driving wheels.

The potential energy of the entire mesh, P, is:

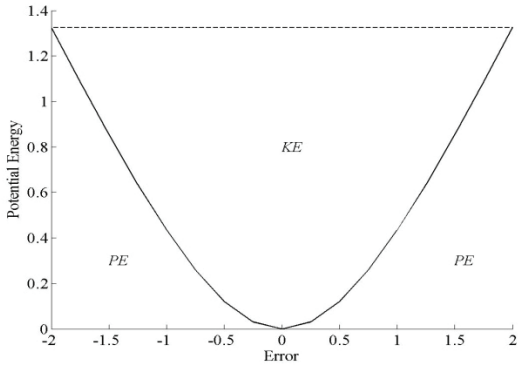
$$P = \sum_{i,j \in R, j \ne k} V_{jk}(r_j, r_k) \quad (20)$$

where R = robots in the mesh connected via virtual springs. Convergence to a stationary state is achieved when P = 0. Based on (1) the mesh will settle into a configuration which is the local minimum for the potential energy, $(l_i - l_0) = 0$.

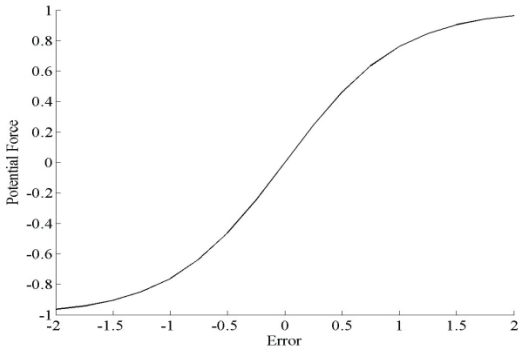
The potential force acting on a robot due to the force from a neighbor, Figure 5b, is the gradient of the potential energy expressed as:

$$f(r_j, r_k) = \nabla V_{jk} = K_{jk} \tanh(E_{jk}) \left(\frac{l_i - l_0}{\|l_i - l_0\|} \right) \quad (21)$$

The total force acting on a robot is the sum of the forces in (21) from all of the robots neighbors connected through virtual springs.



a.



b.

Fig. 5. a) Potential Energy and b) Potential Force

F. Self-Organizing Force with Wireless Sensing F_{SOF}

The self-organizing force, $\mathbf{F_{SOF}}$, for forming the basic virtual spring mesh network is the same as (1) but uses wireless sensing to calculate distances to neighbor robots. All robots are assumed to have the same 802.11 wireless transmission and reception capability and antennas (same transmit and receive antenna gains). The actual spring length is determined from the received signal strength (RSS) value received from another robot on the spring connection. The received RSS value is the transmission power of the other robot minus the path loss as shown in (22).

$$R_x = T_x - L_p \quad (22)$$

where R_x is received signal strength value. T_x is robot transmitter power in dB, and L_p is path loss, and

$$L_p = 33\text{dB} + N*10*\log_{10}(D) + 20*\log_{10}(f) \quad (23)$$

where f is a frequency in gigahertz. N is a path loss exponent, and D is a distance in meters in a free space loss environment, ignoring transmitter and receiver antenna gains [27]. The robots are using 802.11b/g communications at 2.45 GHz. Equation 22 becomes:

$$L_p = 40.2\text{dB} + N*10*\log_{10}(D) \quad (24)$$

A realistic transmit power, T_x, is 23dBm. 40.2dB is the reference loss (RL) at one meter. dBm is the power ratio in decibels (dB) of the measured power referenced to one milliwatt (mW). N equals 3.3 in an office environment. The distance D, which corresponds to the actual spring length between robots, in an office environment where N=3.3 is then:

$$D = l_i = 10^{\left(\frac{-R_x - 17.2}{33}\right)} \quad (25)$$

The RSS value detected from an adjacent robot in the spring mesh decreases exponentially with distance from a WiFi transmitter regardless of transmission power and antenna gain. Substituting (25) into (1) yields:

$$F_{SOF} = \left[\sum_{i \in S} k_s \left(10^{\left(\frac{-R_x - 17.2}{33}\right)} - l_o\right)\hat{u}_i\right] - k_d\dot{x} \quad (26)$$

$\mathbf{F_{SOF}}$ is a 1) a combination of the spring differences (actual spring length and natural spring length) to neighbors times a spring constant, minus 2) the damping constant multiplied by the velocity of the robot. $\mathbf{F_{SOF}}$ is the force that self organizes the swarm and moves the robots into a mesh configuration. The fault tolerant, self repair, scalable, and self adaptive properties of the swarm mesh are a direct result of the physics spring model in $\mathbf{F_{SOF}}$.

Figure 6 depicts an example of R_x values received by robot r_1 from neighboring robots r_2 and r_3. Robots r_2 and r_3 are within wireless range of robot r_1 and therefore r_1 can sense these neighboring robots. $\mathbf{F_{SOF}}$ for r_1 is a summation of the individual forces exerted by r_2 and $r_3$through these sensed R_x values.

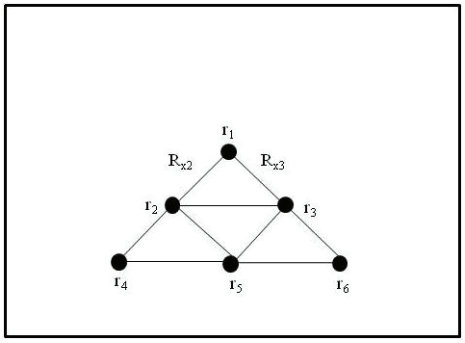


Fig. 6. Self-Organizing Force Example

G. Control Constants

The constants in (1) l_0, k_s, and k_d strongly influence the deployment of the robots. The value of l_0 must be less than the wireless sensing range necessary to detect other robots. In a free space loss environment l_0 is based on (25), where R_x is the minimum, or weakest, RSS value detectable by the wireless communication hardware used on the robot. An l_0 of 100 meters is achievable in an indoor free space loss environment with minimal signal disturbance. When the control laws have converged, the spacing between robots will approach l_0.

The choices for k_s and k_d affect the speed of convergence of the mesh. As k_s is held constant and k_d is increased, the movement of a robot slows, increasing the amount of time for the error E in (2) to approach 0. Shucker [8] has shown that E approaches zero more quickly when k_s is less than 0.5 and k_d varies from 0.125 to 1. The authors of this manuscript have conducted numerous simulations with 100 to 240 robots in an environment with up to 8 obstacles. These results have shown that E→0 faster when 0.4<= k_s <= 0.6 and k_d = 0.1. Consequently k_s of 0.5 and a k_d of 0.1 are used for all of the test cases in this paper.

H. Adaptive Springs

Our robotic mesh network spreads out like water seeking its own level. Robot movement will approach zero as the virtual springs approach their natural length. An adaptive natural spring length (*NSL*) is necessary for the robots to continue spreading through the area from this point forward. This EVSM feature addresses enhancement number 3 (section III.C) with the VSM algorithm.

Adaptive springs start with the edge robots that are on the frontier of the mesh. The edge robots determine if their movement is greater than τ over the last β moves. If this condition is true, the edge robot increases the *NSL* value by δ for each neighbors' virtual springs as in (27). τ, β, and δ values are determined empirically and set to 0.5, 60, and 0.5, respectively.

$$if \sum_{i,j=k}^{\beta} \left(\left\|R_i(x_i, y_i) - R_i(x_j, y_j)\right\|\right) > \tau, NSL \rightarrow NSL + \delta \quad (27)$$

An interior robot that receives a message to use a new NSL value automatically changes their neighbor NSL values. These NSL values then propagate to other interior and edge robots until the new spring lengths have spread through the entire mesh.

I. Energy Costs

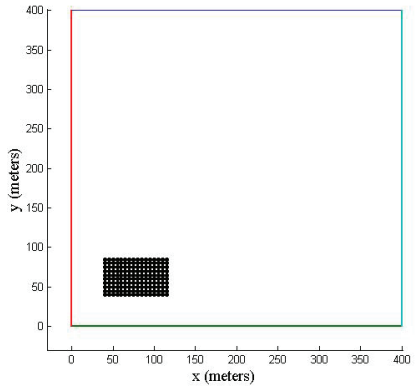
The utility of a mobile robot or mobile sensor is limited by the amount of energy available to perform sensory, communications, and movement functions. A swarm may consist of a heterogeneous mix of robots with different functions. Our assumption is that a mobile swarm of robots will be accompanied by physically larger robots, or "*motherships*" or "*charging bots*", which may deploy the swarm, and will provide a charging capability. Swarm robots will return to their *mothership*, or adaptive dock [28], for charging when their energy supply drops below some threshold. Alternatively swarm robots may signal their *mothership* through the swarm mesh communications channels to pick up the robots just prior to exhausting their energy

supply. Robots may additionally capture energy from the environment with solar panels.

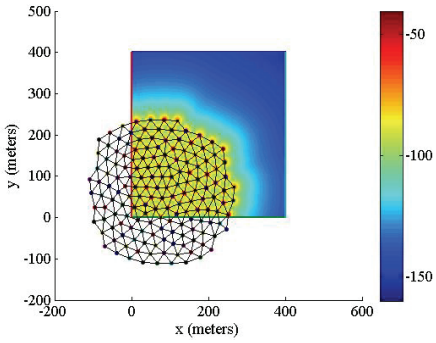
V. Analysis of VSM

The performance of the VSM algorithm in several test environments is demonstrated to provide a baseline for assessing the performance of EVSM. The standard VSM algorithm does not include boundary detection, collision avoidance, or exploration and expansion forces. Figure 7a shows the initial configuration of robots. l_0, k_s, and k_d are 30, 0.5, and 0.1, respectively in Figure 7b. The black dots in Figures 7a and 7b represent the robot positions with walls bounding the area of interest. Signal loss is based on the free space model of (23) with N equal to 2. The color bar on the right of Figure 7b represents signal strength in dBm, depicting RF propagation. Variation in the color is used to depict the intensity of the RF propagation from each robot in a free space loss environment. All units for x and y are in meters. R, the maximum wireless sensing range of a robot, and NSL, the natural spring length, are 40 and 30, respectively, for all simulations in this manuscript.

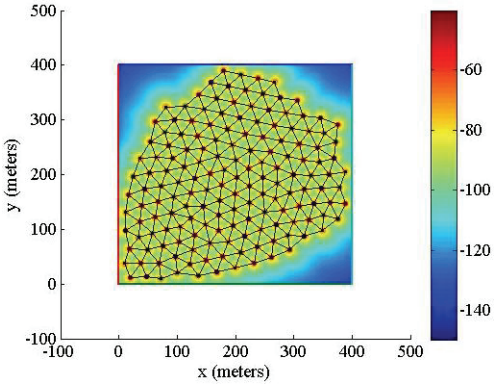
Figure 7b shows the final configuration of a deployment based on the standard VSM algorithm with no boundary detection, or exploration and expansion forces. The robots move outside of the 400 by 400 meter bounded area without detecting wall boundaries. Figure 7c illustrates the VSM algorithm with boundary detection. The robots spread out until the virtual springs reach their natural length where no further movement occurs based on (1).



a) Initial Configuration



b) VSM with No Boundary Detection



c) VSM with Boundary Detection

Fig. 7. Virtual Spring Mesh Algorithm

Figure 8 shows the 0, 4, 6, and 8 office test environments used for EVSM. The results of VSM simulations with boundary detection and added collision detection, using the same office environments from Figure 8, are listed in Table III. F_E and V_E are 1 in all cases. The results are for 2000 iterations of the algorithm and time is in minutes.

TABLE III
VSM (with Collision Detection) Deployments
(2000 Iterations)

# Obstacles	# Robots	C_E	Time
0	160	0.7	42
0	200	0.83	62.8
0	230	0.93	80.4
4	160	0.67	71.6
4	200	0.77	99.2
4	230	0.91	121
6	160	0.65	84
6	200	0.78	113.6
6	230	0.79	137.8
8	160	0.62	131.3
8	200	0.74	127.8
8	230	0.71	154.3

VI. EVSM Simulation Test Cases

In contrast, the EVSM with collision avoidance, exploration and expansion forces, and adaptive springs will be shown to demonstrate the effectiveness of this approach. The EVSM algorithm will be tested in a simulated enclosed building area with 1) regular shaped obstacles/offices or no offices, and 2) arbitrary shaped obstacles/offices. All of the test cases in this paper use this same initial formation, shown in Figure 7a, with additional rows and columns for more robots.

EVSM is capable of dealing with obstacles of virtually any shape. The regular shaped test obstacles include 4, 6, and 8 office areas in a 400 by 400 meter environment. The arbitrary shaped test obstacles include 4 and 6 irregular polygon shaped offices; i.e., polygons whose sides are not all the same length or whose interior angles do not all have the same measure. The measures of effectiveness will be used to evaluate the final positions of 160, 200, and 230 robots shown in Figure 8.

Collision avoidance, exploration, and expansion forces are applied to each of these test cases.

All simulations are run on a Dell computer, Intel Core 2 Quad CPU 2.33 GHz. MATLAB student version 7.8 was used to build all simulation models. Possible robots for implementing the EVSM algorithm include the iRobot SwarmBot (5in x 5in x 5in) or the Pioneer P3-DX (44cm x 38cm x 22cm).

A. Regular Shaped Obstacles/Offices

Open Area with No Interior Offices

A building with a large open space and no interior hard-walled offices is shown in Figure 8a through 8c. 160, 200, and 230 robots are tested in a 400 by 400 meter environment.

0 offices, 160 robots: The robots have begun to spread out and cover the area. An F_E of 1 indicates that all virtual springs have reached their natural length. F_E is greater than one, which indicates that adaptive spring forces are increasing the robot NSL values. Adaptive springs enabled the mesh to cover a larger area than would have been possible otherwise.

0 offices, 200 robots: C_E has increased with additional robots providing 97% coverage. An F_E of 1.03 indicates some of the springs have adapted to slightly higher values than the standard NSL of 30.

0 offices, 230 robots: C_E, F_E, and V_E are 0.92, 0.9, and 1, respectively for 230 robots. C_E and F_E have both decreased from previous deployments indicating that not all robot neighbors are forming triangular meshes due to overcrowding.

Every environment has an optimal number of robots to deploy based on the natural spring length and the available area. When the number of robots deployed is less than the necessary amount, the virtual springs will adapt up to a length of *R-2* for the NSL increasing the amount of covered area. When there are too many robots deployed, C_E and F_E will actually be reduced due to overcrowding.

EVSM Algorithm with Four Offices

The interior of a building with 4 hard wall offices, or obstacles, is shown in Figures 8d, 8e, and 8f. Signal strength drops off on the side of the hard wall sections opposite the robots in Figure 8d due to attenuation from the hard walls.

4 offices, 160→230 robots: The final positions of the robots for the four office 160 robot configuration after 2000 iterations are shown in Figure 8d. C_E, F_E, and V_E are 0.75, 1.02, and 1, respectively. The exploration force draws many of the robots into the center area of hard-walled offices. Once within that region the forces of the virtual spring mesh as well as the forces exerted by the walls visible to the edge robots help to maintain that configuration. An F_E of 1.02 indicates that some of the springs adapted to a larger value than the initial *NSL*. Additional iterations of the algorithm may result in additional spring increases.

Figure 8e shows 200 final robots positions in the same 4 hard-walled office environment after completing the EVSM simulation. C_E, F_E, and V_E are 0.88, 0.94, and 1, respectively. Coverage effectiveness improves with a larger robot swarm because there are more robots to form the mesh and cover a larger area.

C_E reaches a maximum at 90% with 230 robots with no spring adaption as shown in Figure 8f. Additional algorithm iterations may improve both C_E and F_E.

Due to convex rather than concave corners, the robots will not perfectly encircle a fixed obstacle. Note that the robots do not form perfect squares around the interior offices.

Increasing the number of robots deployed for some area improves C_E only up to a point. Then overcrowding occurs and F_E is reduced as well by squeezing additional robots into the area and compressing the virtual springs. An appropriate number of robots must be chosen for a given scenario.

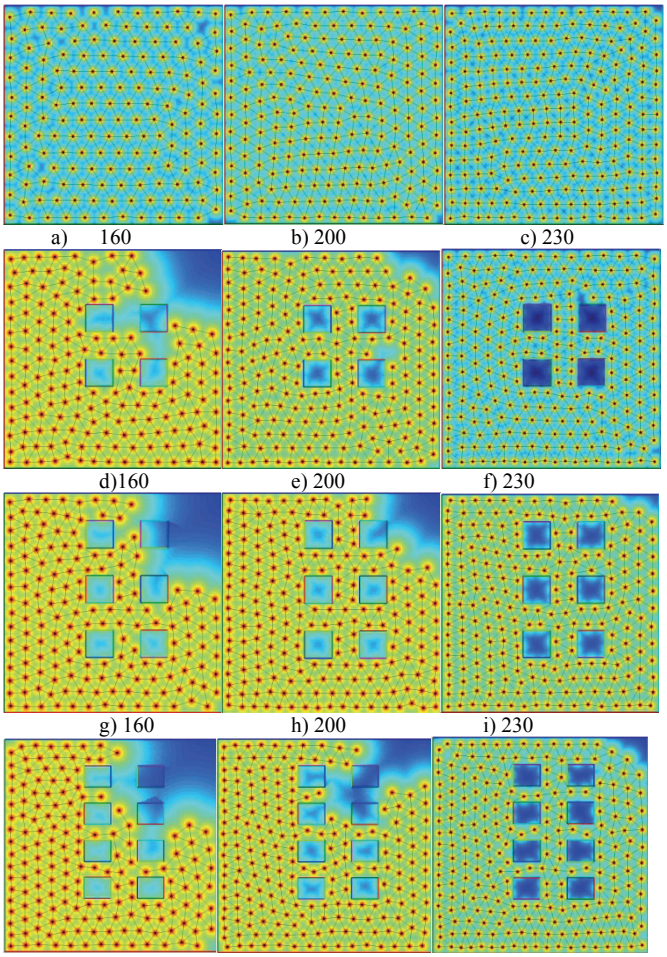


Fig. 8. Final Robot Positions for 0, 4, 6, and 8 Hard-Walled Interior Offices, 2000 Iterations, 400 x 400 Square Meter Area

EVSM Algorithm with Higher Number of Offices

6 offices, 160→230 robots: Figure 8g through 8i show a 160, 200, 230 robot deployment in a 6 hard-walled office area. C_E reaches a maximum at 92% with 230 robots.

8 offices, 160→230 robots: Figure 8k shows a 200 robot deployment in an 8 hard-walled interior office area. An 8 office environment has less available robot space than a 6 office environment, resulting in improved coverage effectiveness for the same number of robots. C_E, F_E, and V_E are

0.8, 0.85, and 0.99, respectively. C_E reaches a maximum at 80% with 200 robots and decreases with 30 additional robots due to overcrowding.

Several test cases have shown that C_E does not approximate the maximum coverage possible. C_E may be improved by enhancements to the adaptive spring algorithm and increasing the number of algorithm iterations. C_{Emax} from (15) is 99% for all office configurations.

A summary of the simulation results for 2000 iterations per simulation run is shown in Table IV. The error, as defined in (2), falls off in exponential fashion with the number of iterations and eventually stabilizes as the virtual springs approach their natural length.

TABLE IV
EVSM SIMULATION RUNS WITH EXPLORATION AND EXPANSION FORCES

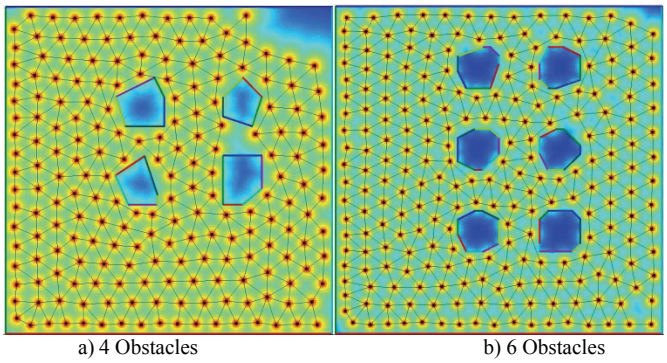
Interior Offices	# Robots	E	C_E	F_E	V_E	t [min]
0	160	4.3	0.89	1.29	1	12.2
0	200	2.45	0.97	1.03	1	17.7
0	230	2.83	0.92	0.9	1	22.9
4	160	2.27	0.75	1.02	1	17.9
4	200	2.44	0.88	0.94	1	25.3
4	230	2.88	0.9	0.86	1	34.7
6	160	2.4	0.74	0.98	1	21.8
6	200	2.94	0.83	0.87	1	30.7
6	230	3.5	0.92	0.81	1	39.2
8	160	2.91	0.71	0.91	1	24.7
8	200	3.58	0.78	0.81	1	33.6
8	230	3.87	0.87	0.78	1	44.9

F_E for the 4, 6, and 8 office configurations has decreased in all cases with an increasing number of robots. There is an insufficient space between obstacles for robots to self organize at the natural spring length, l_0, of 30 meters in the 8 office environment. Therefore the actual spring length, l_i, between robots within the obstacle area will be less than l_0 which will decrease F_E. In the four and six office environments the robots have difficulty in forming perfect equilateral triangle formations in the area between offices and some corners.

Computation time increases from 47% to 49% as the number of robots increases from 160 to 230 for all numbers of interior offices. Larger robot deployments require additional time for the dynamics of the EVSM algorithm to play out.

B. Arbitrary Shaped Obstacles/Offices

The EVSM algorithm will work in environments with any type of obstacle shape. Figure 9a shows a 230 robot deployment in an interior office area with 4 irregular polygon obstacles. C_E, F_E, and V_E are 0.93, 0.94, and 1, respectively. Six eight sided irregular polygons are shown in Figure 9b with metric calculations of C_E, F_E, and V_E as 0.99, 0.82, and 1, respectively. Both simulations were run for 2000 iterations.



a) 4 Obstacles b) 6 Obstacles

Fig. 9. Irregular Polygon Obstacles with 230 Robots

C. EVSM Time and Iterations versus Coverage Effectiveness

Figures 10 and 11 show the time and number of iterations (robot movements), respectively, plotted to achieve different levels of coverage effectiveness. The number of robots (200) and size of area (400 x 400 meters) are held constant. An iteration of the EVSM algorithm is one variable length time step in the simulation. The virtual spring mesh dispersion increases with time and number of iterations, or time step of each robot. As the area covered by the dispersion increases, more time and movement is required to increase the coverage of the mesh (the effect similar to the ripples in a pond). This is due to a greater number of robots required to move in order to expand the entire mesh to achieve greater coverage. Hence iterations in the later algorithm stage take longer than in the beginning. The smaller slope of the ending line segments in Figures 10 and 11 shows more time and iterations are required to achieve an additional coverage than in previous intervals.

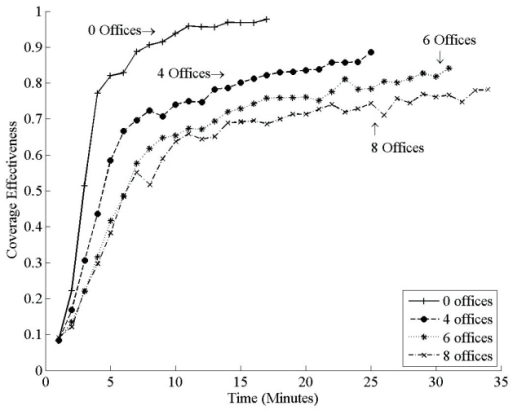


Fig. 10. Time to Reach Coverage Effectiveness

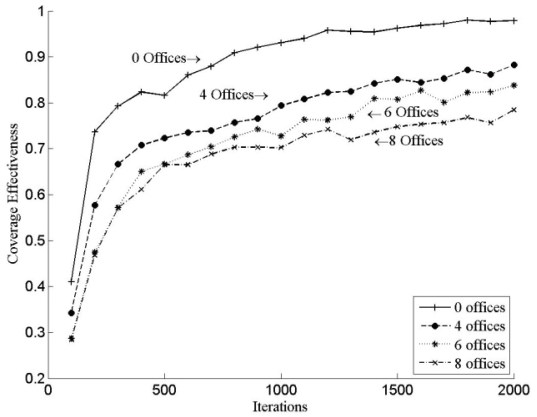


Fig. 11. Number of Iterations to Reach Coverage Effectiveness

D. VSM Time and Iterations versus Coverage Effectiveness

The time and number of iterations for VSM to reach various levels of coverage effectiveness is plotted in Figures 12 and 13, respectively. EVSM attains a higher C_E in less time than VSM in all cases.

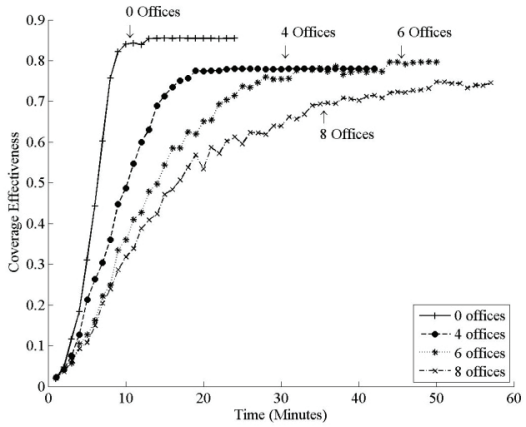


Fig. 12. Time versus C_E for VSM, 200 Robots

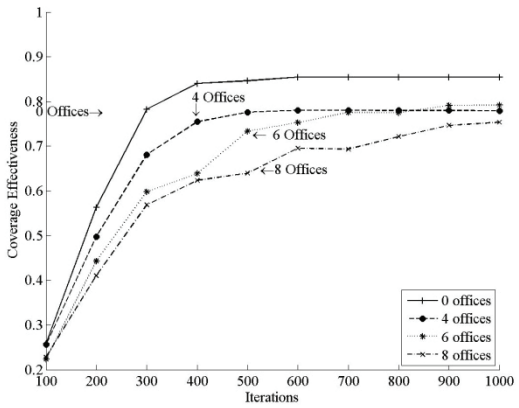


Fig. 13. Iterations versus C_E for VSM, 200 Robots

VII. EVSM WITH WIRELESS SIGNAL LOSS

Propagation of radio waves is characterized by free space loss, attenuation as radio waves pass through solid objects, and scattering and interference. Free space loss is the diminishing of signal power with the geometric spreading of the wave front as noted in (23). Other mathematical transmission models are CCIR model, Hata model, and the Walfisch-Ikegami model [29]. Our approach to capturing the effects of signal loss on the EVSM dispersion algorithm is based on the free space model with low (30%) and high (90%) probability of signal degradation, P_S, and low (25%) to moderate (50%) amount of signal loss, S_V.

Test cases are now repeated with the presence of noise; i.e., a communications disturbance resulting from attenuation and scattering due to physical objects in the area or interference with other communications. The purpose is to examine the relationship between communication disturbances, and number of robots and coverage, and help identify the limitations of our approach.

Table V shows the results of a series of experiments of introducing signal degradation or noise into the exploration environment. 200 robots were used in each simulation run. A P_S of 30% indicates a 3 in 10 chance that the RF signal will be degraded by scattering and interference. An S_V of 25 means the received RSS value is 25% weaker (more negative) than the calculated free space propagation value. A higher P_S and a higher S_V always results in a lower C_E. As S_V approaches 100%, C_E approaches 0.

TABLE V
EVSM SIMULATION RUNS WITH EXPLORATION AND EXPANSION FORCES

Interior Offices	P_S	S_V	C_E	E	F_E	V_E	t [min]
0	30	25	0.43	16.7	0.51	1	27.9
0	30	50	0.21	46.6	0.36	1	37.7
0	90	25	0.13	6	0.15	1	46.9
0	90	50	0.03	9.9	0.04	1	107
4	30	25	0.39	16.5	0.48	1	39
4	30	50	0.22	42.6	0.32	1	48.8
4	90	25	0.15	5.5	0.15	1	50.9
4	90	50	0.04	9.9	0.04	1	116.1
6	30	25	0.41	17.9	0.47	1	45.1
6	30	50	0.2	44.6	0.28	1	55.2
6	90	25	0.16	5.6	0.15	1	54.1
6	90	50	0.04	10.4	0.04	1	120.1
8	30	25	0.38	16.5	0.45	1	49.2
8	30	50	0.22	53.6	0.31	1	59.3
8	90	25	0.16	5.9	0.15	1	55.2
8	90	50	0.04	10.5	0.04	1	123.2

Robots in the presence of high signal degradation will believe their neighbors are further away than they actually are. Therefore significant loss of signal may mean that a robot will not see some neighbors resulting in little or no movement by the robot. Robots that are no longer detectable by their neighbors may become disconnected from the virtual spring mesh. A possible effective approach for dispersing robots in an environment with high signal degradation would be to increase the natural spring length l_o and the wireless sensing range R.

Figure 14 depicts C_E versus iterations with a P_s of 90 and a S_v of 125% for several office environments. C_E increases with the number of offices in the environment because F_{EXPL} and

F_{EXPN} are calculated using remote sensing technology and unaffected by wireless communications disturbances.

Figure 15 shows the results of simulations of 200 robots for 2000 iterations. The robot density increases as the probability and degree of signal degradation increase. As S_V increases the robots have difficulty in sensing their neighbors. Therefore there is little movement on the part of the swarm. As S_V increases C_E approaches zero, F_E grows dramatically smaller because a robots neighbors are outside of the maximum wireless sensing range. The increases in the amount of signal degradation influence the sensed distance to a neighboring robot as noted in (25).

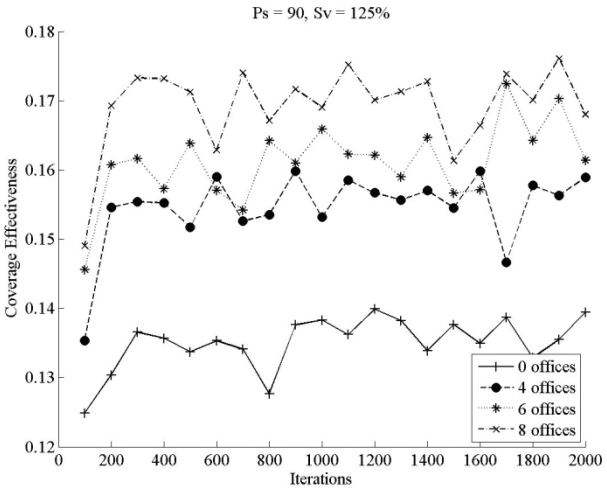
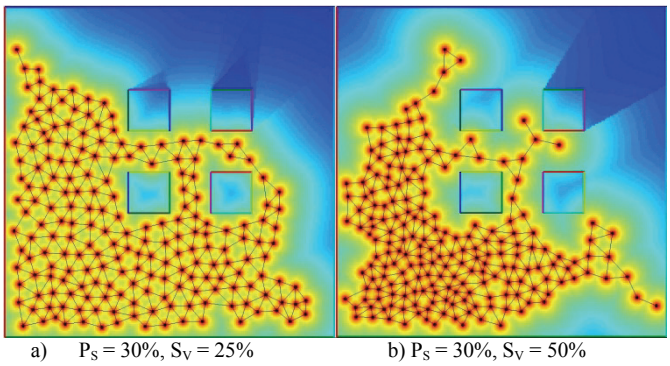


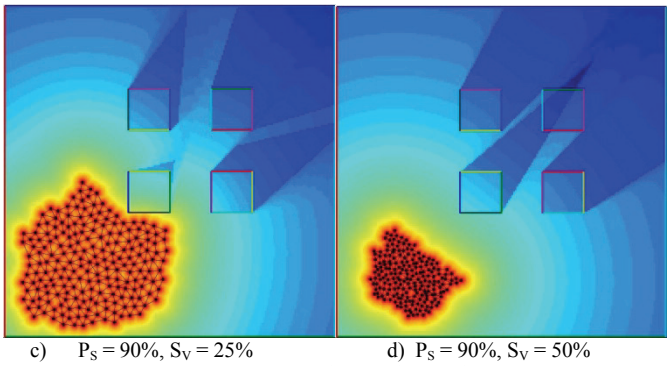
Fig. 14. C_E versus Iterations with Signal Degradation

A. Wireless Data Loss

The propagation of radio waves is affected by free-space loss (geometric spreading of the wavefront), attenuation (absorption of RF signals by physical objects), scattering (RF signals reflected by objects) and other factors. These factors affect the data loss between a robot transmitter and a robot receiver.



a) $P_S = 30\%$, $S_V = 25\%$ b) $P_S = 30\%$, $S_V = 50\%$



c) $P_S = 90\%$, $S_V = 25\%$ d) $P_S = 90\%$, $S_V = 50\%$

Fig. 15. 200 Robots with Signal Degradation

A communications channel between robots or robot and data acquisition computer may operate in one of two modes: 1) error-free transmission but delay, or 2) transmission errors but no delay. The error-free transmission but delay mode is realized by retransmission strategies of a transport layer protocol built on top of a network layer protocol in an OSI layered architecture scheme. Data loss is mitigated by the transport layer protocol via retransmission of damaged packets. Transport layer protocols have also been used to perform remote experiments with robots over the Internet [30]-[31] and industrial robotic platforms with multi-sensor systems [32].

VIII. Conclusion

The EVSM algorithm has proven to be an effective distributed self-organizing algorithm for deploying a MANET in an area with obstacles. The collective behavior of the MANET nodes (global cooperation) emerges from the interactions of the individual nodes with no global control.

The EVSM algorithm presented in this manuscript provides the following advantages over the VSM algorithm: 1) adapts to communications disturbances by varying the density and movement of mobile nodes, 2) dynamically adapts spring lengths enabling the mesh to expand as necessary, 3) provides obstacle avoidance and boundary detection, 4) introduces new control laws for exploration and expansion to provide blanket coverage of an area, and 5) uses new metrics (coverage, formation, and connectivity effectiveness) for assessing the performance of the algorithm. These advantages were demonstrated in test cases for 0, 4, 6, and 8 interior offices in an enclosed building environment with and without signal degradation. The results have shown that broad coverage is achievable for an enclosed area with EVSM's combined exploration and expansion control laws and adaptive springs.

The mobile entities in the EVSM algorithm may have sensors added for detecting chemical, radioactive, or biological elements in the environment. Future work involves testing the EVSM algorithm in new and different environments with varying signal strengths and deployment sizes. New deployments where the mobile entities contain a mix of different characteristics and capabilities for achieving some desired goal will also be explored.

REFERENCES

[1] R. Wood, "Fly, Robot Fly", IEEE Spectrum, Volume 45, Number 3, pp. 25-29, March 2008.

[2] R. Grabowski, L. Navarro-Serment, P. Khosla,, "An Army of Small Robots", Scientific American Reports, Vol. 18 No. 1, pp. 34-39, May 6, 2008.

[3] K. Kroeker, "Medical Nanobots", Communications of the ACM, Vol. 52 No. 9, pp. 18-19, September 2009.

[4] J. Luo, D. Wang, Q. Zhang,"Double Mobility: Coverage of the Sea Surface with Mobile Sensor Networks", Mobile Computing and Communications Review, January 2009, pp. 52-55.

[5] D. Goldman, H. Komsuoglu, D. Koditschek, "March of the Sandbots", IEEE Spectrum, Vol. 46 No. 4, pp. 30-35, April 2009.

[6] W. Gibbs, "Innovations from a Robot Rally", Scientific American Reports, Vol. 18 No. 1, pp. 80-86, May 6, 2008.

[7] B.Shucker, J. Bennett, "Scalable Control of Distributed Robotic Macrosensors", 7th International Symposium on Distributed Autonomous Robotic Systems, June 23-25, 2004.

[8] B.Shucker, J. Bennett, "Virtual Spring Mesh Algorithms for Control of Distributed Robotic Macrosensors", University of Colorado at Boulder, Technical Report CU-CS-996-05, May 2005.

[9] B.Shucker, J. Bennett, "Target Tracking with Distributed Robotic Macrosensors", IEEE Military Communications Conference, October 17-20, 2005, pp. 2617-2623.

[10] "Scientel to Build and Deploy a Meshnetworks' Mobile Wireless System for the City of Buffalo", http://www.encyclopedia.com/doc/1G1-121277642.html, Business Wire, August 16, 2004

[11] H. Ammari, "Fault Tolerant Measures for Large-Scale Wireless Sensor Networks", ACM Transactions on Autonomous and Adaptive Systems, Vol. 4, No. 1, Article 2, January 2009.

[12] S. Dobson, et al, "A Survey of Autonomic Communications", ACM Transactions on Autonomous and Adaptive Systems, Vol. 1, No. 2, December 2006, pp. 223–259.

[13] V. Vemuri, "Enhancing Computer Security with Smart Technology", Auerbach Publications, 2006

[14] A. Howard, J. Mataric, G. Sukhatme, "Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem", 6th International Symposium on Distributed Autonomous Robotics Systems, June 25-27, 2002.

[15] Zou, Y., Chakrabarty, K.," Sensor Deployment and Target Localization in Distributed Sensor Networks", ACM Transactions on Embedded Computing Systems 2004, 3(1), 61–91.

[16] J. Chen, S. Li, and Y. Sun., "Novel Deployment Schemes for Mobile Sensor Networks", Sensors, Vol.7 No. 11, pp.2907-2919, Nov.2007.

[17] M. Ma, Y. Yang, "Adaptive Triangular Deployment Algorithm for Unattended Mobile Sensor Networks", IEEE Transactions on Computers, Vol. 56, NO. 7, July 2007.

[18] N. He, D. Xu, "The Application of Particle Swarm Optimization to Passive and Hybrid Active Power Filter Design", IEEE Transactions on Industrial Electronics, Vol. 56 No. 8, October 2009.

[19] H. Chen, Y. Li, "Enhanced Particles with Pseudolikehoods for Three-Dimensional Tracking", IEEE Transactions on Industrial Electronics, Vol. 56 No. 8, October 2009.

[20] W. Chung, et al, "Safe Navigation of a Mobile Robot Considering Visibility of Environment", IEEE Transactions on Industrial Electronics, Vol. 56 No. 10, October 2009.

[21] N. Uchiyama, et al, "Model-Reference Control Approach to Obstacle Avoidance for a Human-Operated Mobile Robot", IEEE Transactions on Industrial Electronics, Vol. 56 No. 10, October 2009.

[22] R. Morlok, M. Gini, "Dispersing Robots in an Unknown Environment", Distributed Autonomous Robotic Systems 6, Springer, 2007.

[23] J. McLurkin, J. Smith, "Distributed Algorithms for Dispersion in Indoor Environments using a Swarm of Autonomous Mobile Robots", Proceedings of the 6th international Symposium on Information processing in Sensor Networks, pp. 545-546, April 25-27, 2007.

[24] O. Hjelle, M. Daehlen, "Triangulations and Applications", Springer, pp.5-10, 2010.

[25] A. Kansal, W. Kaiser, G. Pottie, M. Srivastava, G. Sukhatme, "Reconfiguration Methods for Mobile Sensor Networks", ACM Transactions on Sensor Networks, Vol. 3, No. 4, October 2007.

[26] Y. Liang, H. Lee, "Decentralized Formation Control and Obstacle Avoidance for Multiple Robots with Nonholonomic Constraints", American Control Conference, Minneapolis, Minnesota, June 14-16, 2006.

[27] "RF Propagation Basics, Sputnik White Paper, April 2004.

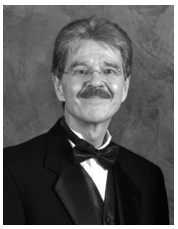
[28] A. Couture-Beil, R. T. Vaughan, "Adaptive mobile charging stations for multi-robot systems," Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS'09), (St. Louis, MO), October 2009

[29] G. Wolfle, P. Wertz, F. Landstorfer, "Performance, Accuracy and Generalization Capability of Indoor Propagation Models in Different Types of Buildings", 10th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 1999.

[30] R. Wirz, R. Marín, J. Claver, J. Fernández, E. Cerveraf, "Transport Protocols for Remote Programming of Network Robots within the context of Telelaboratories for Education: A Comparative Analysis", Proceedings of the 16th International Conference on Computer Communications and Networks, pp. 1315-1320, August 2007.

[31] F. Zeiger, M. Schmidt, K. Schilling, "Remote Experiments With Mobile-Robot Hardware via Internet at Limited Link Capacity", IEEE Transactions on Industrial Electronics, Vol. 56 No. 12, December 2009.

[32] J. García, J. Gómez Ortega, A. García, S. Martínez, "Robotic Software Architecture for Multisensor Fusion System", IEEE Transactions on Industrial Electronics, Vol. 56 No. 3, March 2009.



Kurt Derr (M'04): Kurt Derr, IEEE Member, has a B.S. in Electrical Engineering from Florida Institute of Technology and an M.S. in Computer Science from the University of Idaho. He is currently pursuing a PhD in Computer Science from the University of Idaho. Kurt has worked as a Computer Scientist at the Idaho National Laboratory since 1986 and began teaching courses as Affiliate Faculty at the University of Idaho in 1987. He received an R&D 100 award in 2009 for *RFinity, Mobile Open-Encryption Platform* and has co-authored 3 patents. Kurt has also written a book on object-oriented technology, *Applying OMT*, published in 1995. He has several recent published papers at IEEE conferences on computational intelligence in wireless networking environments.

Milos Manic, PhD (S'95-M'05-SM'06): Dr. Milos Manic, IEEE Senior Member, has been leading Computer Science Program at Idaho Falls and is a Director of Modern Heuristics Group. He received his Ph.D. degree in Computer Science from University of Idaho, Computer Science Dept. He received his M.S. and Dipl.Ing. in Electrical Engineering and Computer Science from the University of Nis, Faculty of Electronic Engineering, Serbia. He has over 20 years of academic and industrial experience, including an appointment at the ECE Dept. and Neuroscience program at University of Idaho. As university collaborator or principal investigator he lead number of research grants with the Idaho National Laboratory, NSF, EPSCoR, Dept. of Air Force, and Hewlett-Packard, in the area of data mining and computational intelligence applications in process control, network security and infrastructure protection. Dr. Manic an Administrative Committee Member for the IEEE Industrial Electronics Society and member of several technical committees and boards of this Society. Dr. Manic has published over hundred refereed articles in international journals, books, and conferences.