ENERGY EFFICIENT COOPERATIVE MOBILE SENSOR NETWORK

MAR CHOONG HOCK
(B.ENG. (HONS, FIRST CLASS), NUS,
M.ENG., NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

NUS GRADUATE SCHOOL FOR INTEGRATIVE
SCIENCES AND ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE
2008

Acknowledgements

First, I thank Agency for Science, Technology and Research (A*STAR) for granting me the A*STAR Graduate Scholarship (AGS) to pursue my PhD research.

Second, I thank my supervisor, Dr Winston Seah, for the supervision and guidance. Also I thank both members of the Thesis Advisory Committee for taking time off their schedules to give me insightful feedback. In particular, I thank Prof Lye Kin Mun for his gems of wisdom and kind advice and A. Prof Ang Marcelo H. Jr. for his gentle encouragement and support.

Third, I thank my loved ones: my wife, Chiew Pei and siblings (Ling Ling and Chong Kiat) for the many joyful moments and emotional supports in my long tedious journey of PhD research.

Fourth, I thank my endearing lab mates: Liu Zheng, Hwee Xian, Inn Inn, Ricky, Junxia, etc for giving me many wonderful moments in the lab and enrich my otherwise prosaic PhD life.

Finally, I thank my former supervisor, Prof Kam Pooi Yuen and those people who have at one time or another gracefully extended both their helping hands and sympathetic ears to me. Although those people remain anonymous in this page, I remember their kindness.

Table of Content

Summary

We research into the challenge of improving the quality of the reconstructed distribution from spatiotemporal monitoring data collected by mobile sensor network. Our approach is to attack the problem from the source, by mobilizing the sensors to harvest data of high information content so that the reconstructed distribution has minimum distortion. We consider four realistic constraints in our design: limitations of wireless communications, limited supply of energy and sensor resources and difficult terrains. Our strategy is to treat each mobile sensor as an intelligent cooperative autonomous agent, capable of processing cooperative shared information independently in order to carry out its harvesting task in an optimal manner. In the greater scheme, the sensors are to be divided into small self-contained cooperative groups for two reasons. First, it improves scalability and facilitates deployment in difficult terrains partitioned by obstacles. Second, it is more robust to communication problems since communications used to facilitate the harvesting tasks are intra-group in nature.

We investigate into the limitations in wireless communications through literature surveys and theoretical analyses. In our analysis, we examine better approaches to organize sensors and design our algorithm so as to alleviate the three main communication problems at the topological, Medium Access Control (MAC) and routing layers. We conclude that the sensors should move orderly where same neighbors are maintained in the neighborhood to prevent routing breakages. Inter-group and multi-hop communications should be minimized. They are taken into consideration in the design of the dissemination protocol of our algorithm.

In our comparative study, we compare the performances of the following using relative global error and total energy consumption: three versions of our cooperative algorithm (cooperative, cooperative-delta and cooperative-orbital harvesting), mobile sensors deployed in Equally Distributed Grid (EDG), three types of independent methods (Broyden-Fletcher-Goldfarb-Shanno, Random Waypoint and our independent delta-harvesting) and static sensors. Our simulation results show that cooperative-orbital algorithm outperforms others. It reduces an average of 738% (with a range of 625% to 885%) more error than mobile sensors deployed in EDG and 35-314% more error than independent methods by consuming 74-81% lesser energy. Our method also has a resource utilization efficiency of 250 times that of static sensors.

In our stability study, we show that the following two methods improve the robustness of optimization: incorporation of an independence phase in our algorithm and division of a group into smaller groups. Therefore, the division of a group into smaller groups has three benefits: easy deployment in difficult terrains, robust communications and stable cooperation. Moreover, we show that our tracking mechanism is stable and the performance is robust against non-ideal communications and sensor failures.

Finally, we have five research contributions. In the optimization mechanism of the algorithm, we adapt the pseudo-Newton algorithm and make four improvements to it as follows: adaptive cooperative search goals in optimization, local RBF interpolation in estimations, dissemination to mitigate the initial value problem and the concept of orientation stabilization to provide adaptive stabilized search direction. Our fifth contribution is the adaptation of the dynamic clustering technique to track continuous distribution robustly.

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Description
1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
AODV	Ad Hoc On-Demand Distance Vector
AWGN	Additive White Gaussian Noise
BFGS	Broyden-Fletcher-Goldfarb-Shanno
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DS/CDMA	Direct Sequence Code Division Multiple Access
EDG	Equally Distributed Grid
ERC	Equal Ratio Combining
FIFO	First-In-First-Out
GPS	Global Positioning System
LDM	Local Delaunay Map
LHS	Left hand side
MAC	Medium Access Control
MANET	Mobile Ad-Hoc Network
MAI	Multi-Access Interference
MRC	Maximal Ratio Combining
PMM	Probabilistic Mobility Model
RHS	Right hand side
RWM	Random Waypoint Mobility
RBF	Radial Basis Function
RWMM	Random Walk Mobility Model
SLAM	Simultaneous Localization and Mapping
TDMA	Time Division Multiple Access
UWA	Underwater Acoustic
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
i.i.d.	Independently and identically distributed
r.m.s.	Root mean square
s.t.	Such that
w.r.t.	With respect to

List of Notations

Notation	Description
c_{ab}, c_{link}	Connectivity Probability between two nodes: *a* and *b*
c_a	Average Connectivity Probability of a node *a*, with any nodes
N, N_s	Number of nodes in the terrain
R_{ab}	Euclidean distance between two nodes: *a* and *b*
S	The length of the square region used in the connectivity analysis expressed in integer number of steps
(x_a, y_a)	Cartesian coordinate of node *a*
$\pi_{x_a}\pi_{y_a}$	Stationary Position probability of node *a*
S(M,λ)	Normalized MAC throughput used in the throughput analysis
M	Number of nodes in a one-hop network neighborhood
λ	Offered Traffic Load
$\pi_i(M,\lambda)$	Stationary probability distribution of the backlogged node
$P_s(i,\lambda)$	Probability of successful packet transmissions given *i* backlogged nodes
$\bar{I}_i$	Average idle period in the channel given given *i* backlogged nodes
T_p	Packet transmission duration
$\overline{\Phi}$	Average number of neighbors
G_{eff}	Effective offered load
P_{MAI}	The probability that a packet is successfully modulated in the presence of Multi-Access Interference in DS/CDMA
P_{RS}	The probability that a packet is successfully received in the Receive state of the MAC protocol
$p_i^{(k)}$	The position of sensor *i* in the k^{th} time step. Sometimes, it is expressed in Cartesian coordinate form $\left(x_i^{(k)}, y_i^{(k)}\right)$.
$\theta_i^{(k)}$	The measurement made by sensor *i* in position $p_i^{(k)}$ in the k^{th} time step.
$s_i^{(k)}$	The state vector of sensor *i* in the k^{th} time step. It is defined as the concatenation of $p_i^{(k)}$ and $\theta_i^{(k)}$. $\therefore s_i^{(k)} = [p_i^{(k)}, \theta_i^{(k)}]$.
$C_{sn}^{(k)}$	The set that represents the states of the sensors belonging to the same cooperative group in the k^{th} time step.
$\Delta p_i^{(k)}\left(C_{sn}^{(k)}\right)$	This is the position control function in the k^{th} time step. It takes $C_{sn}^{(k)}$ as the input and computes the amount of adjustment to be added to the current position, $p_i^{(k)}$ in order to obtain the next position.
$V_{i,sn}^{(k)}$	The set that represents the states of the Voronoi neighbors of sensor *i* in the k^{th} time step, exclusive of sensor *i*.
$LA_i^{(k)}$	The local area of sensor *i* in the k^{th} time step.

$\frac{\partial LA_i}{\partial p_i}(k)$	The first derivative of local area of sensor i in the k^{th} time step. We also denote it as $y(p_i,k)$ for clarity of presentation. That is, $y(p_i,k)=\frac{\partial LA_i}{\partial p_i}(k)$.
$\frac{\partial \theta_i}{\partial p_i}(k)$	The first derivative of sensed value θ_i of sensor i in the k^{th} time step.
$\frac{\partial^2 \theta_i}{\partial p_i^2}(k)$	The second derivative of sensed value θ_i of sensor i in the k^{th} time step.
u_{goal}	Goal function
D_e	Distortion error metric
K_u	Control gain
V	Volume of the tetrahedron
A, B and C	Areas of the triangle projection of the base of the tetrahedron constructed from the four points representing the state information of the four sensors: (x_i, y_i, θ_i), (x_1, y_1, θ_1), (x_2, y_2, θ_2) and (x_3, y_3, θ_3)
$g(k)$	Gradient of V
$H(k)$	Hessian of V
$\nabla \theta_i^{(k)}$	Gradient of temperature of node i computed at the k^{th} time step
$\nabla^2 \theta_i^{(k)}$	Gradient of temperature of node i computed at the k^{th} time step
$\mathbf{I}_n$	n × n Identity Matrix
$\lVert \bullet \rVert$	2-Norm (Magnitude) of the Vector
$\lvert \bullet \rvert$	The matrix determinant. For scalar, it evaluates to the absolute value.
$\varphi(\lVert p - p_h \rVert)$	Radial Basis Function. p_h is a known position. p is the position of interest. We want interpolate (estimate) the temperature at position p
Φ	Interpolation matrix, used in RBF interpolation.
$\underline{\theta}$	Interpolation temperature vector containing all the known temperature
$\underline{w}$	Interpolation weight vector
σ	RBF constant, usually set to a large value for smooth interpolation
D_{ij}	Directional gradient pointing from point i to j
u_{ij}	Unit directional vector pointing from point i to j
D_{st}	Steering direction
u_{st}	Unit steering direction
ε_p	Mean location error
ε_θ	Mean temperature error
$\Delta p_i^{(k)}$	Computed change in position
$\Delta p_{i,st}^{(k)}$	Computed change in position stabilized by u_{st}
$\Psi(\omega)$	An approximate distribution interpolation using the cubic spline function available in MATLAB. ω is the spacing of the known sampling points
$\xi(k)$	Relative global error computed at the k^{th} time step

$E(k)$	Energy consumption per sensor computed at the k^{th} time step
h	The computation interval used to compute the relative global error
σ_{max}	Maximum separation between the hotspot and the tracking cluster
T_0	Total delay in sensor response
T_{comm}	Communication delay
T_θ	Measurement delay of the thermometer
S_{data}	Data throughput
N_{hops}	Maximum number of hops by the data packet to reach destination
V_s	Maximum velocity of the sensor
V_h	Maximum velocity of the hotspot
m_s	Mass of the sensor
u_f	Coefficient of friction
ϕ	Angle of deviation used in the orbital harvesting heuristic

List of Publications

P1. C. H. Mar and W. K. G. Seah, “An analysis of connectivity in a MANET of autonomous cooperative mobile agents under the Rayleigh fading channel,” *Proceedings of the IEEE 61st Semiannual Vehicular Technology Conference, Spring 2005*, Stockholm, Sweden, May 30- Jun 1, 2005, vol. 4, pp. 2606-10.

P2. C.H. Mar and W.K.G. Seah, “DS/CDMA throughput of multi-hop sensor network in a Rayleigh fading underwater acoustic channel,” *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, Vienna, Austria, Apr 18-20, 2006, vol. 2.

P3. C.H. Mar and W.K.G. Seah, “DS/CDMA throughput of multi-hop sensor network in a Rayleigh fading underwater acoustic channel,” *Concurrency: Practice and Experience*, vol. 12, no. 6, pp. 1129-40.

P4. C.H. Mar, W.K.G. Seah, K.M. Lye and Ang H Jr. Marcelo, “An Energy Efficient Cooperative Optimal Harvesting Algorithm for Mobile Sensor Networks,” *Proceedings of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, Cannes, France, Sep 15-18, 2008.

P5. C.H. Mar, W.K.G. Seah, K.M. Lye and Ang H Jr. Marcelo, “Robust Cooperative Data Harvesting Algorithm for Mobile Sensor Networks under Lossy Communications,” *Pending Submission to IEEE Transactions on Systems, Man, and Cybernetics*.

This page is intentionally

left blank.

Chapter 1: Introduction

This thesis is a report on the development of our cooperative control algorithm for the mobile sensors to optimize the harvesting of spatial environmental information with four realistic constraints: limitations of wireless communications, limited supply of energy and sensor resources and to a lesser extent, difficult terrains. The algorithm is inspired partially by nature [1][2] and draws upon the principles from an eclectic mix of cooperation [1]-[4], optimal control [5][6] and statistical decision theories. The following is presented in this chapter. In section 1.1, we describe the background and context of the research. In section 1.2, we specify our research problem. In section 1.3, we enumerate on the significance and contributions of our research. In section 1.4, we justify our use of mobile sensors instead of static sensors in terms of advantages gained. In section 1.5, we present an overview of the methodology used to solve our research problem. In section 1.6, we outline our research scope and aim and breakdown each aim into several objectives to be attained in this research. Finally, in section 1.7, we present the overall organization of this thesis.

1.1 Background and Context

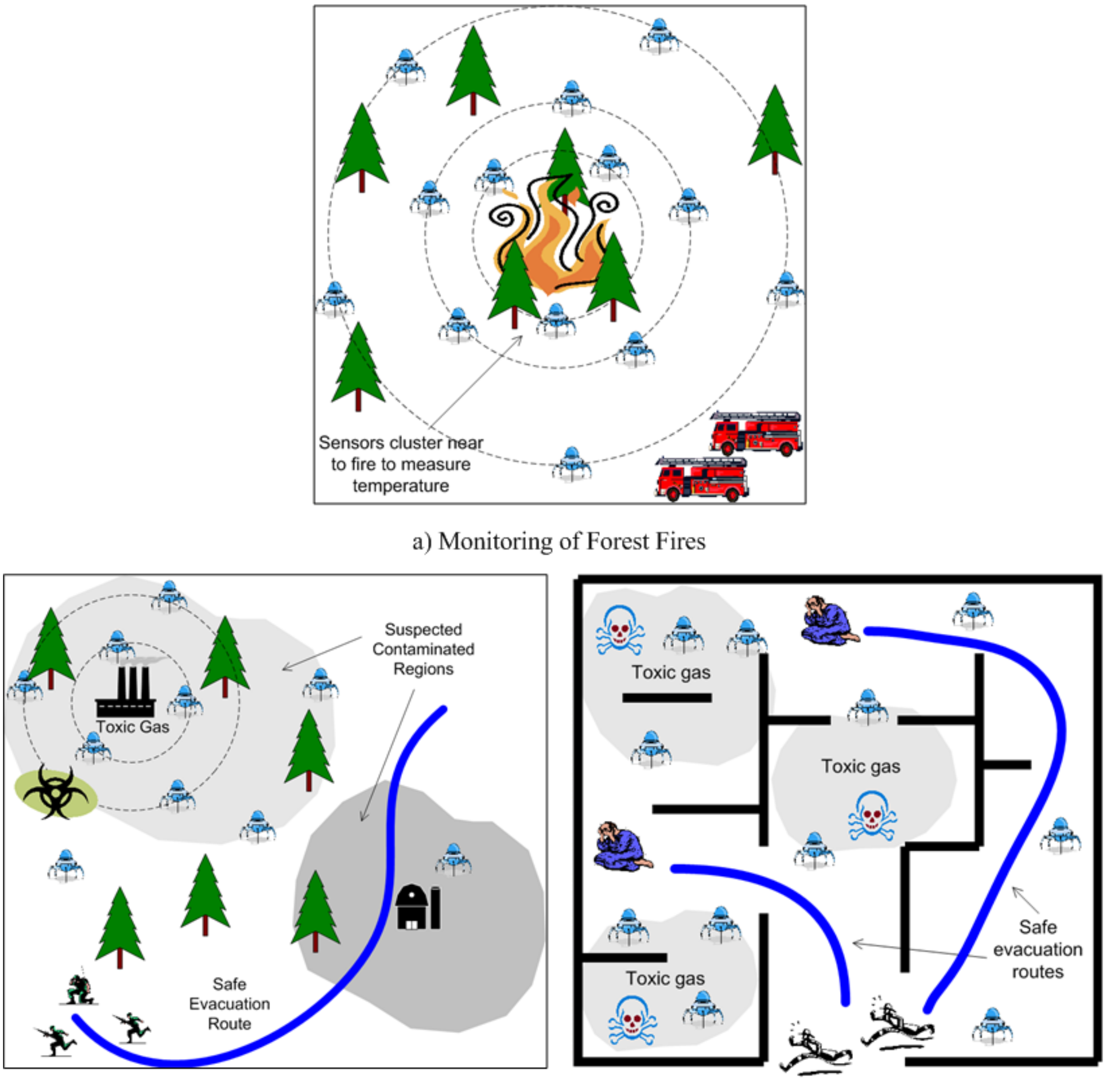
The rapid research and technological advances in wireless communications, sensors and actuators have created exciting and innovative ways of using them that we have never seen before. We envisage a near future where the seamless integration of the abovementioned technologies and devices can make us understand our world better and a safer, efficient and greener place for us to live in. However, many challenges lay ahead, both within each field and in the integration of the fields of research. In the areas of wireless communications, we have challenges ranging from connectivity and reliable communications in the networks due to poor fading channels to security of the networks. In the areas of wireless sensors, challenges typically originated from the paucity of two basic sensors resources: communication bandwidth and energy. Recently, we also witness new fields of research which involved creating smart autonomous actuating devices and robots that can adapt their behaviors according to time-varying sensory inputs. Within these wide overarching research concerns lay our research interest.

In recent years, there is an increasing number of research problems related to the deployment of Wireless Sensor Networks (WSN) [7]-[14][P2][P3] in diverse environments to measure environmental data. These data represent physical quantities that emanate from sources and are diffused in space. For our research, we focus on the use of Mobile Sensor Networks [15]-[20] to harvest such data in an optimal manner so that quality information can be extracted from them. Mobile sensors are sensors that are mounted on vehicular platforms, which could either be land, sea or air based. Thus, they are capable of changing their positions adaptively based on either changes in the topology (for example, due to failed sensors) or internal states of the sensors (for example, low power) or explicit commands from a command centre. Hence, they are more versatile than static sensors. For example, they can be programmed to automatically return to a collection point when they accomplish their mission or when their batteries need to be recharged. Static networks are onerous to gather for disposal or redeployment especially when the sensors are deployed in large quantity in dense vegetations, seabed or hazardous environments. In the long run, battery leaks from uncollected sensors can cause pollutions. However, mobile networks are usually deployed at lower node densities with equal spacing [15]-[18]. As a result, the reconstructed distribution maps are highly distorted and significant amount of post-processing is required to enhance the quality of collected data.

Our networks are to be deployed in environments that are either hazardous or impossible for human intervention. In the future, we believe that many novel applications in the areas of scientific monitoring and disaster management can germinate from such a research. For example, scientists who place high premiums on high quality experimental data to confirm their hypothesis and theoretical models in their quest to unravel the mystery of nature will find such harvested data valuable. Also, in search and rescue scenarios such as fire outbreaks or toxic gas explosions either in outdoor or indoor environments, the use of such data can facilitate operational planning, deployment of human rescuers and subsequent evacuations of casualties. Highly distorted maps may endanger the lives of rescuers. Another possible application is the monitoring of the toxic chemical pollution and the direction that it is spreading. Notice that in all the abovementioned applications, we are interested in both the locations of the sources and their effects on their surroundings. In figure 1.1a to 1.1c, we present three applications for our novel optimal harvesting mobile sensor network.

Figure 1.1a shows the use of our mobile sensor network to monitor forest fires. A fire has occurred in the centre of the figure. As a result, the sensors move in and cluster around the fire to monitor the ambient temperature. Notice that the sensors tend to cluster more tightly when they are nearest to the fire. This is because the temperature gradient is steepest when at the centre. This approach allows us to minimize the distortion error in the measurements given the finite number of sensors and hence ensure high fidelity in the reproduced information. By allowing the sensors to move, we have the advantage of using lower quantity of sensors to achieve the same quality of information as static sensors. If the fire starts to move, the sensors can cluster around and track the fire.

Figure 1.1b shows a military application during biochemical warfare. In the scenario, two regions have been identified as potentially contaminated with toxic biological gases, probably through prior espionage and satellite mapping. The mobile sensor network is deployed to monitor the concentration level of the toxic gas in the two regions. A safe evacuation route is then chosen for the infantry based on which region has the lowest concentration level of toxic gas and direction of movement of the gas.


Sensors cluster near to fire to measure temperature
Suspected Contaminated Regions
Toxic Gas
Safe Evacuation Route
Toxic gas
Toxic gas
Toxic gas
Safe evacuation routes

a) Monitoring of Forest Fires

b) Biochemical Warfare

c) Toxic Gas Explosion

Figure 1.1: Three possible applications

Figure 1.1c shows the use of mobile sensor network in the search and rescue mission in an indoor environment. Here, an explosion in a chemical factory has caused toxic chemical gas leakages in the interior. Time is of the essence and casualties have to be searched and found without endangering the lives of the rescuers. A mobile sensor network is rapidly deployed to measure the concentration level of the toxic gases in the interior. The data is then fed to a command centre to plan the safest evacuation routes for the rescuers to search and evacuate the casualties.

In the greater scheme, we envisage a vast network of self-operating sensor clusters, with mobile routers known as helpers acting as intermediaries to maintain network connectivity such as those described in [8]. Such network can be deployed in vast terrains with many obstacles and barriers. The formation-controlled clusters can initially comb the vast terrain in a systematic and incremental manner during the exploratory phases. Once potentially interesting areas have been detected, the individual clusters can settle down and execute the optimal data harvesting. An example of a network used for monitoring chemical pollution as shown in figure 1.2.

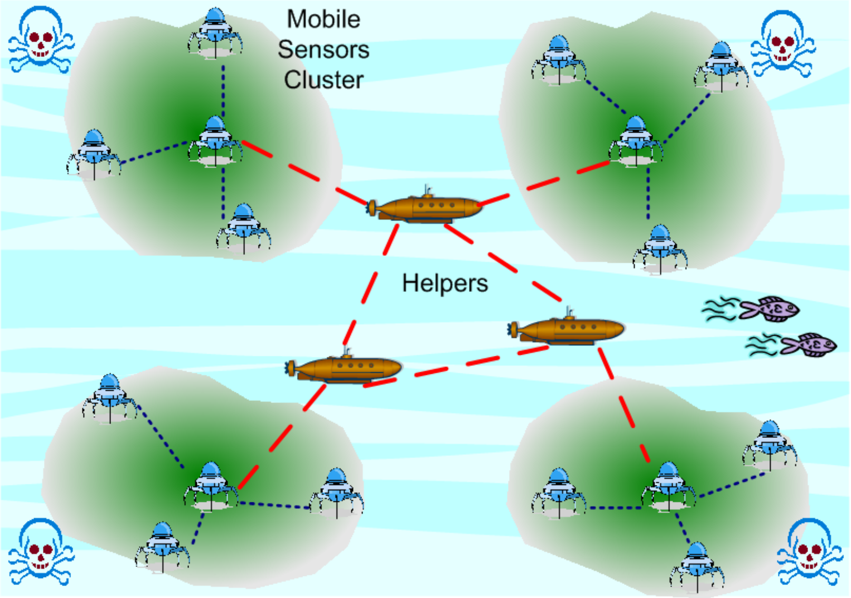
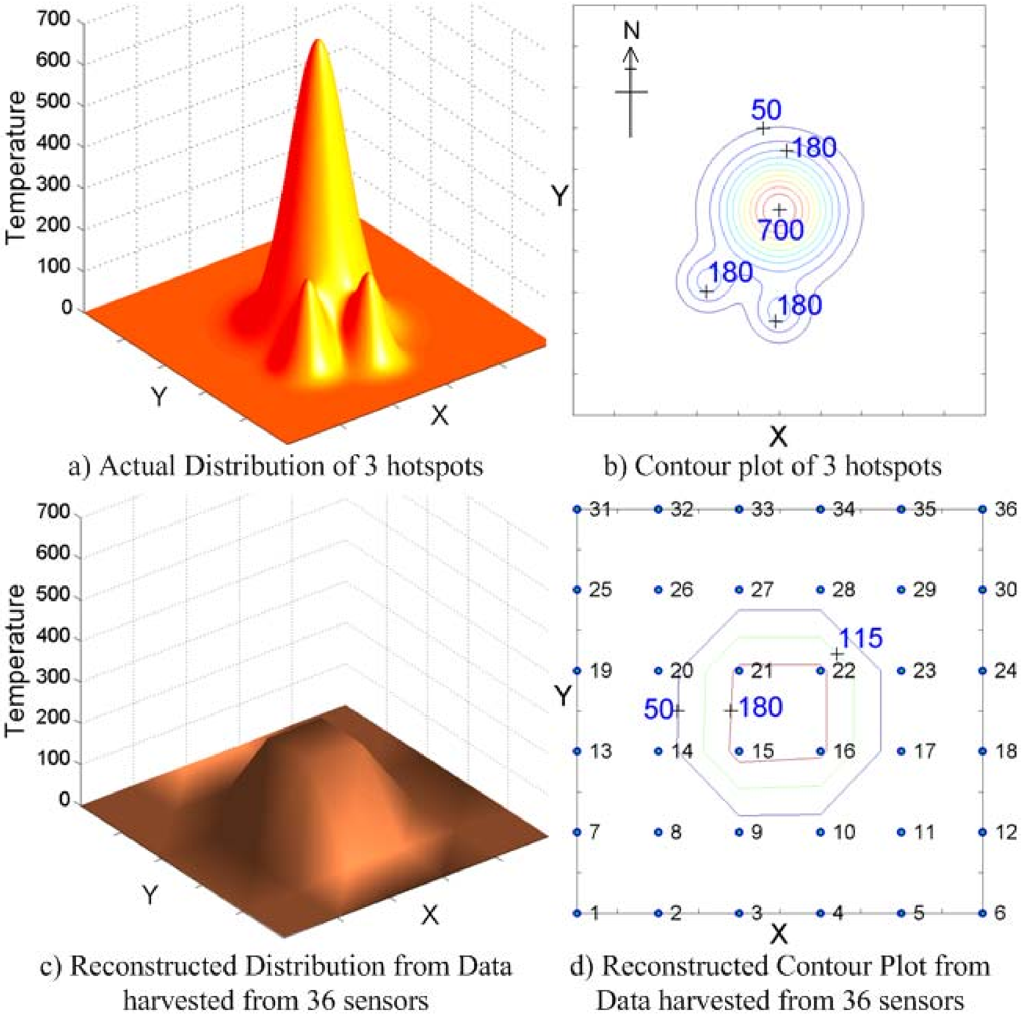


Figure 1.2: Vast oceanic mobile sensor network

1.2 Research Problem

In our research, we want to use a group of cooperative mobile sensors to harvest data from our environment. The data which are associated with the location information can then be used to construct an environmental map of the distribution. Given the sensor, energy and communications resources constraints, we want to optimize their use by placing them in a manner that the data harvested are of high information content with minimum amount of movements and communications. Data with high information content can be used to construct the environmental map with minimal distortion. To better appreciate the problem, we discuss using the forest fire scenario shown in figure 1.3.

In figure 1.3, we show an example of a forest fire that has started to spread its destruction from the center of the terrain. Two smoldering dry bushes have formed at the southern region. This combination causes the fire to move more towards the southwardly direction. The top two sub-figures show the actual temperature distribution and contour plots. We suppose that 36 equally distributed sensors monitor this terrain as illustrated in figure 1.3d. The data harvested are used to reconstruct the two bottom subplots. From the bottom distorted contour plot, the combination of: low maximum temperature of 180°C, the extent of the destruction and the two missing smaller southern hot spots suggest that a recent fire has almost run its course and exhausted its destructive power. It also suggests that the fire spreads symmetrically from the center. If these subplots are used in fire fighting planning, it surely leads to complacency, especially if there are other hotspots in the vicinity to draw attention to. It may also lead to deployment of firemen in the wrong northern location of the terrain to thwart the spread. In this example, we can never extract the distributions of the two smothering bushes from the harvested data, even with post processing.



a) Actual Distribution of 3 hotspots

b) Contour plot of 3 hotspots

c) Reconstructed Distribution from Data harvested from 36 sensors

d) Reconstructed Contour Plot from Data harvested from 36 sensors

Figure 1.3: Forest fire scenario

1.3 Significance and Contributions of Our Research

There are five significant contributions from our research.

Our distributed control algorithm consists of two optimization phases: cooperative and independent, and a tracking mechanism.

In the development of the cooperative phase, a novel approach of using pseudo-Newton method with cooperation is used to propel the sensors rapidly into the optimal positions in an energy-efficient manner [P4][P5]. We make four contributions in the area of cooperative optimization by developing a cooperative version of pseudo-Newton method for our purpose as follows:

1. Optimal placements require the sensors to spread out and position themselves in areas of high curvature where the gradients have different values. Independent Newtonian methods search for a fixed goal–positions of zero gradients. Even if we assume that we can know the values of the gradients to search for in advance and modify the independent methods to handle fixed non-zero gradients, the sensors using the independent methods still cannot spread out properly as they tend to overlap each other in their search and end up chasing after same goals. Therefore, we introduce a novel improvement on the method where the search for positions of high curvature is adaptive and cooperative. It is cooperative because the current position of the sensor is also influenced by the current state information of the neighbors. Consequently, the sensors are better spread out while optimizing and there are no chasings after the same goals among the sensors.
2. Independent pseudo-Newton methods perform badly in harsh environments because of estimation errors incurred due to localization noise. This is exacerbated by the accumulation of past errors in the computations which causes the sensors to persist in the erroneous directions even though current estimates are accurate until the influence of past information has faded in the computations. Therefore, we introduce the memory-less local Radial Basis Function (RBF) interpolation [21][22] to estimate the gradient and hessian values. This is to eliminate the adverse memory effect in harsh environments.
3. The initial value problem in independent optimizations in which the rate and probability of convergence are dependent on the initial position is more severe

for our application. This is because we cannot make a good starting guess for the initial positions of the sensors as we have no advance knowledge of the actual distribution. Therefore, we develop a dissemination mechanism to mitigate the initial value problem.

4. The fixed line search used by some independent methods such as BFGS to stabilize the search is inefficient as it introduces rigidity in the search. In a line search approach, after a direction is determined, the search is conducted along the straight line until a local minimum or maximum point is located. Only then will there be a change of direction. Therefore, we develop the concept of orientation stabilization in which the stabilized direction is adaptive to current states of the neighbors and may vary from one iterative step to another.

Finally, our fifth contribution is from the development of a robust tracking mechanism for our algorithm.

5. We contribute by applying the principle of dynamic clustering onto mobile sensor networks for tracking the continuous distribution. Dynamic clustering was previously used in static sensor network to track discrete targets [9].

1.4 Advantages of Mobile Sensor Network

From our literature survey in chapter 2 on WSN, we are able to identify five advantages that Mobile Sensor Networks offer compared to traditional static sensor networks as follows.

First, a mobile sensor is reusable. An attractive feature that arises from the mobility of the sensors is the ability to command the sensors to gather at a collection point either when we need to send them to another mission or to recharge them. This differs from static sensors that are usually permanently deployed in their environment. Environmental concerns arise when the spent static sensors are not collected or difficult to collect, for example, in a densely forested area or under the sea bed. This is exacerbated by the fact that static sensors are deliberately dispersed with much higher node density than required for minimal connectivity to compensate for uneven dispersion and also for redundancy against sensor failures. The components such as batteries of the spent sensors could pollute the environment. Although mobile sensors are more costly than static sensors, in the long run, it is cheaper to use mobile sensors if the applications require us to frequently re-deploy our sensors. Furthermore, in our times of global warming where environmental costs of cheap disposable plastic bags have caused many countries to restrict or ban their use in place of more expensive, reusable grocery bags, the cheapness of static sensors is a weak justification for their use.

Second, mobile networks have less network problems in the form of congestion or starvation due to lower density in deployment. Due to high density deployments in static sensor networks, congestion in the static sensor networks is an ongoing research issue which we discuss further in chapter 2. Congestion reduces the effectiveness of using the static networks for real-time monitoring due to delayed or lost data packets. It also increases the probability of starvation where a few more aggressive nodes are able to horde the communications for continuous transmission of data. Both congestion and starvation have the secondary effect of degrading the performance of static sensor localization.

Third, mobile sensors can localize with higher accuracies using robotic localization. This is because unlike static sensors, mobile sensors can use heterogeneous fusion of dissimilar measurements (odometry, sonar and laser scanners, etc) to improve the accuracy of its localization. Since reconstructing a high quality distribution require high localization accuracy, in reality, the performance of the static sensor network will be much worse. In real life, another way to achieve even higher accuracy in determining positions is to use Global Positioning System (GPS). It may be argued that the cost is too prohibitive for sensors. However, it must be noted that historically, the cost of hardware is never an insurmountable issue whenever there are huge commercial demands. Commercially, GPS has already been integrated into many small handheld devices such as palmtops and mobile phones, and are available in many modern motor vehicles. In fact, the cost issue is the best argument for the use of mobile sensors instead of static sensors for two reasons. First, based on our simulation in chapter 6, static sensors have to be deployed at a node density that is 250 times greater than mobile sensors using our cooperative algorithm in order to achieve the same level of performance. Since we need to install GPS on every sensor, the total cost of GPS installation on a static sensor network will also be 250 times greater than our equivalent mobile sensor network. Second, as discussed above, mobile sensors have high reusability. Most often, static sensors are deployed permanently in the environment and many of them are lost due to difficulties in recovering them. As a result, installing GPS on static sensors are considered to be an investment only for one time usage, which does not make economic sense.

Fourth, we can control the mobility of mobile sensors based on environmental input to extract data of high information content. Static sensor networks usually require high density of sensors to achieve high quality measurements because of uneven dispersion at deployment and inability to adjust positions in response to environmental changes. Current state of mobile sensor technology focuses on maintaining maximal coverage with equal spacing [15]-[18]. There are no feedback mechanisms to adjust their positions to improve the quality of their measurements.

Fifth, maintaining connectivity in traditional static sensor networks is an issue due to uneven terrain, sensor failures, channel conditions and imperfect methods of sensor deployment. The simplest approach is to deliberately disperse the sensors with higher node density than required to maintain network connectivity. In the process, the redundant nodes cause more problems such as high node interferences and contentions that lead to network congestion. Special data dissemination techniques are then required to deliver the data in a timely manner to a sink node for accurate reconstruction of the distribution. Mobile sensor networks [15]-[18] do not require redundant nodes to maintain global connectivity. They require only the sensors to move in a coordinated manner such that the topological relationships between adjacent neighbors are preserved. This is a special property of the mobility class. The unique characteristic is that in spite of the constant movement of nodes at the physical plane, the Delaunay graph that connects the adjacent neighbors in the topological plane is invariant with time (see figure 1.4). An example of a mobility class that exhibits this property is formation controlled mobility. This property is shown to be a desirable quality based on our throughput analysis of autonomous agents with random mobility. The reason is uncoordinated movements increase route breakages due to disconnections and changes in intermediate nodes which in turn tend to decrease the capacity of the network. Note that the property does not guarantee connectedness because the closest adjacent neighbors may be so far apart that they are out of communication range. However, it simplifies the problem of maintaining global connectivity by reducing it into a problem of maintaining local connectivity with the same neighbors. That is, the nodes need only to ensure that they are connected to their closest neighbors. These are neighbors that surrounds them.

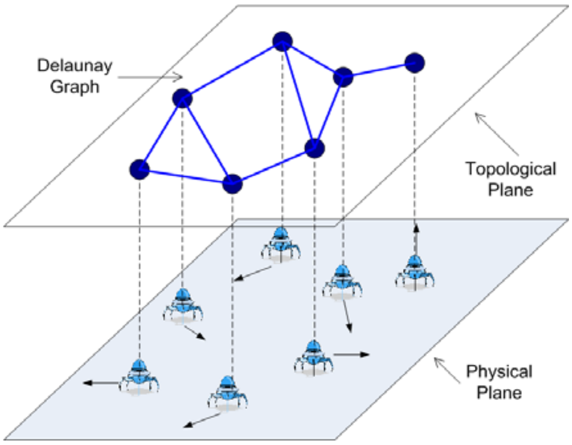


Figure 1.4: The invariance property of Delaunay graph for coordinated movements

A simple scenario in figure 1.5 is used to illustrate this concept. In the scenario, the scouts return to their camp at night after an evening trek. They form a line formation in their movements. Each scout needs only to maintain visual contact with the same neighbor to ensure that the whole line formation remains connected.

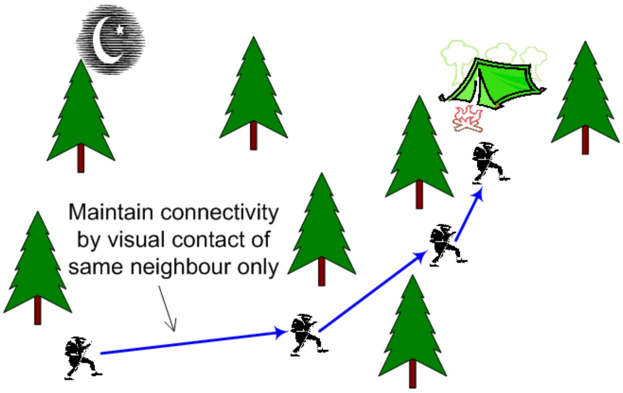

Maintain connectivity by visual contact of same neighbour only

Figure 1.5: Achieving global connectivity by maintaining local connectivity

1.5 Methodology

We use an integrative approach to solve our problem by drawing upon an eclectic mix of principles from various theories such as: cooperation [1]-[4], control [5][6] and statistical decision theory. In our search for a solution, we also draw our inspirations from nature [1][2] and embrace the use of biological principles in our solution. A two-phase method is adopted in order for us to derive our solution.

In order to cooperate, the sensors require wireless communications to exchange cooperative shared information. Therefore, in the first phase, we survey the literature on wireless communications, MANET and sensor networks, and subsequently perform theoretical analyses in order to better understand the principal difficulties and challenges that arise when a network consisting of mobile nodes are deployed in a harsh physical environment. The limitations of wireless communications and networking are taken into consideration in the design of our algorithm.

In the second phase, we design our main algorithm using the top-down approach and by considering the various aspects that will affect our algorithm, inclusive of those insights gained from the first phase. In order for us to use the scarce energy and sensor resources economically, we leverage on cooperation to perform optimal harvesting. We then design our simulation in order to conduct performance studies on the algorithm and identified further improvements to the algorithm.

1.6 Research Scope, Aims and Objectives

The research scope is to develop a distributive cooperative control algorithm to control the movement of the mobile sensors in order to minimize the distortion error when the harvested spatiotemporal environmental data are used to construct the distribution. This is to be done without losing the sensing coverage of the region. To elaborate, each sensor has a finite sensing area due to its sensing range. The sensing coverage refers to the union of the sensing areas of the sensors. The design is to take into consideration the following four realistic constraints. First, the spatiotemporal disconnections of wireless communications as the result of: mobility of the sensors, poor channel conditions in the harsh physical environment and network contentions at the MAC layer due to increasing data traffic load. Second, we have only a finite number of sensors deployed. Third, there is limited energy supply on each sensor. Fourth, difficult terrains where there are physical obstructions and obstacles such as walls.

The following are our research aims and the objectives that we desire to attain for each aim:

1. To investigate the principal difficulties and challenges in wireless communications and networking of MANET and sensor networks.
    a. Survey the connectivity issues in the networks and the various strategies used to mitigate the problem.
    b. Survey the issues in the MAC layer of the networks and the various strategies used to mitigate the problem.
    c. Survey the issues in the routing layer of the networks and the various strategies used to mitigate the problem.
2. To analyze the performance of a MANET in a harsh environment with respect to various parameters.
    a. Theoretically analyze the connectivity of the network taking into consideration the mobility of the nodes and poor channel conditions.

   b. Theoretically analyze the throughput of the network taking into consideration the mobility of the nodes, poor channel conditions and the effects of the MAC and routing layers.

3. To develop the main distributed cooperative control algorithm
   a. Survey the theories from general literatures and literatures related to autonomous mobile robots that are directly relevant to the development of the algorithm such as those related to cooperation, control, mathematical interpolations and decision making in an imperfect knowledge scenario.
   b. Develop the main algorithm for the two-dimensional (2D) network topology scenario.
   c. To design the simulation to study the performance, identify the weakness in the main algorithm and further improve and refine on the main algorithm
      i. Design and conduct simulations for the 2D network topology.
      ii. Further improve and refine on the main cooperative algorithm based on the weaknesses identified during the simulation studies.

1.7 Organization of the Thesis

The thesis is organized as follows. In the next chapter, we discuss our literature survey on on-going research related to wireless communications, MANET, static and mobile sensor networks. In chapter 3, we present the insights gained from our preliminary study and theoretical analyses of various MANETs operating in realistic conditions. Subsequently, those insights gained are used to aid us in the design of our distributed cooperative control algorithm. In chapter 4, we present the two-phase algorithm for the two-dimensional topology scenario. First, we describe our general design considerations. This is followed by detailed discussions of the cooperative and independence phases and the tracking mechanism of the algorithm. The theoretical stability of the algorithm is also analyzed. Finally, we examine the design of our algorithm from the theoretical perspective. In chapter 5, a comprehensive simulation study is carried out. First, we conduct the comparative performance study using two performance metrics: relative global error and total energy consumption per sensor under different scenarios. In the comparative performance study, we compare our three cooperative harvesting algorithms: cooperative, cooperative-delta and cooperative-orbital harvesting with three independent harvesting methods: BFGS, Random Waypoint Mobility (RWM) and independent delta harvesting heuristic. Moreover, we also compare all the abovementioned cooperative algorithms and independent methods with mobile sensors deployed in Equally Distributed Grid (EDG) and static sensors. Second, we examine the optimization and tracking stabilities. Third, we examine the effect of non-ideal communications on the performance. In the final chapter, we conclude our work, where we reiterate and examine all the objectives set up in chapter 1. Additionally, we also explore possible future directions for our work.

Chapter 2: Literature Survey

In this chapter, we present our literature survey. Our survey focuses on three different areas that are relevant to our research problems. The survey on MANET in section one helps us better understand the issues in communications and networking that affect our problem. In section 2.2, we examine the general issues that affect the monitoring and sensing performance of WSN. In section 2.3, we survey on coverage control of mobile sensor networks. Finally, we conclude the chapter.

2.1 Mobile Ad-Hoc Networks

MANETs are multi-hop communication networks that are built on the ad-hoc basis. That is, it is built on-the-fly and torn down rapidly without prior planning, configuring and organizing. Some examples of potential applications are: mobile conferencing, vehicular communication network, emergency and disaster communication services and military networks. It is also most suited for networking in mobile robotic networks [20]. As the name implies, the nodes are mobile, hence the topology of the network changes dynamically. Another notable feature is that the network has no infrastructure. That is, there are no special nodes such as mail, web or authentication servers within the network that provide centralized networking services. Every node is identical in its networking functions. Many researches are focused on improving and augmenting the capabilities of the MAC and routing algorithms so as to provide seamless, non-disruptive services.

Specifically, the focus of our survey is to identify the problems that deteriorate the throughput of the mobile networks and the various methods used to mitigate the problems. From the survey, we identified four problems as follows. The first problem is poor connectivity due to imperfect wireless channel conditions such as fading, node mobility. The second problem is contentions among the nodes for the uses of the communication channels. The third problem is inter-neighborhood interference which gives rise to hidden and exposed node problems. The fourth problem is, in multi-hop communications, whenever there is a need to establish or repair a route, routing overheads are generated.

We generally define *contention* as the competitive node activity occurring among the neighbors **inside** the one-hop neighborhoods that is required to secure the channel for communications. Contentions among neighbors are usually resolved by either having a central node to coordinate and allocate the channel among them or imposing cooperative self-regulating behaviors among the nodes such as “listen-before-transmit” and back-off when collisions occur. This is performed at the MAC layer. We define *interference* as node activity occurring in the regions immediately **outside** the one-hop neighborhoods that disturb the communications inside the one-hop neighborhoods.

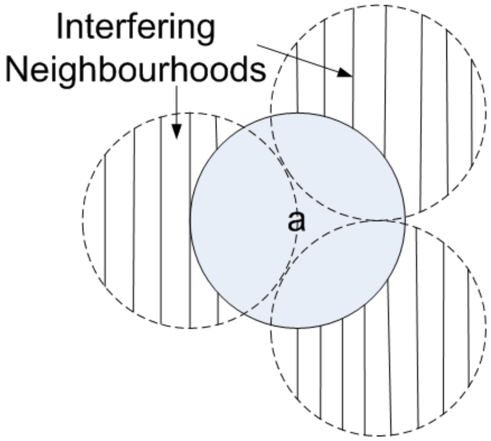


Figure 2.1: Interference in a multi-hop network

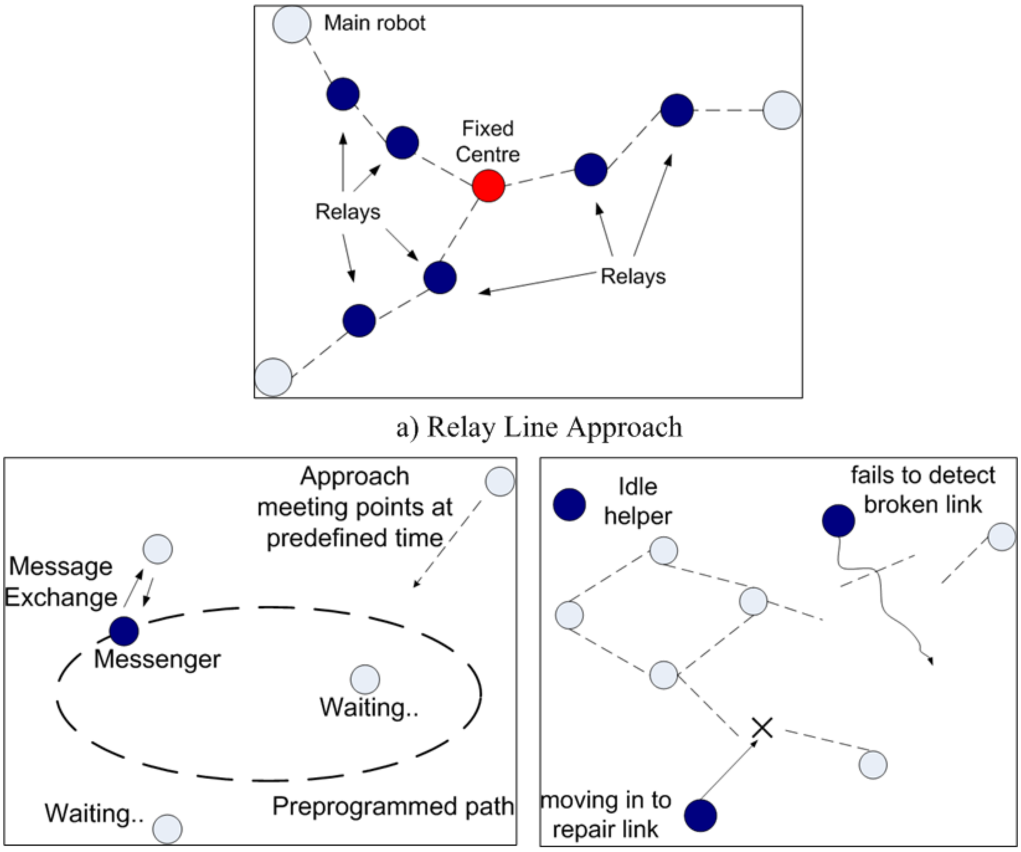
The designers of MAC protocol for MANET face three challenges.

The first challenge is link disconnections due to the unreliable nature of wireless channels (for example, fading and shadowing) [23] and node mobility. The second challenge is the presence of contentions to secure the wireless channel in order to transmit packet. While the first two challenges are not unique and are present also in a one-hop WLAN with a central base station, the third challenge is unique to wireless multi-hop communications. This is the node interference from adjacent overlapping neighborhoods beyond one-hop neighborhood [24][25] as shown in figure 2.1. This type of interference gives rise to the hidden terminal and exposed terminal problems in the literature. Briefly, many MAC protocols such as the popular Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol [25][75][82]-[85] require some form of coordination and cooperation among the nodes in order to improve the efficiency of channel utilization. The only exception is the ALOHA protocol which does not have coordination and cooperation, hence each node transmits autonomously when it has a packet to send. As a result, the ALOHA protocol has the lowest efficiency in channel utilization. CSMA/CA is popular in MANET because the medium control is done in a distributed manner. However, in order to participate in any form of coordination and cooperation, depending on the MAC protocol, the minimum requirement is that all the nodes in the same neighborhood can hear each other. In figure 2.1, the neighborhood that is centered at node *a* is been interfered by 3 overlapping adjacent neighborhoods. We observe that the nodes in the shaded region of the overlapping neighborhoods cannot hear node *a* when node *a* transmits, so they cannot cooperate and remain silent if they have packets to transmit. As a result, we expect this type of interference to reduce the efficiency in channel utilization.

To mitigate the interference, a few approaches have been proposed. In one approach, we allow the nodes to vary their transmission range adaptively. It is known as topology control [28][29]. A second approach is to use a protocol that is more robust to interference of any kinds such as DS/CDMA MAC protocols [24][25][29] which appears to be a promising approach. However, a problem exists in the implementation of DS/CDMA which is traditionally used in mobile cellular networks where there are central infrastructures such as base stations [23] to allocate spreading code and regulates between the transmitting and receiving phases of the half-duplex hardware. To implement DS/CDMA in a multi-hop ad-hoc environment where there is no central infrastructures will require an additional distributed control layer at the MAC which we explore in [P2][P3] and chapter 3. We also analyze the throughput performance between CSMA/CA and DS/CDMA MAC in chapter 3.

Finally, we also examine multi-hop networking as a mean to facilitate the harvesting tasks. A route needs to be established whenever two nodes are several hops away from each other and they need to communicate. This is accomplished by using routing protocols. In a MANET environment where topology changes are frequent, one or several links that formed the route may be broken and as a result, dynamic routings are required. This incurs routing overheads that consume the communication bandwidth, ultimately deteriorating the throughput of the network.

Ad-hoc routing protocols [30]-[44] are the most well researched in MANET. The main challenge in routing protocols is to keep the routes updated because of frequent broken routes. Broken routes in MANET can be due to the change in topology as the nodes move. It could also be due to MAC layer issues such as prolonged unsuccessful link level transmissions when the channel condition is poor or there is interference or contentions are high. This is to be done with as little routing overheads as possible as they consume a fair amount of channel capacity. The protocols can generally be classified as proactive and reactive types. In proactive types, there are periodic route advertisement packets to keep all the route tables updated. However, many of the route updates in the route tables are actually unnecessary. This is especially true when the topology changes are not frequent. To minimize the route overheads, in reactive types such as Ad Hoc On-Demand Distance Vector (AODV) routing protocol [32], routes are constructed on-demand and reconstructed only when it is broken during transmissions.



a) Relay Line Approach

b) Message Ferrying Approach

c) Helper Approach

Figure 2.2: Three different approaches in active routing

Therefore, if multi-hop routing is required in an environment where we have limited communications bandwidth and no control over the node mobility, reactive protocols such as AODV will be more suitable as they attempt to minimize routing overheads, we will investigate further in chapter 3. Clearly, from the discussion, if our topology is invariant with time, there are no routing overheads at steady state as we do not need to reconstruct routes. However, generally for mobile networks, this is not possible as the nodes move with fairly random mobility.

The application of ad-hoc multi-hop networking in mobile robots networks [20] has led to the proposal of active routing [46]-[52]. Active routing uses the fact that we can control the mobility of the nodes to mend or maintain networking routes. In figure 2.2a-c, the dark blue nodes represent the nodes that have their mobility controlled to play the main functions of active routing. In the relay line approach (figure 2.2a), a line of relay robots follows behind a main robot as it moves around. In message ferrying approach (figure 2.2b), a few robots are assigned as postmen. They follow pre-programmed paths to collect and send messages. In one variation, an underwater autonomous vehicle is used to collect or “harvest” information from the underwater sensors and bring it to the surface [51]. The helpers approach (figure 2.2c) has also been proposed. A redundant pool of helpers constantly search for critical links. A critical link is a link that if removed, results in the network being partitioned into two clusters with no communication path from one to another cluster. A depth first search is used to move the helpers to locations where there are critical links.

Furthermore, among the three approaches, only the relay line and message ferrying approach have the invariant topology property as discussed in section 1.4, chapter 1, which is beneficial in minimizing routing overhead. The reason is that only the two approaches require the relaying nodes to coordinate their movements. For the helper approaches, the nodes do not need to coordinate their movements. Therefore, we expect the two approaches to incur the least routing overheads.

Finally, in all examples, the key weakness of the difficulty of multi-hop networking to maintain stable routes in a harsh communication environment with minimal overheads remains. This motivates us to use other forms of communications to facilitate the harvesting tasks.

2.2 Wireless Sensor Networks

WSN consists of cheap miniature wireless networking devices with sensing capability. They are usually deployed in thousands to monitor the environment over a large spatial region. Some of the suggested real-time applications are scientific monitoring, safety and surveillance. They usually send very small data packets by multiple hops to a sink node. Traditionally, many researches focus on WSN with static nodes [9][53]-[68]. However, more recent works look into WSN with mobile nodes [8][12][15]-[18]. The key finding is that there are three main problems that deteriorate their performance. The first problem is poor connectivity due to imperfect wireless channel conditions such as fading and node mobility. The second problem is network congestion due to high node density. The third problem is high localization errors.

Real time monitoring of the environment required timely delivery of sensing data to the sink node. In a network with poor connectivity, data are lost and this leads to unreliable real time monitoring. Poor connectivity is caused by imperfect dispersion in uneven terrain and localized conditions that deplete energy, resulting in early sensors failures. For WSN with static nodes, the main technique is to uniformly disperse the nodes at initial stage with sufficient node density to achieve network connectivity. Early theoretical works focus solely [54]-[59] on the minimum node density required to achieve certain threshold network connectivity. The rationale is that if we know the minimum node density, in actual situation, we can disperse the nodes at a node density much greater than the minimum to guarantee the network connectivity. The two main causes of poor connectivity in real life scenarios that require much higher node density than the minimum during dispersion are as follows:

- *Dispersion at the initial stage*:- Due to uneven geographical terrain and difficulty of controlling the vehicle that is used for dispersing the sensors uniformly, the sensors are not uniformly distributed. As a result, there is a possibility that in some areas the sensors distribution are sparse and the local networks are poorly connected. In the worst case, the local networks in sub-regions can even be partitioned from the rest.
- *Environment*:- Harsh environment can present problems for static sensors. In an underwater environment with strong undercurrents, the static sensors can drift from their original positions and this could lead to changes in network topology, connectivity and coverage area. Excessive drainage of power from communications due to localized channel conditions such as shadowing and fading or high sensing activity and communications contentions can result in early sensor failures. To mitigate the environment effects, redundant nodes are dispersed to reduce the probability of early failures.

Besides the connectivity problem, high node density is also prescribed for good sensing coverage [10] and to reduce localization errors [68]. However, prescribing high node densities as a panacea is not without its side effects. It results in excessive contentions and interference among the many nodes which eventually decreases the capacity of the network. As an example, earlier researchers noted the occurrence of "sensing storms" in monitoring of discrete targets. Sensing storms occur when targets trigger many surrounding sensors within their sensing ranges. As a result, an avalanche of data is sent concurrently leading to congestion, transmission failures and excessive retransmissions. This congestion also affects the performance of localization in static sensor network. The reason is that under ideal communication environments, localization accuracy improves as the node density increases because of the increase in number of neighbors used to determine the position of the sensor. Hence, the localization noise is reduced. However, due to congestion, not all messages sent by the neighbors are received and the average delay in messages arrival increases. In the worst case, the messages arrived beyond the periodic computation intervals. Therefore, the effective number of neighbors communicating with the sensor decreases beyond a critical point as the node density increases. As a result, the localization noise is not decreased and may even increase. Generally, static sensors have poor localization accuracies. Congestion in the networks sets a limit in improving the localization accuracies which is essential for many applications, including ours as it affects the accurate reconstruction of continuous distribution in our application.

The earliest known work to characterize decreases in capacity when node density increases beyond a certain magic number is the simple model developed in [60] that uses the ALOHA MAC protocol. More sophisticated models that use Time Division Multiple Access (TDMA) [69]-[74] and CSMA/CA MAC have been developed [75]. This includes our own works [P1]-[P3] that explore node mobility, different environments (terrestrial vs. acoustic) and the DS/CDMA MAC protocols. To resolve these problems, current research has broadly branched out to three different strategies. The first strategy is to continue with a purely static WSN and find ways to alleviate the side effects. The second strategy is to augment the static WSN with high performance mobile nodes to enhance the connectivity and thus hopefully relax the requirement to disperse the nodes with too high a node density. The third strategy is to use a WSN that is made up of mobile nodes that have their mobility controlled to maintain coverage and connectivity, thus effectively allowing the WSN to operate at the minimum possible node density for network connectivity. We will briefly cover the first two strategies. The third strategy will be discussed in the next section.

One of the characteristics of sensor data is that it is highly spatially correlated. This is especially true in target tracking. Therefore, the data that are transmitted are highly redundant and they may cause congestion in the network during periods of high sensing activity. Congestion in the network affects the punctual arrivals of crucial data to the sink node for decision-making. In the worst case, the crucial data can be lost. High node density in the network increases the amount of spatially correlated data circulating in the network that eventually leads to congestion. In order to resolve this problem, two main techniques are used. The sensing range refers to the average maximum distance that a target can be away from the sensor and still remains detectable by the sensor.

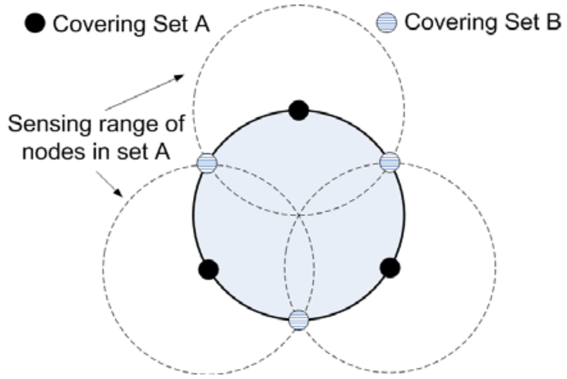


Figure 2.3: Minimum covering set

The first technique is to manage the coverage [10]-[11][61] by a combination of topology control and minimum covering set. The minimum covering set is illustrated in figure 2.3 for a target tracking scenario. There are two independent covering sets A and B that can completely cover the center shaded region in the figure. Therefore, at any time, we only need to switch on the sensors that belong to one covering set thus reducing excessive data transmission. Finally, it is determined theoretically that to ensure connectivity, the communication range must be at least twice the sensing range. The communication range can be adjusted through topology control. The main disadvantage is that this method only works if the sensor has a sensing range. If the sensor is performing measurements at a single point in space, such as temperature measurement, we cannot use this technique. This is especially true if the temperature gradient is not uniformly distributed.

The second technique is to manage the flow of data dissemination [62]-[64] to the sink node through data clustering and aggregation. The idea is to use a hierarchy of intermediate sink nodes to collect regional data and aggregate them as shown in figure 2.4. Aggregation can either be data compression or the averaging operations using filters. Since the regional data are highly spatially correlated, we expect to achieve, for example, high compression ratio in the aggregation process. Therefore, the redundant data circulating in the network is kept to the minimum.

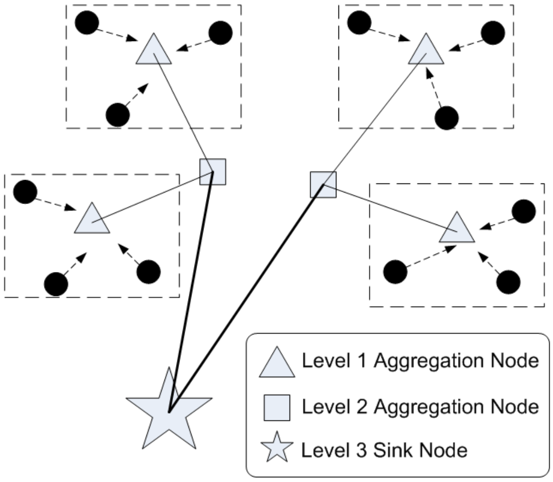


Figure 2.4: Data clustering and aggregation

This approach has several disadvantages. First, the hierarchy of sink nodes introduces single points of failures that reduce the robustness of the WSN. Second, the nodes have to be equipped with better processors in order to perform the data aggregation. Third, network clustering introduces a networking overhead. If the clustering is static, the overhead is a one-off cost otherwise the overhead will be an on-going cost. The benefits of data aggregation can only be fully realized if there is high enough spatial correlation in the data. If the event that we are monitoring varies in space with time such as target-tracking, the amount of benefit that we can reap using a static data clustering is not maximal. One possible approach to realize the benefit is for the cluster to adapt to the environment. For example, in a target tracking scenario, the data cluster has to dynamically form around the target [9] as it moves across the WSN in order to achieve high spatial correlation in the data. In conclusion, in order to reap maximal benefit, we are required to incur additional on-going overhead. Finally, in our application scenario in which we are interested in the continuous distribution of measurements, the aggregation is not beneficial to us. This is because the spatial correlation is low, especially in regions of steep temperature gradients. In short, unlike target tracking in which we are only interested in discrete points in space, in constructing a continuous distribution map over the entire space of interest, every node matters. However, a proper sampling distribution can minimize the distortion.

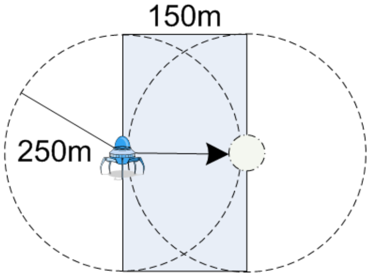


Figure 2.5: Maximum area covered by the mobile node in its search

The second strategy is to augment the static WSN with high-performance mobile nodes [12][65]. Essentially, the general approach is similar to the helpers approach (see figure 2.5) described in the previous section. The high performance mobile nodes attempt to seek out regions of disconnected nodes. They then disperse extra static nodes to enhance the connectivity of the region. In addition, they also act as extra routers in regions of high traffic loads. Currently, this is still a relatively new area of research and its successful development remains to be seen. However, there are three limitations in this approach that have to be overcome in future.

First, the velocity of many present day autonomous vehicles which is about 30 m/s sets an upper limit to the region that the vehicles can cover in their search for failed or disconnected sensors. For example, in figure 2.5, if we can tolerate a delay of 5 s in the disconnection and the vehicle has a sensing radius that is equal to the communication range of about 250 m, even at the most optimistic, the distance that the vehicle can cover is only 150 m which is less than one hop. Routing protocols such as AODV [32] will wait at most 2 s for the MAC protocol to attempt to re-establish the link before it searches for a new route. If there is an alternative route, the whole re-routing process which may involve a search over the entire network for a suitable route will take less than 5 s and during this time, the vehicle can only cover less than one hop in its search. Clearly, for the approach to be a feasible solution, the mobile node must be able to search more than one hop within the acceptable delay tolerance. Therefore, this limits the nodes in our scenario to operate at communication ranges that are less than 150 m. Furthermore, if the WSN is to spread over a large geographical region, in order to achieve a significant reduction in static node density and simultaneously a reasonable delay tolerance, it is likely that many such high performance mobile nodes have to be deployed.

Second, as we decrease the static node density to close to the minimum node density, the role that the mobile nodes play in maintaining the network connectivity becomes more crucial. Ironically, this makes the entire network vulnerable to connectivity problem because the mobile nodes are single points of failures. The approach in [12] is particularly susceptible to this problem. This is because the authors propose to let the high performance mobile nodes carry the main backbone of the data traffic. The idea is to conserve the energy of the static sensors so that their lives can be prolonged. The robustness issue makes the approach unattractive in a military battle scenario where the vulnerable mobile nodes can be deliberately chosen by enemy as targets.

Third, the mobility of the nodes may eventually decrease the capacity of the network. We will discuss this in chapter 3. This is especially true if their movements are uncoordinated and they are frequently chosen as intermediate nodes by the routing protocol. Their movements cause frequent broken routes and route repairs which generate routing overheads. The routing overheads consume a portion of the capacity. The approach in [12] is particularly susceptible to this problem because the main bulk of data traffic passes through unstable routes formed partially from the mobile nodes.

However in these two strategies, the two key environmental-unfriendly weaknesses in static WSN remain unresolved. Although static sensors are cheap and disposable, poor reusability and the need to be deployed in thousands in order to: maintain connectivity, maximize sensing coverage and minimize localization error are especially of great concerns in our times when many countries have already restricted or banned the use of cheap disposable plastic bags in favor of more expensive reusable grocery bags due to increase in global environmental damages. Therefore, we are motivated to survey mobile sensors as suitable substitutes.

2.3 Mobile Sensor Networks

Mobile Sensor Networks have generated much research interests. It is a multi-disciplinary field that draws researchers from diverse areas such as robotic control and networking. We focus on coverage control which is of interest to our research. In coverage control, we control the motion of the mobile sensors to spread out over a region of interest so as to obtain a good sensing coverage. Previously, there is a debate that due to the lower node density of mobile sensor networks, it is unable to provide effective sensing coverage. However, a recent theoretical work [97] demonstrates that mobile sensor networks can achieve coverage that is comparable to static sensor networks by the use of intelligent motion control of the mobile sensors.

Generally, motion control can be centralized or distributed [2]. In centralized motion control, there is usually a coordinating agent that performs the task of ensuring that other agents are moving in order. This approach is not scalable and to solve the scalability issue, a hierarchy of coordinating agents is used to issue the chain of commands when the group gets too large. However, the remedy creates lethargic response and movement of the entire group. The problem tends to worsen as the group gets bigger. Due to the problems in centralized control of a large group of mobile robots, many researches on the mobile sensing coverage of the mobile sensor networks use the distributed approach which is highly scalable. The main idea is that rather than providing a hierarchy of coordinating robots to coordinate the entire group, each robot is programmed with cooperative self-regulating behavior. The idea is biologically inspired and originated from observations of how large insect swarms and flocks of migratory birds are able to coordinate their movements and simultaneously navigate rapidly the whole group through sharp turns. It is observed that the birds do not need to communicate with the leader. Rather they each take reference to the positions of their immediate neighbors when they move. This behavior in their movement is also known as cooperative movement as opposed to autonomous movement. This approach is highly attractive to mobile sensors applications where we need to coordinate the movement of a large group of mobile nodes in respond to rapid changes in the surrounding.

There are many ways to achieve sensing coverage in mobile sensor networks. The simple approaches that use minimum computations are usually very poor in energy efficiency and slow in movement. An example is [98] in which the sensors use a probabilistic mobility model known as *Particle Swarm Optimization*. A pure random mobility method is energy inefficient. In order to overcome the weakness, the model has two parameters which control how far the mobile sensors can move away from the fixed predefined locations in the region of interest. These locations are usually where the routers or fixed sink nodes are located. This solution has two weaknesses. First, network connectivity is not guaranteed. Second, as the sensors are in constant motion, the energy consumption is quite high. Another example using a simple approach is [99], the problem is formulated as sensing holes searching problem. By the exchange of local messages, the sensors eventually discover that there are holes in the region of interest. The sensors then bid among themselves using the *estimated minimum energy consumed to reach the location* as the cost. The sensors that have the lowest bid will then move to the holes. This solution has two weaknesses. First, the communications overheads are quite high. We need more than one message exchange in order to make one move. It is also slow and inefficient. It takes time for the sensors to discover the holes. When a few neighboring sensors move simultaneously to cover one hole, they may leave a hole behind it. As a result, it may result in a never-ending problem of “filling the holes left behind”. A second weakness is that even when the holes are eventually filled up, it is difficult to maintain coverage as some sensors may fail at steady state. To resolve this problem, the authors develop a separate sensor relocation protocol in [100].

Due to the weaknesses in the simple approaches, several papers explored the use of control theoretic approaches.

An example of centralized control approach is used in [101] in the coverage control problem. The scheme is formulated from control theory. However, it has high computational complexity as a double integral has to be evaluated in real time. The scheme also assumes that there is a communication mechanism that delivers the control information to a sink node for the computations. As a result, it is difficult to scale the network to large number of sensors as the computational and communications delay will slow down the system response.

In [15]-[18], the principle of distributed linear control is applied on mobile sensor networks to provide maximum sensing coverage while maintaining connectivity. Each sensor maintains equidistance from all its adjacent neighbors, so that the sensors are spread out evenly. When additional sensors are added into the network, the network automatically expands to accommodate new sensors. However, if there are failed sensors, the neighboring sensors automatically close in to take over the failed sensors, thereby maintaining network connectivity. The connectivity is rapidly restored as the movements of the closing-in sensors are kept to the minimum. This approach automatically resolves the connectivity problem without resorting to the need to deploy sensor networks using high node densities. Another attractive feature is that the nodes are homogeneous. That is, there are no single points of failure and the network is robust. Finally, a further attractive feature is that there are minimum topology changes in the entire network. The topology changes occurred only when sensors fail. Therefore, the re-routing overheads are kept to the minimum. However, the weakness is the rigidity of the networks. In order to fit into coverage regions of different size, a control gain parameter has to be set. To solve this weakness, in [102]-[105], a non-linear distributed approach based on artificial potential field or virtual force is used to maintain coverage. This creates elasticity among the sensors and enables the mobile sensor networks to fit into spaces of any sizes by contracting or expanding. In another variation from [102]-[105], in [106], fuzzy logic and clustering are used to maintain equal spacing among the sensors.

In [107], the authors discuss an interesting problem in a real life multi-disciplinary project known as the *Autonomous Ocean Sampling Network II (AOSN II)* project. Their problem is to locate correctly the local minima and maxima of the distribution in a search region by using the gradient search algorithm. Local minima and maxima are positions on the distribution that have zero gradients. Instead of using one autonomous sensor to perform the independent gradient search, they use a group of sensors. They encounter stability issues in their design and thus decouple the motion control into two levels to stabilize the group.

First, in the intra-group level control, they adopt the potential field method [102]-[105] to maintain the cohesiveness of the formation controlled group. They adopt the potential field method to allow the whole group to expand and contract by a scaling factor in order to increase or decrease their sensing coverage while moving. The expansion and contraction of the group is controlled by a central controller which determines the optimal scaling factor based on the inputs from the members of the group. The central controller then feeds back the scaling factor to the group members for their individual adjustments. The authors define this form of intra-group level control as *cooperative control*.

Second, in the group level formation control, a central controller is in charge of two tasks.

Its main task is to compute the next move using the gradient search algorithm based on all the data collected from the sensors. From the data, the central computer performs Kalman filtering and averaging operations to obtain a representative gradient for the collective group. The authors call this the *virtual body concept*. In the virtual body concept, the sensor group is treated as though it is one single independent entity. A virtual leader is appointed as a representative. The members behave as if they are following the virtual leader. Since the group-level controller treats the group as a single entity, any independent gradient search algorithm can be used directly.

The second task of the central controller is to compute the optimal scaling factor discussed in the above paragraph on cooperative control and feed back to the members for their individual adjustments. The estimation of the gradient from the data collected is subjected to estimation error. Therefore, the authors develop a computational method to minimize the estimation error by formulating it as an optimal formation problem. From their analysis, they conclude that to minimize estimation error, the group must form regular geometrical shapes with equal spacing. For example, a group of four sensors should form a regular tetrahedron. The optimal scaling factor can be obtained from the computational method. Note that in the method, although the group expands and contracts, the spacing of the sensors remains almost equal with some room for slight deviations.

Note that due to the use of centralized control to perform complex computations, it inherits the scalability problem similar to the centralized control discussed in [102].

The main weakness of all the abovementioned approaches [15]-[18][98]-[107] is that the mobile sensors are equally spaced. However, as explained in chapter 1, section 1.2, for our application, in order to maximize the quality of measurements, the sensors must spread out unevenly. Specifically, the sensors must cluster more tightly in regions where the temperature gradients are steep. The second weakness is that their targets for monitoring and searching are single discrete entities. Even in [107], where the problem is monitoring the distribution, the main objective of the sensors as a collective group is to locate the special discrete points in space: local maxima and minima.

A work that examines a problem that requires the sensors to spread out non-uniformly is in [108]. The problem in [108] is that the sensors are given two tasks: to maintain coverage and to monitor special discrete locations that are known in advance. The discrete locations are locations that the sensors need to pay more attention to. The sensors that are closest to the locations will first move to the locations. A modified aggregation algorithm is executed to fill up the sensing holes which are left behind when the original sensors move to the special locations in order to maintain sensing coverage. The aggregation algorithm is essentially an averaging algorithm. In the execution, initially, the closest neighbors of the original sensors will fill up the holes by taking the positions mid-way through the averaging operation. After the closest neighbors move, the new holes that are left over by them are filled up iteratively by the sensors further downstream. The main weakness is that in their problem the locations of the discrete sources are known whereas in our problem the locations of our sources are unknown and the distributions that we want to monitor are continuous in space.

Therefore, due to the abovementioned weaknesses, we are motivated to develop a new approach while embracing the behavioral based paradigm in the design of our cooperative algorithm in chapter 4.

2.4 Conclusion

In this chapter, we presented our literature review.

We first examine the communication and networking issues by surveying the literature on wireless ad-hoc networking of mobile nodes in harsh environments which is most relevant to our application. The key finding is that there are four main problems that deteriorate the performance of the networks. The first problem is poor connectivity due to imperfect wireless channel conditions such as fading, node mobility. The second problem is contentions for the use of channels. The third problem is inter-neighborhood interference. The fourth problem is, in multi-hop communications, whenever there is a need to establish or repair a route, routing overheads are generated.

Next, we examine the issues related to monitoring, detection and sensing by surveying the WSN. The key finding is that there are three main problems that deteriorate their performance. The first problem is poor connectivity due to imperfect wireless channel conditions such as fading and node mobility. The second problem is network congestion due to high node density. The third problem is high localization errors.

We narrow our survey to focus on coverage control of mobile sensor networks. Our three key findings in our survey on the coverage control algorithms are: the coverage control results in equal spacing of mobile sensors over the terrain, the targets that the sensors monitor are assumed to be single discrete entities and sophisticated centralized control schemes are not scalable with the number of sensors. As explained in chapter 1, section 1.2, for our application, in order to maximize the quality of measurements, the sensors are spread out according to the characteristics of the data to be collected, which usually results in a non-uniform or uneven distribution over the area of interest. Specifically, the sensors must cluster more tightly in regions where the temperature gradients are steep. Furthermore, we are monitoring continuous distributions that we do not have advance knowledge of. Therefore, we are motivated to develop a new scalable approach while embracing the behavioral based paradigm in the design of our cooperative algorithm in chapter 4.

Chapter 3: Preliminary Investigation and Analysis

In this chapter, we present our preliminary investigations and analyses of a group of cooperative autonomous mobile agents. Autonomous mobile agents refer to mobile robots or sensors that are capable of performing any missions without requiring a human to manually control their movements from a remote interface. Our emphasis is on the communication and networking aspects of the cooperative mobile agents. In section 3.1, we analyze the effect of mobility and the Rayleigh fading channel on the connectivity of such a network on a terrestrial environment. In section 3.2, we analyze the effect of mobility and the Rayleigh fading channel on the throughput of such a network on a terrestrial environment. In Section 3.3, we analyze the effect of mobility and the Rayleigh fading channel on the throughput of such a network in an underwater environment. In section 3.4, we highlight the main research contributions of our work. Finally, we conclude the chapter with a summary of insights.

3.1 Connectivity Analysis of a MANET of cooperative autonomous mobile agents

We study the connectivity of a Mobile Ad Hoc Network (MANET) of autonomous cooperative mobile agents (e.g. mobile robots) under the Rayleigh fading channel [see List of Publications, P1]. Connectivity is a critical performance parameter of cooperative robots deployed in real-time scenarios such as disaster and rescue scenarios. There are two major factors that affect the connectivity of the MANET. First, the mobility of the nodes causes the separation between any pair of nodes to fluctuate. Second, atmospheric condition and obstacles can cause the transmission range of the nodes to fluctuate. Based on these factors, stochastic analysis is performed to derive the connectivity probability. The connectivity probability represents the fraction of time that a node is connected to at least one other node. This probability is used to study the effect of mobility and fading on the connectivity as the transmission range or number of nodes in the network varies. Such analytical results can form the basis of performance modeling of MANET routing protocols and network optimization.

3.1.1 The Method

Our approach to this study is based on the use of theoretical analysis. The detail of the analysis can be found in our paper [see List of Publications, P1]. In the analysis, we consider the movement of the nodes and the radio channel fluctuations independently. In a high data rate, low velocity environment, the fading is slow-varying [23]. Hence, the fading process is effectively independent of the mobility of the node. We first model the random movement of the nodes using the Probabilistic Mobility Model (PMM) [77]. The radio channel is modeled using the free space propagation model with Rayleigh fading [23]. From these two stochastic models, we derive the connectivity probability of an arbitrary node *a*, c_a as follows:

$$c_a = 1 - \prod_{b=1, b \neq a}^{N} \left(1 - c_{ab}\right) \tag{3.1}$$

N is the node count or the total number of nodes in the region and *b* is the index that represents another node in the region. c_{ab} is the pair-wise one-hop connectivity probability which is given as follows:

$$c_{ab} = \sum_{y_b=0}^{S} \sum_{x_b=0}^{S} \left[\sum_{y_a=0}^{S} \sum_{x_a=0}^{S} \exp\left(-\frac{R_{ab}^2}{2\sigma^2 \eta_0^2} \right) \pi_{x_a} \pi_{y_a} \right] \pi_{x_b} \pi_{y_b} \qquad (3.2)$$

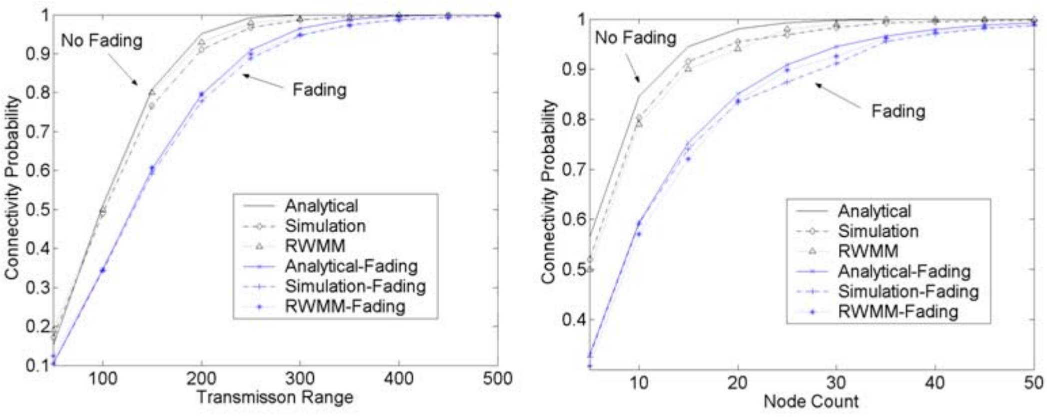
where, R_{ab} is the Euclidean separation of two arbitrary nodes, *a* and *b* and η_0 is the free space transmission range. $\pi_{x_a} \pi_{y_a}$ and $\pi_{x_b} \pi_{y_b}$ are the stationary position probabilities of node *a* and *b* at coordinates (x_a, y_a) and (x_b, y_b) respectively given in our work [see List of Publications, P1].

On a final note, we use the PMM to model the mobility as it leads to tractable result. However, Random Walk Mobility Model (RWMM) is a random mobility model that is popularly used in the MANET simulations. In our paper [see List of Publications, P1], we demonstrate the concept of the statistically equivalent mobility model to show that the conclusion obtained from our results can be approximately generalized to RWMM. This is verified in our simulation results.

3.1.2 Numerical and Simulation Results

In this section, we generate a few plots to study the effects of varying the system parameters on the connectivity of the network. There are two motivations for such studies. First, usually the system parameters represent scarce resources which we will like to minimize given a reasonable quality of connectivity in the network. Second, the minimization of system resources has an additional benefit of reducing interference and channel contentions at the MAC layer [25]. In addition, we will look into the effect of the channel fading on the connectivity of the network. By choosing an appropriate value for the fading parameter, we could characterize the aggregate effect of the radio environment on the connectivity of the network. The details on the value of our parameters are provided in our paper [see List of Publications, P1].

First, we are interested in the connectivity of the nodes in an enclosed room as the transmission range of the nodes increases. One way to achieve an increase in transmission range will be to increase the transmission power of the node. Hence, by looking at the connectivity probability, we will know the amount of transmission power required to achieve a certain level of connection. In figure 3.1a, we show the connectivity probability over different transmission ranges for the cases with and without fading. The node count, *N* is fixed at 25 nodes for this plot. From the graph, the connectivity probabilities increase for both cases as the transmission range increases. When there is no fading, the connectivity probability increases at a faster rate. For example, when there is no fading, the connectivity probability is approximately one when the transmission range is 250m. The average number of hops is about (1000/250 =) 4 hops for this case. However, when there is fading, the transmission range is increased to 380m in order to achieve the same connectivity probability, with an average number of 2.6 hops.



a) Theoretical vs Simulated Connectivity when Transmission Range is varied

b) Theoretical vs Simulated Connectivity when Node Count is varied

Figure 3.1: Study on the effects of varying the transmission range and node count on the connectivity probability.

Second, we are interested in the connectivity of the nodes in an enclosed room as the node count increases. In figure 3.1b, we show the connectivity probability over different node count for the cases with and without fading. The free space transmission range, η_0 is fixed at 250m for this plot. From the graph, the connectivity probabilities increase for both cases as the node count increases. Again, when there is no fading, the connectivity probability increases at a faster rate.

Simulations using PMM and RWMM for both no fading and fading cases are used to verify our analysis. The simulation plots in figure 3.1a and b show that the simulation results converge well to the theoretical results. They also show that PMM provides very good approximation to the RWMM.

3.1.3 Conclusion

Connectivity of MANET is affected by the mobility of the nodes and the fading of the signals in the radio channel due to varying atmospheric conditions and scattering obstacles. In our analysis, we address the issues using stochastic modeling. We derive the connectivity probability and study the effects of varying the system parameters on the connectivity of the network. We hope that from the study, we could find ways to minimize the scarce resources given a reasonable quality of connectivity in the network. This minimization of the scarce resources also has an additional benefit of reducing interference and channel contentions at the MAC layer. In addition, we also study the aggregate effect of the channel fading on the connectivity of the network. It is shown from the numerical result that for the case with fading, we need either a higher transmission power or node count in order to achieve the same level of connectivity compared to the case without fading. Furthermore, we observe that the connectivity of the network is actually not very sensitive to the parameters in the mobility model. In future, this model will serve as a basis to further analytical modeling of MANET routing protocols and network optimization.

Finally, the two implications of the connectivity analysis on the design of our algorithm are as follows:

1. To minimize connectivity problem due to mobility, a possible solution is to ensure that our motion control algorithm results in a coordinated orderly movement with minimal changes of neighbors in the network neighborhoods.
2. To minimize connectivity problem due to channel fading condition, one approach is to use physical layer techniques such as spreading code modulation used in DS/CDMA which is known to be robust to fading channels. Finally, to further improve the robustness of our algorithm to the connectivity problem, our control algorithm should make use of only the information from three closest neighbors out of the other network neighbors. Since, they are the closest, they have better chances of been heard as, statistically, their mean received signal strengths are highest.

3.2 CSMA/CA Throughput Analysis of a MANET of cooperative autonomous mobile agents under the Rayleigh Fading Channel

The deployment of cooperative autonomous mobile agents in harsh terrains deem to be hazardous for humans to accomplish a variety of missions such as search and rescue is an active area of research. An important aspect is the relaying of audio-visual and other real-time information to a control station via multi-hop mobile ad-hoc networking (MANET). Multi-hop networking has many advantages such as energy savings from reduced distance dependent transmission loss and robustness against shadowing due to the ability to communicate around obstacles without line of sight obstructions. It is also purported to achieve capacity gain through spatial diversity. We investigate the feasibility of using MANET for cooperative autonomous mobile agents by looking at a popular configuration that uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC protocol (adopted in IEEE 802.11 MAC) [27] and Ad Hoc On-Demand Distance Vector (AODV) routing protocol [78]. The MANET in our scenario operates under the time-varying connectivity constraints due to node mobility and non-ideal radio channels [79][80]. This time-varying connectivity has a significant impact on the MAC and routing protocols in real life scenarios [80]. However, it is usually not investigated in many theoretical analyses [69]-[75]. Our stochastic model realistically evaluates the impact of the factors. It allows us to identify areas of improvement and propose strategies to optimize the network performance based on the MANET configuration.

Specifically, three salient features are modeled. The first two features are: interference from hidden and exposed node [25] and time-varying connectivity that lead to diminishing spatial diversity gain in multi-hop communications. The third feature is the multi-hop communications overheads in two forms: the increase in routing overheads due to time-varying connectivity that results in congestion and long delays and the additional traffic load each node, acting as intermediate node, is obliged to carry for others.

The performance metric used in our analysis is the average data throughput per node. The average data throughput per node is usually defined as the data rate in bits per second that each node can transmit. For analytical purpose, we normalize the average data throughput per node over the maximum data rate the radio channel can support. Note that the nodes can only transmit at the maximum data rate under these three ideal conditions: the radio channel has no fading, the nodes are not moving and there are only one pair of nodes communicating. The third condition is equivalent to the condition that there is no nodes contending and interfering with each other for the use of the radio channel. Hence, by performing the normalization, we could appreciate quantitatively the deteriorating effect on the MANET communications in a practical non-ideal situation.

3.2.1 Method

We use theoretical analysis to access the performance. In our approach, we model each component of the network separately, namely: The CSMA/CA protocol used in the networking communications, the connectivity model which characterizes the effect of the mobility of the nodes and the Rayleigh fading channel and the effect of the disconnection on the routing layer. From the model, we derive the data throughput per node by studying incrementally more complex scenarios in the following order: *non-overlapping independent neighborhoods*, *overlapping neighborhoods with no multi-hop communications* and finally *overlapping neighborhoods with multi-hop communications*. This allows us to isolate various issues involved in networking and communications.

3.2.1.1 The Analytical Model

In CSMA/CA protocol adopted in IEEE 802.11 MAC is first modeled. The protocol is distinguished by two salient features: 4-way handshaking and the exponential backoff algorithm. Their details can be found in [77][27][93]. In particular, [77] provides the description on the use of the 4-way handshaking to mitigate the hidden node problem. Briefly, the operation of 4-way handshaking is shown in figure. 3.2. The abbreviations are listed in Table II.

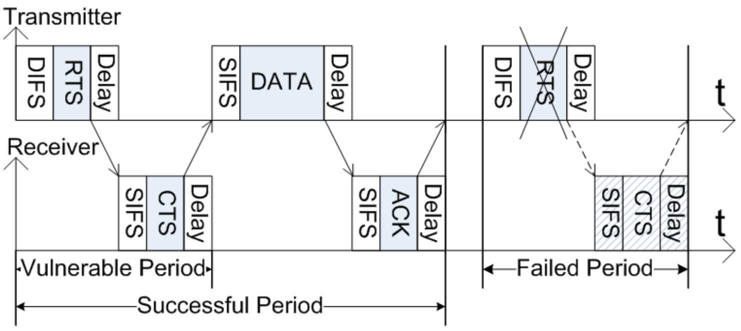


Figure 3.2: Timing diagram for a successful transmission followed by a failed transmission

Table 3.1: Abbreviations in timing diagram

	Description
DIFS	DCF Inter Frame Space. Listening time prior to transmitting a RTS packet.
RTS	Request-To-Send packet. Used by transmitter to reserve the channel.
Delay	Propagation and synchronization delay.
SIFS	Short Inter Frame Space. Listening time prior to transmitting CTS, DATA or ACK.
CTS	Clear-To-Send packet. Used by receiver to indicate which node can send its data.
DATA	Data packet from the transmitter.
ACK	Acknowledgement packet from the receiver to indicate successful receipt.

In figure 3.2, the channel is first reserved using the RTS-CTS packet exchanges between the transmitter and receiver. Failed transmissions occurred when there is a collision of RTS packets when transmitting nodes contend to reserve the channel. The contending nodes realize that there is a collision when there is no CTS packet after the maximum waiting duration. The vulnerable period for collisions is shown in figure 3.2. When a collision is detected, each node executes the exponential backoff algorithm [75][85]. The algorithm determines a random backoff period in time slots that the node must wait before it can retransmit its data packet again. Prior to our analysis, we need the throughput of a slotted CSMA/CA protocol using the exponential backoff algorithm for a one-hop network with *M* nodes having poisson packet arrival rates with mean, λ. This scenario is well analyzed and the result is obtained [75] as:

$$S(M,\lambda)=\sum_{i=0}^{M}\pi_i(M,\lambda)P_s(i,\lambda)T_p \Bigg/ \sum_{i=0}^{M}\pi_i(M,\lambda)\{\bar{I}_i+1+P_s(i,\lambda)T+[1-P_s(i,\lambda)]C\} \tag{3.3}$$

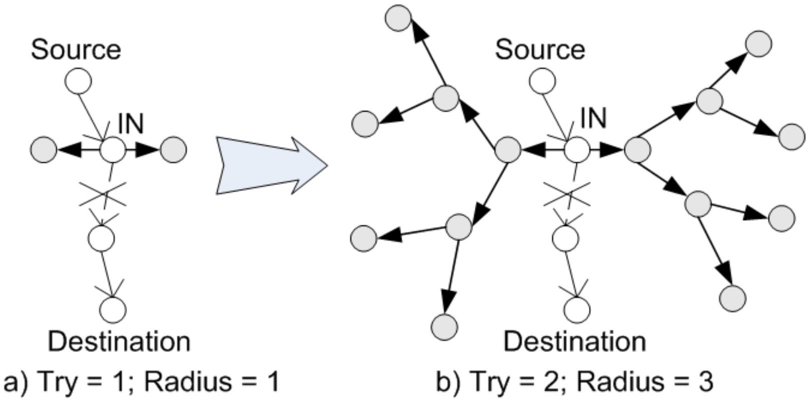
Note that in [75], the notation *S*(*M*) is used instead of *S*(*M,λ*). We add λ to emphasize that the throughput is also dependent on the traffic load. $\pi_i(M, \lambda)$ is the stationary probability distribution of the backlogged nodes given that there are *M* nodes. A node is backlogged if it has a packet to retransmit after a collision. *i* is the state of the Markov chain which represents the number of backlogged nodes in the neighborhood. $P_s(i,\lambda)$ is the probability of successful packet transmissions given that there are *i* backlogged nodes. $\bar{I}_i$ is the average idle period in the channel given that there are *i* backlogged nodes. *T* and *C* are the periods of successful and failed transmissions.

The connectivity model has already been discussed in section 3.1. The time-varying connectivity in the network is due to node mobility and fading channel. The model we used here is identical to those we use in the analysis of connectivity in section 3.1. An important quantity is the average number of temporal neighbors in a time-varying connectivity environment, $\overline{\Phi}$ expressed as:

$$\overline{\Phi}=(N_s-1)c_{link}+1 \tag{3.4}$$

The derivation for $\overline{\Phi}$ is presented in appendix A. Temporal neighbors are neighbors at one discrete time step. N_s is the node count in the square region and c_{link} is the pair-wise link connectivity which is equaled to c_{ab} in equation (3.2)

AODV routing protocol [79] is a reactive routing protocol which builds routing tables on demand adopted by the Mobile Ad Hoc Networking Working Group of the Internet Engineering Task Force (IETF). There are two types of behavior which give rise to routing overheads. First, when either there is no route to a destination or the routes in the routing table expire, the routing protocol initiates a new route search. This behavior dominates at transient state of network where many route tables are empty. It also dominates at low traffic condition where the mean packet inter-arrival time is long and the routes in the table expired. At steady state, with moderate to high traffic conditions, whenever there are unrecoverable link breakages due to disconnections, the routing protocol initiates local repairs which dominate the overheads. Local repairs use an expanding ring broadcast search. It is illustrated in figure 3.3 for first two tries. When a route is found, it is unicast back. The default setting for the search is three tries with increasing ring radii in numbers of hop: 1, 3 and 5. Referring to figure 3.3, intermediate node IN which detects the breakage at downstream always initiates the search at each try. Also the broadcast is in the outward direction because nodes that receive multiple copies of the same search packet suppress sending them.



a) Try = 1; Radius = 1 b) Try = 2; Radius = 3

Figure 3.3: Expanding ring search for the first two tries

To model the effect of the routing protocol, we use an approximate model. From the model we derive the cost, λ_{route}. The detailed is given in appendix A. It is the total AODV route traffic load generated per node per second. λ_{route} is normalized to the data packet transmission time. It is given as:

$$\lambda_{route} \approx (2^{N_{hop}} - 1) r \sum_{i=0}^{2} \sum_{j=0}^{i} I(\omega_{j-1}, \omega_j) \omega_j c_{link} (1 - c_{link})^j \qquad (3.5)$$

N_{hop} is the estimated average number of hops per route. $(2^{N_{hop}} - 1)$ is the count of all combinations of multi-hop routes from 1 to N_{hop} hops. ω_j is the number of local repair packets generated when ring radius is ($2j$+1) hops in current try. r is the ratio of routing to data packet transmission time; it is use to normalize λ_{route}. $I(a,b)$ is the indicator function; it is 1 if ($a \neq b$), 0 otherwise.

3.2.1.2 Discussion on various scenarios used in the derivation

First, we study the hypothetical case of *non-overlapping independent neighborhoods*. In this case, the nodes are mobile and moved in an independent, memoryless and uncoordinated manner described by the PMM. The physical radio channel they used for wireless communications is a Rayleigh fading channel. Therefore, their communications are disrupted by occasionally fading of radio signals. In this scenario, the nodes formed clusters or one-hop neighborhoods and the one-hop neighborhoods are completely isolated and decoupled from each other in relation to communications. We can visualize this scenario as many isolated neighborhoods within an enclosed region where the total number of nodes are fixed. Note that due to node mobility, we expect nodes to constantly move in and out of a neighborhood. However, as PMM is a stable stationary Markovian stochastic process, at steady state, the average number of nodes within each neighborhood is constant. There are two purposes to this assumption. First, it is to eliminate the hidden and exposed node problems associated with the use of the MAC protocol in MANET. Second, as there are no couplings, multi-hop communications is not possible. Therefore, we need not consider the issue of routing overheads at the routing layer of the MANET. Therefore, in this scenario, we could examine solely the effects of node mobility and Rayleigh fading channel on the data throughput.

One of the benefits of multi-hop communications is the spatial diversity gain in capacity. The idea of spatial diversity gain is that isolated neighborhoods form partially independent "communication spaces" for the nodes. As a result, if we can divide a large group of nodes into smaller groups of nodes, the average data throughput per node will increase. This is analogous to a situation that we have a large group of people having a single meeting. In this situation, the opportunity that each person can talk (or the talk-time) is reduced because only one person can talk at one time and this talk-time is been shared by a large group of people. If we divide this group of people into smaller groups and put them in separate isolated rooms and hold the meetings concurrently, we will now have an increase in average talk-time for each person because in each small group, there are now fewer people contending for the talk-time. From this analogy, we see the rationale of using the completely isolated neighborhoods scenario. This is because only through this assumption that the spatial diversity gain is at maximum as there are no inter-group interference.

In our derivation in appendix A, we show that the adverse effect of the mobility of the nodes and the fading radio channel on the data throughput is to almost completely diminish the spatial diversity gain. The first reason is the use of the independent memoryless random mobility model to model the movement of the node. These is analogous to a situation where in a meeting, the uncooperative members leave and join randomly at any time they wish and without any prior reasons (memoryless), resulting in disruptive communications. While this ad-hoc type of situation may sound ridiculous in the human context, it is in fact a common realistic situation where a MANET is expected to operate in. An important conclusion we draw from this situation is that in order for the communications to be effective, especially in relation to the spatial diversity gain, either all the nodes must be stationary or there must be some forms of cooperative behaviors in the movement of the nodes. One possible approach to achieve this is through the use of formation control. This has the effect of making the topology of the network appears stationary or invariant to individual nodes. The second reason is the effect of fading channel. To combat this, we will have to explore other methods of communications that are more robust to fading channel. One possible method is to explore the use of spread spectrum communications.

Second, we study the scenario of *overlapping neighborhoods* with no multi-hop communications. This scenario has been used by [75] to study the throughput in a stationary ad-hoc network by analyzing the hidden node problems. Essentially, in our approach, we calculate the average overlapping area of a typical neighborhood that is overlapped with others. From this, we could quantify approximately the extent of interference from the hidden and exposed nodes. The most important parameter required to compute the average overlapping area is the average transmission range. This parameter is independent of the mobility of the node. As our channel fading is a stationary stochastic process, the average is a constant value. The detailed derivation is in appendix A. The importance of studying this interference is its role in exacerbating the throughput per node when the node density increases. It must be noted that the reduction in throughput as the node density increases is caused by both inter-neighborhood hidden and exposed node interference and intra-neighborhood contentions. In related literature, it is also equivalently studied as the optimal number of neighbors required for maximum throughput per node. Therefore, any MAC protocol that is known to combat interference will be a suitable protocol to solve this problem. One such protocol is the Direct Sequence Code Division Multiple Access (DS/CDMA) which makes use of the property of spread spectrum modulation.

Third, we study the scenario of *overlapping neighborhoods* with multi-hop communications. This is a realistic MANET scenario. Here, we approximately derive the data throughput per node at the routing layer by separating the routing overheads from the actual data stream. This is important because constant route disruptions due to mobility and link disconnections due to fading channel caused the AODV routing protocol to generate routing packets to either repair or search for a completely new route to a multi-hop destination node. This overhead consumes a significant amount of the communication bandwidth. The final expression for the data throughput per node, S_{mhop}^{data} is given as below.

$$S_{mhop}^{data} = (\lambda_{data} / r\lambda_{link})S_{mhop} \tag{3.6}$$

λ_{data} and λ_{link} are the actual data traffic and the total traffic in a link respectively. S_{mhop} is the MAC throughput per node. The detailed derivation is discussed in appendix A.

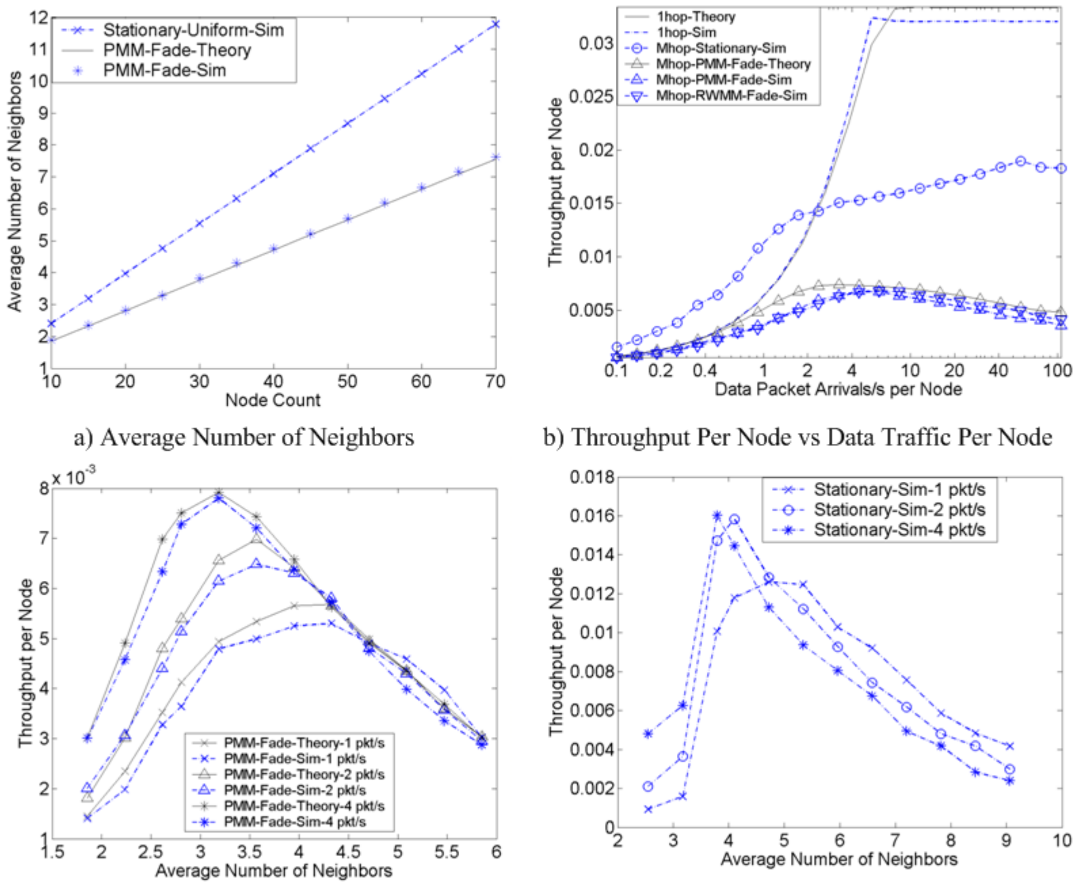
3.2.2 Numerical and Simulation Results

In our Glomosim simulation, at initial time, every node is assigned as source node with its destination node randomly assigned such that no destination node receives from more than one source node. The assignment remains the same throughout each run. The pairs can be more than one hop away from each other. In Table 3.2, common parameters for both numerical computations and simulation are listed. The parameters for the equivalent Random Waypoint Mobility Model

(RWMM) are obtained based on the formula in [see List of Publications, P1]. Figure 3.4a shows the average number of neighbors as node count, N_s is varied. It is used to validate the accuracy of average number of temporal neighbors, $\overline{\Phi}$ in equation (3.4). Our simulation collects the number of neighbors of every node in the square region during each time step. $\overline{\Phi}$ is then obtained by averaging the numbers of neighbors of every node over all time steps. The simulation is run for two cases. In the first case, a uniformly distributed stationary ad-hoc network with a channel that has free-space transmission loss only is simulated. In the second case, a MANET using Probabilistic Mobility Model (PMM) with a Rayleigh fading channel that has free-space transmission loss is simulated. Altogether, 2×10^5 samples are collected for each case. From the figure, we see that our theoretical analysis is well validated by the simulation. We observe that the average number of neighbors is smaller for the case where there is Rayleigh fading in the channel. This is due to attenuation in transmission power caused by random fading.

Table 3.2: Values for the common parameters used in the throughput simulation of a MANET using CSMA/CA and AODV protocols

Parameter	Value	Parameter	Value
Run	40	Route Packet Duration	560 μs
Simulation duration	2,000 s	Data Packet Duration	6,256 μs
Warm up duration for mobility	1,000 s	Transmission Loss Model	Free Space
Square Region dimension	1×1 km^2	Transmission Range for ad-hoc networks, η_0	250 m
Data Packet Size (Max size of TCP data packets)	1460 B	Rayleigh Fading Parameter, σ^2	0.3
RWMM: minimum and maximum speed, [v_{min} v_{max}]	[0 20] m/s	Propagation and synchronization delay	1 slot
RWMM Travel Time	9 s	Slot Time, T_{slot}	20 μs
RWMM Pause Time	2 s	SIFS Duration	1 slot
PMM Time Step	1 s	DIFS Duration	3 slot
PMM Step Size	10 m	ACK/CTS Duration	248 μs
PMM average speed	10 m/s	RTS Duration	272 μs



a) Average Number of Neighbors

b) Throughput Per Node vs Data Traffic Per Node

c) Average Number of Neighbors supported by CSMA/CA for different traffic loads in MANET under a Rayleigh Fading Channel

d) Average Number of Neighbors supported by CSMA/CA for different traffic loads in a stationary Ad-hoc network with No Rayleigh fading

Figure 3.4: Results for the throughput simulation of a MANET using CSMA/CA and AODV protocols

Figure 3.4b shows the throughput per node as the data traffic per node in packets/s, λ_{data} is varied. The node count is 25 for all four cases. The first case is a one-hop network where all the nodes can hear each other. It is used as a benchmark to study the actual spatial diversity gain in ad-hoc networks. The second case is a uniformly distributed stationary ad-hoc network with only space transmission loss. Only hidden and exposed node problems are present in this case due to overlapping transmission regions. The third case is a MANET using PMM with a Rayleigh fading channel. Therefore, it suffers from time-varying connectivity in addition to hidden and exposed node problems. The fourth case is a MANET using RWMM with a

Rayleigh fading channel. This case is plotted to verify the claim in [see List of Publications, P1] that PMM is a good approximation for RWMM when analyzing the average performance. We observe that our theoretical analysis agrees well with the simulation results and PMM is a good approximation for RWMM.

First important observation is in the throughput of stationary ad-hoc network. At low traffic load up to 1 packet/s, there is some spatial diversity gain in relation to one-hop network. However, this gain diminishes rapidly due to increase severity of hidden and exposed nodes at higher traffic loads. Also, simulation and numerical results of MANET with PMM and Rayleigh fading confirm the discussion in section 3.1.2.2 that time-varying connectivity can present a serious problem as it eliminates the spatial diversity gain. There is no significant gain even at low traffic loads. This is exacerbated by the loss due to hidden and exposed node problems. Second important observation is the effect of additional traffic generated by routing packets due to disconnections. Referring to figure 3.2a, the average number of neighbors contending for the channel is about three for MANET with PMM and Rayleigh fading. However, in figure 3.2b, when there is time-varying connectivity, the throughput saturates only slightly better than the one-hop scenario with 25 nodes contending. The stationary case fairs much better than the former two cases as it generates less routing packets. This implies that a network with time-varying connectivity congests more rapidly which results in exponentially increasing delay.

Based on the above observations, two strategies are proposed in the design. First, the robots must be designed to maintain connectivity with the same neighbors when they are moving in a Rayleigh fading channel in order to achieve optimal throughput. This is to mitigate the problem of time-varying connectivity. One possible solution is to divide the large team into small groups of robots and implement formation control for the groups. Second, a MAC protocol such as Direct Sequence/Code Division Multiple Access (DS/CDMA) that is more robust to interference in the overlapping regions can be implemented. This is to mitigate the hidden and exposed node problems.

Figure 3.4c and 3.4d examine the throughput per node as the average number of neighbors is varied for a MANET with PMM and Rayleigh fading channel and a stationary ad-hoc network with no fading for different data traffic loads respectively. The optimal number of neighbors occurs at the point where the throughput is at maximum. In both cases, the same pattern emerges. At low traffic loads, the network can support higher optimal number of neighbors. However, as traffic load increases, the optimal number decreases. For example, for a traffic load of 1 packet/s, optimal numbers for the mobile and stationary ad-hoc networks are about 4.5 and 5.5 respectively. The reason that the optimal number of neighbors the MANET can support is lower is because the effective traffic loads which include route packets generated by the time-varying connectivity is much higher.

We now see two conflicting objectives. As can be seen from the results of our connectivity analysis in section 3.1, network connectivity increases with the node density and eventually saturates. However, an increase in node density also resulted in an increase in both contentions and interference. As a result, the throughput per node decreases. On one hand it is desirable to increase the node density in order to increase connectivity; on the other hand, increasing the node density also resulted in an increase in contentions and interference. Therefore, there exists a balancing point which is known as “the optimal number of neighbor” problem in the literature.

Advancing this reasoning further, we infer that any factor that results in an increase in contentions and interference will also affect the optimal number of neighbors, for example, the total or effective traffic load. That is, if we increase the data traffic per node in the network, it will result in an aggregate increase in interference and contentions. As discussed earlier, our data in figure 3.4c-d supports this hypothesis. In our analysis in appendix A, we show that in multi-hop communications, they are three components that constitute the effective traffic load. The first component is the actual data originated from the node. The second component is the multi-hop data each node is obliged to carry for others. The third component is the routing overheads.

Due to the presence of the second component, the number of optimal neighbors that we can support will always be lower compared to single-hop communications. The significance of this in our optimal harvesting sensor network is that the number of optimal neighbors represents the local node density, which sets a limit to how close we can cluster our nodes in a local region before local communications collapsed and start to inhibit the performance. Since optimal harvesting requires nodes to cluster closer in an information fertile region, it is better to implement only single-hop communications in our cooperative control protocol in order to eliminate the second component. Another characteristic of this component is that it is proportional to the total number of nodes in the network. Clearly, if we required multi-hop communications for other purposes, it is better to limit this component by dividing a large group of sensors into smaller clusters. Occasionally communications can then be confined at the level of cluster-heads.

The third component is affected by disconnections which are in turn affected by random node mobility and channel fading. The solution to this problem has been discussed earlier in section 3.2.1.2. It is to implement motion control such as formation control to reduce the communication disruptions due to frequent change of neighbors. Also, a better protocol that is more robust to fading such as DS/CDMA should be used. Furthermore, DS/CDMA protocol is more robust to any form of interference and therefore can support a higher number of neighbors. A final conclusion is that there is no ultimate optimal number as it depends on the effective traffic loads.

3.2.3 Conclusion

We analyzed the theoretical throughput of a team of cooperative autonomous mobile agents operating in the Rayleigh Fading Channel using the MANET configuration consisting of CSMA/CA MAC (IEEE 802.11 MAC) and AODV routing protocol. Three salient features are modeled: interference from hidden and exposed node, time-varying connectivity due to mobility and Rayleigh fading and multi-hop communications overheads due to routing overheads and additional traffic load each intermediate node carries for others. Our results show that time-varying connectivity, hidden and exposed node problems lead to severely diminished spatial diversity gain in MANET. Additionally, an increase in routing overheads due to time-varying connectivity causes rapid increase in delay due to network congestion. Finally, the optimal number of neighbors for maximum throughput depends on the effective traffic loads.

Finally, the three implications of the DS/CDMA throughput analysis on the design of our algorithm are as follows:

1. Motion control should be implemented to ensure that there are minimum changes in network neighbors.
2. It is better to divide a large group of sensors into smaller groups and implement one-hop communications to facilitate the harvesting tasks.

Occasional multi-hop communications required for other purposes should be confined to the level of cluster-heads.

3. It is better to use a MAC protocol that is more robust to fading and interference such as DS/CDMA.

3.3 DS/CDMA Throughput of Multi-hop Sensor Network in a Rayleigh Fading Underwater Acoustic Channel

In the previous section, we discuss the possibility of using DS/CDMA instead of CSMA/CA as the MAC protocol for the MANET because it is more robust to noise, interference and fading. In this section, we study an underwater multi-hop sensor network in a challenging noisy environment. Asynchronous half-duplex Direct-Sequence Code-Division Multiple-Access (DS/CDMA) is a suitable candidate for the MAC protocol design of Underwater Acoustic (UWA) sensor networks due to its many attractive features. However, its basic form is only suitable to be used in a one-hop network. Our ad-hoc multi-hop network is infrastructure-less without centralized base stations and power control. Therefore, we develop an asynchronous distributed half-duplex control protocol to regulate between the transmitting and receiving phases of transmissions. Furthermore, multi-hop communications is very sensitive to the time variability of fading and the snapping shrimp dominated ambient noise in the harsh underwater environment because a broken link in the multi-hop path is enough to disrupt the communications and initiate new route searches. In our configuration, we use the Ad hoc On-Demand Distance Vector (AODV) routing protocol optimized for UWA networks. Empirical studies show that we can model the channel as slow-varying Rayleigh fading and frequency non-selective. We theoretically analyze the throughput of our configuration by considering three salient features: ability of the receiver to demodulate the data, effect of our control protocol and effect of disconnections on generation of routing packets. The throughput under various operating conditions is then examined. It is observed that at optimal node separation, the throughput is improved by a factor of 10.

3.3.1 Methods

The details of the study and the approach of our analysis can be found in our papers [see List of Publications: P2, P3]. We first model the connectivity of the network and the DS/CDMA system model. As DS/CDMA cannot be directly implemented onto a multi-hop network, we implement an asynchronous half duplex MAC protocol. This portion of the operation is also modeled. We then derive the MAC throughput of the network.

To model the connectivity, we developed a sensor network model for analysis as shown in figure 3.5. We assume that N_s sensor nodes are initially placed uniformly over a region in the square lattice formation. All the nodes are independently and identically distributed (i.i.d.). The spacing of the nodes is $\varepsilon\eta_0$, where $\varepsilon \in (0,1]$. η_0 is the transmission range of the node for a channel which has a transmission loss with 4th power of range.

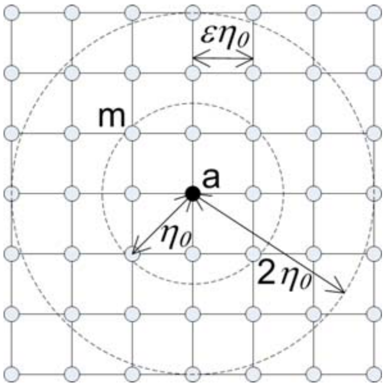


Figure 3.5: Sensor network model

We follow the approach that is used in section 3.1 and 3.2 from which we obtain the average link connectivity probability, c_{link} of two arbitrary nodes: node a and node m as:

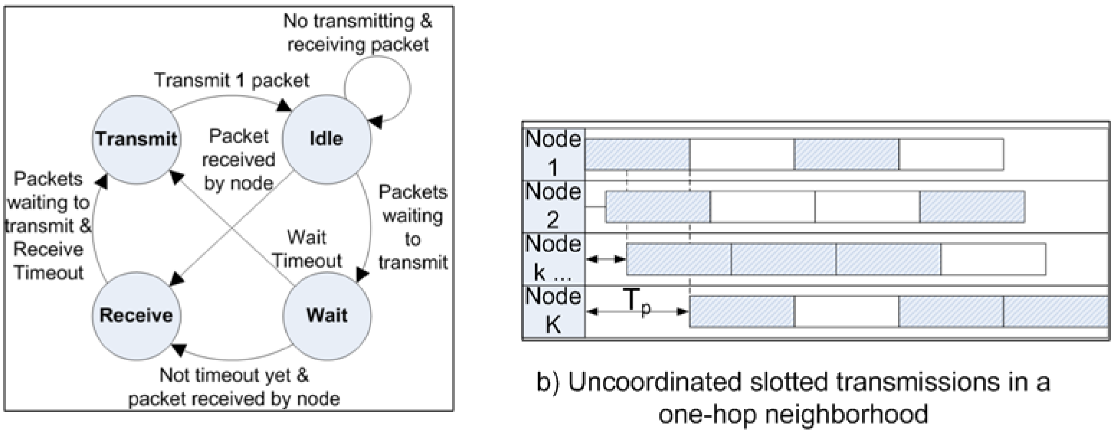
$$c_{link} = N_S^{-1} \sum_{y_a=0}^{\sqrt{N_s}-1} \sum_{x_a=0}^{\sqrt{N_s}-1} \Pr\left(R_{am} \leq \eta \middle| a\right) \qquad (3.7)$$

a and m are the indices for any pair of nodes in the network. R_{am} is the separation between the two nodes, a and m. η is a random variable representing the Rayleigh faded transmission range.

To model the behavior of the DS/CDMA de-spreading operations at the receiver, we use a system model which is partially based on the model in [96]. This model is required for us to determine the probability that a received packet is successfully modulated in the presence of Multi-Access Interference (MAI), Rayleigh fading and the ambient noise through the de-spreading operations denoted as $P_{MAI}(K)$.

Nodes in distributed multi-hop sensor networks cannot transmit and receive simultaneously. Since there are no centralized base stations, we have implemented a distributed control protocol at the MAC layer to regulate between the **Transmit** and **Receive** phases of the node. Our analysis took into account this implementation by modeling its behavior using the Markov Model. This is used to derive the probability that the packet is received successfully without collisions with other packets due to contentions, P_{RS}. In order to do this, we have to examine the state diagram of the protocol, as shown in figure 3.6, to better appreciate the transmitting and receiving behaviors of the node.

The state diagram of our protocol is shown in figure 3.6a. There are two timers in the protocol. The **Wait** timer commences when there is at least one packet waiting to be transmitted. Prior to transmission, the packet is delayed randomly with a uniform distribution between [α, $T_p + \alpha$]. α is a fixed time which is at least equal to the maximum propagation delay of the network. T_p is the packet transmission duration. This is necessary to prevent nodes in the neighborhood from locking into the **Transmit** state since there are no centralized base stations to synchronize between the two different states. This is similar to the half-Aloha MAC protocol implementation in Quadnet. The main difference is that the protocol in Quadnet does not support DS/CDMA.


No transmitting & receiving packet
Transmit 1 packet
Transmit
Idle
Packet received by node
Packets waiting to transmit & Receive Timeout
Packets waiting to transmit
Wait Timeout
Receive
Wait
Not timeout yet & packet received by node
Node 1
Node 2
Node k ...
Node K
Tp

b) Uncoordinated slotted transmissions in a one-hop neighborhood

a) State Diagram

Figure 3.6: State diagram for the synchronous half-duplex protocol

In order to support DS/CDMA in an asynchronous distributed half-duplex environment, two conditions are introduced into our state diagram. First, any simultaneous packets received in one of the branches of the receiver can cause a transition from either the **Idle** or **Wait** state to the **Receive** state in the state diagram. Notice that a transmitting packet arriving at the **Idle** state has to go into **Wait** state. This is because we cannot wait to receive a packet once it has arrived so priority has to be given to receiving the packet. Second, there is a **Receive** timer and it is at least $2T_p$ duration. This timer is required to give the node a chance to transmit its packets if there are packets waiting to be transmitted without been locked in the **Receive** state perpetually. However, beyond $2T_p$ duration, if there is no packet waiting to be transmitted, the node can still remain in the **Receive** state and continue to receive the packets until a packet to be transmitted arrived. The reason that the **Receive** timer cannot be less than $2T_p$ is explained in figure 3.6b. In figure 3.6b, during **Receive** state, the packets from neighboring nodes arrive at different times relative to the first arriving packet. In the worst case, the last packet arrives only after a relative delay of T_p. Therefore, a minimum of $2T_p$ is required for the node to receive multiple concurrent packets. The node only receives a packet successfully in **Receive** state when the packet arrived at least T_p before the end of **Receive** state. Our protocol ensures that the node can receive multiple packets and at the same time, allows the node a chance to transmit.

From the state and the timing diagram analysis, P_{RS} can be derived. This is in turn used to derive the throughput, S_M given as:

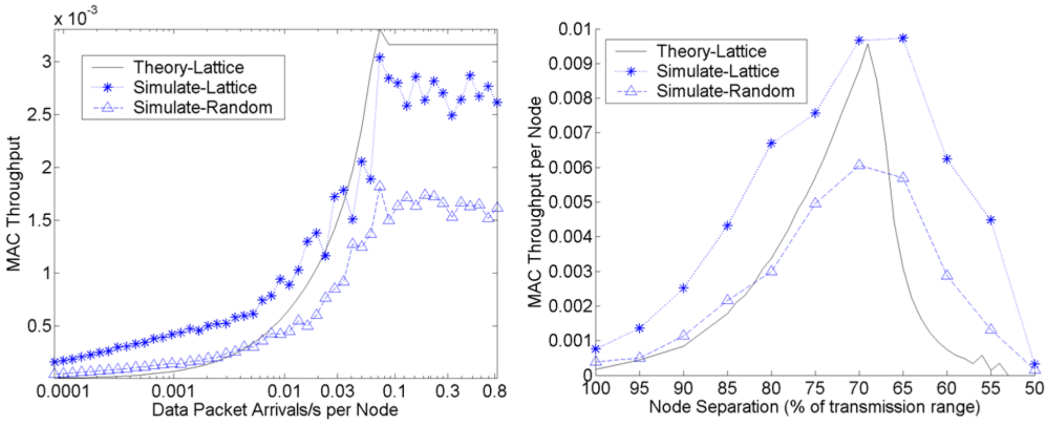
$$S_M = c_{link} \sum_{K=1}^{\overline{\Phi}} \binom{\overline{\Phi}}{K} \left(\frac{G_{eff}}{\overline{\Phi}} \right)^K \left(1 - \frac{G_{eff}}{\overline{\Phi}} \right)^{\overline{\Phi}-K} K P_{MAI}(K) P_{RS}(K) \tag{3.8}$$

The detail of the derivation is found in our papers [see List of Publications: P2, P3].

3.3.2 Numerical and Simulation Results

In this section, we present the numerical and simulation results for our DS/CDMA throughput analysis. The details on the simulation setup can be found in our papers [see List of Publications: P2, P3].

In figure 3.7, the MAC throughput is plotted as the data packet arrival rate per second per node, λ_{data} is varied to about $10 \times \lambda_{max}$. From the plot, we see that the simulation results are well approximated by our theoretical results. When λ_{data} is low (≈ 0.0001 to 0.02), the simulation throughput is slightly higher. This is due to the additional amount of route packets generated for new route search when the routing table expired which is not accounted for in the analysis. An important observation is that the throughput saturates at near the maximum packet transmission rate, λ_{max}. As the average delay increases exponentially beyond saturation, this implies that we can transmit at high data rates with low average delays.



a) Throughput for multi-hop sensor network scenario with Rayleigh fading channel and node separation equals to the transmission range

b) Throughput per node for Data Packet Arrival rate = 0.07 packets/s per node

Figure 3.7: Results for the throughput simulation of an UWA multi-hop sensor network using DS/CDMA and AODV protocols

This is different from CSMA/CA MAC protocol which saturates more rapidly due to contentions and retransmissions. The result from a more realistic topology is also included. Here, each node *a*, with a position, (x_a, y_a) is randomly displaced with uniform distribution within the boundary given by $(x_a \pm 0.5\varepsilon\eta_0, y_a \pm 0.5\varepsilon\eta_0)$ where $\varepsilon\eta_0$ is defined in section 3.3.1 as the spacing of the nodes. It is observed that the throughput has been scaled by about half due to the poorer connectivity of the network. However, the characteristic saturation point of the curve remains the same. This is consistent with our discussion that the saturation is caused by the time-varying connectivity because it affects the amount of AODV overheads generated. It can be inferred further that the throughputs of different network topologies differ by only constant fractional scaling factors without changes in their general characteristics, provided that the topologies do not vary in time. In sum, our theoretical throughput provides an upper limit when different static topologies are examined.

In figure 3.7b, the MAC throughput per node is plotted as the node separation is reduced from 100% to 50% of the transmission range, η_0. It is observed that initially the throughput increases up to the optimal separation of about 65% and 70% of η_0. At optimal separation, there is an improvement of a factor of 10 in the MAC throughput. The initial increase in throughput is due to improve network connectivity as the separation is reduced. However, the throughput decreases when the node separation is further reduced beyond the optimal separation. This is because of increase in interference from other nodes due to increase in the number of neighbors. Also, the throughput of the random topology is about half of that of the square lattice topology. This is consistent with the preceding discussion. In particular, the improvement factor at optimality remains at 10.

3.3.3 Conclusion

The theoretical MAC throughput of UWA multi-hop ad-hoc sensor networks with the use of an improved asynchronous half-duplex DS/CDMA protocol is analyzed and evaluated. Our ad-hoc multi-hop network is infrastructure-less which means that there are no centralized base stations. Hence, we have developed an asynchronous distributed half-duplex MAC protocol to regulate between the transmitting and receiving phases of transmissions. Our analysis accounts for two salient features of our implementations. First, the ability of the correlators in the DS/CDMA receiver to demodulate or de-spread the data in a Rayleigh fading channel with alpha-stable distributed ambient noise when no power control is implemented. Second, the effect of our asynchronous distributed half-duplex MAC protocol. Our networks can be deployed in shallow coastal waters for surveillance purposes.

However, they suffer from frequent disconnections due to Rayleigh multipath fading and high snapping shrimps dominated ambient noise. This is made worse by congestion of the channel due to the control packets generated to search for new routes. Unlike other works which simply abstracts the additional traffic load generated by the AODV protocol; it is explicitly accounted for in our analysis. Our results show that even with frequent disconnections, our network can transmit at close to maximum packet rate with low delay which is important for real-time applications. Furthermore, when the node separation is reduced, the throughput initially increases due to improve connectivity. It then decreases due to increase interference from increasing number of neighbors. The optimal node separation occurs between 65% and 70% of the transmission range for the Rayleigh fading parameter, $\sigma^2 = 0.3$. The MAC throughput at optimal node separation is increased by 10 times.

3.4 Conclusion

In this chapter, we investigate and analyze the communication and networking aspects of the MANET operating in realistic harsh environment where the radio transmission is subject to fading and noise. This gives us better insights into our problem and prepares us for the development of our cooperative control algorithm. In our theoretical analysis on communications, we examine the connectivity, MAC and routing issues to better understand on how to organize the sensors and design the control algorithm as follows:

The first problem is poor connectivity due to node mobility and fading channel conditions. When poor connectivity is due to independent random node mobility, the solution is to ensure that our motion control algorithm results in a coordinated orderly movement with minimal changes of neighbors in the network neighborhoods. For the poor fading channel condition, one approach is to use physical layer techniques such as spreading code modulation used in DS/CDMA which is known to be robust to fading channels. Finally, to further improve the robustness of our algorithm to the connectivity problem, our control algorithm makes use of only the information from three closest neighbors out of the other network neighbors. Since, they are the closest, they have better chances of been heard as, statistically, their mean received signal strengths are highest.

The second problem is contentions and interference. One approach is to organize the sensors into small cooperative groups and design our algorithm in such a way as to minimize inter-group communications. Another approach is to use DS/CDMA which is known to be robust to interference in the MAC layer.

The third problem is route breakages in networking resulting in excessive routing overheads and network congestion. Our analysis shows that there are two factors that affect the effective data traffic: presence of disconnections and routing overheads that are proportional to the total number of nodes in the network. Therefore, it is better to organize our sensors into a few small isolated cooperative groups and use intra-group communications to facilitate the harvesting tasks. The routing overheads can be further minimized if most communications are one-hop communications.

Our theoretical analysis on the wireless communication issues focused on a connectivity model that takes into account: random mobility and Rayleigh fading, an asynchronous half-duplex MAC layer for DS/CDMA in the multi-hop ad-hoc network environment and an analytical model to analyze the throughput of the DS/CDMA in UWA channel with Rayleigh fading and alpha-stable distributed noise.

Chapter 4: The Cooperative Control Algorithm

In this chapter, we discuss our main cooperative control algorithm which is used to control the movement of the mobile sensors in order to optimize the harvesting of spatial environmental information under energy constraints. In section 4.1, we explain the general design of the algorithm which takes into consideration the communication and energy constraints of mobile sensors operating in harsh environments. Our design is drawn upon the principles from an eclectic mix of cooperation, optimal control and statistical decision theories. In section 4.2, we present our cooperative algorithm. In section 4.3, we discuss our algorithm from the theoretical perspective. This is followed by the conclusion of the chapter.

4.1 General Overview

In this section, we give a general overview by emphasizing the three key elements in the algorithm. First, we discuss our choice in organizing the mobile sensor group to perform the data harvesting task. Second, we discuss our method in controlling the mobility of the nodes. Third, we discuss the information processing aspect.

4.1.1 Organization of the Mobile Sensor Group

There are three main approaches we can organize our group to perform the data harvesting task. The organization of the group is important because it determines the cooperative relationship among the sensors.

Figure 4.1a shows a coordination structure. In a coordination structure, each sensor in the group collects environmental information and reports this together with its current location to a controller. We define the concatenated environmental and location information as the state information as it is sufficient to drive the sensor to the next position. After aggregation and processing of all the information, the controller issues a command stating the next positions that the sensors should move to.

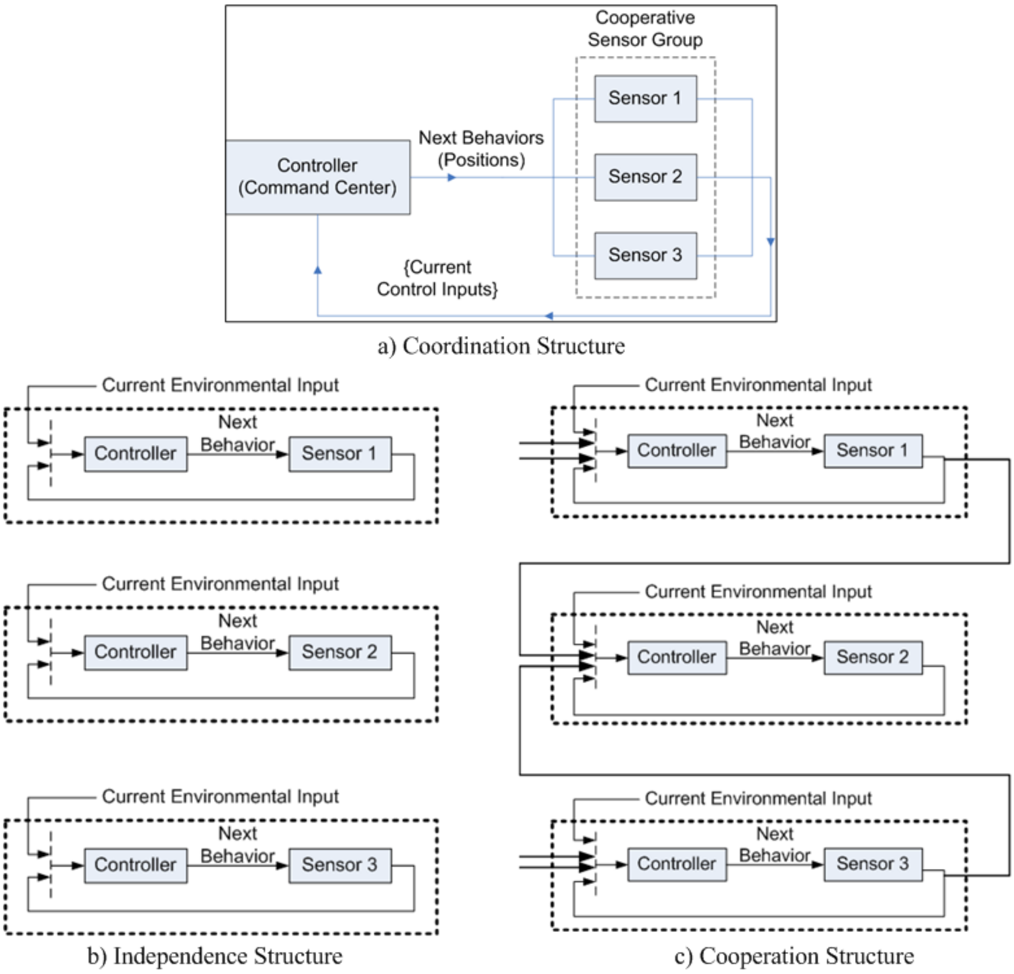


Figure 4.1: Different ways of organizing our mobile sensor group

In this structure, the controller has the collective global view of the terrain that the sensors are deployed in. In other words, it has almost full knowledge of the environment. It also bears all the processing loads. The sensors are essentially dumb sensors as they do not perform any processing and passively wait for the controller to drive their next positions. Note that the controller is an abstract concept and refers to a role that the agent is playing. In figure 4.1a, it is shown as the command center since this is the sink node where all the information in the network is sent and therefore the most natural node to take on this role. However, there is nothing to prevent any sensors in the group to take on this role if it has the required processing capacity. If this structure is adopted, our algorithm will be a centralized control algorithm. This is one of the popular structures mentioned in the robotic literature. The relationship between the sensors and the sink node is sometimes described as the "master-slave" relationship in the literature.

However, this approach has two disadvantages. First, since all the processing is performed in a single node, it must have a very powerful processor to reduce processing delay. This is especially true if the group size is large as the processing load and therefore delay increases exponentially with the number of state information inputs. Second, the group is highly dependent on the reliability of the wireless communications to perform its task. A loss in the communication packets could delay the movement of the group as the individual member has to wait passively for the packets to be resent before it knows what to do next. In a large network, multi-hop communications is required. If we characterize the reliability based on link failure probability, the reliability will drop by a multiplier every time the data packet transverses one hop towards its destination. That is, the probability that the packet is lost increases with the number of hops that the packet is required to traverse to reach the destination node. Moreover, excessive uses of multi-hop communications cause flooding and congestion of the network, locking up the communication resources.

Due to these two reasons, this approach is not scalable with the number of nodes, especially in a harsh communication environment.

Since the reliability of communications is a constraint in a harsh environment, one possible approach is to completely do away with communications. Figure 4.1b shows the independence structure. This structure is made possible due to the advance in processor technology, leading to smart sensors. Each sensor is described to be autonomous as it decides the next move independently based on its own current state input and past state inputs in its memory. With this definition, the sensors in figure 4.1a are non-autonomous. This approach results in a distributed control which is scalable with the number of sensors. Essentially each sensor plays the role of its own controller. This type of control is sometimes known as a self-regulated or self-controlled system model. However, this approach suffers from two weaknesses. First, as it no longer has shared information from others, its knowledge of the surroundings is limited to its own past memories. In a fast changing environment, the outdated memory it uses for processing could lead to erroneous response. Second, if we incrementally use the past information that is further back in time, there is more delay in the system response.

Finally, figure 4.1c shows the cooperation structure. An autonomous agent is described to be cooperative if its state behavior is partially influenced by the state behaviors of others. The level of cooperation will be determined by the level of influence. Notice that in figure 4.1c, sensor 2 obtains cooperatively shared state information from both sensor 1 and sensor 3. Intelligence can be defined briefly as the ability of the agent to digest information to its best advantage in adapting its behavior in order to achieve its goal in an optimal manner. Since this form of structure is able to acquire the most information at any time, the sensors in the cooperation structure are the more intelligent thus they exhibit also the more optimal behavior than the independent sensors.

We adopted the structure in figure 4.1c for our algorithm. Therefore, our main algorithm takes the form of a distributed control algorithm. This structure requires communications among the sensors. We restrict the communications in the cooperative relationship to one-hop communications for three reasons. First, as mentioned above, multi-hop communications are more unreliable and could lead to congestion and delay as discussed in chapter 3, section 3.2.2, eventually starving up scarce communication resources for other purposes. Second as the sensors only move around their neighborhoods in each time step, the information provided by the neighbors is more important. Third, too much unnecessary information results in processing delays and erroneous behaviors. This is especially true when the extra information is either unreliable, outdated or of low information content. We will discuss more in section 4.1.3.

4.1.2 Motion Control

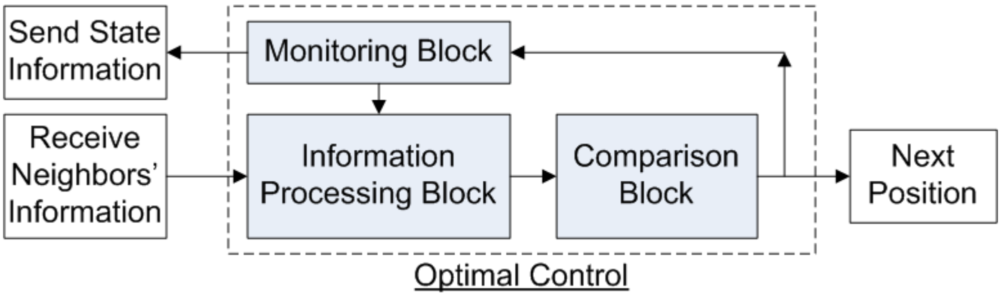


Figure 4.2: Cooperative optimal control

In Figure 4.2, we present a more detailed block diagram at the individual sensor level. The general scheme of our cooperative control block is similar to the optimal control block presented in [5]. The obvious difference is the use of state information from the neighbors. Generally, the state information is dependent on the context of the problem. We will define the state information for our context in the following section. Optimal control requires the sensor to monitor the environment changes. It also uses the current position as the collective input to the information processing block to generate the next move. The information processing block is to be implemented in the algorithm. Also, there is a comparison block in the optimal block. This is the block where the measure known as a metric is compared with a control goal to determine whether the system is performing optimally. The metric acts as a figure of merit for the sensor and is usually computed from measurable quantities that are observable by the sensor. The comparison block is also implemented within the algorithm. Note, also that the next position is completely determined by the current position and the measurement of the sensor and those of its neighbors. There is no usage of past historical state information. As a result, we can have a faster system response time.

4.1.3 Information Processing

There are three aspects of information that will affect the motion behavior of the sensor.

First aspect is the reliability of the information. This is usually modeled as noise in the data. Usually, noise filtering is required to alleviate the problem. In our context, there are two main types of noises. The first is the measurement noise in localization and sensor measurements. This is usually mitigated by the use of noise filtering, for example, particle filters. The second is channel noise. This is usually mitigated by the use of physical layer communication filters. As they have been researched by many others, they are beyond our research scope.

The second aspect is the timeliness of the information. That is whether the information is outdated. In our context, this occurs when the distribution is non-stationary. That is, external events are changing too fast that past information is useless and may even be misleading. In fact, many available pseudo Newton's search methods in the literature are known to be badly behaved if the past state information is not accurate, since they assume stationary distributions. Usually, the problem of outdated information can be mitigated by increasing the frequency of measurements (or equivalently, shortening the iteration time step size). Furthermore, if past information is used, usually some forms of expiration techniques are required for us to phase out the past information.

The third aspect is the information content of the data. The concept of information content is fairly abstract. Generally, the information content of the data is which that gives value to the user of the data. Therefore, it is a relative concept, depending on the context of the problem. In our context, the user is the sensor. From the perspective of the sensor, the data of higher information will be of higher value because it can better aid the sensor in more accurate and faster search of the optimal position. To digress, Shannon in his work on information theory coined the term *entropy* for the information content and attempts to define a generalized method to measure it. It leads to research on source coding which is a method used in communications and compression techniques. Essentially, the main motivation behind the techniques is for us to pack as much information as possible on a per data bit basis.

We elaborate further on the importance of the third aspect in our problem. In our problem, the sensor does not have a-priori knowledge of the environment except that the distribution is continuous in space. At each time step, it relies on its own discrete measurements together with its neighbors' to interpolate in order to estimate the gradient and hessian which is required for the sensor to navigate its surrounding.

This is analogous to the echolocation used by the blind bat to navigate the cave. By relying on a series of discrete sonar pulses it emits and from the time and phase differences of the discrete echoes, it estimates a continuous space which allows it to navigate safely in the cave and capture the prey. Note that performing this estimation does not require the bat to know in advance (a-priori knowledge) the map of the cave and the actual shapes, sizes and locations of the obstacles and preys. The same challenge is presented in our problem in that our sensors are also blind; they essentially make use of the discrete sensor data to navigate and will behave erroneously when the information content of the data is low.

4.2 The Algorithm

Our optimization design consists of two phases. The first phase requires the sensors to concurrently spread out and optimally place themselves such that (s.t.) there are more sensors in information fertile regions while maintaining sensing coverage. The sensing coverage refers to the union of the sensing range of all the individual sensors. In our context, our sensors attempt to sense the temperature of a source. If a source is located too far from a sensor, the temperature variation will be too small to be detectable. The second phase is where the optimally placed sensors move independently to harvest data around their localities. Moreover, each sensor has two states: **Optimize** and **Track**. This is to cater to possible movements in the sources in practical situations. The high-level framework of our algorithm with the three components is shown in figure 4.3

```
For Current Iteration, $k$:
1. If (State = Optimize)
    a. If (Phase = Cooperative) {Cooperative Optimal Placement}
    b. Else If (Phase = Independent) {Independent Harvesting}
2. Else If (State = Track)
    a. {Tracking}
3. End
Next Iteration, ($k$+1)
```

Figure 4.3: The high-level framework of our algorithm

We adopt the pseudo-Newton method to optimize the sensors' placements in a distributed manner. However, four problems are encountered. First, independent Newtonian methods search for a fixed goal – positions of zero gradients. Even after modifying the independent methods to handle fixed non-zero gradients, the sensors using the independent methods cannot spread out properly as they tend to overlap each other in their search and end up chasing after the same few goals. Second, independent pseudo-Newton method performs badly in a harsh environment because of estimation errors incurred due to localization noise. This is exacerbated by the accumulation of past errors which causes the sensors to persist in the erroneous direction even though current estimate is accurate until the influence of past information has faded. Third, the initial value problem in independent optimization in which the rate and probability of convergence is dependent on the initial position is more severe for our application. This is because we cannot make a good starting guess for the initial positions of the sensors as we have no advanced knowledge of the actual distributions. Fourth, the fixed line search used by some independent methods such as BFGS to stabilize the search is inefficient. Also, a fifth problem is encountered when there are moving sources. Movements of multiple sources at sufficiently high speed and different directions may cause confusions and disorderly behaviors among the sensors as their temperature measurements are the superposition of the effects of multiple sources.

4.2.1 Cooperative Optimal Placements

```
For Current Iteration, $k$:
S2. Extract Search Information from Neighbors.
S3. Perform Cooperative Pseudo-Newton Optimization.
S4. Perform Stabilization Procedure.
S5. Disseminate New State Information.
Next Iteration, $(k+1)$
```

Figure 4.4: The main cooperative control algorithm

Our cooperative optimization algorithm is shown in figure 4.4. We will elaborate the three main blocks: Cooperative Pseudo-Newton Optimization (S2), Stabilization (S3) and Dissemination and Extraction (S4, S1) in the sub-sections. Without loss of generality, we discuss our problem in the context of temperature measurements.

4.2.1.1 Cooperative Pseudo-Newton Optimization (S2)

In this section, we first consider the one-dimensional (1D) case in order to elucidate the main concept. We then extend our results to the two-dimensional (2D) case.

4.2.1.1.1 One-Dimensional Case

Figure 4.5 illustrates our problem in 1D case. In the top left hand side, five sensors are placed equally along a single line to measure the variation of a scalar quantity of interest, θ. Without loss of generality, we assume θ to be temperature. On the top right hand side, the data harvested are used to reconstruct a distribution map using linear interpolation. In the bottom sub-plots, we place the sensors in an optimal manner so that the same number of samples produces a better map with a correct peak.

Our optimization strategy is described as follows. Each node asynchronously and continuously adjusts its position w.r.t. its two adjacent neighbors until it determines through a metric that it has reached an optimum position. The two stationary boundary nodes are used as reference nodes. In actual implementation, these reference nodes are also beacons in the context of localization [14][66]-[68]. Furthermore, in order for us to perform these adjustments, the nodes will have to broadcast their state information at regular intervals. The state information of node *i*, consists of the location information concatenated with the temperature in the form, [x_i, θ_i]. We limit this broadcasting of state information to the local neighborhood to prevent excessive consumption of communication bandwidth.

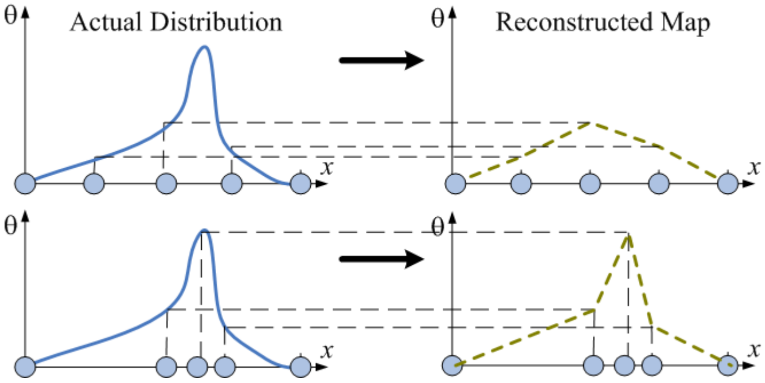


Figure 4.5: Quality enhanced reconstructed distribution map using optimally spaced sensors

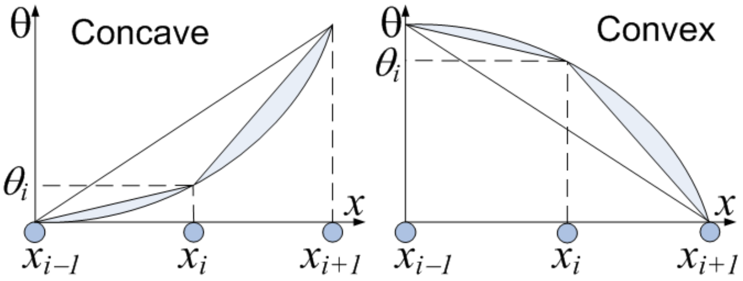


Figure 4.6: Local distortion metrics

In order for us to implement a distributed form of optimal control, we need to identify a goal that is observable and measurable at the individual node level. We divide our sensors into many small groups of three that are coupled together so that every node is a member of two adjacent groups. Figure 4.6 shows two possible scenarios that a group of three sensors: $\{i-1, i, i+1\}$ can handle in a small segment of the distribution. This is based on the assumption that there is sufficient number of sensors so that practically, every three nodes will see either a simple convex or concave curve.

We shall now formulate our problem from the control perspective. In a control problem, we want to control and adjust the behavior of a system so that the behavior moves towards one that can attain a desired outcome or system goal. The principle requires that the selected behavior is controllable and observable. In our context, the system is the individual mobile sensor and the behavior of the mobile sensor that we desire to control is the position of the sensor. We will first define our cooperative control problem from control-theoretic perspective using the difference equation since our control is performed at discrete time step instead of continuous time. The general expression for our cooperative control problem is defined as follows:

$$p_i^{(k+1)} = p_i^{(k)} + \Delta p_i^{(k)}\left(C_{sn}^{(k)}\right) \tag{4.1}$$

Where:

$p_i^{(k)}$ The position of sensor *i* in the k^{th} time step.

$\theta_i^{(k)}$ The measurement made by sensor *i* in position $p_i^{(k)}$ in the k^{th} time step.

$s_i^{(k)}$ The state vector of sensor *i* in the k^{th} time step. It is defined as the concatenation of $p_i^{(k)}$ and $\theta_i^{(k)}$. $\therefore s_i^{(k)} = [p_i^{(k)}, \theta_i^{(k)}]$.

$C_{sn}^{(k)}$ The set that represents the states of the sensors belonging to the same cooperative group in the *k*th time step.

$\Delta p_i^{(k)}\left(C_{sn}^{(k)}\right)$ This is the position control function in the *k*th time step. It takes $C_{sn}^{(k)}$ as the input and computes the amount of adjustment to be added to the current position, $p_i^{(k)}$ in order to obtain the next position.

$V_{i,sn}^{(k)}$ The set that represents the states of the Voronoi neighbors of sensor *i* in the *k*th time step, exclusive of sensor *i*.

The control equation describes the dependency of the position of sensor *i* with the states of all the sensors within the same cooperative group of sensor *i*. It can be implemented in a distributive manner by having each sensor performs the computation in (4.1) based on information gathered from its cooperative group. Therefore, there is no central authority that gathers information on behalf of the group and assigns the set of next positions for the mobile sensors in the cooperative group.

Finally, before we move on, it is instructive for us to examine the basic structure of the simplest form of (4.1). We look at the structure of the simplest linear autonomous control equation:

$$p_i^{(k+1)} = p_i^{(k)} + K_u \left\{ u_{goal} - u_i\left(s_i^{(k)}\right) \right\} \qquad (4.2)$$

In (4.2), the control function consists of a goal function, u_{goal}. The principle dictates that the control goal must be observable and measurable by the system (that is, the sensor *i*). At position $p_i^{(k)}$, sensor *i* computes a value, $u_i\left(s_i^{(k)}\right)$ which is dependent only on the current state of sensor *i* for autonomous control. The amount of adjustment or control requires for the behavior (that is, the position) of sensor *i* depends on the deviation of $u_i\left(s_i^{(k)}\right)$ from the goal. The larger the deviation, the more adjustment we need to make to its position. The value K_u is known as the control gain and is a constant in linear control. If K_u is too small, the system takes a long time to converge to its desired behavior and if K_u is too large, the system is highly oscillatory and may never converge to its desired behavior. Clearly, from the discussion, before we can solve (4.2), we must first define and express our control goal in the form that is both observable and measurable.

Let $\hat{\theta}$ be the piecewise linear approximation map of θ based on the sensor readings. We shall now formally define our measure of distortion by the global distortion error metric, D_e in two variations as follows:

$$D_e = \int_E \left|\theta - \hat{\theta}\right| dx \tag{4.3a}$$

$$D_e = \sum_{i=1}^{N_S - 1} \left| \int_{p_i}^{p_{i+1}} \theta dx - \frac{1}{2}(\theta_{i+1} + \theta_i)(p_{i+1} - p_i) \right| \tag{4.3b}$$

The second variation of D_e in (4.3b) which is derived from (4.3a) is more important for a practical reason because it expresses the global distortion error as the sum of local distortion errors. This form suggests that it is sufficient for the distributed algorithm to use the local distortion error to adjust the individual node positions in order to achieve the global minimum.

Referring to Figure 4.7, D_e can be visualized as the area between the actual temperature distribution θ and the approximate map $\hat{\theta}$ from the sensor reading. θ is composed of a concave region follow by a convex region. Clearly, our objective is to select the positions of the mobile sensors, *P* such that the total area is minimized.

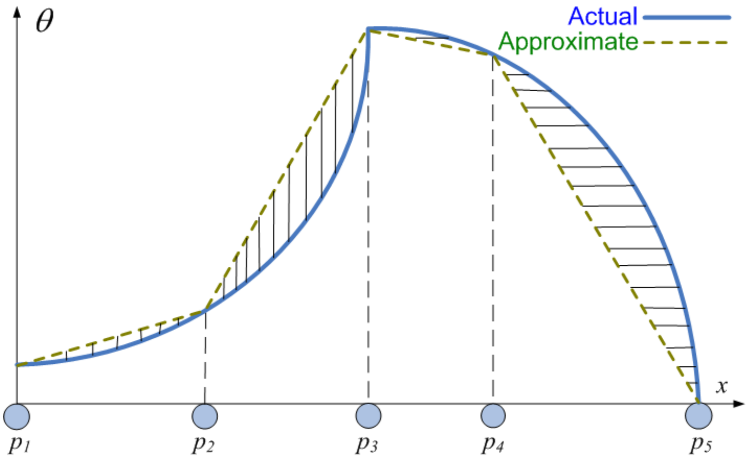


Figure 4.7: Distortion error

An important observation is that for concave/convex regions, our approximate map $\hat{\theta}$ is lower/upper bounded by θ. For example, in the concave region, to minimize the distortion, we only need to choose the position p_2 so that it minimizes the local area of $\hat{\theta}$. This is because we know that this local area of $\hat{\theta}$ must be lower bounded by the local area of θ and vice versa. The second observation is that whether the region is concave or convex can be determined by testing the sign of the second derivative of the locality. The local area of the approximate map as seen by sensor *i*, LA_i, is given as:

$$LA_i = \frac{1}{2}\left[(\theta_i + \theta_{i-1})(p_i - p_{i-1}) + (\theta_{i+1} + \theta_i)(p_{i+1} - p_i)\right] \tag{4.4}$$

Referring to Figure 4.7 again, our optimization strategy can be briefly described as follows:

- For the **concave** region, sensor 2 should adjust its position p_2 with respect to its 2 Voronoi neighbors in order to minimize LA_2. Since, we know that our approximate map is lower bounded by θ.

- For the **convex** region, sensor 4 should adjust its position, p_4 with respect to its 2 Voronoi neighbors in order to maximize LA_4. Since we know that our approximate map is upper bounded by θ.

We now examine the condition for minimum/maximum. We differentiate (4.4) w.r.t. p_i.

$$\begin{aligned}\frac{\partial LA_i}{\partial p_i} &= \frac{1}{2}\left[\frac{\partial \theta_i}{\partial p_i}(p_i - p_{i-1}) + (\theta_i + \theta_{i-1})(1) + \frac{\partial \theta_i}{\partial p_i}(p_{i+1} - p_i) + (\theta_{i+1} + \theta_i)(-1)\right] \\ &= \frac{1}{2}\left[\frac{\partial \theta_i}{\partial p_i}(p_{i+1} - p_{i-1}) + (\theta_{i-1} - \theta_{i+1})\right]\end{aligned} \tag{4.5}$$

To obtain the condition for minimum/maximum, we set $\frac{\partial LA_i}{\partial p_i} = 0$,

$$\frac{\partial LA_i}{\partial p_i} = \frac{1}{2}\left[\frac{\partial \theta_i}{\partial p_i}(p_{i+1} - p_{i-1}) + (\theta_{i-1} - \theta_{i+1})\right] = 0$$

$$\Leftrightarrow \frac{\partial \theta_i}{\partial p_i}(p_{i+1} - p_{i-1}) + (\theta_{i-1} - \theta_{i+1}) = 0$$

$$\therefore \frac{\partial \theta_i}{\partial p_i} = \frac{(\theta_{i+1} - \theta_{i-1})}{(p_{i+1} - p_{i-1})} \tag{4.6}$$

The condition in (4.6) states that in order for us to obtain a minimum /maximum point, we should adjust the position of sensor *i* such that the gradient at the tangent point (p_i, θ_i) is equal to the gradient of the secant line bounded by the Voronoi neighbors of sensor *i*. In other words, the tangent is parallel to the secant line at the minimum/maximum. The optimal condition for Figure 4.7 is illustrated in Figure 4.8.

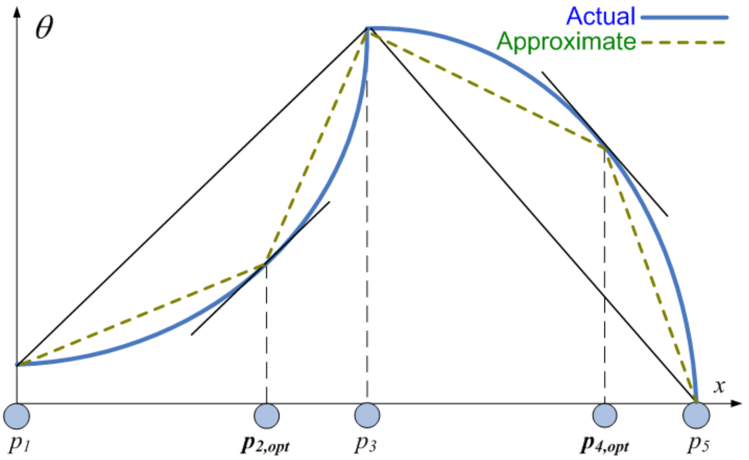


Figure 4.8: Optimal condition of minimum distortion error

For a continuous function, (4.6) holds and we can always find an intermediate point that has a tangent which is parallel to the secant line based on Mean Value Theorem. Therefore, it shows us that it is possible for intermediate node *i* to adjust its position in between its two neighbors to minimize distortion without changing the relative topological relationship with its Voronoi neighbors. Applying the argument inductively, it is feasible for us to develop a distributive algorithm that preserves the network topology.

We now take the second derivative of LA_i using (4.5),

$$\therefore \frac{\partial^2 LA_i}{\partial p_i^2} = \frac{1}{2}\frac{\partial^2 \theta_i}{\partial p_i^2}\left(p_{i+1} - p_{i-1}\right) \tag{4.7}$$

Note that by virtue of the ordering $\left(p_{i+1} - p_{i-1}\right)$ is always positive. For the concave region, $\frac{\partial^2 \theta_i}{\partial p_i^2} > 0 \Rightarrow \frac{\partial^2 LA_i}{\partial p_i^2} > 0$. Therefore, the optimal point is a minimum and vice versa. This is in agreement with our observation in Figure 4.7 and 4.8 that we should minimize/maximize the area in the concave/convex region in order to minimize distortion because they are bounded by θ.

However, we need to address two issues. First, since it is unlikely that node *i* reaches this goal initially, we need an adaptive approach to progressively move node *i* in the correct direction and speed in order to reach this goal. Second, the temperature gradient is not directly measurable since our sensors do not have a-priori knowledge of the actual distribution. It has to be estimated. We will resolve these two issues for the 2D case. The detailed derivation can also be found in appendix B.

4.2.1.1.2 Two-Dimensional Case

In 2D, it can be shown that our optimization requires concurrent maximization of the volumes of the tetrahedrons bounded by the sensors and their three neighbors. In figure 4.9, we show sensor *i* in the center of the group enclosed by three nearest surrounding neighbors labeled 1, 2 and 3 in the anti-clockwise direction. Although we have illustrated the concept of neighborhood couplings using equidistant nodes, it is also applicable for non-equidistant nodes.

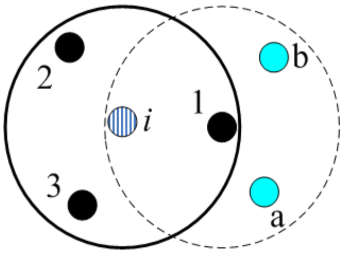


Figure 4.9: Neighborhood couplings

The neighbors themselves form the centers of other groups due to the couplings. Particularly, sensor 1 is the center of another neighborhood and sensor *i* is its neighbor. There are several practical benefits in the use of nearest neighbors. Wireless communications tends to be unreliable and the sensors are subjected to time-variable link failures [79][80][see List of Publications: P1-P3]. Statistically, the closer the neighbors are, the better is the received signal strength. Therefore, it makes our application robust. Also, the sensor is constrained to move within the triangular enclosure formed by the three neighbors in each step so that the network topology is invariant with time. This has the beneficial effect of reducing multi-hop routing overheads. This is useful as our network is likely to serve other purposes that require multi-hop communications.

Recall that in the derivation for the 1D case, we examine the area of the distribution under a node bounded by two referencing neighbors. Therefore, based on current state information, the node chooses the best position to move towards so as to either maximize or minimize the area of the distribution. The individual goal turns out to be equivalent to seeking a position such that the temperature gradient in that position is equal to the mean gradient taken over the two reference points and the intermediate node.

We use the same argument to obtain the expression for the 2D case, by considering the volume bounded by an intermediate node and three referencing neighbors which is the volume of the tetrahedron with four vertices: (x_1, y_1, θ_1), (x_2, y_2, θ_2), (x_3, y_3, θ_3) and (x_i, y_i, θ_i). Let V be the volume of the tetrahedron, we have,

$$V = \frac{1}{3!}\begin{vmatrix} x_i & y_i & \theta_i & 1 \\ x_1 & y_1 & \theta_1 & 1 \\ x_2 & y_2 & \theta_2 & 1 \\ x_3 & y_3 & \theta_3 & 1 \end{vmatrix}$$

$$\Rightarrow 6V = x_i[y_1(\theta_2 - \theta_3) - y_2(\theta_1 - \theta_3) + y_3(\theta_1 - \theta_2)] - x_1[y_i(\theta_2 - \theta_3) - y_2(\theta_i - \theta_3) + y_3(\theta_i - \theta_2)]$$

$$+ x_2[y_i(\theta_1 - \theta_3) - y_1(\theta_i - \theta_3) + y_3(\theta_i - \theta_1)] - x_3[y_i(\theta_1 - \theta_2) - y_1(\theta_i - \theta_2) + y_2(\theta_i - \theta_1)]$$

(4.8)

To maximize/minimize V, set $\frac{\partial V}{\partial x_i} = 0$ and $\frac{\partial V}{\partial y_i} = 0$

$$6\frac{\partial V}{\partial x_i} = [y_1(\theta_2 - \theta_3) - y_2(\theta_1 - \theta_3) + y_3(\theta_1 - \theta_2)]$$

$$-x_1\left[-y_2\frac{\partial\theta_i}{\partial x_i}+y_3\frac{\partial\theta_i}{\partial x_i}\right]+x_2\left[-y_1\frac{\partial\theta_i}{\partial x_i}+y_3\frac{\partial\theta_i}{\partial x_i}\right]-x_3\left[-y_1\frac{\partial\theta_i}{\partial x_i}+y_2\frac{\partial\theta_i}{\partial x_i}\right]=0$$

$$\Rightarrow\frac{\partial\theta_i}{\partial x_i}=-\begin{vmatrix}y_1 & \theta_1 & 1\\ y_2 & \theta_2 & 1\\ y_3 & \theta_3 & 1\end{vmatrix}\begin{vmatrix}x_1 & y_1 & 1\\ x_2 & y_2 & 1\\ x_3 & y_3 & 1\end{vmatrix}^{-1} \tag{4.9}$$

Also,

$$6\frac{\partial V}{\partial y_i}=-x_1\left[(\theta_2-\theta_3)-y_2\frac{\partial\theta_i}{\partial y_i}+y_3\frac{\partial\theta_i}{\partial y_i}\right]$$

$$+x_2\left[(\theta_1-\theta_3)-y_1\frac{\partial\theta_i}{\partial y_i}+y_3\frac{\partial\theta_i}{\partial y_i}\right]-x_3\left[(\theta_1-\theta_2)-y_1\frac{\partial\theta_i}{\partial y_i}+y_2\frac{\partial\theta_i}{\partial y_i}\right]=0$$

$$\Rightarrow\frac{\partial\theta_i}{\partial y_i}=\begin{vmatrix}x_1 & \theta_1 & 1\\ x_2 & \theta_2 & 1\\ x_3 & \theta_3 & 1\end{vmatrix}\begin{vmatrix}x_1 & y_1 & 1\\ x_2 & y_2 & 1\\ x_3 & y_3 & 1\end{vmatrix}^{-1} \tag{4.10}$$

Let $A=\begin{vmatrix}x_1 & y_1 & 1\\ x_2 & y_2 & 1\\ x_3 & y_3 & 1\end{vmatrix}$, $B=\begin{vmatrix}y_1 & \theta_1 & 1\\ y_2 & \theta_2 & 1\\ y_3 & \theta_3 & 1\end{vmatrix}$ and $C=\begin{vmatrix}x_1 & \theta_1 & 1\\ x_2 & \theta_2 & 1\\ x_3 & \theta_3 & 1\end{vmatrix}$

$$\therefore\frac{\partial\theta_i}{\partial x_i}=-BA^{-1}\text{ and }\frac{\partial\theta_i}{\partial y_i}=CA^{-1}$$

Substituting the above into the gradient equation, we have our objective function as:

$$\nabla\theta_i=\begin{bmatrix}\frac{\partial\theta_i}{\partial x_i} & \frac{\partial\theta_i}{\partial y_i}\end{bmatrix}=A^{-1}[-B\quad C]^T \tag{4.11}$$

Let $u_{goal}=A^{-1}[-B\quad C]$.

$$\therefore\nabla\theta_i=u_{goal} \tag{4.12}$$

Therefore, simply stated, the equation requires the node to search for a point in the triangular region bounded by its three neighbors such that the gradient at that point is equal to the mean gradient taken over the triangular region.

Since, we are looking at a continuous temperature distribution, an appropriate class of search methods will be the descent method. Newton search method is the search method in this class that offers a rapid second order of convergence. The rapid convergence is due to the use of more information (first and second order derivatives) in the search method. The assumption in this method is that we have a-priori perfect knowledge of the distribution. We will look at the modifications required for the case where we do not have a priori perfect knowledge of the distribution. A discussion of the Newton Search Method can be found in [5].

Ignoring the constant term, $\frac{1}{6}$ in (4.8), we let

$$\begin{aligned} V' &= x_i[y_1(\theta_2-\theta_3)-y_2(\theta_1-\theta_3)+y_3(\theta_1-\theta_2)] \\ &-x_1[y_i(\theta_2-\theta_3)-y_2(\theta_i-\theta_3)+y_3(\theta_i-\theta_2)] \\ &+x_2[y_i(\theta_1-\theta_3)-y_1(\theta_i-\theta_3)+y_3(\theta_i-\theta_1)]-x_3[y_i(\theta_1-\theta_2)-y_1(\theta_i-\theta_2)+y_2(\theta_i-\theta_1)] \end{aligned}$$

The idea of Newton's method is to maximize V' w.r.t. $p_i = (x_i, y_i)$ by maximizing the quadratic approximation of V' about $p_i = (x_i, y_i)$. This maximization is performed at each iterative step of the algorithm. Specifically, using a second-order Taylor series expansion around the point, $p_i = (x_i, y_i)$, we may write:

$$\Delta V'(p_i) = V'(p_i^{(k+1)}) - V'(p_i^{(k)}) \approx g^T(k)\Delta p_i^{(k+1)} + \frac{1}{2}(\Delta p_i^{(k)})^T H(k)(\Delta p_i^{(k)}) \quad (4.13)$$

where ,

$$g(k) = \nabla V'(k) = \begin{bmatrix} \frac{\partial V'(k)}{\partial x_i} & \frac{\partial V'(k)}{\partial y_i} \end{bmatrix}^T$$

$$H(k) = \nabla V'(k) = \begin{bmatrix} \frac{\partial^2 V'(k)}{\partial x_i^2} & \frac{\partial^2 V'(k)}{\partial x \partial y} \\ \frac{\partial^2 V'(k)}{\partial y \partial x} & \frac{\partial^2 V'(k)}{\partial y_i^2} \end{bmatrix}$$

For a continuous function, $\frac{\partial^2 V'(k)}{\partial x \partial y} = \frac{\partial^2 V'(k)}{\partial y \partial x}$, therefore, the Hessian, *H(k)* is symmetric. The change $\Delta V'(p_i)$ is minimized or maximized when $\nabla[\Delta V'(p_i)] = \underline{0}$. Therefore, differentiating (4.13) w.r.t Δp_i to obtain $\nabla[\Delta V'(p_i)]$, we have,

$$\nabla[\Delta V'(p_i)] = g(k) + H(k)\Delta p_i^{(k)} = \underline{0}$$

$$\Rightarrow \Delta p_i^{(k)} = -H^{-1}(k)g(k) \qquad (4.14)$$

$$\therefore p_i^{(k+1)} = p_i^{(k)} + \Delta p_i^{(k)} = p_i^{(k)} - H^{-1}(k)g(k) \qquad (4.15)$$

We express the first and second partial derivatives of $V'(k)$ as the temperature derivatives and substitute them into *g*(*k*) and *H*(*k*) as follows:

$$\frac{\partial V'}{\partial x_i} = \begin{vmatrix} y_1 & \theta_1 & 1 \\ y_2 & \theta_2 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix} + \frac{\partial \theta_i}{\partial x_i} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \qquad (4.16)$$

$$\frac{\partial V'}{\partial y_i} = \frac{\partial \theta_i}{\partial y_i} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} - \begin{vmatrix} x_1 & \theta_1 & 1 \\ x_2 & \theta_2 & 1 \\ x_3 & \theta_3 & 1 \end{vmatrix} \qquad (4.17)$$

$$\frac{\partial^2 V'}{\partial x_i \partial y_i} = \frac{\partial^2 \theta_i}{\partial x_i \partial y_i} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \qquad (4.18)$$

Using: $A = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$, $B = \begin{vmatrix} y_1 & \theta_1 & 1 \\ y_2 & \theta_2 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix}$ and $C = \begin{vmatrix} x_1 & \theta_1 & 1 \\ x_2 & \theta_2 & 1 \\ x_3 & \theta_3 & 1 \end{vmatrix}$ and substituting (4.16) to (4.18) into *g*(*k*) and *H*(*k*), we have,

$$g(k) = [B \quad -C]^T + A\nabla\theta_i^{(k)} \qquad (4.19)$$

$$H(k) = A\nabla^2\theta_i^{(k)} \qquad (4.20)$$

The form presented in (4.15) is the form used in Newton Search Method. We write (4.15) in the standard control equation form for the purpose of stabilization.

$$\Delta p_i^{(k)} = -H^{-1}(k)g(k) = \left\{u_{goal} - \nabla\theta_i^{(k)}\right\}K_u \tag{4.21}$$

where $K_u = (\nabla^2\theta_i^{(k)})^{-1}$ is the control gain.

The goal which is the gradient to be searched is cooperative because it is adaptive and dependent on the current state information of the neighbors. This novel approach allows the sensors to concurrently spread out and optimize their positions. Now, (4.21) represents the ideal case where the distribution is known a-priori to the sensors. In our problem, the distribution is unknown and thus the current gradient and hessian of the temperature, $\nabla\theta_i^{(k)}$ and $\nabla^2\theta_i^{(k)}$ cannot be directly evaluated. We do not have any a-priori information on the temperature distribution except that θ_i is smooth and continuous.

To obtain an approximation of θ_i, we perform a local interpolation of the measurements available at the four points: p_i, p_1, p_2 and p_3 using the RBF Interpolation and derive the approximate partial derivatives. The RBF interpolation is chosen for our 2D formulation because it can be done with three neighbors that are not confined to a square grid. A bilinear interpolation in 2D would require a point to be surrounded by at least four neighbors located on a rectangular grid.

Let $\|\bullet\|$ be the norm of a vector. Let $\varphi(\|p - p_h\|) = \exp\left(-\|p - p_h\|^2 / 2\sigma^2\right)$, where $h = i$, 1, 2, and 3. We choose the RBF in Gaussian form although other equivalent choices are available [22] for the ease of manipulating the expression mathematically to a compact form. The temperature of an arbitrary point p, inside the region surrounded by the three neighbors is given by,

$$\theta(p) \approx \sum_{h=i,h=1}^{3} w_h \varphi(\|p - p_h\|) \tag{4.22}$$

where, the weight vector, $\underline{w} = [w_i \quad w_1 \quad w_2 \quad w_3]^T$ is obtained from the interpolation matrix, Φ as follows:

$$\underline{w} = \Phi^{-1}\underline{\theta} \tag{4.23}$$

where, $\Phi = \begin{bmatrix} \varphi_{ii} & \varphi_{i1} & \varphi_{i2} & \varphi_{i3} \\ \varphi_{1i} & \varphi_{11} & \varphi_{12} & \varphi_{13} \\ \varphi_{2i} & \varphi_{21} & \varphi_{22} & \varphi_{23} \\ \varphi_{3i} & \varphi_{31} & \varphi_{32} & \varphi_{33} \end{bmatrix}$ and $\varphi_{lh} = \varphi(\|p_i - p_h\|)$, $l = i$, 1, 2, and 3. Also,

$\underline{\theta} = [\theta_i \quad \theta_1 \quad \theta_2 \quad \theta_3]^T$

We can now obtain the gradients from (4.22) by differentiation as follows:

$$\nabla\theta(p) \approx -\sigma^{-2}\begin{bmatrix} \sum\limits_{h=i,h=1}^{3} w_h(x - x_h)\varphi(\|p - p_h\|) & \sum\limits_{h=i,h=1}^{3} w_h(y - y_h)\varphi(\|p - p_h\|) \end{bmatrix}^T \tag{4.24}$$

$$\nabla^2\theta(p) \approx \sigma^{-4}\begin{bmatrix} \sum\limits_{h=i,h=1}^{3} w_h\left[(x - x_h)^2 - \sigma^2\right]\varphi(\|p - p_h\|) & \sum\limits_{h=i,h=1}^{3} w_h(x - x_h)(y - y_h)\varphi(\|p - p_h\|) \\ \sum\limits_{h=i,h=1}^{3} w_h(x - x_h)(y - y_h)\varphi(\|p - p_h\|) & \sum\limits_{h=i,h=1}^{3} w_h\left[(y - y_h)^2 - \sigma^2\right]\varphi(\|p - p_h\|) \end{bmatrix}$$

(4.25)

Now we substitute $p = p_i^{(k)}$ into (4.24) and (4.25). Let $\mathbf{I}_2$ be the 2×2 identity matrix. Let $d'_{ih} = -\sigma^{-2}[(x_i - x_h) \quad (y_i - y_h)]^T$ and $d''_{ih} = (d'_{ih}d'^{\,T}_{ih} - \sigma^{-2}\mathbf{I}_2)$. Equation (4.22), (4.25) and (4.26) can then be expressed in compact matrix form as,

$$\theta_i \approx \sum_{h=i,h=1}^{3} w_i\varphi_{ih} \tag{4.26}$$

$$\nabla\theta_i \approx \sum_{h=i,h=1}^{3} w_i d'_{ih}\varphi_{ih} \tag{4.27}$$

$$\nabla^2\theta_i \approx \sum_{h=i,h=1}^{3} w_i d''_{ih}\varphi_{ih} \tag{4.28}$$

To summarize, the main equations to compute for our cooperative optimization step are given as:

Estimation of gradient and Hessian:

$$\begin{aligned}
&\nabla \theta_i^{(k)} \approx \sum_{j=1}^{3} w_i d'_{ij} \varphi_{ij}; \quad \nabla^2 \theta_i^{(k)} \approx \sum_{j=1}^{3} w_i d''_{ij} \varphi_{ij} \\
&\varphi_{ij} = \exp(-\left\| p_i^{(k)} - p_j^{(k)} \right\|^2 \Big/ 2\sigma^2) \\
&d'_{ij} = -\sigma^{-2} (p_i^{(k)} - p_j^{(k)})^T; \quad d''_{ij} = (d'_{ij} d'^{\,T}_{ij} - \sigma^{-2} \mathbf{I}_2) \\
&\underline{w} = [w_i \quad w_1 \quad w_2 \quad w_3]^T = \Phi^{-1} \underline{\theta}
\end{aligned} \tag{4.29}$$

where: $p_i^{(k)}$ =($x_i^{(k)}$, $y_i^{(k)}$) is the position of sensor *i* at *k*th step. $\underline{\theta} = [\theta_i \quad \theta_1 \quad \theta_2 \quad \theta_3]^T$,

$\Phi = [\varphi_{lh}]$ is the 4×4 interpolation matrix and $\mathbf{I}_2$ is the 2×2 identity matrix.

Computation of next position to move:

$$\Delta p_i^{(k)} = \left\{ u_{goal} - \nabla \theta_i^{(k)} \right\} K_u \tag{4.30}$$

where $K_u = (\nabla^2 \theta_i^{(k)})^{-1}$ is the control gain.

A novel feature in our approach is that it does not use past information. Together with the use of dissemination discussed in section 4.2.1.3, our approach can search more accurately and faster than independent approaches. This feature allows us to eliminate the adverse memory effect which slows down the optimization due to two causes. First, in a harsh environment with localization errors, use of past information tends to cause accumulated estimation error. Second, in the simulation, the use of (4.29) becomes more accurate as the sensor and its three neighbors converge to optimal positions. Thus, the use of poor quality past information would have actually hindered the search. Particularly, it is observed in the simulation for sensors using BFGS, past information acts as a burden in harsh environment as they are given weights in the estimation and their influences take time to fade. Consequently, sensors using BFGS tend to persist stubbornly in the erroneous direction for a longer time than those using our approach.

The detailed derivation of this subsection can also be found in appendix C.

4.2.1.2 Stabilization (S3)

Stabilization of search is required initially. This is because the sensors are not near optimal positions. Therefore, the cooperatively shared state information those sensors provide among themselves as search information has low information content. This problem is also present in independent pseudo-Newton algorithms. To elaborate, in the early stage of an independent search, past and current information are collected near the vicinity of the search. As a result, they are low in content and do not contribute much to the estimation. This problem is especially severe when the initial position is far from the optimal point, leading to erratic and badly behaved search. Generally, independent algorithms adopt conservative approaches in the initial stage. The approach used in BFGS method is to initialize the hessian as the identity matrix. Consequently, the initial search is the stable but slow gradient descent search. However, note that this method of initialization may still lead to erratic search if the initial position is too far from the search goal. To further stabilize the search, a constrained line search is performed. In a line search, an approximate local optimal point is searched along a straight line. Only when it is found that there is a change of direction. Inefficiency occurs in the initial stage as most likely a wrong direction is chosen. In the worst case, the search continues along this direction until the boundary of the constrained feasible space is reached.

We resolve this issue by making our stabilized direction adaptive. This is inspired by how flocks of birds fly or move in search of food. The movement of each bird is guided by the leading neighbors in the general flow of the larger movement. Essentially, we capitalize on the cooperatively shared information and adopt the method of Maximal Ratio Combining (MRC) in statistical decision making theory to select the most reliable steering direction. Our objective is to make a choice in a manner that minimizes the cost of the decision error. The cost is assumed to be quantifiable. It is also assumed that we could measure the reliability of a choice, usually through a metric. The MRC strategy is considered to be the most appropriate when there is a group of reliable choices and the standard deviation is not too large among them. In this strategy, we use a weighted average of the choices. The average is weighted such that the most reliable choice is given the highest weight. It is considered the best among the three possible strategies (see Appendix C for other strategies and detailed derivations) because it provides diversification of risks as well as minimization of the cost of decision error due to the use of unreliable choice.

In our problem, a good measure of reliability is the magnitude of the mean temperature gradient along the path leading to one of the neighbors. This is because it indicates the likelihood that there are minimum or maximum points in that direction. We would use this to help the node decides on the best direction to move. First, we define the mean directional derivative from point p_i to p_j as follows:

$$D_{ij} = \frac{\theta_j - \theta_i}{\left\| p_j - p_i \right\|} u_{ij} \tag{4.31}$$

where u_{ij} is the unit steering directional vector pointing from point i to point j, given as:

$$u_{ij} = \frac{p_j - p_i}{\left\| p_j - p_i \right\|} \tag{4.32}$$

The mean directional derivative D_{ij} describes the average temperature change when a point moves along the straight path from point i to j.

To perform the orientation stabilization, we use the mean directional derivative as a criterion to choose a steering direction. Let D_{st} and u_{st} be the directional derivative and the unit directional vector for the steering node respectively. D_{st} is obtained from MRC as follows:

$$D_{st} = \sum_{j=1}^{3} \left\| D_{ij} \right\|^2 u_{ij} \tag{4.33}$$

u_{st} is then obtained as follows: $u_{st} = D_{st} \left\| D_{st} \right\|$

$\Delta p_i^{(k)}$ obtained in (4.30) is then restrained to move along u_{st} as:

$$\Delta p_{i,st}^{(k)} = \left| u_{st}^{\mathrm{T}} \Delta p_i^{(k)} \right| u_{st} \tag{4.34}$$

We name our concept in (4.34) as *orientation stabilization*. This stems from our simulation studies of the trajectory plots that the sensors appear erratic and disorientated without orientation stabilization. At the later stage, as the general flow of movement for the group as a whole stabilizes and moves towards convergence, the directions provided by $\Delta p_i^{(k)}$ and u_{st} become gradually aligned. That is, the restraint imposed by the steering direction, u_{st}, will eventually become redundant.

Two additional practical constraints are imposed to stabilize the cooperative group. First, the sensors are limited to move up to a maximum radius from their initial positions. This constraint is imposed to prevent the sensors from attracting too near to a hotspot. Without this constraint, it may result in an eventual loss of sensing coverage if all the sensors are attracted and concentrated in one spot in the terrain. Sufficient sensing coverage in the terrain is a practical requirement as spontaneous events may occur in random locations. Second, due to the limits in the accuracies of the temperature and location measurements, we impose constraints to prevent the sensors from being over-sensitive and jittery about their positions. Let ε_p and ε_θ be the limits in the accuracies of the localization and temperature measurements. The sensors stop if either one of the inequalities is satisfied:

$$\left| \Delta p_{i,st}^{(k)} \right| < \varepsilon_p \text{ ; } \left| (\nabla \theta_i^{(k)})^{\mathrm{T}} (\Delta p_{i,st}^{(k)}) \right| < \varepsilon_\theta \tag{4.35}$$

We briefly discuss the stability of our network under heat sources jitter. When they jitter, the optimal positions that the sensors are locked in will also shift. For stability, the sensors need to restore back to the positions in the next step. Since our optimization is based on the Newton's Method, the search will converge to the goal in one step if the locality of the distribution is approximately quadratic. It is shown in detail in appendix D by using the Taylor's expansion around the locality that a stable convergence region centered at the optimal position with a radius, ρ>0 will always exist.

4.2.1.3 Dissemination and Extraction (S4, S1)

In section 4.2.1.2, we discuss about the importance of using cooperatively shared information of high content in the search as it leads to stable, efficient and more accurate search. The weakness of independent search is, if the initial position is in the locality where the distribution is essentially flat, it leads to erratic search with slow convergence or even divergence. This is known as the initial value problem. Our cooperative optimization mitigates this problem by obtaining information from neighbors which are spread out further. However, in a situation where the distribution is flat over a large region, this approach may not be sufficient. To mitigate the problem further, we develop a communication mechanism inspired by the foraging behavior of social animals and insects such as ants that could pass on not only neighborhood information, but better information further upstream without using multi-hop routing in order to alleviate network congestion.

Figure 4.10a shows the dissemination mechanism of the state information. It corresponds to S4 of figure 4.4. As with orientation stabilization, the gradient is used as the proxy for the content of the information. A receiving sensor will now have the flexibility of working with either the primary or secondary source of information using the extraction mechanism shown in figure 4.10b. It corresponds to S1 in figure 4.4.

```
1.For each network neighbor $j$ :
  a.Compute the average gradient w.r.t. node $i$: $\lambda_{ij} =(\theta_j^{(k)}- \theta_i^{(k)})\div \| p_j^{(k)}- p_i^{(k)} \|$
  b.Identify a subset $\Gamma$ of all gradients such that: $\Gamma=\{ |\lambda_{ij}| > \lambda_{\text{threshold}}\}$
2.Identify node $l$ such that its gradient belong to $\Gamma$ and at the same time, it is
  nearest to node $i$.
  a.If (found) broadcasts state information of node $i$ and node $l$.
  b.If (not found) broadcasts only state information of node $i$.
```

Figure 4.10a: Dissemination mechanism (S4)

```
1.Determine 3 nearest neighbors
  a.Construct the Local Delaunay Map of the network neighbors.
  b.Perform Delaunay triangulation over the network neighbors.
  c.Locate the triangle that current position is in. The vertices of the triangle are
    the 3 nearest neighbors.
2.For Each Nearest Neighbor
  a.Compute the average gradient $\lambda_{ij,\text{pri}}$ w.r.t. node $i$, based on primary
    information.
  b.If ($\lambda_{ij,\text{pri}} > \lambda_{\text{threshold}}$) includes primary information for optimization.
  c.Else computes $\lambda_{ij,\text{sec}}$ based on secondary information
   i. If ($\lambda_{ij,\text{sec}} > \lambda_{ij,\text{pri}}$) includes secondary information for optimization.
  Next Neighbor
```

Figure 4.10b: Extraction mechanism (S1)

The rule for using the information is: always use the primary source of information unless, the content of primary source of information is below a threshold, $\lambda_{\text{threshold}}$. This rule is put in place because in multiple hotspots scenario, there may be smaller hotspots in the vicinity of the sensors. Even if the secondary source of information indicates that there is a larger hotspot in a faraway region, we would prefer to mobilize our sensor to explore the smaller hotspots in the vicinity first. This rule also prevents all the sensors from converging greedily to the largest hotspot in the region, ignoring other hotspots. The same principle also applies in figure 4.10a for selecting secondary information for dissemination.

Finally, as the dissemination piggybacked on the existing communications over a one-hop neighborhood used to facilitate harvesting tasks, it does not require additional time slots for transmission. For a network with N_s sensors, the total message overhead is O(N_s). Note that the dissemination and communication protocols discussed for facilitating the harvesting task are implemented at the higher layer. In actual implementation, there exist supporting network functions in wireless sensor network such as: MAC [94] and physical layers to handle non-ideal physical communication channels.

4.2.1.4 Effect of stabilization and dissemination on the trajectory

We show some trajectory plots from our simulation study in order to better appreciate how the control algorithm operates and the importance of various components in the control algorithm. The simulation study is conducted using MATLAB.

In figure 4.11a, we show an example of a plot where there is no orientation stabilization. We observe that the sensors appear to be moving in disorientated and disorderly manners. In figure 4.11b, we observe the effect of orientation stabilization on another example, notice now that the sensors are able to move in a more orderly manner. The twists and turns in the movements are within normal expectations as without perfect knowledge of the whole terrain, the trajectory will not be smooth. This is analogous to two methods of driving a car from the source to a destination location. If the car has a pre-installed GPS system with preloaded maps, the system can plan the shortest and smoothest path for us to reach the destination. Otherwise, we can only plan the next step forward in small increments, based on current available information and on some occasions where we hit a dead end, we will need to reverse.

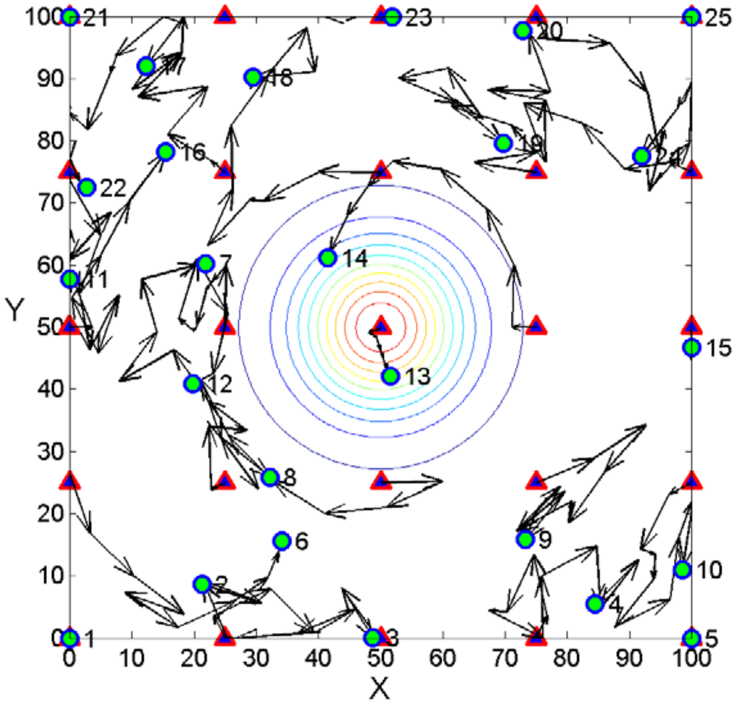


Figure 4.11a: An example of a trajectory plot of the movements of the 25 mobile sensors without orientation stabilization

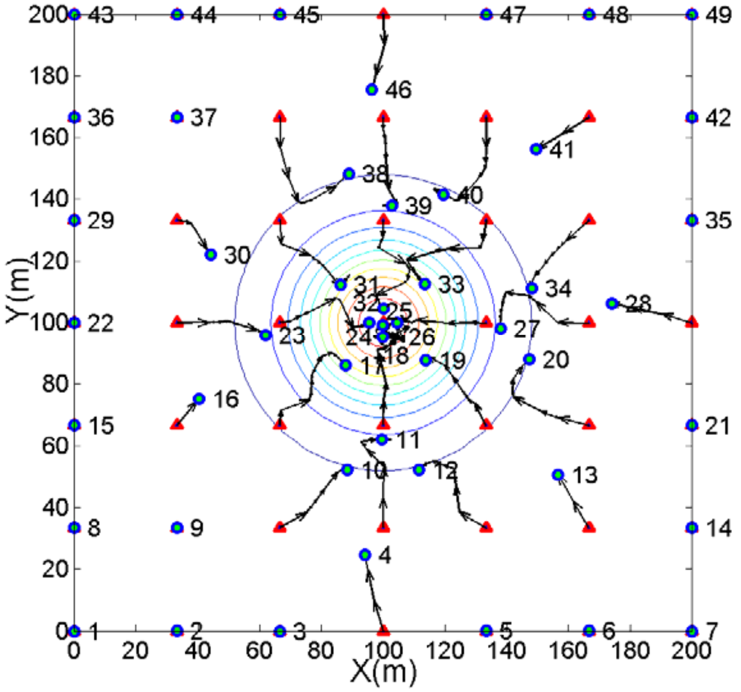


Figure 4.11b: An example of trajectory plot of the movements of the 49 mobile sensors with orientation stabilization

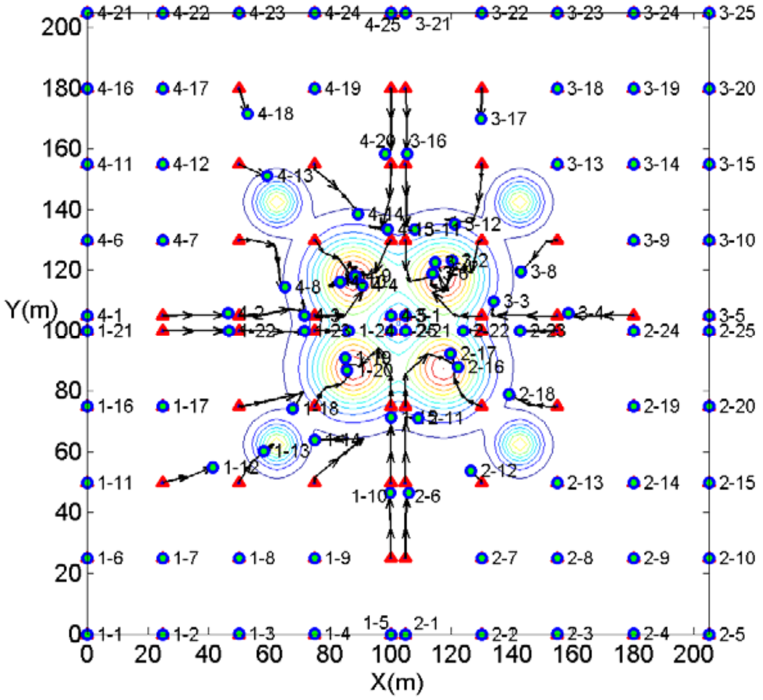


Figure 4.12a: An example of trajectory plot of the movements of 4 groups of 25 mobile sensors without information dissemination for the first 7 iterations

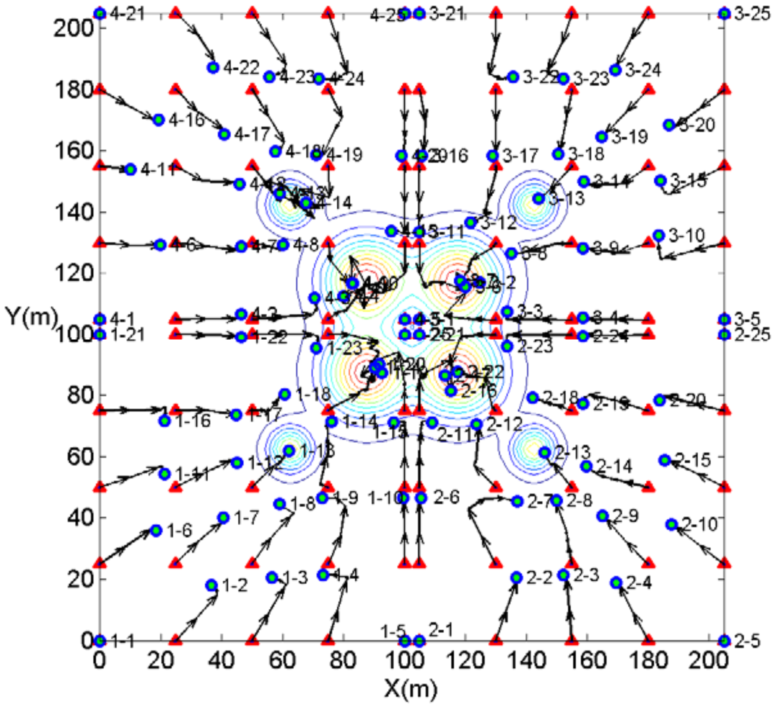


Figure 4.12b: An example of trajectory plot of the movements of 4 groups of 25 mobile sensors with information dissemination for the first 7 iterations

In figure 4.12a, we look at an example where a large proportion of the sensors are far from the hotspots and there is no information dissemination. The sensor group as a whole is not very responsive in the first seven iterations. In figure 4.12b, we implemented the information dissemination. It can now be observed that the sensors are more responsive and accurate.

4.2.1.5 Performance bounds

The performance of our algorithm is limited by the estimation of the gradient and hessian in (4.4). Therefore, a lower bound can be obtained by considering an ideal algorithm where the sensor has perfect knowledge of the distribution, sources and states of its neighbors. Likewise, in the worst case, the sensor has no knowledge of its surrounding. Both the direction and magnitude of $\Delta p_i^{(k)}$ will be random with uniform distribution. As our approach is memory-less, such a sensor will move with RWM within the circular constrained region. By examining RWM, we can derive an approximate upper bound. At the Kth iteration, the sensor would have collected (K+1) data points. As the data points are positioned in a continuous space, there are infinite numbers of combinations. Therefore, instead, we will derive the approximate bound by examining the two extreme scenarios.

Given that the sensor has no knowledge and it is allowed to choose the positions s.t. as many points as possible lands on the information fertile region, it makes the least decision error statistically if the data points are distributed with equal spacing.

To elaborate, the problem of optimal data harvesting of non-adaptive mobile sensors can be formulated into the problem of optimal decision making in statistics as follows:

1. To minimize the mean distortion error, we want to place as many sensors as possible in regions where there are hotspots.
2. No a-priori knowledge of the locations and sizes of the hotspots.

In a statistical situation where we are faced with many choices and we have completely no knowledge of which of the choices are more reliable than others, the best approach is to give equal weights to all the choices in order to minimize the risk of decision error. This is akin to our situation where there are infinitely many positions for us to choose from and we have no way to know reliably the hotspots are located in which positions. Therefore, the optimal way to harvest the data points is to give equal weights to all the positions by distributing the discrete number of harvesting points equally over the terrain.

Finally, the worst scenario occurs if it wagers all the points approximately in one same location.

Let $\theta(x,y)$ be the actual distribution. Let $\Psi(\omega)$ be the distribution interpolated from points with EDG positions of spacing, ω using the cubic-spline interpolation available in MATLAB. Let h be the spacing used in error computations. Let x_{max} and y_{max} be the maximum values of the x and y coordinates of the terrain. The approximate upper bound for relative global error, ξ_{ub} in percentage can be computed by averaging the two abovementioned scenarios:

$$\xi_{ub} = 50\sum_{i=0}^{m}\sum_{j=0}^{n}[|\theta(ih,jh)-\Psi(\omega)|+|\theta(ih,jh)-\Psi(\omega/\Gamma)|] \div \sum_{i=0}^{m}\sum_{j=0}^{n}\theta(ih,jh) \qquad (4.9)$$

where, $m=x_{max}/h$, $n=y_{max}/h$ and $\Gamma = \lfloor(\mathrm{K}+1)^{0.5}\rfloor$.

4.2.2 Independent Optimal Harvesting

Once the sensors are locked respectively in their optimal positions, there is no further data harvesting. To further harvest data to reduce the error, we introduce a coordinator to monitor and usher in the independence phase in which the sensors move around their localities to further harvest data of high information content. In this approach, the role of the coordinator is taken by the same sensor that plays the cluster-head or aggregation node in networking. It also participates in harvesting as a sensor can take on multiple roles. Our approach also allows for integration with the networking functions as the information is piggybacked along with other essential services, resulting in a more efficient use of the communication bandwidth. Our coordination protocol is shown in figure 4.13.

We briefly describe the protocol in figure 4.13. The coordinator maintains an array of First-In-First-Out (FIFO) queues. Each queue stores the positions of the members in the last three iterations. The purpose of these queues is to compute the moving averages of the last three positions. This allows us to smooth out the jittery effects of the movements of the sensors.

For practical purposes, we say that the sensors have stopped when the change in positions of the sensors is smaller than the location error, ε_p. Although, we want the members to transit to the independence phase only after all of them have stopped, this may not be feasible in those special cases where a few sensors may be oscillating about their equilibrium positions. This occurs at a time when to continue the cooperative phase is infeasible because the harvesting only produces minimum error reduction at the expense of continual energy consumption. Thus, it is better for the coordinator to usher in the independence phase without waiting for them to conserve energy. A variable, *nos_of_sensors_threshold* is used to set the maximum number of sensors to wait before the group transits.

In the independence phase, there is no communications. This is because in this stage of data harvesting where the sensors are near their optimal positions, the use of cooperative shared information would have no further large gains except to consume communication resources (bandwidth and energy). Thus, it is more frugal to release these resources for other purposes. We develop two possible heuristics for the independence phase: delta and orbital-harvesting.

```
Procedure Coordinator ()
Initialize: nos_of_sensors_threshold, moving_average_q = FIFOqueue
(qsize=3), members_moving_average = array (moving_average_q)(arraysize =
numberofmember),Phase="Cooperate"
1.sensor_count = 0;
2.For Each Member
  a.moving_average_q =  members_moving_average (current_member);
  b.Compute average_change_position = current_position - average (elements
    in moving_average_q);
  c.Remove first element in moving_average_q;
  d.Insert current_ position into moving_average_q;
  e.if (average_changePosition <
    locationerror){sensor_count=sensor_count+1};
  Next Member
3.If (sensor_count > nos_of_sensors_threshold & Phase = "Cooperate")
  a.Set Phase to "Independent";
  b.Inform members to transit;
4.Else {Set Phase to "Cooperate"};
End Procedure Coordinator
Procedure Member ()
Initialize: Phase = "Cooperate"
1.If (Phase of Coordinator's Message = "Independent")
  a.Set Phase to "Independent";
  b.Set InitialIndependentStateInformation =
    FinalCooperativeStateInformation;
  c.Execute Independent Heuristic;
2.Elseif (No Coordinator's Message and Phase= "Cooperative") {Execute
  Cooperative Algorithm};
3.Elseif (No Coordinator's Message and Phase= "Independent"){Execute
  Independent Heuristic};
End Procedure Member
```

Figure 4.13: Pseudo-code for the coordination protocol

In delta-harvesting (Figure 4.14a), the sensors spiral outwards from their optimal positions. There are two reasons behind this approach. First, in reality, due to estimation and measurement errors, it is unlikely that we are exactly at the optimal positions. The heuristic increases the probability that the actual optimal positions are harvested successfully in the localities. Second, additional harvesting in region where temperature variation is greater reduced the distortion error of the reconstructed distribution.

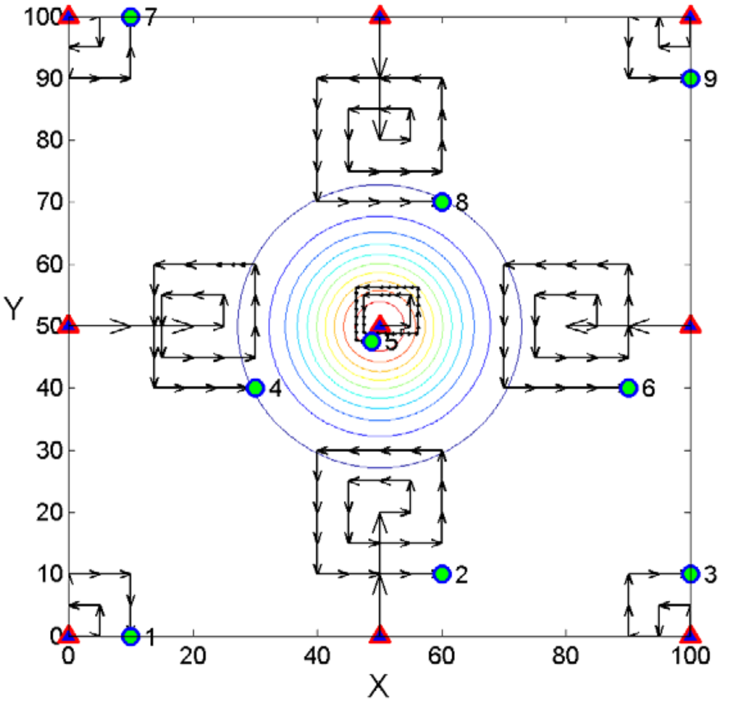


Figure 4.14a: The trajectory for the delta-harvesting heuristic

In figure 4.14a, notice that sensor 2, 3, 4 and 8 have been propelled towards the information fertile region during the cooperative phase. At each step, the sensor computes the gradient either in the *x* or *y* direction and compared with *GradientThreshold*. If (current gradient > *GradientThreshold*), the step size is halved up to a maximum of three times and vice versa. The method is similar to adaptive delta modulation in [89]. Notice that sensor 5 moves with a smaller step size as it is inside the information fertile region where the temperature variation is the greatest.

Due to the high energy consumption of delta-harvesting, we implement two measures to curb their movements in order to conserve energy. First, the sensors stop moving when there are negligible detected temperature variations after moving for a while. This behavior is observed in sensor 1, 3, 7 and 9. Second, a radius constraint is imposed s.t. the sensors stop moving when the maximum radius is reached. This behavior is observed in sensor 2, 4, 6 and 8.

The pseudo-code for the delta-harvesting heuristic is shown in figure 4.14b-d. Due to the tree-like decision structure used to control the orderly spiraling movements, we develop the main portion of the algorithm in the compact recursive form. Altogether, it consists of three functions: main function which examines the initial condition and calls the recursive function accordingly, the recursive function which determines the next direction and the adaptive step size function which determines the next step size. The algorithm is self-explanatory in figure 4.14b-d.

```
Function Main () {
1.Initialize: StepSize, TerrainBoundaries, CurrentRingBounds = {LeftBound,
  RightBound, BottomBound, TopBound}
2.For Each Iteration, k {
  a.Record Current Position in Cartesian Coordinate Form and Temperature of
    sensor i: $[x_i^k, y_i^k, \theta_i^k]$
  b.Compute previous change in position and temperature: $[\Delta x_i^{k-1}, \Delta y_i^{k-1}, \Delta\theta_i^{k-1}]$
    $= [x_i^k, y_i^k, \theta_i^k] - [x_i^{k-1}, y_i^{k-1}, \theta_i^{k-1}]$
  c.If (k = 1) {
     i. Test the next move in the following anti-clockwise order: {Right, Top, Left,
        Bottom}, to check whether it is blocked by the TerrainBoundaries. Repeat
        this test until a successful direction is found.
    ii. Increase CurrentPosition and CurrentRingBounds by one StepSize in the
        direction of the successful move }
  d.Else {
     i. Initialise: nTry = 3, PreviousTryDirection = None, CurrentMoveDirection.
    ii. Set PreviousDirection {Right, Top, Left, Bottom} based on $[\Delta x_i^{k-1}, \Delta y_i^{k-1}]$.
   iii. Compute new adaptive step size: $[\Delta x_i^k, \Delta y_i^k]$ = AdaptiveStepSize ($\Delta x_i^{k-1}$,
        $\Delta y_i^{k-1}$, $\Delta\theta_i^{k-1}$)
    iv. Call Function ExpandingRing ($\Delta x_i^k$, $\Delta y_i^k$, TerrainBoundaries,
        CurrentRingBounds, CurrentPosition, PreviousDirection,
        PreviousTryDirection, nTry) }
  } (Next Iteration, k)
} (End Function Main)
```

Figure 4.14b: Pseudo-code for the main function of the delta-harvesting heuristic

```
Function ExpandingRing ($\Delta x$, $\Delta y$, TerrainBoundaries, CurrentRingBounds,
CurrentPosition, CurrentTryDirection, PreviousTryDirection, nTry) {
1. Choose the corresponding element from CurrentRingBounds to test depending
   on the PreviousDirection. (For eg., if PreviousDirection = Right, RightBound
   is chosen from CurrentRingBounds)
2. TestPosition =  CurrentPosition + PreviousDirection × [$\Delta x$, $\Delta y$]
3. If (TestPosition ≤ CurrentRingBounds) (Move Forward) {
  d.Return NewPosition = TestPosition }
4. If (TestPosition > CurrentRingBounds) (We are blocked!) {
  a.If (nTry > 0) (We first try to spiral without expanding the ring boundaries by
    trying the other orthogonal directions) {
     i. Choose nextCurrentTryDirection from the following list in the anti-
        clockwise order: {Right, Top, Left, Bottom}depending on the value of
        CurrentTryDirection. (For eg., if CurrentTryDirection = Right,  then
        nextCurrentTryDirection = Top).
    ii. Recursively Call Function ExpandingRing ($\Delta x$, $\Delta y$, TerrainBoundaries,
        CurrentRingBounds, CurrentPosition, newCurrentTryDirection,
        CurrentTryDirection, (nTry-1)) }
  b.Else (We have exhausted our try and is still block in the orthogonal directions,
    therefore we expand the ring boundaries) {
     i. Increase one of the elements in CurrentRingBounds either by $\Delta x$ or $\Delta y$
        depending on PreviousDirection. (For eg., if PreviousDirection = Right,
        RightBound is chosen from CurrentRingBounds and new RightBound =
        RightBound + $\Delta x$). Denote this current element as testRingBound.
    ii. Also choose the elements in TerrainBoundaries depending on
        PreviousDirection. Denote this current element as testTerrainBound.
   iii. Set NewPosition = TestPosition
    iv. If (testRingBound > testTerrainBound) {
        • Set testRingBound = testTerrainBound
        • Adjust NewPosition upto the terrain boundary limits }
     v. Return NewPosition }
  } (End If Step 4)
} (End Function ExpandingRing)
```

Figure 4.14c: Pseudo-code for the recursive function of the delta-harvesting heuristic

```
Function AdaptiveStepSize ($\Delta x$, $\Delta y$, $\Delta\theta$) {
1.Initialize: $maxStepSize$, $factor$, $nReduce$, $gradientthreshold$
2.$minStepSize = maxStepSize \div (factor)^{nReduce}$
3.If ($\Delta x \neq 0$) {
  a.Estimate $d\theta/dx \approx \Delta\theta/\Delta x$
  b.If ($d\theta/dx > gradientthreshold$) {
    i. $\Delta x = \textbf{Max}((\Delta x \div b), minStepSize)$ }
  c.If ($d\theta/dx < gradientthreshold$) {
    i. $\Delta x = \textbf{Min}((\Delta x \times b), maxStepSize)$ }
  }
4.If ($\Delta y \neq 0$) {
  a.Estimate $d\theta/dy \approx \Delta\theta/\Delta y$
  b.If ($d\theta/dy > gradientthreshold$) {
    i. $\Delta y = \textbf{Max}((\Delta y \div b), minStepSize)$ }
  c.If ($d\theta/dx < gradientthreshold$) {
    i. $\Delta y = \textbf{Min}((\Delta y \times b), maxStepSize)$ }
  }
5.Return [$\Delta x$, $\Delta y$]
} (End Function AdaptiveStepSize)
```

Figure 4.14d: Pseudo-code for the adaptive step size function of the delta-harvesting heuristic

In orbital-harvesting shown in figure 4.15a, the sensors orbit around heat sources by moving along the isotherms or contours. Again, we observe that sensor 2, 4, 6 and 8 have been propelled towards the information fertile region during the cooperative phase. Note that Matlab has problems displaying the quivers. However, we can still observe the orbital band formed by the trajectories. The orbital band is shown as many small dots representing the positions of the sensors over time. It resembles the asteroid belt of the solar system.

In the orbital-harvesting heuristic shown in figure 4.15b, the sensors compute the source direction and locked into the current temperature as the reference temperature. The source directions are estimated by the difference between the final and initial cooperative positions. The initial harvesting direction of the movement is set perpendicular to the source direction. Ideally, we want the sensors to move perpendicular to the source direction all the time if the orbit is circular in shape. Since, in practical situations, the contours may come in any shape, we implement a self-correcting control loop to make our sensors adaptive. Essentially, after every small step, the sensor measures the current temperature. If the current temperature is greater than the reference temperature, it means that currently, it has deviated from the contour and is moving towards the source and vice versa. It will then adjust its current direction by an additional angle ϕ so as to maintain an approximate orbit.

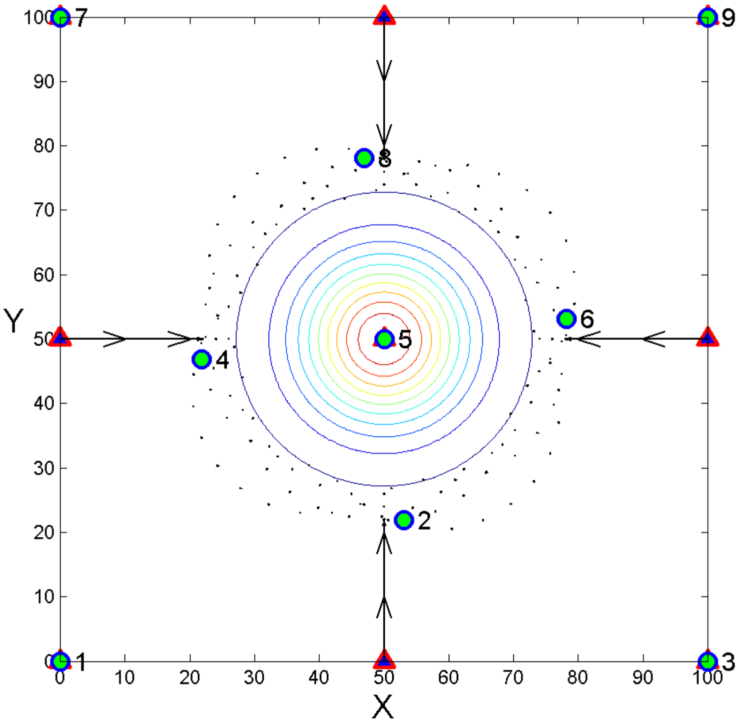


Figure 4.15a: The trajectory for the orbital-harvesting heuristic

These constant adjustments result in the characteristic zigzag appearance. As a result, their trajectories create an uncertainty orbital band. The reason for the creation of the band is the same as the first reason for the delta-harvesting heuristic. That is, in reality, due to estimation and measurement errors, it is unlikely that we are exactly at the optimal positions. Therefore, the orbital band increases the probability that the actual optimal positions are harvested. However, the main motivation for this heuristic is to smooth out the distortion in the *xy*-plane which causes the reconstructed distribution to look pyramid-like.

```
Heuristic Orbital harvesting
Initialize: Orbiting direction = Rotate 90 degree from source direction,
Stepsize, $\phi$
1. Compute: source direction = Final cooperative position – Initial cooperative
   position
2. Set Reference Temperature = Final cooperative measurement  // We will
   lock into this temperature when moving
For each iteration
3. If (current Temperature < Reference Temperature) // We are deviating too
   far away from the source direction
   a. Rotate current orbiting direction by $\phi$ towards source direction
   b. Move in current orbiting direction by stepsize
4. Else // We are deviating too close to the source direction
   a. Rotate current orbiting direction by $\phi$ away from the source direction
   b. Move in current orbiting direction by stepsize
End
```

Figure 4.15b: Pseudo-code for the orbital-harvesting heuristic

4.2.3 Tracking Mechanism

We have incorporated a tracking mechanism in our algorithm. Consequently, our sensor has two states: **Optimize** and **Track**. There are three motivations behind the mechanism. First, although we have shown that the sensors at optimal positions are stable, the sizes of the surrounding convergence regions are determined solely by environmental parameters such as the intensity of the sources. It is desirable from the perspective of control that stability and speed of tracking is determined also by system parameters. Second, in reality, the hotpot may move anytime during optimization. Consequently, there is confusion among the sensors as some over-respond while others under-respond to the movement. In scenarios where many hotspots move in different directions and at different speeds, further confusion arises when the sensors have to decide on which hotspots to follow. Third, the movement of the hotspot may result in some variation of intensity of the source, for example, if the source is a fire.

We make two assumptions. First, the generation of hotspots over time is rare or accompanied by extinguishing of hotspots. This is because constant generation will eventually deplete the sensors allocated to track them. Second, the hotspots move within the terrain in random manner and stop so that optimization can resume. This is a valid assumption because in reality, when we are monitoring forest fires or underwater toxic leakage from sunken containers, it is unlikely that they have large constant movements. They are more likely to be jittery in the random manner around the locality. The dynamic clustering, tracking and state transitions of the algorithm are discussed in the subsections. Finally, although the concept of dynamic clustering is used by static sensor networks to track discrete targets [9], our novelty is in its use in mobile sensor networks to track distributions that are continuous in space.

4.2.3.1 Dynamic Clustering

The sensors start at the **Optimize** state and the same dissemination mechanism used in figure 4.10a-b is leveraged on to perform dynamic clustering simultaneously with the optimization. The main idea is to have the sensors clustered around the individual hotspots based on their proximity to the hotspots. The cluster formation begins with the election of the cluster-head based on the lowest sensor id. The criteria for the choice of the cluster-head is that it must be a local maximum w.r.t. its neighbors. The local maximum is close to the actual peak of the hotspot and the most prominent landmark that is robust to intensity variation of the source. The cluster-head id is used as the cluster id for the members to identify with.

Sender ID	State	Time	Nos Info	Cluster ID	LM Flag	Primary				– –	Tertiary			
						ID	X	Y	θ		ID	X	Y	θ

Figure 4.16a: Format of communication packet

Figure 4.16a shows the format of the communication packets. **State** indicates the state of the sender. **NosInfo** indicates the number of state information transmitted in the variable size packet. **ClusterID** indicates the current cluster that the sender belongs to. **LMFlag** indicates whether the state information of a local maximum is transmitted. If **LMFlag**=1, the state information of the local maximum is always appended as the last information in the transmitted packet.

```
If ($\theta_i$ > AmbientThreshold)
1.Perform Clusters Collection: ClustersArray.
  First Pass: Elect Tentative New Cluster-Heads
  a.Construct LDM from the primary and secondary state information obtained
    from the received packets.
  b.For Each Sensor  in LDM
     i.  If (Local Maximum w.r.t. adjacent neigbors) store the sensor id and its
         state information into ClustersArray
     Next Sensor
  Second Pass: Collect Existing Cluster-Heads
  a.If (it is a member of existing cluster) insert cluster id and state information of
    local maximum that it is tracking into ClustersArray.
  b.For Each Packet with(LMFlag=1)
     i.  Store the sensor id and state information of the last appended information
         in the packet into ClustersArray.
    Next Packet
  Third Pass: Merge Same Clusters
  a.Scan ClustersArray for clusters tracking the same local maximums – those
    with same state information. Only the smallest cluster id is retained in
    ClustersArray, the others are removed.
2.Choose the nearest cluster in ClustersArray to join.
```

Figure 4.16b: Dynamic clustering algorithm

The clustering algorithm is shown in figure 4.16b. Sensor *i* first checks whether its current temperature, θ_i is above the ambient temperature threshold, *AmbientThreshold*. If ($\theta_i \leq$ *AmbientThreshold*), it is far from the hotspots and will remain in sentry position for sensing coverage. This is to cater for the contingency of spontaneous generation of new hotspots in the locality. Otherwise, it collects potential new clusters and existing clusters from the information in the received packets into *ClusterArray*. It then joins the nearest tracking cluster. Merging of clusters is required because due to the delayed dissemination of the packets that are more than one hop away, it is possible for new clusters to form around a local maximum that is tracked by existing clusters during transience.

4.2.3.2 Transition from Optimize to Track state

Once a cluster is formed around a local maximum, the tracking operation is essentially a re-centering operation except for the case of crossover of multiple hotspots which will be discussed later. There are two conditions for transition. A normal condition occurs when the cluster-head discovers that it is no longer a local maximum, that is, the peak has shifted. It sets its state to **Track** and informs its member by sending the packet as follows: **State**=**Track**, **NosInfo**=1, **ClusterID**=SenderID, **LMFlag**=0. Upon receiving this packet, each member sets its state to **Track,** pauses optimization and detects whether it is a local maximum. If it is a local maximum and previously it is not, it sends the packet as follows: **State**=**Track**, **NosInfo**=1, **ClusterID**= CurrentClusterID, **LMFlag**=1 appending its state information. Otherwise, it disseminates the received packet. Note that it only disseminates on condition that its current cluster ID is the same as **ClusterID** in the packet and its original state is **Optimize**. Therefore, total message overhead is O(N_c) where N_c is the number of members in a cluster. The special condition occurs if there is a blind spot problem.

A blind spot problem occurs when a small hotspot hides in between the sensors. It may occur in the initial stage of optimization and lead to sluggish sensor movements because of small temperature variations. This is detected by examining the temperature of the local maximum. If the temperature is below or equal to

AmbientThreshold defined in section 4.2.3.1, the special condition is triggered and the problem is handled by the algorithm in section 4.2.3.4.

4.2.3.3 Behavior at Track state

In **Track** state, each member concurrently detects for the local maximum and monitors for received packets. When a member detects that it is a local maximum and previously it is not, it sends a packet as follows: **State**=**Track**, **NosInfo**=1, **ClusterID** = CurrentClusterID, **LMFlag**=1 with its state information. Upon receiving a packet, other members perform tracking shown in figure 4.17.

```
1.If ($\theta_i \leq AmbientThreshold$) De-join the cluster, return to Sentry Position and
  Set State = Optimize.
2.Else
  a.Perform Clusters Collection with timestamp filtering: ClustersArray.
  b.Compute change of position, $\Delta p_i^{(k)}$ w.r.t. the position of cluster-head, $p_{track}^{(k)}$
    based on the position of the local maximum, $p_{lm}^{(k)}$ for each element in
    ClustersArray: $\Delta p_i^{(k)} = p_{lm}^{(k)} - p_{track}^{(k)}$.
  c.Select the element with the smallest , $\Delta p_i^{(k)}$.
  d.If ($\Delta p_i^{(k)}$ violates maximum cooperative constraint) De-join the cluster, return
    to Sentry Position and Set State = Optimize.
  e.Otherwise
    i.   Perform the Move.
    ii.  If (New Position is Local Minimum) Transmit Packet as: State= Track,
         NosInfo=1, ClusterID=CurrentClusterID, LMFlag=1.
    iii. Else retransmit the packet containing the local maximum.
```

Figure 4.17: Tracking algorithm

The *Clusters Collection* procedure in step 2a is similar to step 1 of figure 4.16b except that now only the most updated packet based on timestamp is used to track the current position of the local maximum. Step 2b causes the members to perform a mathematical translation parallel to each other, as a result, the relative positions of the members remain unchanged and optimization can resume after tracking ends.

Three exceptional conditions are handled in figure 4.17. In the context of our application where we foresee no large movements in the sources, the exceptional conditions are rare events which are incorporated into our algorithm for contingency purposes. The first two exceptional conditions (step 1 and 2d) cause sensor *i* to de-join the cluster, return to its sentry position which is the initial position and transit to **Optimize** state. They are for the same purpose of maintaining minimal sensing coverage for spontaneous events discussed in section 4.2.1.2. Additionally, step 2d prevents the sensors from performing the energy inefficient wild goose chase, the sensors return to their initial sentry positions once they move beyond the maximum radius mentioned in section 4.2.1.2 on cooperative constraints.

In figure 4.18, we illustrate the unstable condition that leads to loss of sensing coverage if we do not impose a maximum cooperative constraint radius mentioned in section 4.2.1.2 and allow the sensors to perform wild goose chases. Initially, the sensors are well spread out (Figure 4.18a). Later, they cluster around the hotspot at the top region which starts to move downwards (Figure 4.18b). Without imposing the maximum radius, it will lead to a collapsing cluster with loss of coverage as shown in figure 4.18c as the sensors chase after the hotspot. In figure 4.18d, with the cooperative constraint in place, the sensors at the top region do not follow the hotspot once it moves out of range. Instead, they return to their original sentry positions.

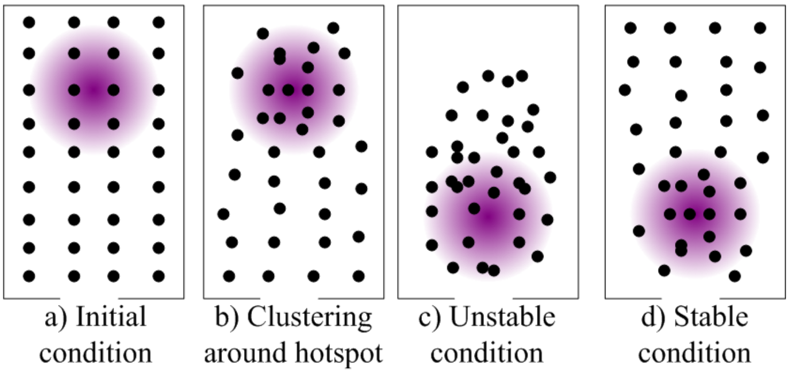
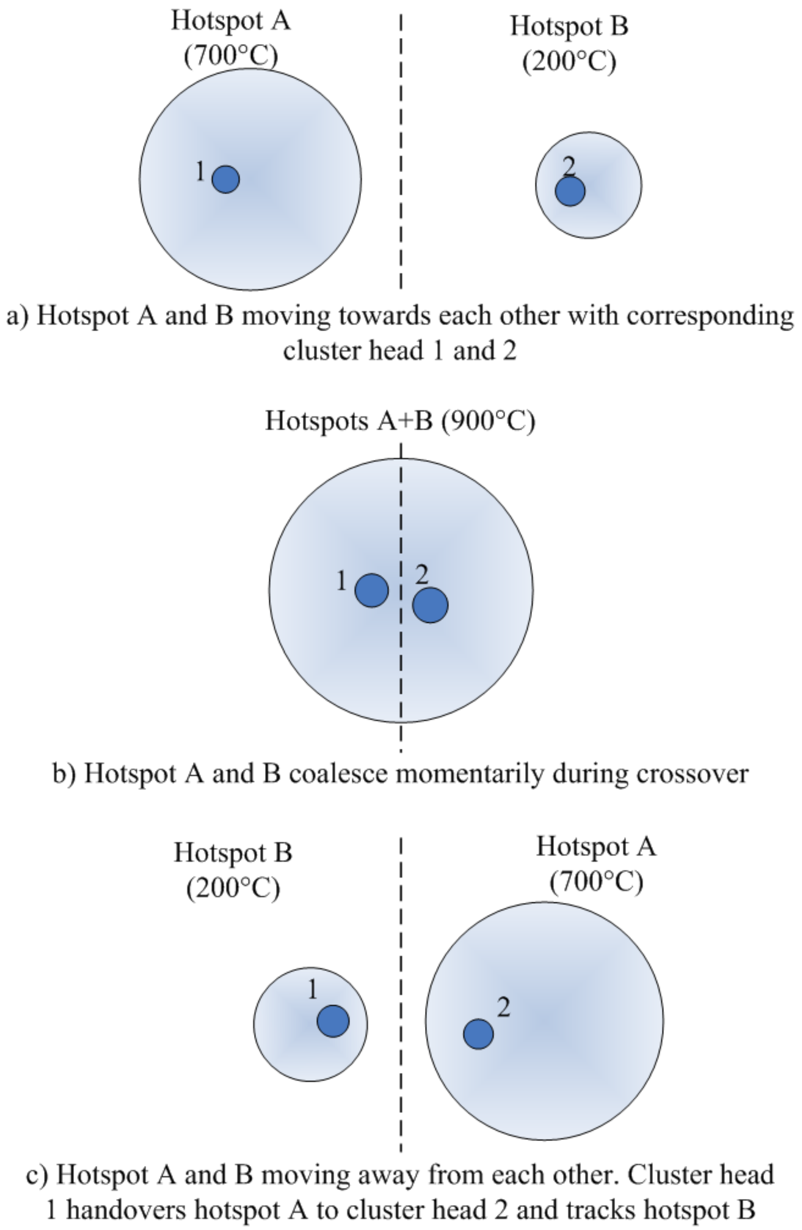


Figure 4.18: Stability condition during tracking



a) Hotspot A and B moving towards each other with corresponding cluster head 1 and 2

b) Hotspot A and B coalesce momentarily during crossover

c) Hotspot A and B moving away from each other. Cluster head 1 handovers hotspot A to cluster head 2 and tracks hotspot B

Figure 4.19: Crossover condition of hotspots and handover effect of tracking algorithm

Step 2c handles the third exceptional condition known as the crossover problem. In the crossover problem, two or more hotspots move towards each other and momentarily coalesce as illustrated in figure 4.19a-b. Afterwards, they separate and continue moving as illustrated in figure 4.19c. Consequently, the trackers are confused on which hotspots they initially track. In our algorithm, the cluster chooses the hotspot that is currently nearest to the position of the cluster-head regardless of which hotspot the cluster initially tracks, that is, the smallest $\Delta p_i^{(k)}$. This creates a handover effect (Figure 4.19c) where the cluster handovers the hotspot that is moving away from the sentry positions of the members towards the maximum radii and receives the hotpot that is moving towards their sentry positions. One benefit is that it reduces the likelihood of cooperative constraint violations.

These exceptional conditions may prolong the optimization depending on the stage of optimization in exchange for the benefit of sensing coverage and stability when the cluster resumes optimization. This is because some sensors that are originally converging towards their optimal positions may be reset to their sentry positions. Therefore, the optimization now begins at an earlier stage. Finally, sensor *i* only disseminates the received packet once when it starts to move. If it has moved and later, it receives the same packet again, it does not disseminate. Therefore, total message overhead is again, $O(N_c)$.

4.2.3.4 Transition from Track to Optimize state

A cluster-head initiates a normal transition to **Optimize** state when it is currently a local maximum and duration, *TimeLimit* has elapsed without further change. At this stage, the blind spot problem where the source is hiding between the sparsely distributed sensors adjacent to the cluster-head may occur (see Figure 4.20).

In reality, it is unlikely that the hotspot is undetectable, as the heat of the source permeates through space. Thus, there is always small temperature variation that is sufficient to be detected by a modern electronic thermometer with temperature sensitivity of around 1°C. However, small temperature variations lead to slow responses during optimization. This is improved by first locating the source of the hotspot. The resultant increase in magnitude of the gradient will hasten the response.

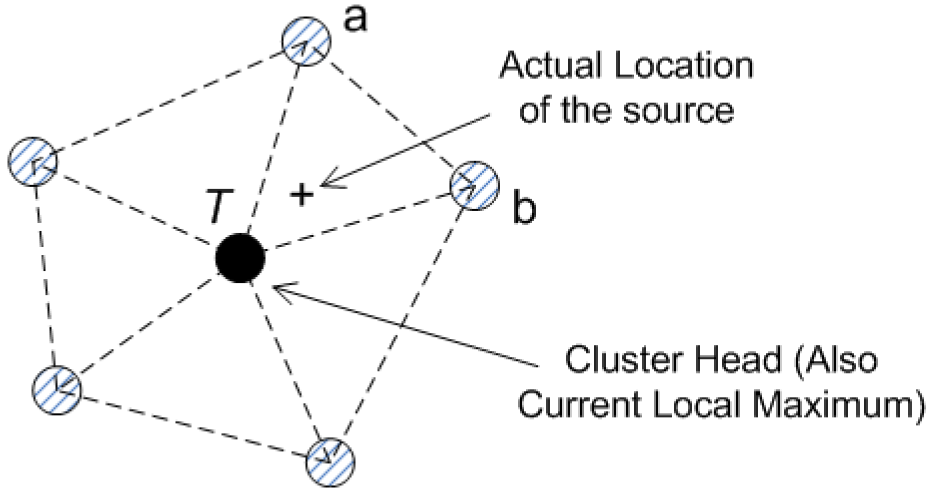


Figure 4.20: Blind spot problem

In figure 4.20, we illustrate the blind spot problem. If we are to construct a Local Delaunay Map (LDM) over the cluster, the location of the source can be narrowed to triangle *Tab* which has the cluster-head as one of the vertices. This is because we can test it by summing up the temperatures at the vertices of the triangle for all the triangles with the cluster-head as one of the vertices. The source will be located in the triangle with the greatest sum. We can narrow down the position of the source by taking the weighted centroid of the triangle using the temperatures as the weights. This is because the source should be nearer to the vertices with higher temperatures. This method for locating the initial search position is used in our algorithm for the cluster-head to search for the peak in figure 4.21.

Once an initial value for the search is computed, the cluster-head can move to this position. Using the stationary sensors: *a* and *b* as two neighbors and its previous position at *T* as the third virtual neighbor, it performs a rapid modified cooperative optimization inside triangle *Tab*. The cluster-head is silent until the search is completed since it is energy-inefficient for its members to move together. Once the center of the source is located, the cluster-head transmits one final packet to inform others the location of the center and set its state to **Optimize**. Upon receiving this packet, the members make one final translational moves, set their states to **Optimize** and resume optimization. As in section 4.2.3.2, they only disseminate the received packets if their original states are still **Track** and not yet transit to **Optimize**.

1. Construct the LDM from the positions of Network Neighbors **and own position**, $p_i^{(k)} = (x_i^{(k)}, y_i^{(k)})$.
2. Search the adjacent triangles (Those with one of the vertices as $p_i^{(k)}$) to locate the triangle which the sum of the temperatures at its vertices is the largest, Δ_{max}. The hotspot is hiding inside this triangle.
3. Compute initial search position as the weighted centroid of Δ_{max} with vertices j = 1, 2, 3 as: $p_i^{(k+1)} = (\Sigma\theta_j^{(k)} p_j^{(k)}) \div \Sigma\theta_j^{(k)}$
4. Perform cooperative search using (4.3)-(4.7) by setting the vertices j of Δ_{max} as the fixed surrounding neighbors and a fixed u_{goal} = [0 0] (a local maximum is a stationary point with zero gradient).
5. Set its state=**Optimize** and transmit packet: **State**=**Optimize**, **NosInfo**=1, **ClusterID**=SensorID, **LMFlag**=1 appending its current state information.

Figure 4.21: Cluster-head peak search algorithm

4.2.3.5 Stability of Tracking Mechanism

We now briefly discuss the stability of the tracking mechanism. The detailed derivation and analysis are given in appendix E. Let the maximum speeds of the hotspot and the sensor be: V_h, V_s, respectively. Let the total delay in the response of the sensor be T_0. It is shown in appendix E that the maximum separation, $\sigma_{\max}$ between the centers of the tracking cluster and the hotspot is bounded by:

$$\sigma_{\max} = T_0 V_h \div (1 - V_h V_s^{-1}) \qquad (4.10a)$$

The total delay is given as:

$$T_0 = T_\theta + N_{hops} T_{comm} \qquad (4.10b)$$

Where, N_{hops} is the maximum number of communication hops in the networks, T_θ is the measurement delay and T_{comm} is the communication delay. T_θ is determined from the specification of the thermometer. A fast electronic thermometer has a delay that is less than 1 s. T_{comm} is given as: $T_{comm} = P \div S_{data}$ where P is the packet length in bits and S_{data} is the the data throughput per node. S_{data} [see List of Publications: P2, P3] [94] is affected by channel conditions such as: noise, fading, shadowing, the MAC protocol used, data traffic load, maximum communication rate, etc.

Therefore, the tracking mechanism is stable as long as $V_s > V_h$. Furthermore, as V_h increases and approaches V_s, $\sigma_{\max}$ also increases.

4.2.4 Our Research Contributions

In this subsection, we summarize our five research contributions highlighted in section 4.2 during the detailed description of our algorithm.

In adapting the independent pseudo-Newtonian method to our problem, we encounter several problems which lead to our contributions.

First, optimal placements require the sensors to spread out and position themselves in areas of high curvature where the gradients have different values. Independent Newtonian methods search for a fixed goal–positions of zero gradients. Even if we assume that we can know the values of the gradients to search for in advance and modify the independent methods to handle fixed non-zero gradients, the sensors using the independent methods still cannot spread out properly as they tend to overlap each other in their search and end up chasing after same few goals. Therefore, we introduce a novel improvement on the method where the search for positions of high curvature is adaptive and cooperative. It is cooperative because the current position of the sensor is also influenced by the current state information of the neighbors. Consequently, the sensors are better spread out while optimizing and there are no chasings after the same goals among the sensors.

Second, independent pseudo-Newton methods perform badly in harsh environments because of estimation errors incurred due to localization noise. This is exacerbated by the accumulation of past errors in the computations which causes the sensors to persist in the erroneous directions even though current estimates are accurate until the influence of past information has faded in the computations. Therefore, we introduce the memory-less local RBF interpolation [21][22] to estimate the gradient and hessian values. This is to eliminate the adverse memory effect in harsh environments.

Third, the initial value problem in independent optimizations in which the rate and probability of convergence are dependent on the initial position is more severe for our application. This is because we cannot make a good starting guess for the initial positions of the sensors as we have no advance knowledge of the actual distribution. Therefore, we develop a dissemination mechanism to mitigate the initial value problem.

Fourth, the fixed line search used by some independent methods such as BFGS to stabilize the search is inefficient as it introduces rigidity in the search. In a line search approach, after a direction is determined, the search is conducted along the straight line until a local minimum or maximum point is located. Only then will there be a change of direction. Therefore, we develop the concept of orientation stabilization in which the stabilized direction is adaptive to current states of the neighbors and may vary from one iterative step to another.

Finally, our fifth contribution is from the development of a robust tracking mechanism for our algorithm, we contribute by applying the principle of dynamic clustering onto mobile sensor networks for tracking the continuous distribution. Dynamic clustering is previously used in static sensor network to track discrete targets [9].

4.3 Theoretical perspective on our design

In analyzing and improving the design of our cooperative algorithm, we look at our problem from two different theoretical perspectives.

In the microscopic perspective, we view our problem as individual sensors attempting to maximize their metrics when they have only partial knowledge in a cooperative setting. In game theory and related theories in philosophy and sociology such as spontaneous order and social contract theories, one approach to achieve order or global stability is by ensuring that individual agents adhere to certain common rules. Our use of orientation stabilization to restraint the movements of individual sensors in order to achieve orderly behaviors at the global systemic level can be seen as the application of such a bottom-up approach. This approach has the advantage that it can be implemented in a distributed algorithm. To digress, it is interesting to note that many social animals and insects also appear to use this approach in achieving order. Spontaneous order theory also seems to provide an interesting explanation for swarm intelligence. It is through these views that we see the important role of information, both in providing global stability and improving the optimality. Both the use of orientation stabilization which steers the sensor more towards the neighbors that have higher information content and information dissemination can be seen as our strategies to improve the knowledge of the individual sensors on their surroundings.

In the macroscopic perspective, we look at the collective behavior of the whole group in response to external stimuli. Our cooperative sensor group is analogous to a mechanical system where many individual metal balls are coupled together by metal springs. In our context, the couplings refer to the communication channels that the sensors have with each other. While these couplings are where the cooperative shared information is transmitted, it is also through these couplings that local instabilities can propagate and negatively affect the overall performance of our sensor harvesting group. This form of instability occurs when the energy consumption of the sensor is translated to useless oscillation or agitation. From control theory, we are made aware of the presence of a finite Lyapunov's stability limit in all constrained dynamic systems. At a systemic level, we can view these as the natural behavior of the system as we gradually drive or load the system toward its stability limit. It is usually observed as a deterioration of performance in the system when we increasingly load the system.

To reconcile the later perspective with the former one, our simulation study suggests that the stability limit is affected by the availability of quality information to the sensors as a whole. By quality, we refer to all three aspects of information as discussed in section 4.1.3. This stems from the observation that after the implementation of the information dissemination block in the main algorithm, there is a decrease in the oscillations of the sensors and simultaneous increase in responsiveness and more accurate forward movements among the sensors. In other words, the performance has improved because at the systemic level, the energy consumption is now converted to actual work done in the system.

4.4 Conclusion

In this chapter, we discuss our novel two-phase cooperative optimal harvesting algorithm that attacks the problem from the source, by mobilizing the sensors to harvest data with high information content through optimal placements of sensors. In the overview, we explain generally the three possible structures: coordination (or centralized control), independence (or distributed control) and cooperative (or distributed control with cooperative shared information), to organize our mobile sensors and how it affects the design of our algorithm. The motion control and information processing aspects of the algorithm are also discussed.

Next, we elaborate in detail the two phases of our optimization algorithm: cooperative and independence phases and the tracking mechanism. In the cooperative phase, our novel approach is to introduce cooperation into the pseudo-Newton method and adapt the method to propel the sensors rapidly into the optimal positions. In the independence phase, we develop two alternative versions of independent heuristics: delta and orbital-harvesting. We also develop a novel tracking mechanism that uses dynamic clustering of mobile sensors to track continuous distributions. The stability of our optimization and tracking mechanisms is then theoretically analyzed. Furthermore, we examine and discuss briefly the design of our algorithm from the theoretical perspectives.

Finally, our five research contributions, we have adopted: adaptive cooperative search goals in optimization, local RBF interpolation in estimations, dissemination to mitigate the initial value problem, the concept of orientation stabilization to provide adaptive stabilized search direction and the principle of dynamic clustering onto mobile sensor networks for robust tracking of continuous distribution.

Chapter 5: Performance Studies

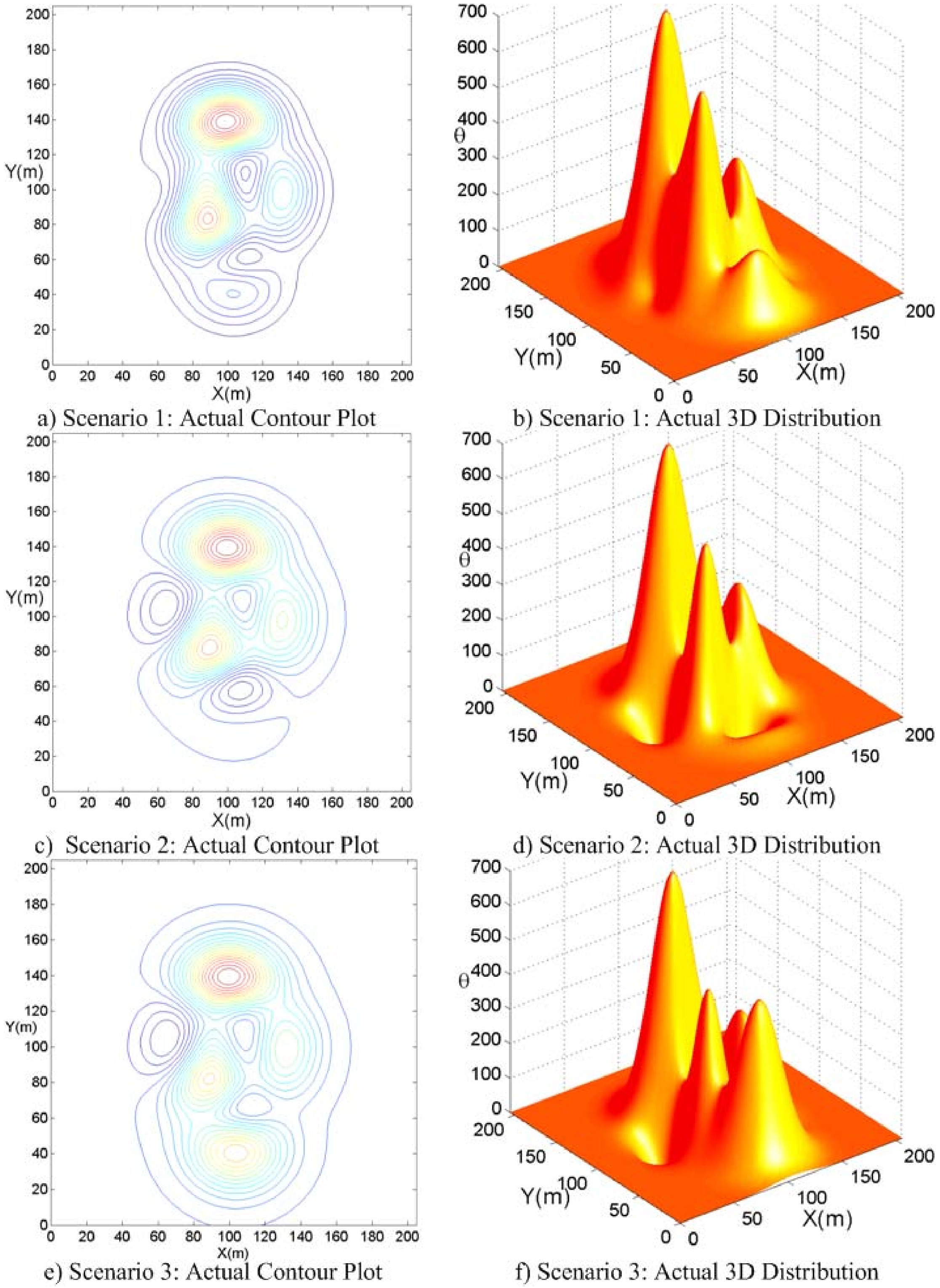
In this chapter, we examine the performance of our cooperative algorithm using simulations. The comparative study with mobile sensor networks using other algorithms will be based on two metrics: relative global error and total energy consumption per sensor. The comparative study with static sensor network is based on resource utilization efficiency. In section 5.1, we present the overview where the followings are discussed: setup, assumptions, parameters and metrics used in the simulations. In section 5.2, we present the results of our comparative study with mobile sensor networks using other algorithms and static sensor networks. In section 5.3, we present a more focused study on the optimization and tracking stability of our algorithm. In section 5.4, we examine the effect of non-ideal communication channels on the performance. Finally, in section 5.5, we conclude the chapter.

5.1 General Overview

In this section, we discuss the followings: setup, assumptions, parameters and metrics used in the simulations.

5.1.1 Simulation Setup

Altogether nine simulation scenarios grouped into three cases are simulated: hills and valleys with irregular shapes and sizes (Figure 5.1), sparse terrains with four hotspots (Figure 5.2) and sparse terrains with eight hotspots (Figure 5.3). Figure 5.2-5.3 are used for more controlled simulations in order for us to better understand the stability of our algorithm. Furthermore, the sparseness of the terrains is used to demonstrate the ability of the sensors to cooperatively locate the hotspots through information dissemination and orientation stabilization. We have also hidden some hotspots in the blind spots of the network to test the ability of the sensors to locate the hotspots.



a) Scenario 1: Actual Contour Plot

b) Scenario 1: Actual 3D Distribution

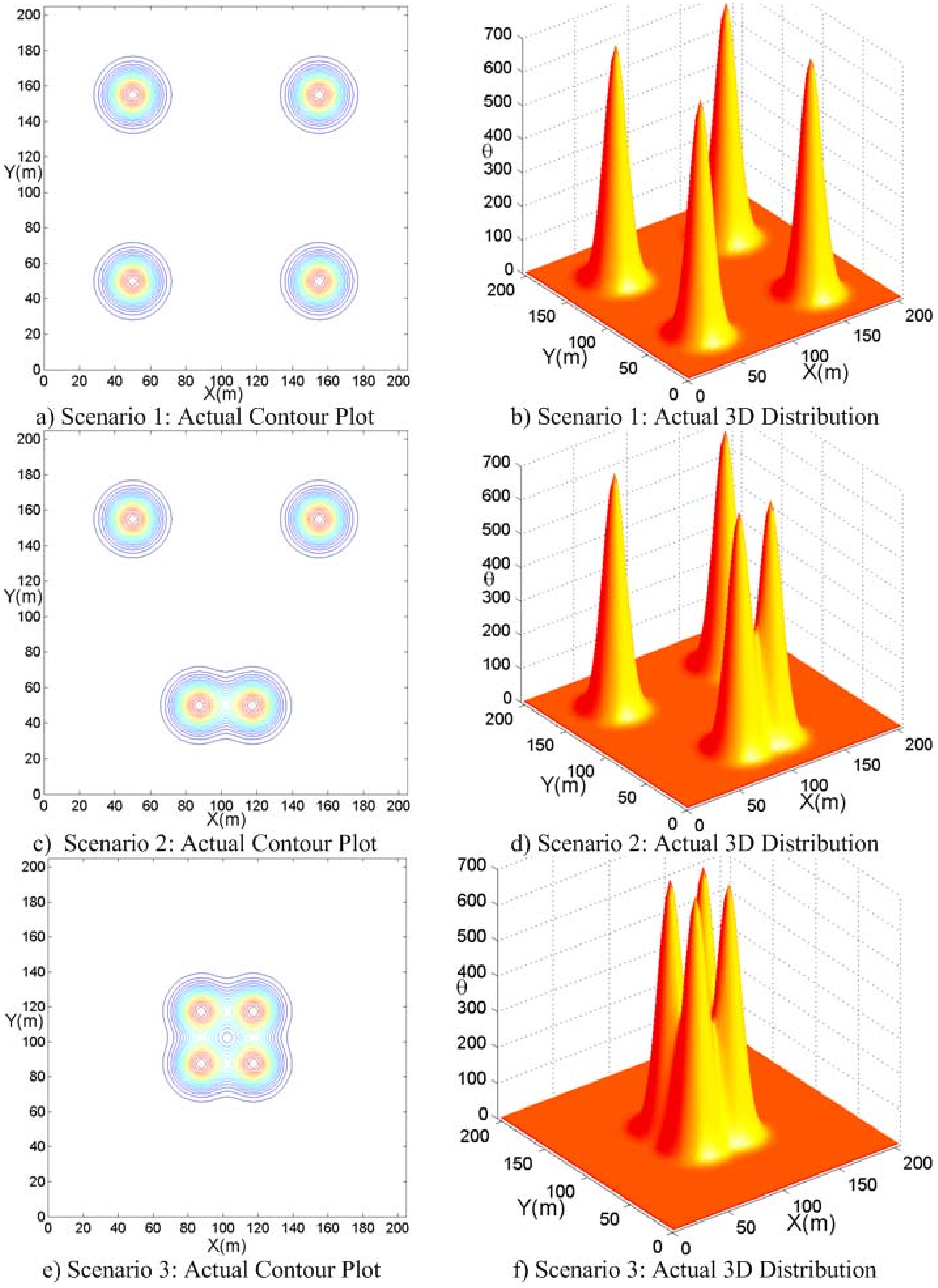
c) Scenario 2: Actual Contour Plot

d) Scenario 2: Actual 3D Distribution

e) Scenario 3: Actual Contour Plot

f) Scenario 3: Actual 3D Distribution

Figure 5.1: Scenarios with hills and valleys of irregular shapes



a) Scenario 1: Actual Contour Plot

b) Scenario 1: Actual 3D Distribution

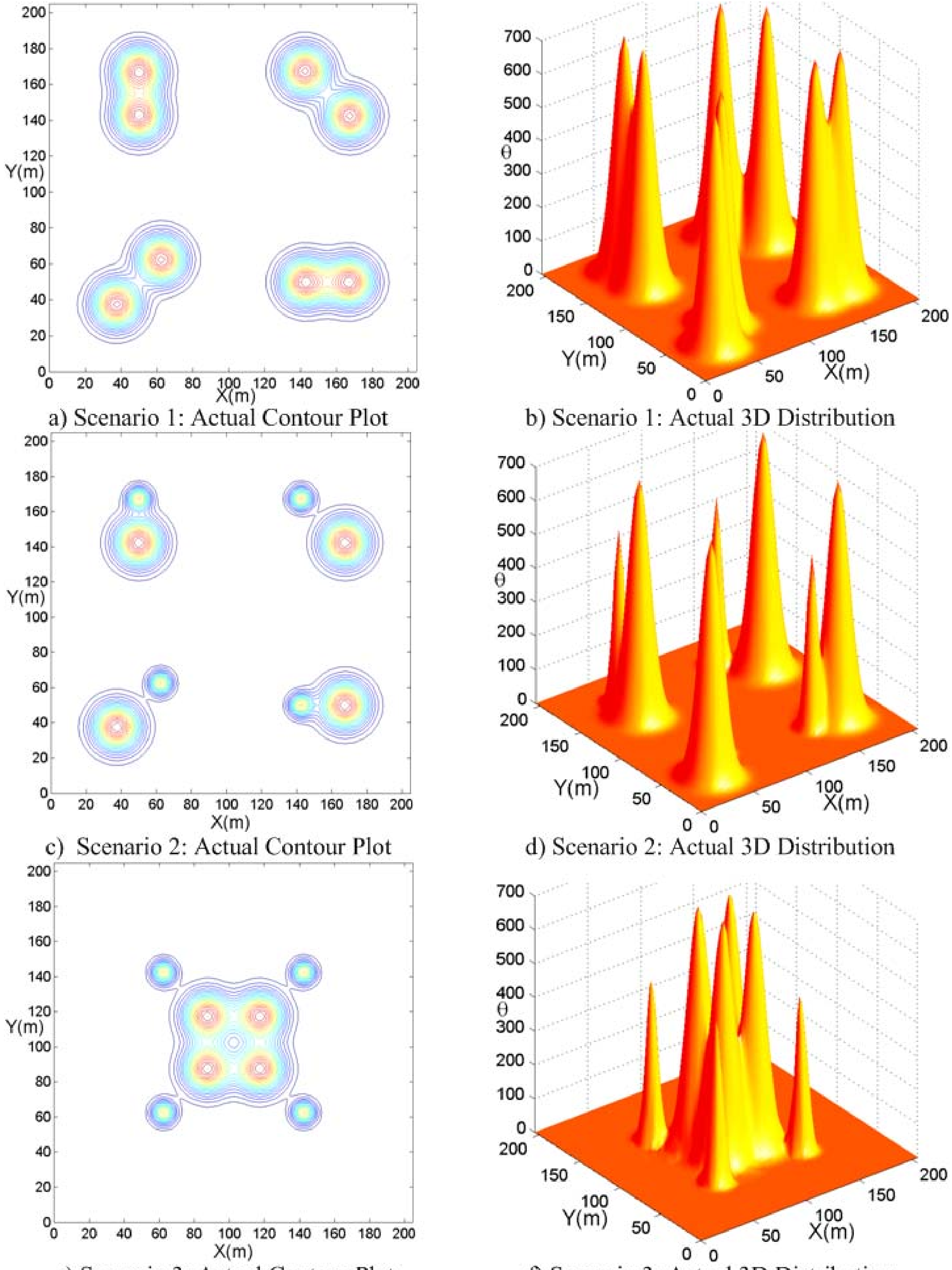
c) Scenario 2: Actual Contour Plot

d) Scenario 2: Actual 3D Distribution

e) Scenario 3: Actual Contour Plot

f) Scenario 3: Actual 3D Distribution

Figure 5.2: Scenarios with 4 hotspots



a) Scenario 1: Actual Contour Plot

b) Scenario 1: Actual 3D Distribution

c) Scenario 2: Actual Contour Plot

d) Scenario 2: Actual 3D Distribution

e) Scenario 3: Actual Contour Plot

f) Scenario 3: Actual 3D Distribution

Figure 5.3: Scenarios with 8 hotspots

Two versions of our cooperative algorithms described in chapter 4 are simulated: Cooperative-Delta and Cooperative-Orbital. Furthermore, in order for us to understand the benefit of the second independence harvesting phase, we develop a pure cooperative version where the second independence harvesting phase is omitted for comparison purpose. There are two versions of this pure cooperative version. In the practical version, we have no advance knowledge of the distribution. Therefore, the gradient and hessian are estimated using (4.4) in chapter 4. In the ideal version, the sensors have complete knowledge of the distribution. The gradient and hessian are computed directly from the distribution without estimation error. Therefore, this would represent the lower bound for our pure cooperative algorithm.

We examine the relative performance of mobile sensors using our cooperative algorithm w.r.t. those deployed in EDG and using the following independent methods: RWM, BFGS and our independent delta harvesting heuristic.

In RWM, the sensors randomly select a speed uniformly distributed between 0 and V_{max} m/s and a direction uniformly distributed between 0 and 360°. The movements of the sensors are confined within circular regions centered at the initial EDG positions. The confinement is to simulate the cooperative constraints mentioned in chapter 4, section 4.2.1.2. This is to simulate the worst case scenario of the algorithm where no information is acquired by the sensors.

Also, we modify an independent Pseudo-Newton algorithm using BFGS to accept a fixed non-zero gradient as a search goal. The goals for the sensors are obtained by running the ideal cooperative algorithm and computing the gradients at the optimal positions at the 20th iterative step. Furthermore, BFGS requires the current gradient as the input to calculate the hessian. Like the pure cooperative version of our algorithm, we have two versions. In the ideal version, the current gradient is computed from the actual distribution. In the practical version, the current gradient is estimated using a five-point stencil maneuver shown in figure 5.4.

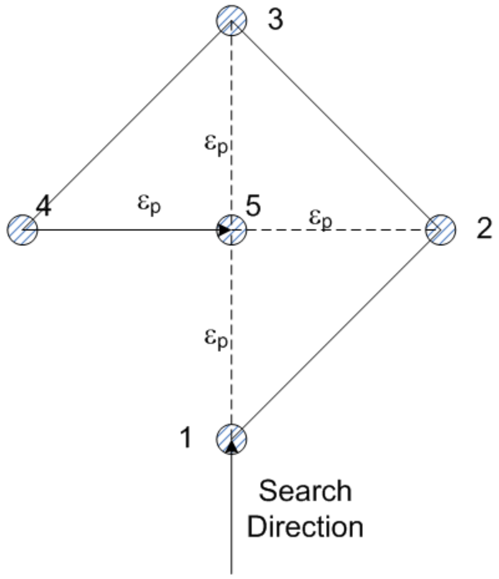


Figure 5.4: Five-point stencil maneuver

In figure 5.4, the actual destination of the sensor is shown as location 5. Prior to reaching location 5, the sensor has to stop at location 1 to 4 to perform measurements. The current gradient is then computed using numerical differentiation [92]. To be accurate in the estimation of current gradient without interpolations, we want location 1-4 to be as near as possible to the center location 5. This is limited by the position accuracy, ε_p. As the sensor does not travel in a straight path from location 1 to 5 and instead takes a longer path, there is an increase in energy consumption as well as a decrease in effective speed. Furthermore, five measurements are required per iteration. Therefore, the measurement delay is increased by a factor of five. Nevertheless, we will ignore the delay for conservative performance studies.

The independent delta heuristic described in figure 4.11b-d, chapter 4 can be used alone. Therefore, we use it as one of the independent methods for comparison purpose. Figure 5.5 illustrates the trajectories of the sensors executing the algorithm for a simple scenario where there is one hotspot and nine sensors. The delta-harvesting heuristic has a simple discrete type of adaptive mechanism where as long as the gradient is above a certain threshold, it will continuously reduce its speed by half up to a maximum of three times and vice versa. The direction is also constrained to spiral outwards to prevent directional instability and to ensure that the region near the sensor is explored thoroughly first before venturing further. Notice that all the sensors harvest data by spiraling outwards. The nodes at the boundaries are confined to move within the boundaries. Once the sensors touch the boundaries, they are reflected back into the terrain of interest. Also, the sensors in the boundaries move faster than node 5 at the center as they are in the localities that have essentially flat temperature gradients, indicating that they are far from information fertile region.

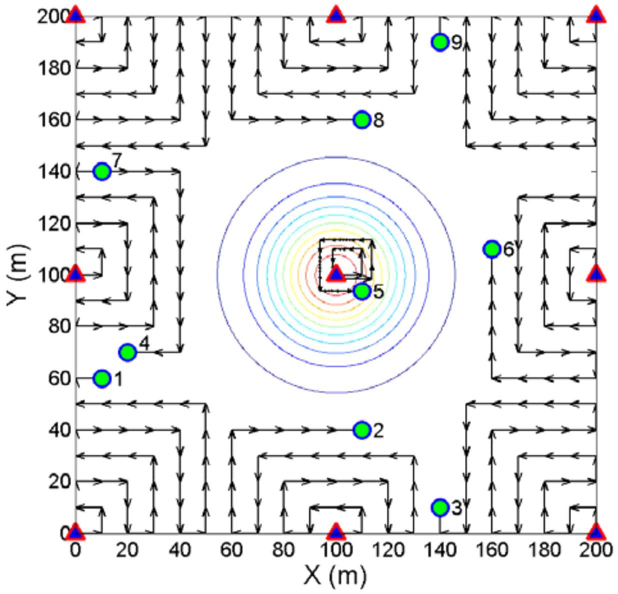


Figure 5.5: Trajectory plot of 9 sensors using the independent delta-heuristic

Comparing with the trajectories of the sensors using the cooperative-delta algorithm in figure 4.11a, chapter 4, we observe that the main difference is that there is no cooperative phase to propel the sensors towards the information fertile region. A second difference is that we set the maximum speed of the independence phase by half in cooperative-delta algorithm to conserve energy.

Our simulation is done using Matlab. For fair comparison, the same cubic spline interpolation available in Matlab is used to reconstruct the distributions from the data harvested by the various algorithms. The interpolation function provides a simple localization noise filtering by averaging.

Finally, we consider a common initial setup where we have four small groups of 25 sensors distributed over four quadrants of the terrain. Each group has its own coordinator and executes their harvesting tasks in isolation from each other. There is no sharing of cooperative state information between the four groups. This is to simulate actual deployments. For the stability study, we also examine an initial setup where we have 100 sensors equally distributed in the square terrain. In this setup, we assume that there is only one coordinator. The purpose is to examine the effect of dividing one large group into four smaller groups on the overall performance for the cooperative algorithms.

5.1.2 Assumptions

The followings are the assumptions used in our simulations.

First, for fair comparison with our cooperative network, the sensors start initially with EDG positions. The assumption is that all mobile sensors would have executed any one of the maximal coverage algorithms [15]-[18]. Note that by assuming that the mobile network is equally distributed, our estimate on the relative performance is conservative. This is because this is the best that a non-adaptive mobile sensor network can achieve in real life as explained in chapter 4, section 4.2.1.5.

Second, there is sufficient memory for the sensors to remember the previous measurements. All the memories collected will be used for the reconstruction of the distribution. We will assume that the distribution is stationary to ignore the need to expire outdated data in the memory. This will not lead to unfair comparisons as the same assumption is used for all algorithms. The main reason is that past information is essential for independent algorithms to both ensure the consistency of their performance and improve their performance. This is especially true for the RWM and independent delta harvesting heuristic. As an example for the independent delta harvesting heuristic in figure 5.6, sensor 6 is moving away from the hotspot after entering it. If we use only the current measurements of all the sensors to construct the distribution for the scenario in figure 5.6, it will lead to the impression that the independent algorithm has erratic and poor performance. In practical situations, even if the hotspot does move, it will move very slowly. If the expiration of data is required, it can be implemented using any number of expiration techniques, for example, using a FIFO queue of a certain length in which old data can be gradually pushed out of the queue.

Third, there is a networking or dissemination mechanism to send the data of the sensors to a final sink node for reconstruction. We will assume that each cluster-head aggregates the information of the sensors at a fixed interval and sends via multi-hop communications to the final sink node. This approach aligns well with the greater scheme discussed in chapter 1, section 1.1. Furthermore, our analysis in chapter 3, section 3.2, shows that the best approach to organize the communications used to facilitate the harvesting task is to have more intra-group communications and minimum inter-group communications. This is consistent with theoretical results reported in the literature such as [69].

Fourth, the mobile sensor is able to obtain its position with higher accuracy than the static sensor through robotic localization. This is because unlike static sensors, mobile sensors can use heterogeneous fusion of measurements (odometry, sonar and laser scanners, etc) to improve the accuracy of its position as mentioned in chapter 1, section 1.4. Particularly, in [90] published in year 1999, an accuracy of less than 5 cm has already been reported from actual mobile robot experiments. For static sensors, [66] published in year 2005 obtained an accuracy of 9 cm for simulation results and 2.27m for experimental results which is two orders of magnitude worst than the accuracy reported in robotic localization. Hence, in reality, the performance of the static sensor network will be much worse. Nevertheless, in our simulation, we use an accuracy of 1 m for all mobile sensor networks in consideration of harsh environments.

5.1.3 Metrics

We use two metrics in our performance studies: relative global error and total energy consumption per sensor.

The relative global error provides the quantitative measure of the total distortion in the distribution. It is expressed as a percentage. Let $\theta(x,y)$ and $\hat{\theta}(x,y,k)$ represent the actual distribution and interpolated bi-variate distribution at the k^{th} iteration respectively. Let h be the step size used in the error computations. Let x_{max} and y_{max} be the maximum values of the x and y coordinates of the terrain. Also, $m=x_{max}/h$, $n=y_{max}/h$. The relative global error in the k^{th} iteration, $\xi(k)$ is computed as:

$$\xi(k) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n}\left|\theta(ih,jh) - \hat{\theta}(ih,jh,k)\right|}{\sum_{i=0}^{m}\sum_{j=0}^{n}\theta(ih,jh)} \times 100\% \qquad (5.1)$$

We compute the total energy consumption per sensor in joules using a linear energy consumption model. This metric is used to demonstrate that cooperative algorithms consume far lesser energy than independent algorithms. We assume that all sensors are of the same weight of one kilogram. Note that since we are interested in the relative performance, the actual weight or energy unit is not of significant concern. In our energy consumption model, we assume that the sensors are moving along a fairly leveled ground with only the frictional forces to overcome. We also assume that the air resistance is negligible due to the low speed and small footprint of the sensors. The total energy consumption at the k^{th} iteration is given in (5.2). The constants introduced in the equation are: u_f which is the coefficient of friction, the smaller the value of u_f, the less energy is consumed, a typical value for u_f is 0.1. m_s is the mass of the sensor and g is the gravitational acceleration which has the value of 10 m/s^2. $d_n(t)$ is the distance travelled in the time interval between $(t-1)^{\text{th}}$ and t^{th} iteration. As communications consume energy in magnitude of microjoules [94], it is ignored. The total energy consumption per sensor in joules (J), $E(k)$ is given as:

$$E(k) = \frac{1}{N_s} \sum_{t=1}^{k} \sum_{n=1}^{Ns} m_s g \mu_f d_n(t) \qquad (5.2)$$

As an extra note, this model gives a more conservative estimate in the relative performance gain of our cooperative algorithms for two practical reasons. First, in practical situations, the battery provides the energy for the mobile sensor. This battery occupied a significant proportion of the body weight of the sensors. The more energy the sensor needs, the heavier will be the sensor. As a result, we need even more batteries (or equivalently, energy) just to carry the extra burden generated. In other words, the energy consumption is significantly higher and non-linear in practical situations because our model does not account for the extra increase in the weight of the batteries when the energy consumption increases. Second, if we examine the trajectory plot of the independent delta-harvesting heuristic in figure 5.5, most of the time, the sensors are travelling at maximum speed. In practical situations where air and especially water resistances are not negligible. The additional energy consumption required to overcome the resistances are proportional to the squared or even cubed of the speed, depending on the empirical models used. Therefore, the practical energy consumption of the independent algorithms is much higher than cooperative algorithms then suggested in (5.2).

5.1.4 Simulation Parameters

The values of parameters used for our simulation is listed in table 5.1.

Table 5.1: Values of the parameters for the performance studies

Parameter	Value
Number of Runs	20
Dimension of Terrain	205 × 205 m^2
Number of Sensors, N_s	100
Constrained Radius	40 m
Maximum Speed of Sensor, V_s	10 m/s
Duration of 1 Iterative Step	1 s
Radial Basis Function constant, σ	100
Mean Location Accuracy, ε_p	1 m
Temperature Accuracy, ε_θ	1 °C
Error Computation Step Size, h	2.5 m
Coefficient of friction, u_f	0.1
Weight of the Sensor, $m_s g$	1 kg × 10 m/s^2
Coordination – *nos_of_sensors_threshold*	85%
Maximum Harvesting Speed at Independence Phase	5 m/s
Delta-Harvesting – *GradientThreshold*	10 °C /m
Cooperative-Delta – Maximum Harvesting Radius	25 m
Cooperative-Delta – no gradient detected stop radius	12.5 m
Cooperative-Orbital – Angle of Deviation, ϕ	±15°
Tracking – Measurement Delay, T_θ	1 s
Tracking – *AmbientThreshold*	35 °C
Tracking – Communication Delay, T_{comm}	0.1 s
Tracking – *TimeLimit*	1 s

5.2 Comparative Study

In this section, we first examine the relative average performance in terms of relative global error and total energy consumption per sensor of mobile sensors using three different versions of our cooperative algorithms: Cooperative, Cooperative-Delta and Cooperative-Orbital with mobile sensors deployed in EDG and those using the independent algorithms: RWM, BFGS, and our delta-harvesting. Next, we examine the relative average performance of the mobile sensors using the abovementioned harvesting algorithms w.r.t. a static sensor network in terms of resource utilization efficiency.

5.2.1 Relative Performance with Mobile Sensor Networks using different harvesting algorithms

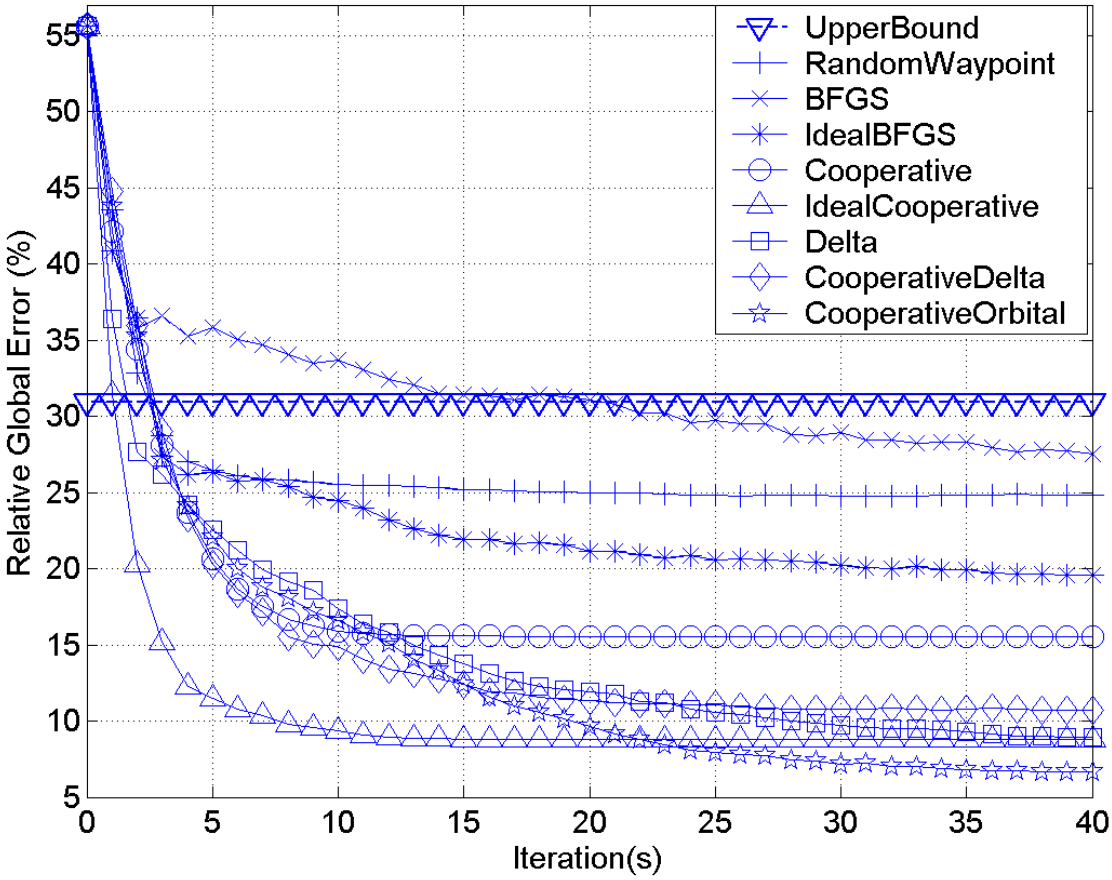


Figure 5.6a: Relative global errors for the different algorithms for the 9 scenarios

Figure 5.6a-b examine the average relative global error and total energy consumption per sensor. The plots are obtained by taking the average performance over all the nine scenarios. As can be seen from figure 5.6a, our approximate upper bound, ξ_{ub} in (4.12), chapter 4, for 40 iterations is tight.

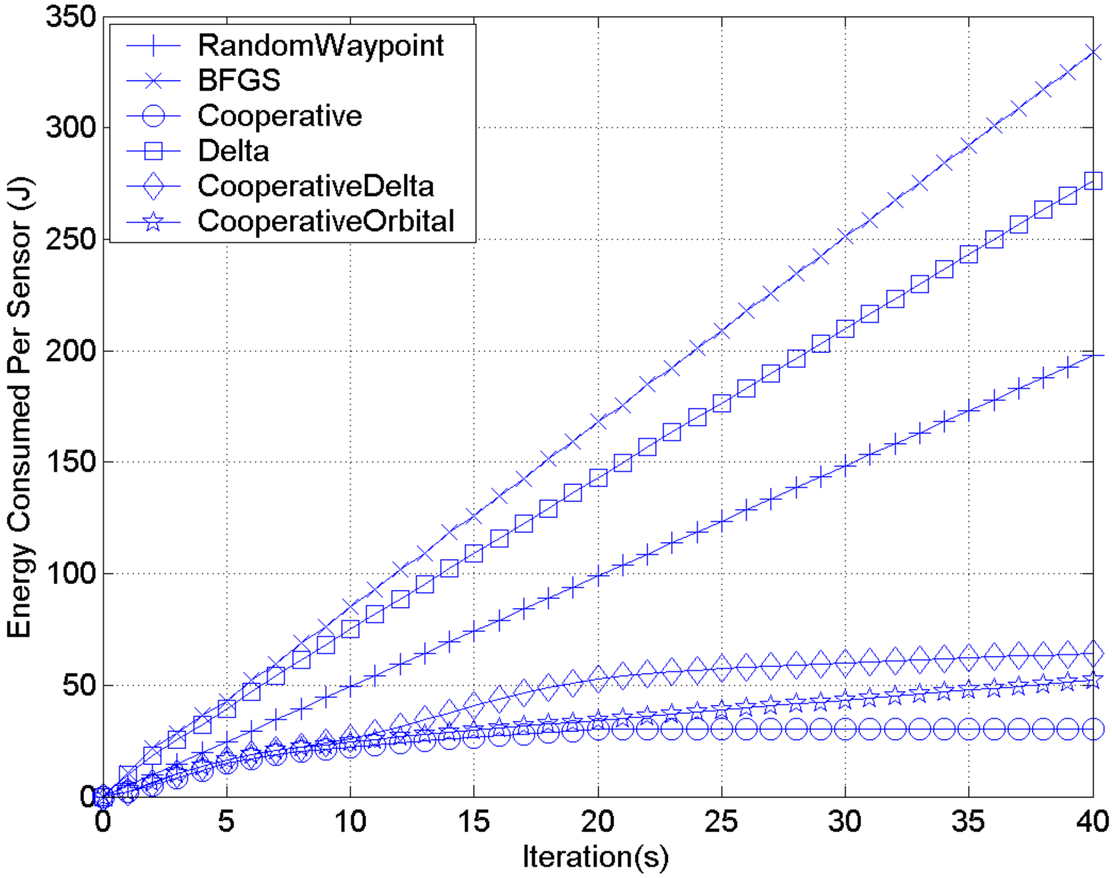


Figure 5.6b: Total energy consumption per sensor for the different algorithms for the 9 scenarios

First, we examine the performance of RWM. In figure 5.6a, we see that RWM is slow, especially beyond the 10th iterative step. In general, sensors moving with random mobility models tend to cluster near the centers of the circular regions. Over time, the sensors tend to repeatedly harvest data points that they have harvested previously around the same locality. As a result, their performance diminishes rapidly over each iterative step. Moreover, in scenarios from figure 5.2-5.3 where there are sparse regions, there are sensors that harvest data of low information content. In figure 5.6b, the average total energy consumption of RWM increases linearly. In fact, the analytical average total energy consumption per sensor at the k^{th} iterative step, $\xi(k)$ can be obtained trivially by observing that the average speed of the sensor is 5 m/s for each step for all scenarios. This is because their mobility is not dependent or adaptive to the environment.

Second, we examine both the ideal and practical versions of BFGS in figure 5.6a. BFGS is generally slow and only the ideal BFGS performs slightly better than RWM. The slow performance of the ideal BFGS is because of the followings. First, the hessian is estimated from current and past gradients. Thus, it has a warm up time before it acquires enough past gradients of high information content. Second, a fixed line search is used for stability which constrains the movement to a straight line inefficiently for a period of time. This is especially true when the search direction is at an angle from the goal. Third, as there is no communications, in some runs, in regions where optimal positions are near and their gradients have approximately the same value, the sensors ended up chasing after the same search goals. In addition to the general problems of BFGS, the poor performance of the practical BFGS is also caused by the followings. First, the five-point stencil maneuver in each step reduces the effective speed of the sensor, slowing the algorithm further. Second, due to the memory effect, the estimation error of the gradient in each step accumulates in the estimation of hessian. From figure 5.6a, this memory effect causes the practical BFGS to perform worse than the memory-less RWM when ε_p = 1 m. Thus, we infer that the use of past information is infeasible for BFGS in a harsh environment. In figure 5.6b, we noted that practical BFGS consumes the most energy. This is because the five-point stencil maneuver consumes energy. We clearly see that the extra energy effort and delay incurred in taking the extra measurements are ineffective in improving the performance w.r.t. RWM. We expect the BFGS to be even slower in a realistic environment as we have ignored the measurement delay in our simulation.

Third, we examine the independent delta-harvesting heuristic without the cooperative phase (labeled as Delta) in figure 5.6a. Although it is simpler than BFGS, it outperforms BFGS. It even outperforms both the pure cooperative and cooperative-delta algorithms towards the final stage. Note that pure cooperative algorithm is still faster than the independent delta-harvesting heuristic before the 12th iterative step. However, after the 12th iterative step, sensors using the independent delta heuristic can continuously reduce the global error because those using the pure cooperative algorithm are locked into the optimal positions. For the cooperative-delta algorithm, it outperforms the independent delta-harvesting heuristic up to 24th iterative step. This is because, in the interval between the 12th and 24th iterative step, the cooperative-delta has a head-start in using the delta-harvesting heuristic to harvest the data in the information fertile region as the sensors are propelled into this region at around the 12th iterative step. Beyond, the 24th iterative step, the independent delta-harvesting heuristic, which is moving at twice the maximum speed of cooperative-delta algorithm during the independence phase (see Table 5.1), is able to catch up with cooperative-delta algorithm in terms of error reduction. However, independent delta-harvesting heuristic achieves this with greater brute force and lesser intelligent as can be seen in figure 5.6b. Independent delta-harvesting heuristic consumes more energy than cooperative algorithms in general. Particularly, it also has higher energy consumption than RWM. Referring to the trajectory in figure 5.5, we observed that sensors in region with gradients of almost zero magnitudes traveled at a constant maximum speed of 10 m/s compared to the average speed of 5 m/s by sensors with RWM. Hence, we infer that the sparser the region, the higher is the energy consumption for independent delta-harvesting heuristic.

Fourth, we examine the ideal and practical versions of the pure cooperative algorithm in figure 5.6a. The practical version is closer to the ideal cooperative algorithm than RWM. This shows that (4.4) in chapter 4 provides a good estimation. It also performs better than BFGS. Furthermore, before the 12th iterations, the practical cooperative algorithm is able to reduce the global error at a faster rate compare to independent algorithms. In figure 5.6b, we observe that pure cooperative algorithm consumes the least amount of energy. The ability to make use of cooperatively shared information to economically move the sensors to conserve energy has resulted in high energy efficiency. The faster rate of error reduction and high energy efficiency are the two motivations for using cooperation in our algorithm. The ideal cooperative algorithm shows that without any errors, we could achieve a global error as low as 8% at the 15th iterative steps. Cooperative-delta and cooperative-orbital algorithms are our practical way to achieving a better global error reduction.

Fifth, we examine the cooperative-delta algorithm in figure 5.6a. By introducing an independence phase into the algorithm, we overcome the limitation of practical pure cooperative algorithm which has no further error reduction after the sensors are locked into optimal positions at the 12th iterative step thereby achieving an error close to that of the ideal pure cooperative algorithm. However, it is still not good enough w.r.t. the independent delta-harvesting heuristic. The effectiveness of delta-harvesting at the independence phase is reduced because the sensors are crowded closer when they are propelled into the information fertile region. Thus, there are more overlaps during delta-harvestings. Another reason is that the delta-harvestings in the independence phase of the cooperative-delta algorithm are done at half the maximum speed of 10 m/s to conserve energy. This is because the cooperative-delta algorithm inherits the undesirable high energy consumption property of the delta-harvesting heuristic. In chapter 4, section 4.2.2, we discuss ways to curb the movements of the sensors using the delta-harvesting heuristic in the independence phase in order to conserve energy. Figure 5.6b shows that even with the curbing, the cooperative-delta algorithm consumes the highest energy in the cooperative category. Nevertheless, it still consumes much lesser energy than the independence category.

Sixth, we examine the cooperative-orbital algorithm in figure 5.6a. This algorithm performs outstandingly compared to the rest of the algorithms. It is able to achieve an error less than that of the ideal pure cooperative algorithm. In figure 5.6b, it consumes lesser energy than the cooperative-delta algorithm while slightly higher energy than the pure cooperative algorithm. However, for that extra small amount of energy consumed w.r.t. the pure cooperative algorithm, we are able to achieve an additional 8-9% reduction in error. Therefore, we believe it is a worthwhile effort.

We now examine more closely the average overall relative performance of the cooperative-orbital algorithm tabulated in Table 5.2.

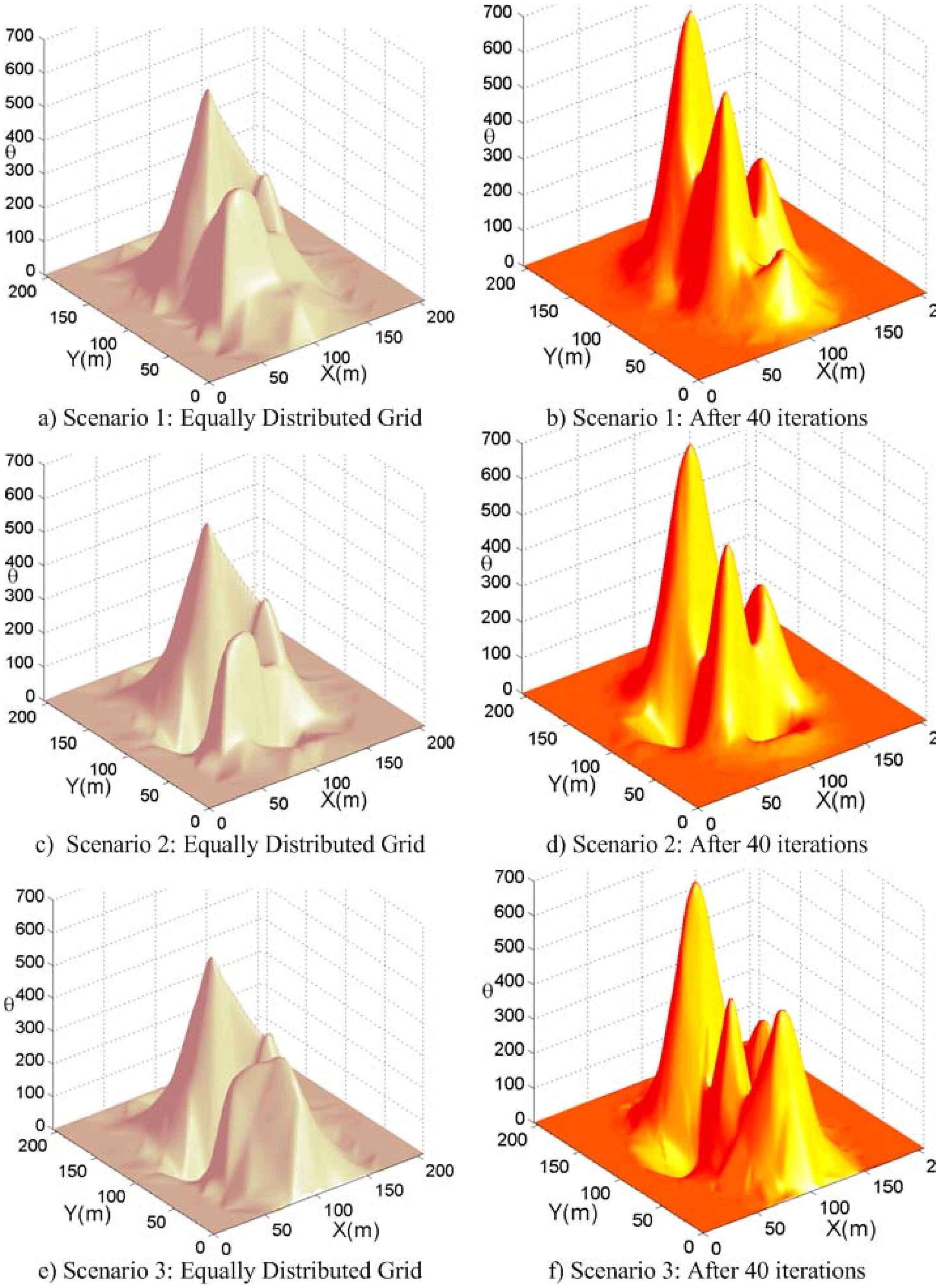
Table 5.2: Relative performance of cooperative-orbital algorithm

	Relative Global Error	Energy Consumption
EDG	+737.5%	N.A.
Independent RWM	+273.8%	−73.6%
Independent BFGS	+314.3%	−84.4%
Independent Delta-Harvesting	+34.9%	−81.1%
Cooperative	+133.7%	+72.9%
Cooperative-Delta	+60.8%	−18.8%

From Table 5.2, we see that cooperative-orbital algorithm reduces 737.5% more error (with a range of 625% to 885%) than mobile sensors deployed in EDG.

Therefore, the use of our optimal harvesting algorithm in mobile sensor network can improve data harvesting. In comparison with the best approach in the independent category, it reduces 34.9% more error while consuming 81.1% lesser energy than independent delta-harvesting. In the cooperative category, it reduces 133.7% more error while consuming 72.9% more energy than cooperative algorithm. In absolute terms, that represents a mere increase from 30.2J to 52.1J which is still a substantial saving comparing with independent delta-harvesting which consumes 275J (Figure 5.6b). Moreover, as explained in the previous paragraph, this additional increase of 21.9J is used to reduce 8-9% more error than pure cooperative algorithm. Therefore, our cooperative-orbital algorithm is an energy efficient way to improve the performance of mobile sensor networks. As explained in section 5.1.4, our energy comparison is conservative. In reality, sensors using independent methods consume even more energy. Consequently, we can achieve a much higher energy efficiency gain.

Figure 5.7-5.9 show the reconstructed distributions from data harvested from EDG sensors and those obtained after executing our cooperative-orbital algorithm for 40 iterations. The actual distributions are in figure 5.1-5.3. Prior to the execution of our algorithm, mobile sensors deployed in EDG extracted the distribution with wrong shape, height and number of peaks and valleys. After the execution, approximately correct shape, height and number of peaks and valleys are obtained. In particular, in figure 5.7, scenario 1 and 2 prior to the execution of our algorithm appear identical (Figure 5.7a and 5.7c). After execution of our algorithm, the two distributions are markedly different as scenario 1 has a small hill and no valleys (Figure 5.7b) whereas scenario 2 has no small hill and two valleys (Figure 5.7d). Furthermore, in some scenarios where hotspots are hidden at the blind spots of which the most prominent is scenario 3 in figure 5.9, mobile sensors deployed in EDG fail to detect the four smaller hotspots (Figure 5.9e). However, our cooperative-orbital algorithm is able to extract the correct number of hotspots (Figure 5.9f).



a) Scenario 1: Equally Distributed Grid

b) Scenario 1: After 40 iterations

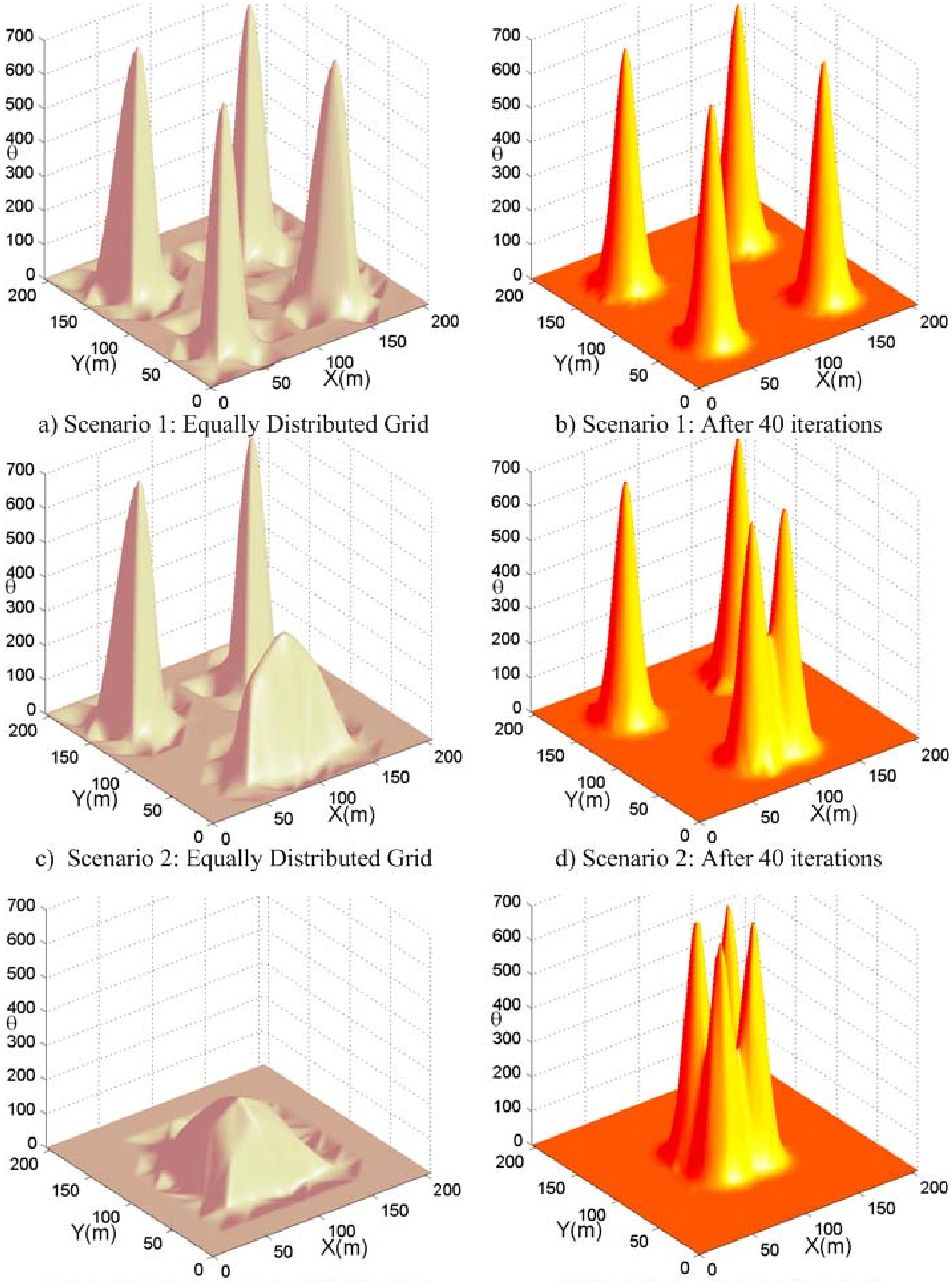
c) Scenario 2: Equally Distributed Grid

d) Scenario 2: After 40 iterations

e) Scenario 3: Equally Distributed Grid

f) Scenario 3: After 40 iterations

Figure 5.7: Reconstructed distributions of scenarios with hills and valleys of irregular shapes using data obtained from cooperative-orbital algorithm



a) Scenario 1: Equally Distributed Grid

b) Scenario 1: After 40 iterations

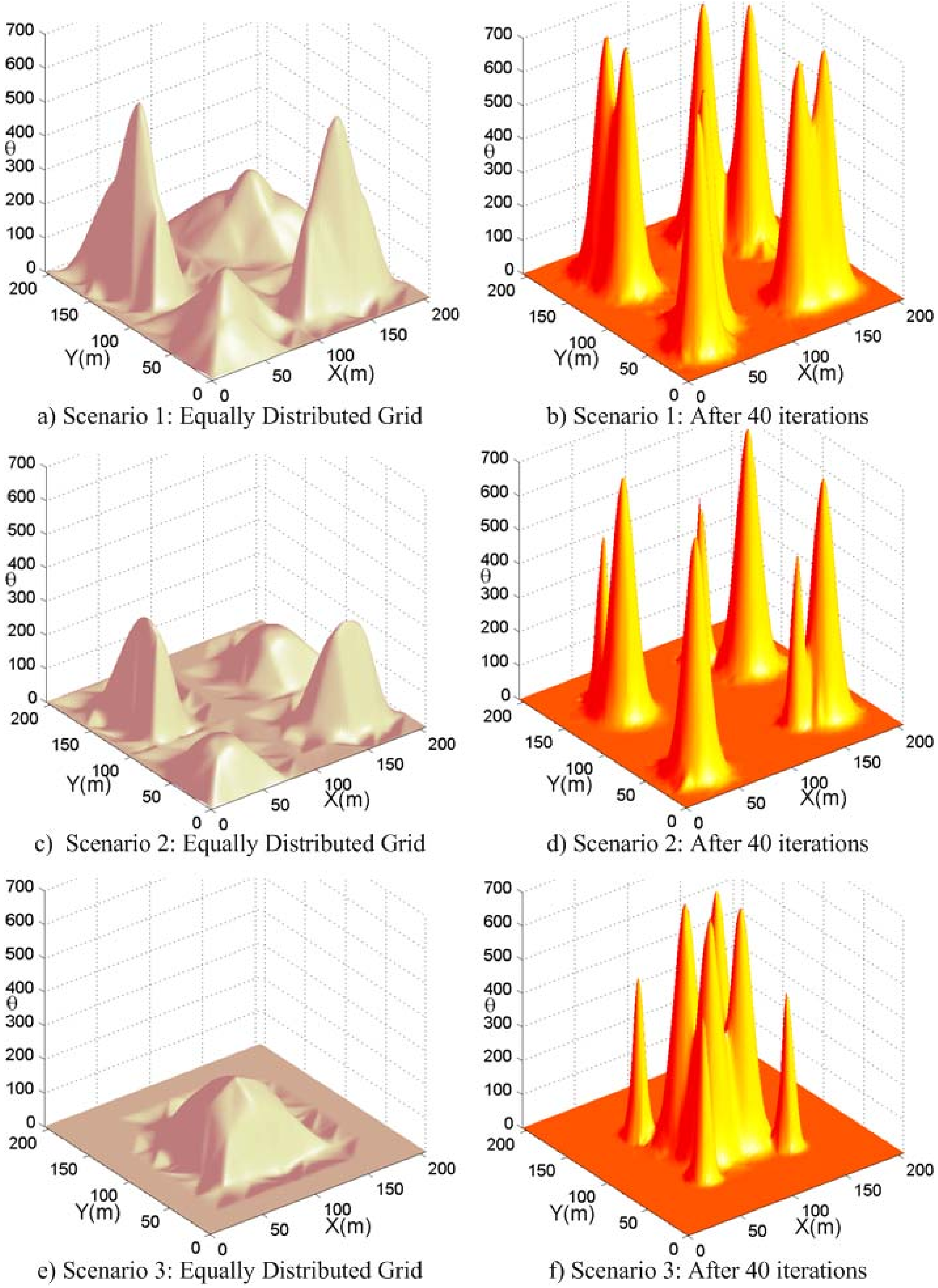
c) Scenario 2: Equally Distributed Grid

d) Scenario 2: After 40 iterations

e) Scenario 3: Equally Distributed Grid

f) Scenario 3: After 40 iterations

Figure 5.8: Reconstructed distributions of scenarios with 4 hotspots using data obtained from cooperative-orbital algorithm



a) Scenario 1: Equally Distributed Grid

b) Scenario 1: After 40 iterations

c) Scenario 2: Equally Distributed Grid

d) Scenario 2: After 40 iterations

e) Scenario 3: Equally Distributed Grid

f) Scenario 3: After 40 iterations

Figure 5.9: Reconstructed distributions of scenarios with 8 hotspots using data obtained from cooperative-orbital algorithm

.

5.2.2 Relative Performance with Static Sensor Networks

We examine the performance of static w.r.t. mobile network. The static sensors are uniformly distributed to simulate actual deployment. For realistic and conservative comparison, we use the best experimental ε_p of 2.27m [66] which is achieved at the expense of high communication costs. Moreover, in accordance to static sensor localization [66], the reduction in ε_p as node density increases is simulated. A linear model is used to estimate the localization error of the static sensor network. Generally, assuming all parameters in the localization algorithm are identical, the square of the mean localization error, ε_p^2 will be inversely proportional to the node density, ρ_s (exclusive of the four anchor nodes). In particular, using the method in [66], it is also proportional to area, d_r^2 of the neighbors' selection region (assuming that the static sensors are uniformly distributed) as follows:

$$\varepsilon_p = \varepsilon_{0,p}\sqrt{\frac{\rho_{0,s} d_{0,r}^2}{\rho_s d_r^2}} \tag{5.3}$$

Note that d_r is the neighbors' selection range which is the radius of the circular selection region centered at the sensor. This model allows us to compute the localization error based on any actual experiment conducted. In particular, in the experiment conducted in [66], the values for the mean location error, node density and neighbors' selection range are respectively as follows: $\varepsilon_{0,p}$ = 2.27 m, $\rho_{0,p}$ = 44 ÷ 14^2 sensors/m^2 and $d_{0,\mathrm{r}}$ = 8.5 m. Note that the experiment is conducted in the open plan office building where communication channel is good and using the typical specification for the static wireless sensors. The minimum mean error of 2.27m is obtained using maximum communication costs. For conservative purpose, we will use this data and vary only the node density. As a result, our localization error for the static sensors decreases linearly as the node density increases. In reality, localization error saturates at higher density and may even increase [95]. The reasons that this approach is conservative are as follows:

1. The error is obtained at maximum communication costs in a good communication environment.
2. In reality, as the number of sensors increases, as a result of network congestion, the effective node density as seen by the network is much lower.
3. The localization method in general required dissimilar data to increase diversity and hence improve the estimate. Starvation reduces the diversity of data that is required to obtain a good estimate. Congestion at higher density increases the probability of starvation as more aggressive nodes tend to hoard the communications.
4. The approach to combat harsh communication environment is to reduce the neighbor selection range, $d_{0,\mathrm{r}}$ to limit the data collected for estimation from neighbors that are inside the circular selection region centered at the sensor [66]. This is because the closer the neighbors are to the sensor, the better is the received signal strength and the more reliable is the data. In a harsh communication environment, we will have to reduce the radius of the selection region, $d_{0,\mathrm{r}}$ further below 8.5 m in order for us to obtain a better estimate.

In comparing the relative performance of the mobile to static networks, we use the **Resource Utilization Efficiency** which is defined as the ratio of *the number of static sensors required to achieve the same relative global error as the mobile sensors executing a specific harvesting algorithm for 40 iterative steps* to *the number of the mobile sensors used in the harvesting*.

Figure 5.10 shows the average relative global error over the same nine scenarios used in section 5.2.1 for the static sensor network. The errors obtained for the various harvesting algorithms at the 40th iterative step in section 5.2.1 are plotted into figure 5.10 together with their resource efficiency gains. From figure 5.10, without any post-processing, at least 2.5×10^4 static sensors are required to obtain the same performance as mobile sensors using the cooperative-orbital algorithm. Therefore, mobile sensors using the cooperative-orbital algorithm is 250 times more efficient than static sensors in utilizing resources. This suggests that mobile sensor networks using our optimal harvesting algorithm are an attractive environmental friendly and reusable substitute to some static sensor applications. It also suggests that our approach is cost effective enough for us to install sophisticated equipments such as GPS per sensor, resulting in a virtue cycle of further improving their accuracies and efficiencies.

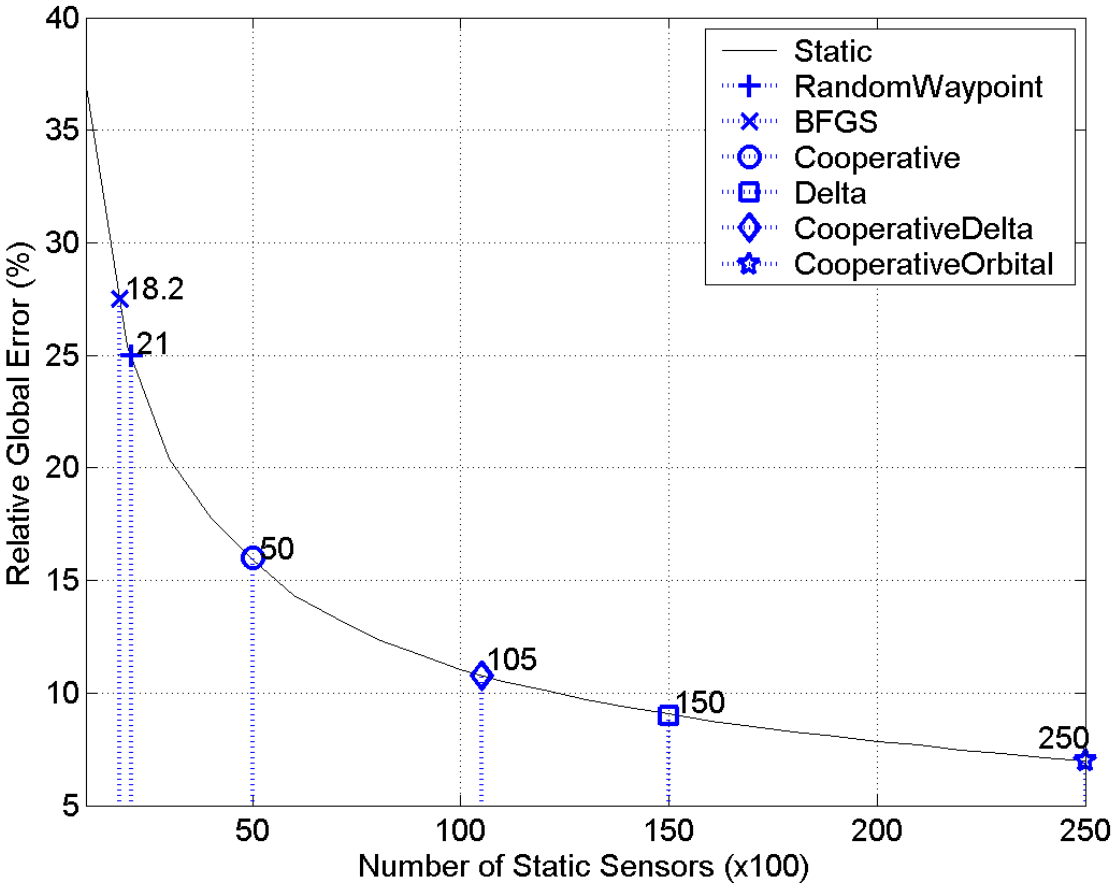


Figure 5.10: Relative global error of static sensor network

5.3 Stability Study

In this section, we examine the stability of the cooperative optimization and tracking mechanism in our algorithm separately.

5.3.1 Optimization Stability

Our approach makes use of gradient and hessian information to navigate the terrain. Physically, these are translated as the steepness and curvature of the distribution. Therefore, we postulate that an increase in steepness and undulating surface in the distribution will result in increase in oscillatory or agitated behaviors in the sensors. From control perspective, the rapid rate of change in state information input due to the undulating surface when the system is in a dynamic state of flux put a strain on or load the system. We can study the strain on the system by the deterioration in its performance in error reduction and energy consumption. The simulation results from 4-hotspot and 8-hotspot scenarios (Figure 5.2-5.3) are used for our study. For clarity of presentation, the error and energy consumption spreads of the scenarios (Figure 5.11a-b) are used. The error and energy consumption spreads are obtained by subtracting the individual metrics at the 40th iteration for each scenario by the average value over the six scenarios. By doing so, the spreads are centered at the zero point on the vertical axis. We can then observe clearly the widening and narrowing of spreads in all the methods.

First, we examine the RWM method in data harvesting. From figure 5.11a, we observe that it has the widest error spread. However, this does not imply that sensors using the RWM method are under strain due to an increase in undulating surface when the hotspots are increased from four to eight. We observe further that the order of performance is random over the six scenarios for the RWM method in figure 5.11a.

Sensors using RWM method do not make use of any external environmental information to decide their next moves. Thus, they are insensitive to any external stimuli. Generally, the wide spread or volatility is merely a reflection of the unpredictable nature of random harvesting method. The insensitivity of RWM method to environment is confirmed by observing the energy consumption spread in figure 5.11b. This method has zero energy consumption spread because the average speed for the sensor is constant at 5 m/s for all the scenarios, regardless of the number of hotspots.

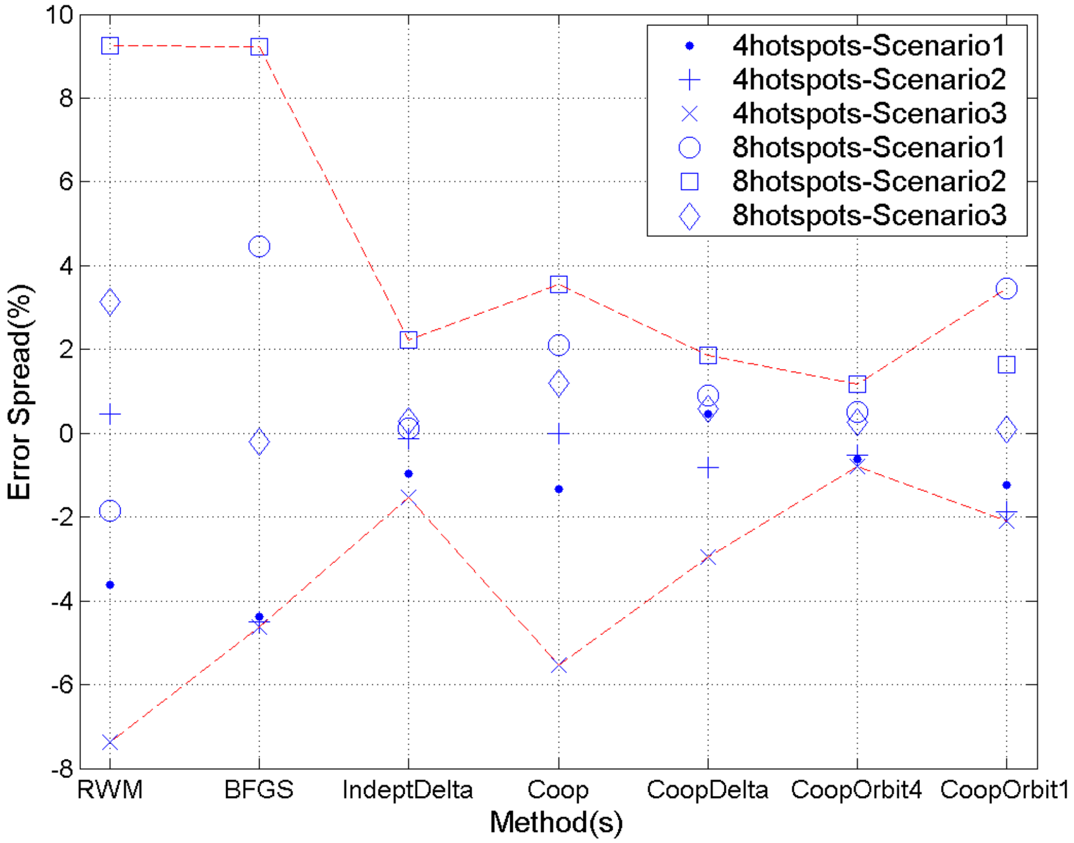


Figure 5.11a: Error spread for different methods

Second, we observe a trend in all the intelligent harvesting methods: BFGS, Independent Delta-Harvesting (labeled as IndeptDelta), Cooperative (labeled as Coop), Cooperative-Delta (labeled as CoopDelta) and Cooperative-Orbital (labeled as

CoopOrbit4) in figure 5.11a. Generally, the relative global error increases when the number of hotspots is increased from four to eight. In particular, BFGS appears to be least tolerant to increase in strain as the increase in the number of hotspots causes a large deterioration in performance. We observe that our cooperative-orbital algorithm is most robust to increase in the number of hotspots as it has the narrowest spread. This is followed by our independent delta heuristic in the independence category. When we cross-examine with figure 5.6a, we observe that there is a strong correlation between the error reduction performance of the system and its error spread (stability). That is, the cooperative-orbital algorithm is the best performing algorithm with the largest error reduction and simultaneously the smallest error spread. This is followed by the independent delta-harvesting heuristic, cooperative-delta, pure cooperative and BFGS algorithm in the performance order from best to worst. Particularly, if we focus on error spreads among those in the cooperative category, the incorporation of an independence phase in our cooperative algorithm helps to improve the robustness of optimization.

Next, we examine figure 5.11b. All the cooperative methods exhibit similar trends in that an increase in the number of hotspots causes the energy consumption to increase. This is generally due to increase in oscillatory behaviors when the system is under strain as mentioned in chapter 4, section 4.3. BFGS also follows similar trend although there is one aberration (8 hotspots – Scenario 3). The aberration is due to sensitivity of BFGS to a number of other factors such as initial positions of the sensors and accumulated estimation error from past information. Finally, independent delta-harvesting heuristic exhibits an opposite trend w.r.t. other intelligent methods. This has been explained in section 5.2.1 and is merely a peculiarity of the heuristic. The reason is that the sensors will move at maximum speed in the locality with flat temperature gradients. We observe in figure 5.2 and 5.3 that the 4-hotspot scenarios have a larger proportion of the terrain that has flat temperature gradients than 8-hotspot scenarios. As a result, there are more sensors residing in regions of flat distribution harvesting at maximum speed in the 4-hotspot scenarios than those in the 8-hotspot scenarios. Similarly, the smaller energy spread of cooperative-delta algorithm is also due to the individual peculiarity. That is, it is because of the special energy curbing measure taken in the independence phase of the algorithm as mentioned in chapter 4, section 4.2.2.

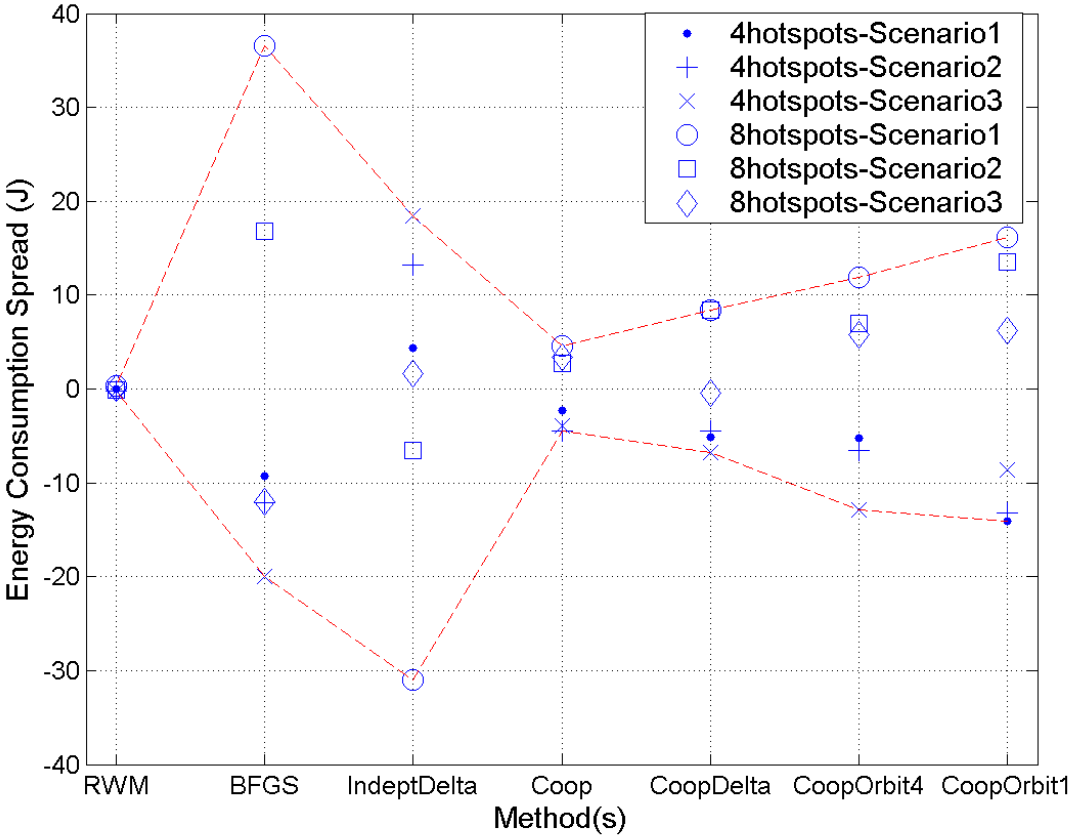


Figure 5.11b: Energy consumption spread for different methods

Generally, in figure 5.11b, there is a trend showing that when the sensors using cooperative intelligent method are under environmental strain, the energy consumption increases. However, there is no strong correlation between the magnitude of energy consumption spread and stability of the algorithm among the different algorithms due to the peculiarity of the individual algorithms. Particularly, zero energy consumption spread in sensors using RWM method does not imply that the sensors are stable and predictable.

Third, as mentioned in chapter 1, section 1.1, in the greater scheme, the sensors are deployed in small groups in consideration of difficult terrains where there may be partitions. Therefore, we are interested in the effect of deploying the sensors in four small groups (labeled as CoopOrbit4 in figure 5.11a-b) w.r.t. deploying them as one large single group (labeled as CoopOrbit1 in figure 5.11a-b). We present the result for sensors using our cooperative-orbital algorithm as similar trends are observed for those that use the cooperative and cooperative-delta algorithms.

In figure 5.11a-b, we observe that sensors that are deployed in four small groups have smaller spreads than those in one large single group. Therefore, we infer that those deployed in four small groups are more stable than those that are deployed in one large single group. Note that although, in the previous paragraph, we explain that there is no correlation between energy spread and stability when we compare with different methods due to individual peculiarities, we are now comparing with the same method.

The reason that those deployed in four small groups are more stable than those that are deployed in one large single group is because of the coupling effect due to the sharing of the cooperative state information as mentioned in chapter 4, section 4.3. In the previous section, we have seen the importance of cooperative shared state information in improving the error performance and energy efficiency of the cooperative mobile sensor groups. However, the same communication channels also allow the oscillations of individual sensors under strain to be propagated globally. By dividing the sensors into four small groups and ensuring that there is no inter-group transmission of cooperative shared information to facilitate the harvesting task, we have decoupled and isolated the four groups. Consequently, any intra-group instability is contained. From the result, we infer that while increasing the number of sensors in a cooperative group allows us to reap the benefits of increasing cooperative shared information, it will eventually be counterbalanced by the detrimental coupling effects. Therefore, to conclude, our greater scheme of using small groups of sensors has three benefits: easier deployment in difficult terrains, more robust communications (discussed in chapter 3) and more stable cooperative behaviors among the sensors.

5.3.2 Tracking Stability

Figure 5.12 shows the average separation between the centers of the tracking clusters and the sources when four hotspots with constant speeds and uniformly distributed random directions move independently and continuously. It is observed that the average separation is bounded and stable over time. Moreover, as the speed of the hotspot, V_h approaches the maximum speed of the sensor, V_s the separation increases. These are corroborated by our stability analysis of the tracking mechanism in chapter 4, section 4.2.3.5 and appendix E.

Finally, we briefly discuss on the issues of reconstructions of distributions. Past data are useful in the reconstructions as they reduce global distortions. However, their direct use in non-stationary distributions leads to ghost or smeared images due to: space-time distortion and discrete time monitoring process. One method to mitigate the problem is to expire past data after some time. A more effective method requires post-processing which is beyond the scope of this thesis. In the simple case of one hotspot, we can apply affine transforms on the past data to use them. In cases with many sources of varying intensities, more sophisticated post-processing is required in order to use the past data. This is because the temperature at any point is the superposition of the effects of all sources.

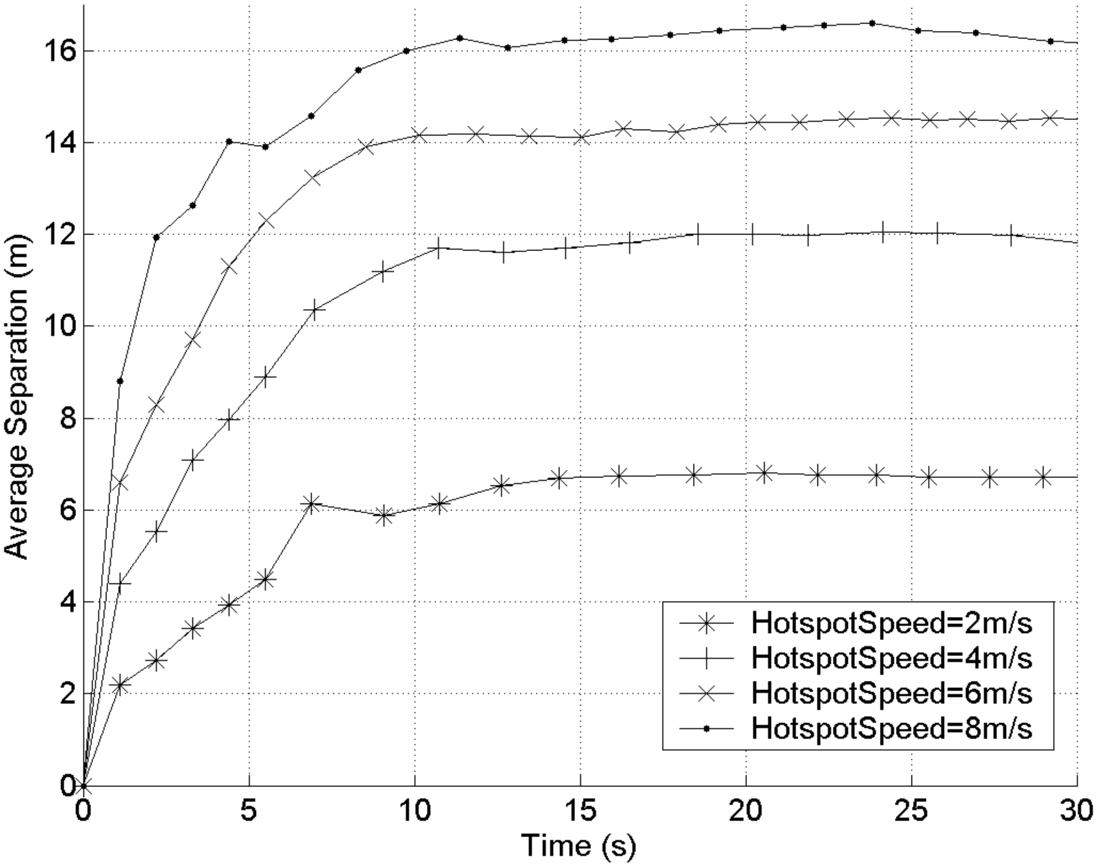


Figure 5.12: Average separations between the centers of the tracking clusters and the hotspots

5.4 The effect of non-ideal communications and sensor failures

In this section, we examine the effect of non-ideal communications on the performance and permanent sensor failures. Initially, we examine the scenario where there are no sensor failures. This is followed by the scenario with sensor failures.

5.4.1 Effect of non-ideal communications

Based on our work in chapter 3, the most suitable MAC protocol for our network is the DS/CDMA protocol. Two communication scenarios discussed in our theoretical analyses in chapter 3 are examined.

The first scenario is a terrestrial scenario. The terrestrial channel with Additive White Gaussian Noise (AWGN) described in [96] is used. The theoretical probability of a successful packet transmission by a sensor using DS/CDMA when there are N concurrent transmissions in a one-hop neighborhood can be obtained in [96]. To obtain N, we modify our simulation program to include the effect of AWGN and run the program to obtain the average number of network neighbors, Φ. Φ is also affected by the type of node mobility as discussed in chapter 3. We then assume the worst case scenario where all the network neighbors transmit concurrently. Therefore, N = Φ. Based on: N = 8, the maximum packet length (see chapter 4, figure 4.13a) of 64 bytes and the 16-bit Walsh-Hadamard code, the probability of a successful packet transmission by a sensor is 0.7177.

The second scenario is the harsh underwater scenario discussed in chapter 3. The UWA channel with Rayleigh fading and alpha-stable distributed noise is used. The probability of a successful packet transmission by a sensor using DS/CDMA can be obtained from our work in [see List of Publications: P2, P3]. Using similar modifications, we obtain N = Φ = 5. The smaller number for Φ is due to the harsher communication environment. Based on: N = 5, the maximum packet length of 64 bytes and the 16-bit Walsh-Hadamard code, the probability of a successful packet transmission by a sensor is 0.4592.

The nine hotspot scenarios with the mobile sensors harvesting using our cooperative-orbital algorithm are then run for the two communication scenarios. Our result is shown in figure 5.13. Generally, the performance of our algorithm deteriorates in the non-ideal communication scenarios. In the terrestrial scenario, the global error increases slightly from 6.64% to 8.12%. In the harsh underwater scenario, the global error increases more rapidly from 6.64% to 12.01%. Particularly, the deterioration is most severe in approximately the first 15 iterative steps when the sensors are in the cooperative phase. As there are no communications in the independence phase, the non-ideal communication channels appear to have negligible effect on the performance. All in all, our algorithm is still fairly robust against non-ideal communications. This is due to the beneficial diversity effect when there are more than three network neighbors which help to alleviate the deteriorating effect of the non-ideal communications shown in figure 5.14.

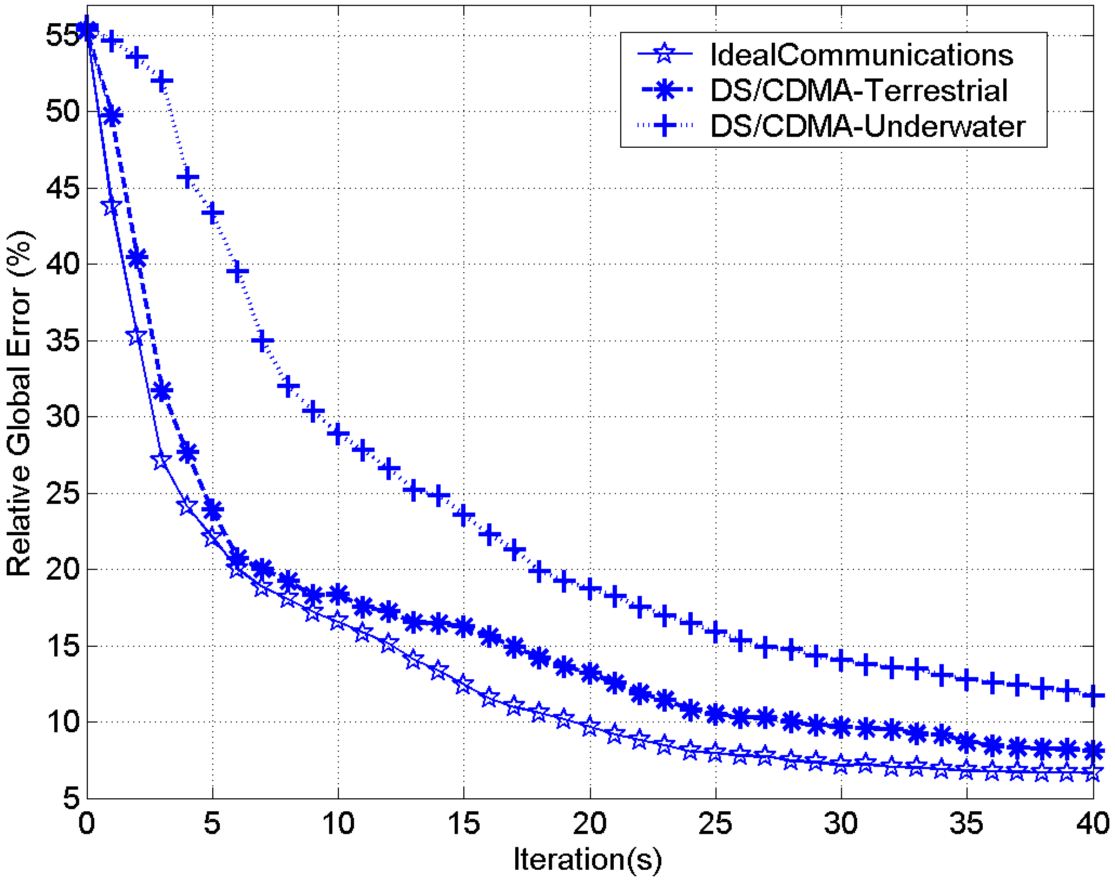


Figure 5.13: Relative global errors for the terrestrial and underwater DS/CDMA communication scenarios

In our algorithm, a sensor stops moving if it cannot receive packets from at least three enclosing network neighbors. We now suppose that in one of the iterative steps during the cooperative phase, sensor 3 in figure 5.14 fails to transmit to sensor *i*.

Sensor *i* is still able to receive the transmissions from the three network neighbors: 1, 2 and 4. As a result, as far as sensor *i* is concerned, sensors: 1, 2 and 4, are its three closest enclosing neighbors. Therefore, it still can continue to harvest data. Furthermore, notice that sensor 4 is nearer to the hotspot than sensor 3 and therefore, has a greater temperature difference w.r.t. sensor *i*. Consequently, in this case, the non-ideal communications actually improve the performance!

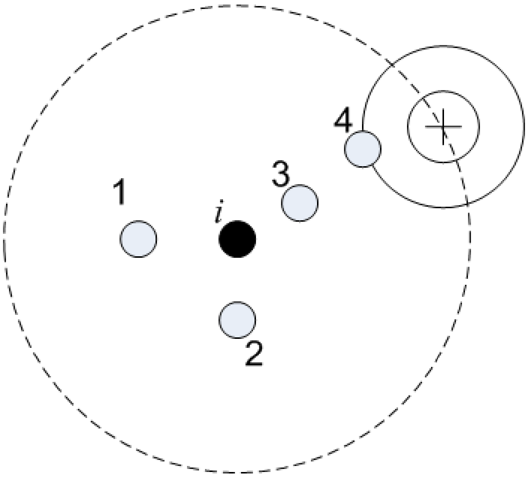


Figure 5.14: Beneficial diversity effect when there are more than three network neighbors

The diversity effect increases when the average number of neighbors increases beyond three. This is because there are a greater number of permutations. This explains the more rapid deterioration in performance in the underwater scenario because the diversity effect is weaker due to the smaller average number of neighbors. Generally, we can increase the average number of neighbors by increasing the transmission power. However, we cannot increase the power indefinitely as it causes MAI in DS/CDMA communications discussed in chapter 3 to increase. An increase in MAI causes the probability of a successful packet transmission to decrease. We can reduce MAI by better synchronizations in the transmissions using base stations. However, this may not be a feasible solution in a harsh underwater environment.

In future work, we can investigate into the use of predictive techniques in our algorithm to predict the likely positions of network neighbors which temporally cannot communicate. This may further improve the performance of our algorithm in non-ideal communications scenarios.

5.4.2 Effect of sensor failures

We now examine the addition of sensor failures on the performance of the cooperative-orbital algorithm. In the setup, at beginning of the simulation, each sensor is randomly failed with a given failure probability. The sensors that have failed will remain so throughout the simulation and no longer participate in the harvestings.

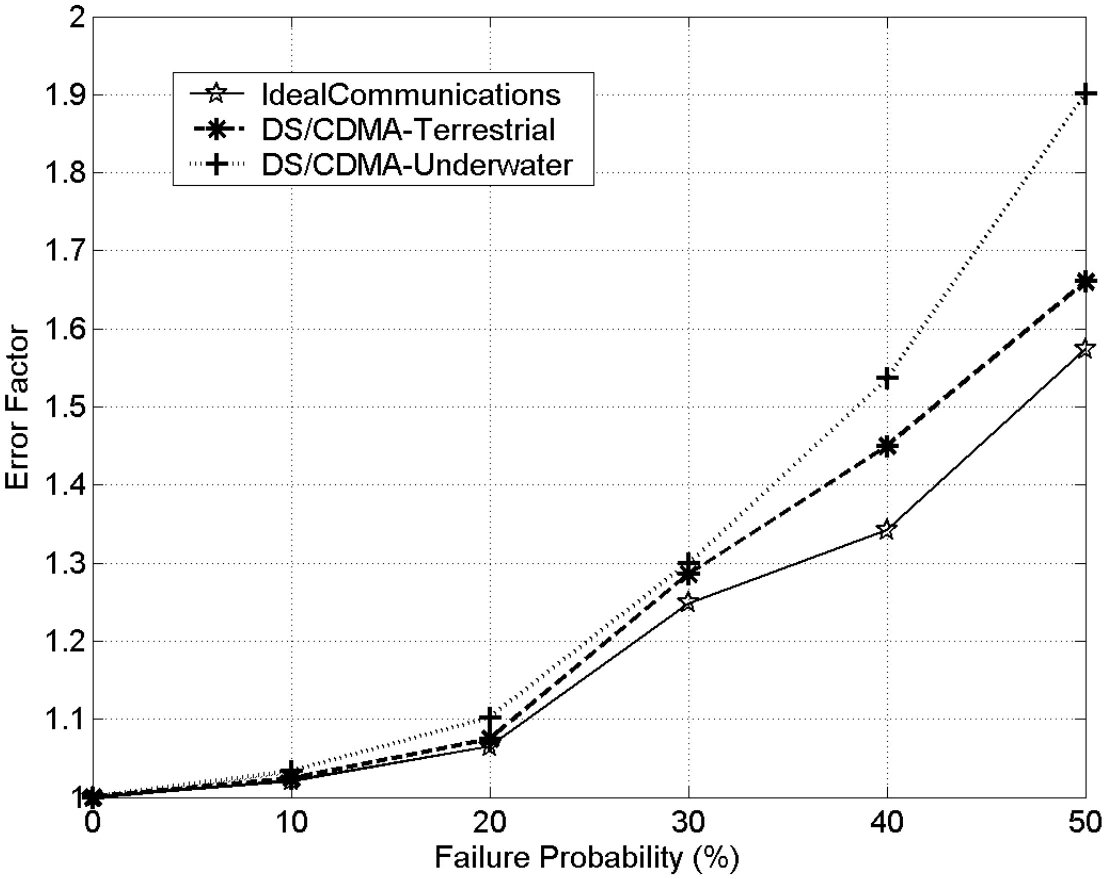


Figure 5.15: Effect of sensor failures on the error reduction performance

The simulation result is shown in figure 5.15 for the three cases: ideal communications, terrestrial with DS/CDMA communications and underwater with

DS/CDMA communications. The error factor for each scenario is obtained by dividing the relative global errors for each failure probability by the relative global error when there are no sensor failures at the 40th iteration. The result shows that there is a gradual degradation of performance as the failure probability increases. Particularly, the degradation increases more rapidly when the failure probability is greater than 20%. The gradual rather than abrupt nature of the degradation shows that the network is robust and has the ability to realign itself and operate continuously without been affected by missing sensors.

5.5 Conclusion

In this chapter, we studied the performance of our cooperative algorithm. In our comparative study, the results show that our cooperative-orbital algorithm is an energy efficient way to optimize the performance of mobile sensor networks because it reduces 738% more error (with a range of 625% to 885%) than mobile sensors deployed in EDG and 35-314% more error than independent methods while consuming 74-81% lesser energy. It is also an attractive environmental friendly and reusable substitute to some static sensor applications because it utilizes resources 250 times more efficiently than static sensors. Furthermore, the high resource utilization suggests that our approach is cost effective enough for us to install sophisticated equipment such as GPS on each mobile sensor to further improve their accuracies and efficiencies. In the stability study, the simulation results show that incorporation of an independence phase helps to improve the robustness of optimization. Also, we observe that the approach of dividing the sensors into small groups leads to more stable cooperative behaviors among the sensors. Our simulation results further show that the tracking mechanism is stable. Finally, we show that when our sensors use the

DS/CDMA MAC protocol to communicate, the performance is robust against non-ideal terrestrial and underwater communication channels and permanent sensor failures.

Chapter 6: Conclusion

To conclude, in our work, we have studied the problem of improving the quality of reconstructed distribution of environmental spatiotemporal data harvested by mobile sensors that can communicate with each other using wireless communications. Our approach is to attack the problem from the source, by mobilizing the sensors to harvest data of high information content so that the reconstructed distribution has minimum distortion. Four realistic constraints are considered in our design: limitations of wireless communications, limited supply of energy, limited sensor resources, and to a lesser extent, difficult terrains. In order for us to use the scarce energy and sensor resources economically, we have leveraged on cooperation to perform optimal harvesting. The possible presence of difficult terrains with barriers and obstacles means it is more feasible to divide our sensors into separate groups.

In our literature survey, we first examined the communication and networking issues by surveying the literature on wireless ad-hoc networking of mobile nodes in harsh environments which is the most relevant to our application. The key finding is that there are four main problems that deteriorate the performance of the networks as follows: poor connectivity due to node mobility and imperfect wireless channels, node contentions, hidden and exposed node problems and excessive routing overheads are generated from broken links. The weakness of maintaining a stable route for multi-hop networking in a harsh environment motivates us to explore alternatives in communications for our cooperative mobile sensor network.

Next, we examined the issues related to monitoring, detection and sensing by surveying work on wireless sensor networks. The key finding is that there are three main problems that deteriorate their performance as follows: poor connectivity due to node mobility and imperfect wireless channels, network congestions and high localization errors. The survey also showed that there are two environmental-unfriendly weaknesses in static WSN: poor reusability and the need to be deployed in large number. As environmental issues are great concerns of our times, we are motivated to survey mobile sensors as suitable substitutes.

Our survey on coverage control of mobile sensor networks showed that there is an increasing trend to adopt the behavioral based paradigm which is biologically inspired. In this approach, each individual sensor adopts a cooperative, self-regulating and adaptive behavior that leads to orderly group movements. The advantages are scalability and ability to respond rapidly in group movements. Our three key findings are: coverage control results in equal spacing of mobile sensors over the terrain, the targets that the sensors monitor are assumed to be single discrete entities and sophisticated centralized control schemes are not scalable with the number of sensors. For our application, in order to maximize the quality of measurements, the sensors are likely to be spread out unevenly. Specifically, the sensors will cluster more tightly in regions where the temperature gradients are steep, and vice versa. Furthermore, we are monitoring continuous distributions that we do not have advance knowledge of. Therefore, we are motivated to develop a new scalable approach while embracing the behavioral based paradigm in the design of our cooperative algorithm.

In our theoretical analysis on communications, we examined three problems. The first problem is poor connectivity due to node mobility and fading channel conditions. When poor connectivity is due to independent random node mobility, the solution is to ensure that there is coordinated orderly movement. For the poor fading channel condition, one approach is to use DS/CDMA which is known to be robust to fading channels. Finally, to further improve the robustness of our algorithm to the connectivity problem, our algorithm makes use of only three closest neighbors out of the other network neighbors. The second problem is contentions and interference. One approach is to organize the sensors into small cooperative groups and design our algorithm in such a way as to minimize inter-group communications. Another approach is to use DS/CDMA which is known to be robust to interference in the MAC layer. The third problem is route breakages in networking resulting in increased routing overheads that may lead to network congestion. We concluded that it is better to organize our sensors into small isolated cooperative groups and use intra-group communications to facilitate the harvesting tasks. The routing overheads can be further minimized if most communications are one-hop communications. We also analyzed the throughput of an underwater multi-hop ad-hoc sensor network using DS/CDMA in UWA channel with Rayleigh fading and alpha-stable distributed noise.

In the development of our cooperative optimal harvesting algorithm, we assumed each mobile sensor to be an intelligent cooperative autonomous agent, capable of processing cooperative shared information independently in order to carry out its harvesting task in an optimal manner. The optimization consists of two phases: cooperative and independence phases. In the cooperative phase, our novel approach is to introduce cooperation into the pseudo-Newton method and adapt the method to propel the sensors rapidly into the optimal positions. In the independence phase, we developed two alternative versions of independent heuristics: delta and orbital-harvesting. Additionally, we incorporated a tracking mechanism so that our mobile sensors can track moving hotpots.

In the development of our algorithm, we have adopted: adaptive cooperative search goals in optimization, local RBF interpolation in estimations, dissemination to mitigate the initial value problem, the concept of orientation stabilization to provide adaptive stabilized search direction and the principle of dynamic clustering in mobile sensor networks for robust tracking of continuous distribution.

Our performance study is divided into two aspects: comparative study (with other algorithms) and stability study. Additionally, we also examined the effect of non-ideal communications of the performance.

In the comparative study, the simulation results showed that our cooperative-orbital algorithm is an energy efficient way to optimize the performance of mobile sensor networks because it reduces 738% more error (with a range of 625% to 885%) than mobile sensors deployed in EDG and 35-314% more error than independent methods such as: BFGS, RWM and Independent delta-harvesting while consuming 74-81% lesser energy. It is also an attractive environmental friendly and reusable substitute to some static sensor applications because it utilizes resources 250 times more efficiently than static sensors. Furthermore, the high resource utilization suggests that that our approach is cost effective enough for us to install sophisticated equipment such as GPS on each mobile sensor to further improve their accuracies and efficiencies.

In the stability study, the simulation results showed that our cooperative-orbital algorithm is most robust compared to other methods. Therefore, incorporation of an independence phase helps to improve the robustness of optimization. Furthermore, we observed that our approach of dividing the sensors into small groups for ease of deployment in difficult terrains and robustness in harsh communication environments has one additional benefit. It leads to more stable cooperative behaviors among the sensors. Therefore, for these three reasons, the recommended approach is to deploy the sensors into small self-contained cooperative groups for data harvesting. Moreover, the simulation results showed that the tracking mechanism of our algorithm is stable.

Finally, we showed that when our sensors use the DS/CDMA MAC protocol to communicate, the performance is robust against non-ideal terrestrial and underwater communication channels. It is also robust against sensor failures.

6.1 Future Work

There remain many areas that we can explore further in this research as follows.

First, we can investigate ways to improve the communication, control and information processing mechanisms of our algorithm in the following areas:

1. Bandwidth and energy efficient approaches to disseminate information can be explored.
2. Approaches in communications to facilitate the harvesting task can also be explored.
3. Distributed control methods to stabilize the cooperative groups while simultaneously increase the responsiveness of the groups can be explored.
4. Predictive methods to further improve on the robustness of the sensors in harsh communication environment.
5. Since in reality, we may have more than three network neighbors, we can explore the use of cooperative techniques to process the information in order to improve its reliability. If heterogeneous data are available, we can even use data fusion techniques.

Second, alternative methods to optimize data harvesting such as non-computational optimization can be explored. We can even draw inspirations from nature or agriculture.

Third, it will be interesting to investigate other ways or combinations to organize the sensor groups so as to make them more robust and efficient in the harvesting tasks. This may lead to novel ways to control the mobile sensors.

Fourth, some issues need further theoretical analysis. One possible issue will be the optimal number of sensors to be deployed in a terrain of a particular size and the factors that affect this optimal number. Another possible issue is optimal number of sensors per group if we are to divide our sensors into several small groups. Some empirical investigations will likely be required in order to provide empirical data and highlight practical problems. This is to ensure that the theoretical analysis is as realistic as possible. Another possible theoretical analysis is the global convergence analysis that takes into account of different factors such as: link failures, sensor noises and random distributions of system parameters.

Fifth, we acknowledge that it is not possible to explore all issues given the limited time and that our research spans several areas. For example, further investigations to examine the data throughput and energy consumption in communications may be needed by implementing our harvesting algorithm into a simulation program that simulates actual wireless sensor hardware specifications and networking protocols. In actual implementations, there will be other practical issues depending on the applications and environments. An example will be further investigations into the interaction between the lower layers and application layer.

APPENDICES

Appendix A: CSMA/CA Throughput Analysis of a MANET of cooperative autonomous mobile agents under the Rayleigh Fading Channel

A.1 Theoretical Model

CSMA/CA protocol adopted in IEEE 802.11 MAC is distinguished by two salient features: 4-way handshaking and the exponential backoff algorithm. Their details can be found in [75][27][86]. In particular, [75] provides the description on the use of 4-way handshaking to mitigate the hidden node problem. Briefly, the operation of 4-way handshaking is shown in Figure A1. The abbreviations are listed in Table A1.

In Figure A1, the channel is first reserved using the RTS-CTS packet exchanges between the transmitter and receiver. Failed transmissions occurred when there is a collision of RTS packets when transmitting nodes contend to reserve the channel. The contending nodes realize that there is a collision when there is no CTS packet after the maximum waiting duration. The vulnerable period for collisions is shown in Figure A1. When a collision is detected, each node executes the exponential backoff algorithm [75][86]. The algorithm determines a random backoff period in time slots that the node must wait before it can retransmit its data packet again. Prior to our analysis, we need the throughput of a slotted CSMA/CA protocol using the exponential backoff algorithm for a 1-hop network with *M* nodes having poisson packet arrival rates with mean, *λ*. This scenario is well analyzed and the result is obtained [75] as:

$$S(M,\lambda)=\sum_{i=0}^{M}\pi_i(M,\lambda)P_s(i,\lambda)T_p \Bigg/ \sum_{i=0}^{M}\pi_i(M,\lambda)\{\bar{I}_i+1+P_s(i,\lambda)T+[1-P_s(i,\lambda)]C\} \qquad \text{(A1)}$$

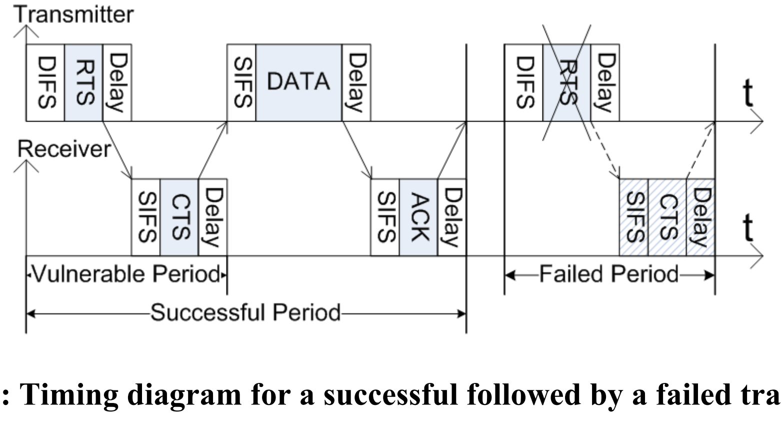


Figure A1: Timing diagram for a successful followed by a failed transmission

Note that in [75], the notation *S*(*M*) is used instead of *S*(*M,λ*). We add *λ* to emphasize that the throughput is also dependent on the traffic load. $\pi_i(M)$ is the stationary probability distribution of the backlogged nodes given that there are *M* nodes. A node is backlogged if it has a packet to retransmit after a collision. *i* is the state of the Markov chain which represents the number of backlogged nodes in the neighborhood. $P_s(i)$ is the probability of successful packet transmissions given that there are i backlogged nodes. $\bar{I}_i$ is the average idle period in the channel given that there are i backlogged nodes. T and C are the periods of successful and failed transmissions.

Table A1: Abbreviation in the timing diagram

	Description
DIFS	DCF Inter Frame Space. Listening time prior to transmitting a RTS packet.
RTS	Request-To-Send packet. Used by transmitter to reserve the channel.
Delay	Propagation and synchronization delay.
SIFS	Short Inter Frame Space. Listening time prior to transmitting CTS, DATA or ACK.
CTS	Clear-To-Send packet. Used by receiver to indicate which node can send its data.
DATA	Data packet from the transmitter.
ACK	Acknowledgement packet from the receiver to indicate successful receipt.

The time-varying connectivity in the network is due to node mobility and fading channel. In [P1], the node mobility model based on the Probabilistic Mobility Model (PMM) and Rayleigh fading channel radio model are used to derive the connectivity probability. The result in [P1] also considered the free space transmission loss. The result for the pair-wise link connectivity probability, c_{ab} in [P1] denoted as c_{link} is given as follows:

$$\begin{aligned} c_{link} &= E\{E[\Pr(R_{ab} \le \eta | b,\eta)]\} = E[E(c_{ab|b,\eta})] = \\ &= \sum_{y_b=0}^{S} \sum_{x_b=0}^{S} \pi_{x_b} \pi_{y_b} \left[\sum_{y_a=0}^{S} \sum_{x_a=0}^{S} \exp[-R_{ab}^2 \big/ (2\sigma^2 \eta_0^2)] \pi_{x_a} \pi_{y_a} \right] \end{aligned} \quad \text{(A2)}$$

where, R_{ab} is the Euclidean separation of two arbitrary nodes, a and b and η is the transmission range. In a fading channel, η is a random variable that is dependent on the fading statistic. $\Pr(R_{ab} \le \eta | b,\eta)$ is the pair-wise connectivity probability conditioned on the position of node b and η also denoted as $c_{ab|b,\eta}$. The discrete nature of the expectations is due to the discrete nature of PMM. At each time step, the node can either move to one of the adjacent north, south, east, west, northeast, northwest, southeast, and southwest positions by one discrete step or remain in the same position. The node movement at each time step is given by a constant transition probability matrix. As a result, the position probability of the node in the enclosed region at steady state is stationary. The result is based on node mobility in an enclosed square region. For analysis purpose, without loss of generality, the discrete step is normalized to 1. The length of the square region is then expressed as S number of discrete steps. $\pi_{x_a}\pi_{y_a}$ and $\pi_{x_b}\pi_{y_b}$ are the stationary position probabilities of node a and b at coordinates (x_a, y_a) and (x_b, y_b) respectively given in [P1].

An important quantity is the average number of temporal neighbors in a time-varying connectivity environment, $\overline{\Phi}$. Temporal neighbors are neighbors at one discrete time step. Let N_s be the node count in the square region and $\Phi_{ab|b}$ be the conditional random variable representing the number of temporal neighbors of node *b*. The probability distribution and conditional expectation of $\Phi_{ab|b}$ and $\overline{\Phi}$ are derived as:

$$\Pr(\Phi_{ab|b} = \phi) = \binom{N_s - 1}{\phi} c_{ab|b,\eta}^{\phi} (1 - c_{ab|b,\eta})^{N_s - \phi - 1} \quad 0 \le \phi < N_s$$

$$E(\Phi_{ab|b}) = (N_s - 1) c_{ab|b,\eta} \tag{A3}$$

$$\overline{\Phi} = E[E(\Phi_{ab|b})] + 1 = (N_s - 1) E(c_{ab|b,\eta}) + 1 = (N_s - 1) c_{link} + 1$$

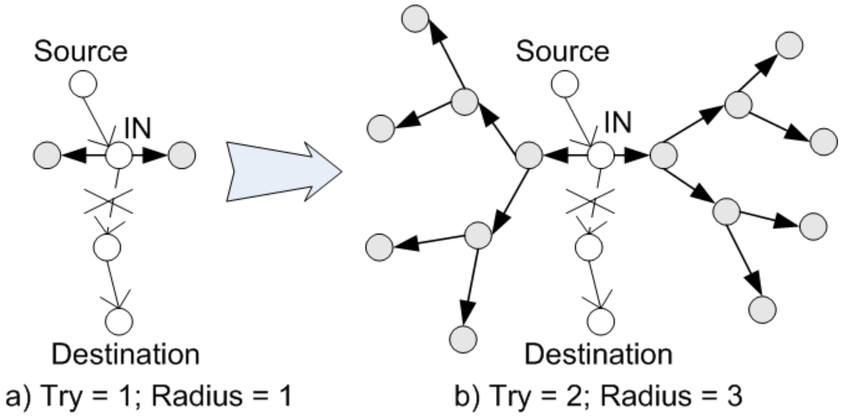


Figure A2: Expanding ring search for the first two tries.

AODV routing protocol [78] is a reactive routing protocol which builds routing tables on demand adopted by the Mobile Ad Hoc Networking Working Group of the Internet Engineering Task Force (IETF). There are two types of behavior which give rise to routing overheads. First, when either there is no route to a destination or the routes in the routing table expire, the routing protocol initiates a new route search. This behavior dominates at transient state of network where many route tables are empty. It also dominates at low traffic condition where the mean packet inter-arrival time is long and the routes in the table expired. At steady state, with moderate to high traffic conditions, whenever there are unrecoverable link breakages due to disconnections, the routing protocol initiates local repairs which dominate the overheads. Local repairs use an expanding ring broadcast search. It is illustrated in Figure A2 for first two tries. When a route is found, it is unicast back. The default setting for the search is three tries with increasing ring radii in numbers of hop: 1, 3 and 5. Referring to Figure A2, intermediate node IN which detects the breakage at downstream always initiates the search at each try. Also the broadcast is in the outward direction because nodes that receive multiple copies of the same search packet suppress sending them.

In our approximate model, we assume that the network is at steady state and only local repairs are initiated by the routing protocol. This assumption does not hold under severe disconnections where there are new route searches. Each node is equally likely to initiate a local repair when it detects a disconnection with disconnection probability given by $(1-c_{link})$. We ignore the cost of unicast packets as it is negligible compares to that of broadcast packets. Our cost is represented by λ_{route}. It is the total AODV route traffic load generated per node per second. λ_{route} is normalized to the data packet transmission time. It is derived as:

$$\begin{aligned}
&\lambda_{route} \approx (2^{N_{hop}} - 1) r \textstyle\sum_{i=0}^{2} \sum_{j=0}^{i} I(\omega_{j-1}, \omega_j) \omega_j c_{link} (1 - c_{link})^j \\
&N_{hop} \approx 0.5(1 + \sqrt{2} S \eta_0^{-1}) \qquad\qquad (\text{A4}) \\
&\omega_j \approx \min(\overline{\Phi}^{2j+1} \big/ 2^{2j}, N_s), \omega_{-1} = 0
\end{aligned}$$

N_{hop} is the estimated average number of hops per route. η_0 is the nominal transmission range when there is no Rayleigh fading defined in [P1]; it is dependent on the transmission loss only. $(2^{N_{hop}} - 1)$ is the count of all combinations of multi-hop routes from 1 to N_{hop} hops. ω_j is the number of local repair packets generated when ring radius is $(2j+1)$ hops in current try. r is the ratio of routing to data packet transmission time; it is use to normalize λ_{route}. $I(a,b)$ is the indicator function; it is 1 if $(a \neq b)$, 0 otherwise. $\min(a,b)$ is the minimum function that returns the minimum of $\{a,b\}$.

A.2 Derivation of Data Throughput per Node

In this section, we initially develop the throughput in an ideal ad-hoc network where the 1-hop neighborhoods are isolated. This allows us to isolate the effect of time-varying connectivity. We then consider the case where the neighborhoods are overlapped in which we examine the hidden and exposed node problems. Finally, we present the data throughput which accounts for the additional overheads in multi-hop communications.

One of the benefits of multi-hop communications is the spatial diversity gain in capacity. First, we examine the upper limit of this gain by using an ideal ad-hoc network. In Figure A3, we reduce the transmission ranges of all the nodes such that a 1-hop network is reduced into an ideal ad-hoc network where there are four isolated neighborhoods. Furthermore, we assume that the nodes are stationary and the channel is non-fading. Finally, there are only intra-neighborhood communications. Clearly, under these ideal assumptions, the spatial diversity gain is at maximum. In fact, it is observed in [69] from info-theoretic perspective that for a stationary multi-hop network, if there are more intra than inter-neighborhood communications, the capacity can be increased further. We derive the maximum gain, $G_{S,max}$ by looking at the throughputs per node for the 1-hop and ideal ad-hoc network: S_{node}^{1}, S_{node}^{ideal}. Let Φ be the average number of neighbor in a 1-hop neighborhood of the ideal ad-hoc network. $G_{S,max}$ is derived using equation (A1) as:

$$\begin{aligned}
&G_{S,\max} = S_{node}^{ideal} \big/ S_{node}^{1} = [S(\Phi, \lambda)/\Phi]/[S(N_S, \lambda)/N_S] \approx N_S / \Phi \\
&(\because \max S(\Phi, \lambda) \approx \max S(N_S, \lambda) \approx 0.8) \qquad\qquad (\text{A5})
\end{aligned}$$

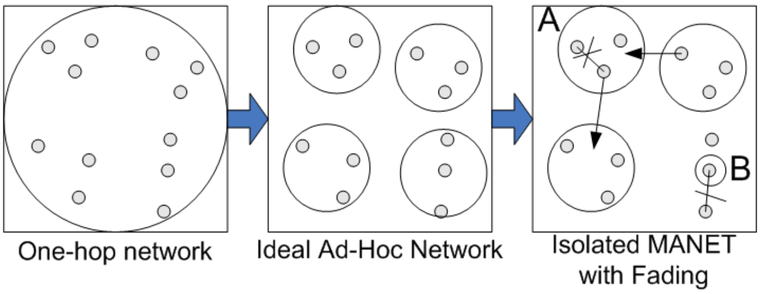


Figure A3: Evolution of ideal ad-hoc network

We now remove the stationary nodes and ideal channel assumptions by considering Probabilistic Mobility Model (PMM) and Rayleigh fading channel. However we assume again that there are only intra-neighborhood communications and the neighborhoods are isolated when communications take place. The stationary nature of $\overline{\Phi}$ in equation (A3) implies that at steady state, the number of nodes leaving and entering a neighborhood is the same. In Figure A3, in neighborhood A, there is a lost communications when a node leaves the neighborhood A. In neighborhood B, the center node loses its communications to the neighbor because of the diminishing transmission range due to fading. Packet transmissions are successful between a pair of nodes in a neighborhood only when they are temporally connected. We can now derive the throughput per node, *S_{node}* under this scenario. Using equation (A3), it is derived as:

$$\begin{aligned} S_{node} &= c_{link} S(\overline{\Phi},\lambda) / \overline{\Phi} + (1-c_{link}) \cdot 0 \\ &= c_{link} S(\overline{\Phi},\lambda) / [(N_s - 1)c_{link} + 1] \approx S(\overline{\Phi},\lambda) / N_S \end{aligned} \tag{A6}$$

Substituting equation (A6) into (A5), the maximum gain under time-varying connectivity is $G'_{S,max}$≈1! The reason for the loss of spatial diversity gain is due to the node mobility and fading. While the average number of neighbors is the same in the temporal neighborhood, the neighbors are always changing due to movement of nodes in and out of the neighborhood. Hence, there are disruptions to the communications. This is exacerbated by the fading fluctuations. This is different from the stationary case where nodes can be assured of communicating to the same neighbors without disruptions. Nevertheless, we will verify our analysis by simulation. On the positive note, there is still much to gain from ad-hoc communications because of improvement in delay. This is because the number of nodes contending for the channel in the smaller neighborhood is reduced. Therefore, the mean waiting time per node before it gets to use the channel is reduced also by the original factor, $G_{S,max}$ (equation (A5)).

Overlapping neighborhoods give rise to two known problems when CSMA/CA protocol is used: hidden and exposed nodes. For clarity of presentation, we assume again that there are only intra-neighborhood (1-hop) communications. Note that such interference is a feature of communications in an environment with overlapping neighborhoods. It is present even when there is no multi-hop communications. We define an interfering neighborhood as the adjacent neighborhood of the typical neighborhood and they overlap each other. The typical neighborhood is defined in [75]. We will adopt the approach of using a typical neighborhood in our analysis and test the validity of this approach through simulation. In Figure A4, we see that a typical neighborhood (shaded circle) is wholly covered by at most three independent interfering neighborhoods. In Figure A5, we separate the interfering neighborhood into two disjoint areas: the overlapping exposed node area (horizontal stripes) which it shares with the typical neighborhood and the complementary hidden node area (vertical stripes). Briefly, a hidden node problem occurs (see Figure A5) whenever node *b* tries to transmit to node *a* and node *c* from the hidden node area also transmits. Hence, there is a collision at node *a*. An exposed node problem occurs when node *a* wants to transmit to node *b* and is blocked from transmission because of on-going transmission in node *c*. Node *a* may be exposed to up to three independent regions.

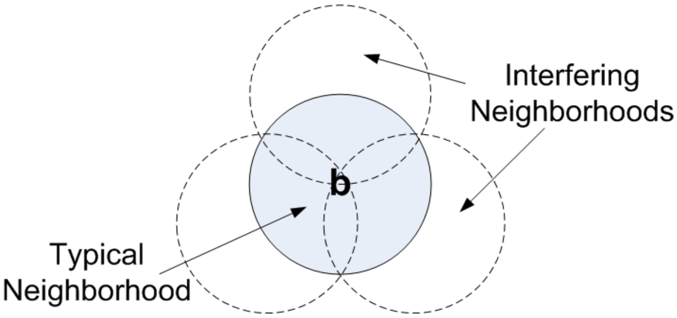


Figure A4: Maximum independent interfering neighborhoods covering a typical neighborhood.

First, we evaluate the average numbers of exposed and hidden nodes in one interfering neighborhood. In Figure A5, node *b* is centered at the typical neighborhood. Therefore, we first evaluate the number of exposed nodes. Let η_{xy} be the pair-wise transmission range of node *x* transmitting to node *y* and R_{xy} be pair-wise separation of node *x* and *y*. Let p_{exp} be the probability that an arbitrary node *a* is an exposed node.

$$\begin{aligned} p_{\exp} &= \Pr\{(R_{ab} \le \eta_{ab}) \cap (R_{ca} \le \eta_{ca}) \cap (R_{cb} > \eta_{cb})\} \\ &= \Pr\{(R_{ab} \le \eta_{ab}) \cap (R_{ca} \le \eta_{ca})\} - \\ &\Pr\{(R_{ab} \le \eta_{ab}) \cap (R_{ca} \le \eta_{ca}) \cap (R_{cb} \le \eta_{cb})\} = c_{link}^2 - c_{abc} \end{aligned} \qquad \text{(A7)}$$

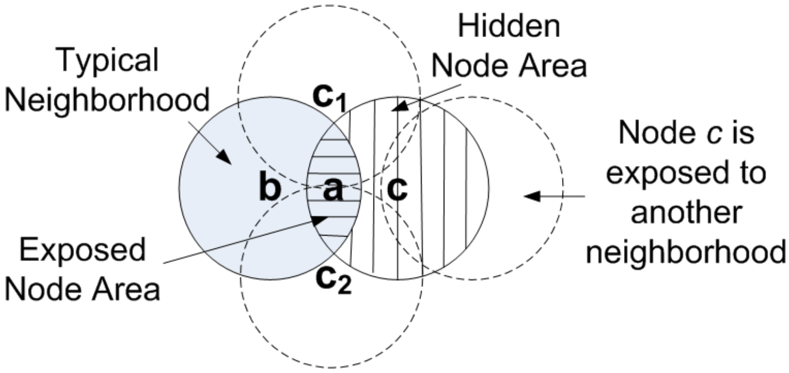


Figure A5: Exposed and hidden node areas

The first term is simplified by noting that the joint probability is independent as they are not transmitted to the same node. The second term, c_{abc} represents the probability that node *a* is the neighbor of node *b* and *c* and simultaneously, node *b* and *c* are also neighbors of each other. We first derive the c_{abc} conditioned on the transmission ranges: η_{ab}, η_{cb} and η_{ca} using the approach of [P1] as follows:

$$c_{abc}(\eta_{ab},\eta_{cb},\eta_{ca}) = \sum\nolimits_{y_c=0}^{S}\sum\nolimits_{x_c=0}^{S}\pi_{x_c}\pi_{y_c}\sum\nolimits_{y_a=0}^{S}\sum\nolimits_{x_a=0}^{S}\pi_{x_a}\pi_{y_a} \\ \cdot\,\Theta(R_{ca},\eta_{ca})\sum\nolimits_{y_b=0}^{S}\sum\nolimits_{x_b=0}^{S}\pi_{x_b}\pi_{y_b}\Theta(R_{ab},\eta_{ab})\Theta(R_{cb},\eta_{cb}) \qquad \text{(A8)}$$

where $\Theta(R_{xy},\eta)$ is the indicator function; it is 1 if ($R_{xy} \le \eta$), 0 otherwise. We will take the expectation of $c_{abc}(\eta_{ab}, \eta_{cb}, \eta_{ca})$ over η_{ab}, η_{cb} and η_{ca}. However, two of the random variables are dependent.

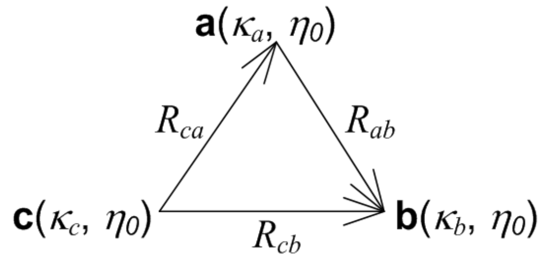


Figure A6: Relationship among fading gains, free space transmission ranges and separations of node *a*, *b* and *c*.

In Figure A6, node *a*, *b* and *c* experience independent and identically distributed (i.d.d.) fading gains: κ_a, κ_b and κ_c respectively. As all the nodes have the same transmission powers, the free space transmission ranges, η_0 are the same. Both node *a* and *c* are directed towards the node *b*. Node *b* is connected to node *a* and *c* if:

$$\Pr[(\eta_{ab} > R_{ab})\cap(\eta_{cb} > R_{cb})] = \\ \Pr[(\kappa_b\eta_0 > R_{ab})\cap(\kappa_b\eta_0 > R_{cb})] = \Pr[\kappa_b\eta_0 > \max(R_{ab}, R_{cb})]\,.$$

Taking the expectation of equation (A8), we have,

$$c_{abc} = \sum\nolimits_{y_c=0}^{S}\sum\nolimits_{x_c=0}^{S}\pi_{x_c}\pi_{y_c}\sum\nolimits_{y_a=0}^{S}\sum\nolimits_{x_a=0}^{S}\pi_{x_a}\pi_{y_a}\ \Pr(\eta_{ca} > R_{ca}) \\ \sum\nolimits_{y_b=0}^{S}\sum\nolimits_{x_b=0}^{S}\pi_{x_b}\pi_{y_b}\ \Pr[(\eta_{ab} > R_{ab})\cap(\eta_{cb} > R_{cb})] \Leftrightarrow$$

$$\therefore c_{abc} = \sum_{y_c=0}^{S}\sum_{x_c=0}^{S}\pi_{x_c}\pi_{x_c}\sum_{y_a=0}^{S}\sum_{x_a=0}^{S}\pi_{x_a}\pi_{x_a}\ \exp[-R_{ca}^2\big/(2\sigma^2\eta_0^2)] \\ \sum_{y_b=0}^{S}\sum_{x_b=0}^{S}\pi_{x_b}\pi_{x_b}\ \exp[-\max(R_{ab}^2, R_{cb}^2)\big/(2\sigma^2\eta_0^2)] \qquad \text{(A9)}$$

Let $\widetilde{E}_n$ be the random variable representing the number of exposed nodes. $\widetilde{E}_n$ is binomial when conditioned over the positions of node *b* and *c* with the binomial probability, p_{exp}. Hence using the similar approach for the evaluation of average number of neighbors in equation (A3), the average numbers of exposed nodes is derived. The average numbers of exposed and hidden nodes in an interfering neighborhood: E_n and H_n are derived as:

$$E_n = (N_s - 2)p_{\exp}; \quad H_n = \overline{\Phi} - E_n \tag{A10}$$

The exposed nodes in the typical neighborhood have two effects. First, as the transmissions of exposed nodes are suppressed by hidden nodes outside the typical neighborhood, the throughput of the typical neighborhood is reduced. Second, due to fewer neighbors contending for the channel, the throughput is increased. These two effects can be characterized by estimating the effective number of neighbors due to exposed node problem, $\overline{\Phi}_{eff}$. To compute $\overline{\Phi}_{eff}$ we need to estimate the probability that a node is idle in one time slot. The average idle probability, p_{idle}, is derived as follows:

$$p_{idle} = \sum_{i=0}^{\overline{\Phi}} \pi_i(\overline{\Phi})(1-g)^{\overline{\Phi}-i}(1-v_i)^i \tag{A11}$$

where, $\pi_i(\overline{\Phi})$ is known as the stationary probability distribution of the backlogged nodes given that there are $\overline{\Phi}$ nodes. g is the probability of a new packet arrival in a time slot. It is a function of the traffic load. v_i is the retransmission probability of a backlogged node in a time slot given that there is currently i backlogged nodes in the neighborhood. v_i is dependent on the type of retransmission policy which is, in our case, the exponential backoff. The formula for $\pi_i(\overline{\Phi})$, g and v_i are obtained in [75]. The effective number of neighbors $\overline{\Phi}_{eff}$ is estimated as:

$$\overline{\Phi}_{eff} \approx (\overline{\Phi} - E_n)(1 - p_{idle}^{3H_n}) + \overline{\Phi} p_{idle}^{3H_n} \tag{A12}$$

Hidden node problem is investigated in [75]. Briefly, in [75], a uniformly distributed stationary network is used. The hidden node problem is then accounted for by examining the probability that a node in the hidden node area from one interfering neighborhood will transmit to a typical neighborhood, p_{ht}. It is assumed that transmissions of hidden nodes from one interfering neighborhood can collide with all nodes in the typical neighborhood. Referring to Figure A4 and A5, their assumptions appear too conservative as hidden nodes from one interfering neighborhood can only interfere in their respective exposed node area and three independent interfering neighborhoods are required to cover a typical neighborhood. Another observation made in Figure A5 is, while node c is a hidden node, it is itself exposed to another neighborhood. This suggests that the probability of hidden node transmissions is much lower. Due to the differences with [75], we extend the work of [75] by re-deriving p_{ht} for our scenario. In Figure A5, a collision occurs if node c transmits while node a is receiving packets from node b. The vulnerable period (see Figure A1), T_h in units of time slot, is a period when the hidden node transmissions will cause a collision. The moment when there is a hidden node transmission in one of the time slots, the other hidden nodes will not transmit in the subsequent time slots. This is because they will sense the channel to be busy in the subsequent slots. Let $p_{ht,eff}$ be the effective hidden node transmission probability over the typical neighborhood. p_{ht} and $p_{ht,eff}$ are derived as:

$$p_{ht} \approx T_h \delta_h (1-\delta_h)^{T_h-1} / 3; \quad p_{ht,eff} = 1-(1-p_{ht})^3 \tag{A13}$$

where, δ_h is the probability that at least one hidden node transmits data in one time slot and is derived in [75].

There are two effects of multi-hop communications which reduce the data throughput per node. First, multi-hop communications require multi-hop routes to be setup by routing protocols. This incurs overheads from route packets generated. Route packets generated consume a portion of the link capacity. Second, each node is obliged to carry the traffic of others in addition to its own data load. Hence, the effective traffic load per node, λ_{eff} is higher than the actual data traffic load arriving at each node, λ_{data}. Therefore, these two effects tend to saturate the throughput more rapidly.

The network traffic consists of superposition of different traffic streams. Therefore, at any instance of time, the nodes play three roles: as source, intermediate and destination nodes simultaneously for different traffic streams. The nodes are assumed to be identically distributed in our analysis. The 1-hop link traffic per node, λ_{link} is derived by examining the traffic carried for two cases: when there is no link disconnection and when there is link disconnection during one data packet transmission derive as:

$$\lambda_{link} = \lambda_{data}\left[(\lambda_{route}+1)(1-c_{link})+c_{link}\right] \tag{A14}$$

The ambient traffic λ_{amb} beyond 1-hop neighborhood that a node is obliged to carry is computed by aggregating the link traffic λ_{link} over the h-hop neighborhood. As h increases, the traffic is attenuated by the factor, c_{link}^{h}. The number of nodes in a h-hop neighborhood, $\Phi_{est,h}$ is estimated by using similar figures ratio. Therefore,

$$\begin{aligned} \lambda_{amb} &= \lambda_{link} \textstyle\sum_{h=2}^{N_{hop}} (\Phi_{est,h} - \Phi_{est,h-1}) c_{link}^{h} \\ \Phi_{est,h} &= \min(h^2 \overline{\Phi}, N_s) \end{aligned} \tag{A15}$$

The effective traffic load per node, λ_{eff} is derived as:

$$\lambda_{eff} = \lambda_{link} + \lambda_{amb} \tag{A16}$$

By analyzing the typical neighborhood, the total throughput per node, S_{mhop} is derived by combining all the factors affecting it, based on discussions from section IV-A to IV-C, using equation (A1), (A6), (A12), (A13) and (A16).

$$S_{mhop} = \overline{\Phi}_{eff}^{-1} c_{link} (1-p_{ht,eff}) S(\overline{\Phi}_{eff}, \lambda_{eff}); \quad g = T_{slot} \lambda_{eff} \tag{A17}$$

Notice, that we have replaced M, λ by $\overline{\Phi}_{eff}, \lambda_{eff}$ in the original expression for $S(\cdot)$ in equation (A1). λ_{eff} affects $S(\cdot)$ through g which is the probability of a new packet arrival in a time slot defined in section IV-B. T_{slot} is the time of one slot in seconds. Finally the data throughput per node, S_{mhop}^{data} is derived from the proportionality of traffics as:

$$S_{mhop}^{data} = (\lambda_{data} / r\lambda_{link})S_{mhop} \tag{A18}$$

Appendix B: Derivation of the Motion Control Equations for One-Dimensional Topology

In a control problem, we want to control and adjust the behavior of a system so that the behavior moves towards one that can attain a desired outcome or system goal. The principle requires that the selected behavior is controllable and observable. In our context, the system is the individual mobile sensor and the behavior of the mobile sensor that we desire to control is the position of the sensor. We will first define our cooperative control problem from control-theoretic perspective using the difference equation since our control is performed at discrete time step instead of continuous time. The general expression for our cooperative control problem is defined as follows:

$$p_i^{(k+1)} = p_i^{(k)} + \Delta p_i^{(k)}\left(C_{sn}^{(k)}\right) \qquad \text{(B1)}$$

Where:

$p_i^{(k)}$	The position of sensor *i* in the k^{th} time step.
$\theta_i^{(k)}$	The measurement made by sensor *i* in position $p_i^{(k)}$ in the k^{th} time step.
$s_i^{(k)}$	The state vector of sensor *i* in the k^{th} time step. It is defined as the concatenation of $p_i^{(k)}$ and $\theta_i^{(k)}$. $\therefore s_i^{(k)} = \left\lfloor p_i^{(k)}, \theta_i^{(k)} \right\rfloor$.
$C_{sn}^{(k)}$	The set that represents the states of the sensors belonging to the same cooperative group in the k^{th} time step.
$\Delta p_i^{(k)}\left(C_{sn}^{(k)}\right)$	This is the position control function in the k^{th} time step. It takes $C_{sn}^{(k)}$ as the input and computes the amount of adjustment to be added to the current position $p_i^{(k)}$ in order to obtain the next position.
$V_{i,sn}^{(k)}$	The set that represents the states of the Voronoi neighbors of sensor *i* in the k^{th} time step, exclusive of sensor *i*.

The control equation describes the dependency of the position of sensor *i* with the states of all the sensors within the same cooperative group of sensor *i*. It can be implemented in a distributive manner by having each sensor performs the computation in equation (B1) based on information gathered from its cooperative group. Therefore, there is no central authority that gathers information on behalf of the group and assigns the set of next positions for the mobile sensors in the cooperative group.

At one end of the spectrum, we observe that for autonomous control, $C_{sn}^{(k)} = \left\{ s_i^{(k)} \right\}$. That is, in autonomous control, each sensor adjusts its future behavior purely on what it observes alone. This is different from a cooperative control problem in which there is sharing of state information within the cooperative group. In doing so, we achieve a higher level of efficiency and effectiveness in control. At the other end of the spectrum, the cooperative group encompasses the whole sensor network. However, the dissemination of global state information in pure ad-hoc multi-hop network is very costly and non-scalable. Therefore, it is wiser for us to examine the smallest possible communications group to see whether we can still solve our control problem with such a group. In fact, we show in the subsequent sections that the smallest group that is required for us to achieve our control goal is $C_{sn}^{(k)} = V_{i,sn}^{(k)} \cup \left\{ s_i^{(k)} \right\}$.

Finally, before we move on, it is instructive for us to examine the basic structure of the simplest form of equation (B1). We look at the structure of the simplest linear autonomous control equation:

$$p_i^{(k+1)} = p_i^{(k)} + K_u \left\{ u_{goal} - u_i \left(s_i^{(k)} \right) \right\} \qquad \text{(B2)}$$

In equation (B2), the control function consists of a goal function, u_{goal}. The principle dictates that the control goal must be observable and measurable by the system (that is, the sensor *i*). At position $p_i^{(k)}$, sensor *i* computes a value, $u_i \left(s_i^{(k)} \right)$ which is dependent only on the current state of sensor *i* for autonomous control. The amount of adjustment or control requires for the behavior (that is, the position) of sensor *i* depends on the deviation of $u_i \left(s_i^{(k)} \right)$ from the goal. The larger the deviation, the more adjustment we need to make to its position. The value K_u is known as the control gain and is a constant in linear control. If K_u is too small, the system takes a long time to converge to its desired behavior and if K_u is too large, the system is highly oscillatory and may never converge to its desired behavior. One approach to ensure fast convergence and simultaneously alleviate the oscillatory behavior is to use a damper as in the PID controller. Another approach is to use adaptive gain control. In the later case, our controller is no longer linear. Clearly, from the discussion, before we can solve equation (B1), we must first define and express our control goal in the form that is both observable and measurable.

The goal of our sensor network is to sense and reproduce an accurate map of the distribution of the temperature. Since we have a constraint of N_s sensors, we could only sample the distribution of the temperature at discrete point in space. However, we are allowed to position the sensors arbitrarily to obtain the most accurate map. Note that this problem is different from traditional curve fitting problem. In curve fitting problem the sampling points are evenly spaced and we are allowed to increase the number of points in order to improve the accuracy.

In Figure B1a-b, we use 2 different placement strategies of the sensors to obtain the mappings of the temperature distribution, θ in the 1-D enclosed space, *E*. A visual comparison between Figure B1a and Figure B1b shows that by using uneven placement of sensors, we are able to reproduce a more accurate map of θ.

Finally, we observe that the approximate maps appear crude and jagged. It is true that there are techniques to smooth out the jagged edges. However, this is not our main concern, as ours is to ensure that there is minimum distortion or loss of information in the raw data collection. Notice that once the information is lost as in Figure B1a, no technique can restore back the loss even if we can smooth the approximate map at the later stage.

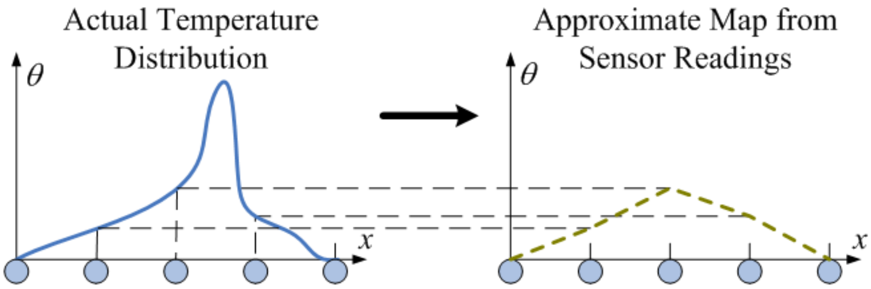

Actual Temperature Distribution
θ
x
Approximate Map from Sensor Readings
θ
x

Figure B1a: Mapping the temperature distribution, θ using sensors placed evenly in the 1-D enclosed region, *E*

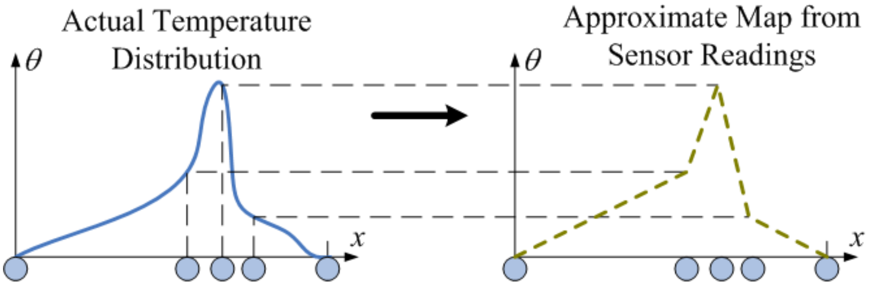

Actual Temperature Distribution
θ
x
Approximate Map from Sensor Readings
θ
x

Figure B1b: Mapping the temperature distribution, θ using sensors placed optimally in the 1-D enclosed region, *E*

Let $\hat{\theta}$ be the piecewise linear approximation map of θ based on the sensor readings. We shall now formally define our measure of distortion by the global distortion error metric, D_e in two variations as follows:

$$D_e = \int_E \left|\theta - \hat{\theta}\right| dx \tag{B3a}$$

$$D_e = \sum_{i=1}^{N_S-1} \left| \int_{p_i}^{p_{i+1}} \theta dx - \frac{1}{2}(\theta_{i+1} + \theta_i)(p_{i+1} - p_i) \right| \tag{B3b}$$

The second variation of D_e in equation (B3b) which is derived from equation (B3a) is more important for a practical reason because it expresses the global distortion error as the sum of local distortion errors. This form suggests that it is sufficient for the distributed algorithm to use the local distortion error to adjust the individual node positions in order to achieve the global minimum.

Referring to Figure B2, D_e can be visualized as the area between the actual temperature distribution θ and the approximate map $\hat{\theta}$ from the sensor reading. θ is composed of a concave region follow by a convex region. Clearly, our objective is to select the positions of the mobile sensors, P such that the total area is minimized.

An important observation is that for concave/convex regions, our approximate map $\hat{\theta}$ is lower/upper bounded by θ. For example, in the concave region, to minimize the distortion, we only need to choose the position p_2 so that it minimizes the local area of $\hat{\theta}$. This is because we know that this local area of $\hat{\theta}$ must be lower bounded by the local area of θ and vice versa. The second observation is that whether the region is concave or convex can be determined by testing the sign of the second derivative of the locality. The local area of the approximate map as seen by sensor *i*, is given as:

$$LA_i = \frac{1}{2}\left[(\theta_i + \theta_{i-1})(p_i - p_{i-1}) + (\theta_{i+1} + \theta_i)(p_{i+1} - p_i)\right] \qquad \text{(B4)}$$

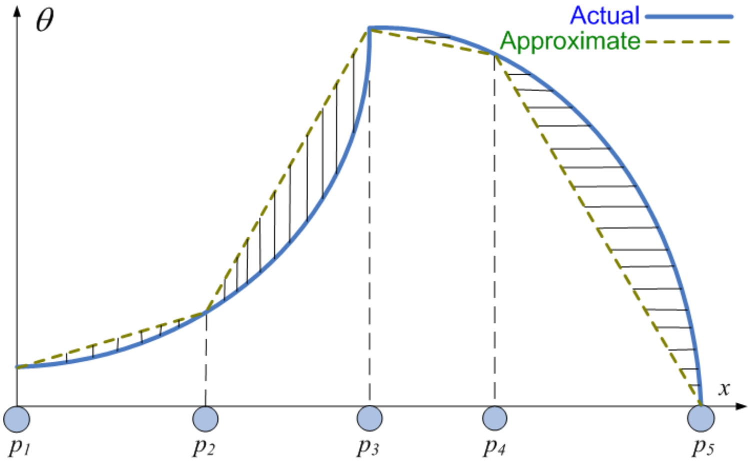


Figure B2: Distortion error

Referring to Figure B2 again, our optimization strategy can be briefly described as follows:

- For the **concave** region, sensor 2 should adjust its position p_2 with respect to its 2 Voronoi neighbors in order to minimize LA_2. Since, we know that our approximate map is lower bounded by θ.
- For the **convex** region, sensor 4 should adjust its position, p_4 with respect to its 2 Voronoi neighbors in order to maximize LA_4. Since we know that our approximate map is upper bounded by θ.

We now examine the condition for minimum/maximum. We differentiate equation (B4) w.r.t. p_i.

$$\frac{\partial LA_i}{\partial p_i} = \frac{1}{2}\left[\frac{\partial \theta_i}{\partial p_i}(p_i - p_{i-1}) + (\theta_i + \theta_{i-1})(1) + \frac{\partial \theta_i}{\partial p_i}(p_{i+1} - p_i) + (\theta_{i+1} + \theta_i)(-1)\right]$$
$$= \frac{1}{2}\left[\frac{\partial \theta_i}{\partial p_i}(p_{i+1} - p_{i-1}) + (\theta_{i-1} - \theta_{i+1})\right] \qquad \text{(B5)}$$

To obtain the condition for minimum/maximum, we set $\frac{\partial LA_i}{\partial p_i} = 0$,

$$\frac{\partial LA_i}{\partial p_i} = \frac{1}{2}\left[\frac{\partial \theta_i}{\partial p_i}(p_{i+1} - p_{i-1}) + (\theta_{i-1} - \theta_{i+1})\right] = 0$$
$$\Leftrightarrow \frac{\partial \theta_i}{\partial p_i}(p_{i+1} - p_{i-1}) + (\theta_{i-1} - \theta_{i+1}) = 0$$

$$\therefore \frac{\partial \theta_i}{\partial p_i} = \frac{(\theta_{i+1} - \theta_{i-1})}{(p_{i+1} - p_{i-1})} \tag{B6}$$

The condition in equation (B6) states that in order for us to obtain a minimum /maximum point, we should adjust the position of sensor *i* such that the gradient at the tangent point (p_i, θ_i) is equal to the gradient of the secant line bounded by the Voronoi neighbors of sensor *i*. In other words, the tangent is parallel to the secant line at the minimum/maximum. The optimal condition for Figure B2 is illustrated in Figure B3.

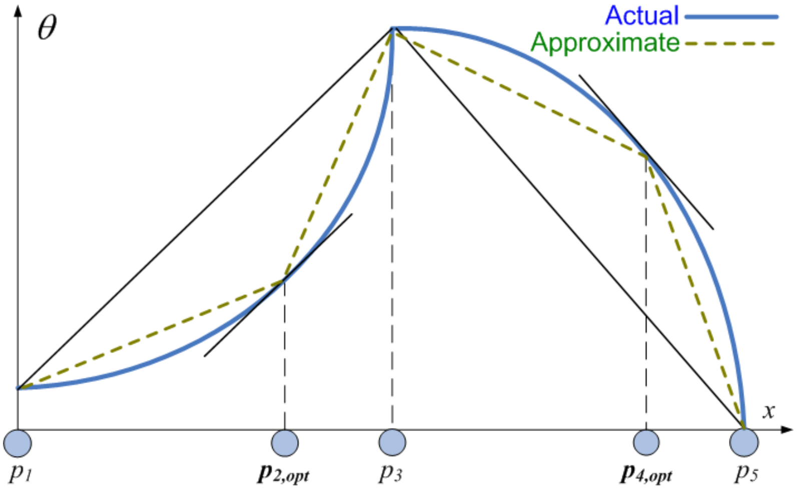


Figure B3: Optimal condition of minimum distortion error

For a continuous function, equation (B6) holds and we can always find an intermediate point that has a tangent which is parallel to the secant line based on Mean Value Theorem. Therefore, it shows us that it is possible for intermediate node *i* to adjust its position in between its two neighbors to minimize distortion without changing the relative topological relationship with its Voronoi neighbors. Applying the argument inductively, it is feasible for us to develop a distributive algorithm that preserves the network topology.

We now take the second derivative of LA_i using equation (B5),

$$\therefore \frac{\partial^2 LA_i}{\partial p_i^2} = \frac{1}{2}\frac{\partial^2 \theta_i}{\partial p_i^2}(p_{i+1} - p_{i-1}) \tag{B7}$$

Note that by virtue of the ordering $(p_{i+1} - p_{i-1})$ is always positive. For the concave region, $\frac{\partial^2 \theta_i}{\partial p_i^2} > 0 \Rightarrow \frac{\partial^2 LA_i}{\partial p_i^2} > 0$. Therefore, the optimal point is a minimum and vice versa. This is in agreement with our observation in Figure B2 and B3 that we should minimize/maximize the area in the concave/convex region in order to minimize distortion because they are bounded by θ.

We are now in the position to derive an expression for the control equation stated in equation (B1) for our cooperative mobile sensors. Upon examination of equation (B6), we find that it is a suitable control goal for us to use to control the mobile sensor. The form in equation (B6) suggests that we could use the efficient Newton's method of zero-finding [91] with order of convergence of 2 to compute the next position that sensor *i* should take. Let us first define the following variables:

$LA_i^{(k)}$	The local area of sensor *i* in the *k*th time step.
$\frac{\partial LA_i}{\partial p_i}(k)$	The first derivative of local area of sensor *i* in the *k*th time step. We also denote it as $y(p_i,k)$ for clarity of presentation. That is, $y(p_i,k)=\frac{\partial LA_i}{\partial p_i}(k)$.
$\frac{\partial \theta_i}{\partial p_i}(k)$	The first derivative of sensed value θ_i of sensor *i* in the *k*th time step.
$\frac{\partial^2 \theta_i}{\partial p_i^2}(k)$	The second derivative of sensed value θ_i of sensor *i* in the *k*th time step.

From equation (B5) and (B7), and the Newton's method we obtain the main result as follows:

$$p_i^{(k+1)} = p_i^{(k)} - y(p_i,k)\left[\frac{\partial y(p_i,k)}{\partial p_i}\right]^{-1}$$

$$\Leftrightarrow p_i^{(k+1)} = p_i^{(k)} - \frac{1}{2}\left[\frac{\partial \theta_i}{\partial p_i}(p_{i+1}-p_{i-1})+(\theta_{i-1}-\theta_{i+1})\right]\left\{\frac{1}{2}\frac{\partial^2 \theta_i}{\partial p_i^2}(p_{i+1}-p_{i-1})\right\}^{-1}$$

$$\Leftrightarrow p_i^{(k+1)} = p_i^{(k)} - \left[\frac{\partial \theta_i}{\partial p_i} - \frac{(\theta_{i+1}-\theta_{i-1})}{(p_{i+1}-p_{i-1})}\right]\left(\frac{\partial^2 \theta_i}{\partial p_i^2}\right)^{-1}$$

$$\therefore p_i^{(k+1)} = p_i^{(k)} + \left(\frac{\partial^2 \theta_i}{\partial p_i^2}(k)\right)^{-1}\left[\frac{\left(\theta_{i+1}^{(k)}-\theta_{i-1}^{(k)}\right)}{\left(p_{i+1}^{(k)}-p_{i-1}^{(k)}\right)} - \frac{\partial \theta_i}{\partial p_i}(k)\right] \qquad \text{(B8)}$$

Note that there is an important difference between the original Newton's method and equation (B8). In one time step, the adjustment is made using 3 simultaneous measurements (obtained from cooperation). In the original form, there is only 1 measurement (corresponding to autonomous behavior) in one time step. The original method has to make use of additional values from the past 2 time steps. In other words, an autonomous approach will converge 3 times slower than the cooperative approach since it requires 3 time steps to obtain the same amount of measurements as 1 time step in the cooperative approach.

Since, we have the readings for 3 sensors: (*i* – 1), *i*, (*i* + 1), we should fully utilize the values to accurately evaluate $\frac{\partial \theta_i}{\partial p_i}(k)$ and $\frac{\partial^2 \theta_i}{\partial p_i^2}(k)$. We first construct the Lagrange polynomial [92] through the 3 sensor positions:

$$l_2(p)=\frac{(p-p_i)(p-p_{i+1})}{(p_{i-1}-p_i)(p_{i-1}-p_{i+1})}\theta_{i-1}+\frac{(p-p_{i-1})(p-p_{i+1})}{(p_i-p_{i-1})(p_i-p_{i+1})}\theta_i$$
$$+\frac{(p-p_{i-1})(p-p_i)}{(p_{i+1}-p_{i-1})(p_{i+1}-p_i)}\theta_{i+1} \tag{B9a}$$

Taking the first and second derivatives of $l_2(p)$ w.r.t. p,

$$l_2'(p)=\frac{2p-p_i-p_{i+1}}{(p_{i-1}-p_i)(p_{i-1}-p_{i+1})}\theta_{i-1}+\frac{2p-p_{i-1}-p_{i+1}}{(p_i-p_{i-1})(p_i-p_{i+1})}\theta_i$$
$$+\frac{2p-p_{i-1}-p_i}{(p_{i+1}-p_{i-1})(p_{i+1}-p_i)}\theta_{i+1} \tag{B9b}$$

$$l_2''(p)=\frac{2\theta_{i-1}}{(p_{i-1}-p_i)(p_{i-1}-p_{i+1})}+\frac{2\theta_i}{(p_i-p_{i-1})(p_i-p_{i+1})}$$
$$+\frac{2\theta_{i+1}}{(p_{i+1}-p_{i-1})(p_{i+1}-p_i)} \tag{B9c}$$

Therefore, $\frac{\partial\theta_i}{\partial p_i}(k)$ and $\frac{\partial^2\theta_i}{\partial p_i^2}(k)$ are evaluated as follows:

$$\frac{\partial\theta_i}{\partial p_i}(k)\approx l_2'(p_i)=\frac{p_i^{(k)}-p_{i+1}^{(k)}}{\left(p_{i-1}^{(k)}-p_i^{(k)}\right)\left(p_{i-1}^{(k)}-p_{i+1}^{(k)}\right)}\theta_{i-1}^{(k)}$$
$$+\frac{2p_i^{(k)}-p_{i-1}^{(k)}-p_{i+1}^{(k)}}{\left(p_i^{(k)}-p_{i-1}^{(k)}\right)\left(p_i^{(k)}-p_{i+1}^{(k)}\right)}\theta_i^{(k)}+\frac{p_i^{(k)}-p_{i-1}^{(k)}}{\left(p_{i+1}^{(k)}-p_{i-1}^{(k)}\right)\left(p_{i+1}^{(k)}-p_i^{(k)}\right)}\theta_{i+1}^{(k)} \tag{B10a}$$

$$\frac{\partial^2\theta_i}{\partial p_i^2}(k)\approx l_2''(p_i)=\frac{2\theta_{i-1}^{(k)}}{\left(p_{i-1}^{(k)}-p_i^{(k)}\right)\left(p_{i-1}^{(k)}-p_{i+1}^{(k)}\right)}$$
$$+\frac{2\theta_i^{(k)}}{\left(p_i^{(k)}-p_{i-1}^{(k)}\right)\left(p_i^{(k)}-p_{i+1}^{(k)}\right)}+\frac{2\theta_{i+1}^{(k)}}{\left(p_{i+1}^{(k)}-p_{i-1}^{(k)}\right)\left(p_{i+1}^{(k)}-p_i^{(k)}\right)} \tag{B10b}$$

We shall now interpret the main result in equation (B8). Comparing equation (B8) with (B1) and (B2), we observe that:

$$\Delta p_i^{(k)}\left(C_{sn}^{(k)}\right)=K_u\left(u_{goal}-u_i\right) \tag{B11}$$

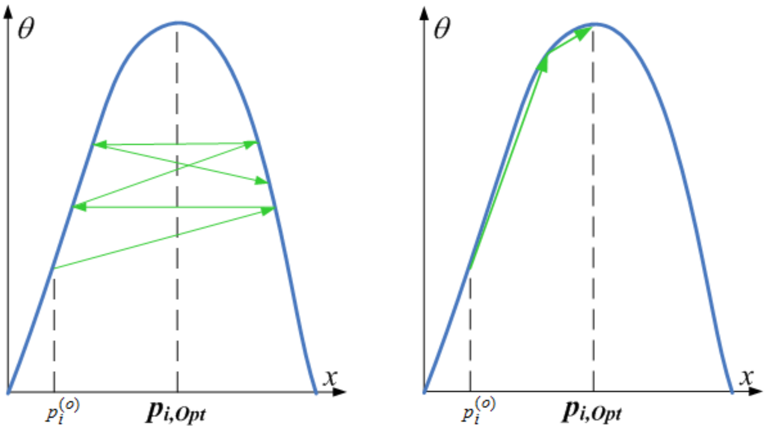
Where,

$$K_u=\left(\frac{\partial^2\theta_i}{\partial p_i^2}(k)\right)^{-1} \tag{B12a}$$

$$u_{goal}=\frac{\left(\theta_{i+1}^{(k)}-\theta_{i-1}^{(k)}\right)}{\left(p_{i+1}^{(k)}-p_{i-1}^{(k)}\right)} \tag{B12b}$$

$$u_i = \frac{\partial \theta_i}{\partial p_i}(k) \tag{B12c}$$

The first observation is our position control function, $\Delta p_i^{(k)}\left(C_{sn}^{(k)}\right)$ in equation (B11) is a non-linear function of the set of state variables, $C_{sn}^{(k)}$. The second observation is that its structure fits the general structure of a control equation. In particular, we know that the amount of position adjustment depends on the deviation of u_i from u_{goal}. Obviously, the adjustment stops when the control goal is reached: $u_i = u_{goal}$. The control goal is in alignment with the optimality condition stated in equation (B6). The third observation is that the control gain, K_u is adaptive. It is the reciprocal of $\frac{\partial^2 \theta_i}{\partial p_i^2}$. $\frac{\partial^2 \theta_i}{\partial p_i^2}$ is a measure for the curvature of θ at p_i. $\frac{\partial^2 \theta_i}{\partial p_i^2}$ is positive when the curve is concave and otherwise. Furthermore, the absolute magnitude of $\frac{\partial^2 \theta_i}{\partial p_i^2}$ indicates the sharpness in the turn of the curve at p_i. When the absolute magnitude of $\frac{\partial^2 \theta_i}{\partial p_i^2}$ is small, the curve is relatively gentle and flat and vice versa. Hence, K_u is large when the curve is gentle and small when the curve is sharp.



a) Constant K_u: Sensor i overshoots the optimum point and backtracks many times.

b) Adaptive K_u: The sharper the curve, the slower the velocity of sensor i.

Figure B4: Adjustments in the region of high curvature

This is intuitively satisfying as shown in Figure B4a-b. In Figure B4a, K_u is constant. As a result, the sensor i tends to overshoot the optimum point, $p_{i,opt}$ and has to backtrack its position several times, resulting in an oscillatory behavior with large amount of energy expended. In Figure B4b, K_u is adaptive. This time, K_u acts as a braking mechanism, it decelerates the sensor i around the sharp turn so that it approaches $p_{i,opt}$ cautiously. The movement is smoother than Figure B4a and follows closely with the shape of the curve. This allows the algorithm to converge rapidly. In fact, it converges with an order of convergence of around 2, since it is adopted from Newton's method.

Appendix C: Derivation of the Motion Control Equations for Two-Dimensional Topology

Recall that in the derivation for the 1D case, we examine the area of the distribution under a node bounded by two referencing neighbors. Therefore, based on the current state information, the node chooses the best position to move towards so as to either maximize or minimize the area of the distribution. The individual goal turns out to be to seek a position such that the temperature gradient in that position is equal to the mean gradient taken over the two reference points and the intermediate node.

We use the same argument to obtain the expression for the 2D case, by considering the volume bounded by an intermediate node and three referencing neighbors.

The volume of the tetrahedron with four vertices: (x_1, y_1, θ_1), (x_2, y_2, θ_2), (x_3, y_3, θ_3) and (x_i, y_i, θ_i), V is given as:

$$V = \frac{1}{3!}\begin{vmatrix} x_i & y_i & \theta_i & 1 \\ x_1 & y_1 & \theta_1 & 1 \\ x_2 & y_2 & \theta_2 & 1 \\ x_3 & y_3 & \theta_3 & 1 \end{vmatrix}$$

$$\Rightarrow 6V = x_i\begin{vmatrix} y_1 & \theta_1 & 1 \\ y_2 & \theta_2 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix} - x_1\begin{vmatrix} y_i & \theta_i & 1 \\ y_2 & \theta_2 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix} + x_2\begin{vmatrix} y_i & \theta_i & 1 \\ y_1 & \theta_1 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix} - x_3\begin{vmatrix} y_i & \theta_i & 1 \\ y_1 & \theta_1 & 1 \\ y_2 & \theta_2 & 1 \end{vmatrix}$$

$$\begin{aligned} \Rightarrow 6V = {} & x_i\left[y_1(\theta_2 - \theta_3) - y_2(\theta_1 - \theta_3) + y_3(\theta_1 - \theta_2)\right] - x_1\left[y_i(\theta_2 - \theta_3) - y_2(\theta_i - \theta_3) + y_3(\theta_i - \theta_2)\right] \\ & + x_2\left[y_i(\theta_1 - \theta_3) - y_1(\theta_i - \theta_3) + y_3(\theta_i - \theta_1)\right] - x_3\left[y_i(\theta_1 - \theta_2) - y_1(\theta_i - \theta_2) + y_2(\theta_i - \theta_1)\right] \quad \text{(C1)} \end{aligned}$$

To maximize/minimize V, set $\frac{\partial V}{\partial x_i} = 0$ and $\frac{\partial V}{\partial y_i} = 0$

$$\begin{aligned} 6\frac{\partial V}{\partial x_i} = {} & \left[y_1(\theta_2 - \theta_3) - y_2(\theta_1 - \theta_3) + y_3(\theta_1 - \theta_2)\right] \\ & - x_1\left[-y_2\frac{\partial \theta_i}{\partial x_i} + y_3\frac{\partial \theta_i}{\partial x_i}\right] + x_2\left[-y_1\frac{\partial \theta_i}{\partial x_i} + y_3\frac{\partial \theta_i}{\partial x_i}\right] - x_3\left[-y_1\frac{\partial \theta_i}{\partial x_i} + y_2\frac{\partial \theta_i}{\partial x_i}\right] = 0 \end{aligned}$$

$$\begin{aligned} & \Rightarrow \left[y_1(\theta_2 - \theta_3) - y_2(\theta_1 - \theta_3) + y_3(\theta_1 - \theta_2)\right] = \\ & x_1\left[-y_2\frac{\partial \theta_i}{\partial x_i} + y_3\frac{\partial \theta_i}{\partial x_i}\right] - x_2\left[-y_1\frac{\partial \theta_i}{\partial x_i} + y_3\frac{\partial \theta_i}{\partial x_i}\right] + x_3\left[-y_1\frac{\partial \theta_i}{\partial x_i} + y_2\frac{\partial \theta_i}{\partial x_i}\right] \end{aligned}$$

$$\Rightarrow \left[y_1(\theta_2-\theta_3)-y_2(\theta_1-\theta_3)+y_3(\theta_1-\theta_2)\right]=$$
$$\frac{\partial\theta_i}{\partial x_i}\left[x_1(-y_2+y_3)-x_2(-y_1+y_3)+x_3(-y_1+y_2)\right]$$

$$\Rightarrow \begin{vmatrix} y_1 & \theta_1 & 1 \\ y_2 & \theta_2 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix} = -\frac{\partial\theta_i}{\partial x_i}\left[x_1(y_2-y_3)-x_2(y_1-y_3)+x_3(y_1-y_2)\right]$$

$$\Rightarrow \begin{vmatrix} y_1 & \theta_1 & 1 \\ y_2 & \theta_2 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix} = -\frac{\partial\theta_i}{\partial x_i}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

$$\therefore \frac{\partial\theta_i}{\partial x_i} = -\begin{vmatrix} y_1 & \theta_1 & 1 \\ y_2 & \theta_2 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}^{-1}$$

Also,

$$6\frac{\partial V}{\partial y_i} = -x_1\left[(\theta_2-\theta_3)-y_2\frac{\partial\theta_i}{\partial y_i}+y_3\frac{\partial\theta_i}{\partial y_i}\right]$$
$$+x_2\left[(\theta_1-\theta_3)-y_1\frac{\partial\theta_i}{\partial y_i}+y_3\frac{\partial\theta_i}{\partial y_i}\right]-x_3\left[(\theta_1-\theta_2)-y_1\frac{\partial\theta_i}{\partial y_i}+y_2\frac{\partial\theta_i}{\partial y_i}\right]=0$$

$$\Rightarrow -x_1(\theta_2-\theta_3)-x_1(-y_2+y_3)\frac{\partial\theta_i}{\partial y_i}$$
$$+x_2(\theta_1-\theta_3)+x_2(-y_1+y_3)\frac{\partial\theta_i}{\partial y_i}-x_3(\theta_1-\theta_2)-x_3(-y_1+y_2)\frac{\partial\theta_i}{\partial y_i}=0$$

$$\Rightarrow \frac{\partial\theta_i}{\partial y_i}\left[-x_1(-y_2+y_3)+x_2(-y_1+y_3)-x_3(-y_1+y_2)\right]$$
$$=x_1(\theta_2-\theta_3)-x_2(\theta_1-\theta_3)+x_3(\theta_1-\theta_2)$$

$$\Rightarrow \frac{\partial\theta_i}{\partial y_i}\left[x_1(y_2-y_3)-x_2(y_1-y_3)+x_3(y_1-y_2)\right]=\begin{vmatrix} x_1 & \theta_1 & 1 \\ x_2 & \theta_2 & 1 \\ x_3 & \theta_3 & 1 \end{vmatrix}$$

$$\Rightarrow \frac{\partial\theta_i}{\partial y_i}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = \begin{vmatrix} x_1 & \theta_1 & 1 \\ x_2 & \theta_2 & 1 \\ x_3 & \theta_3 & 1 \end{vmatrix}$$

$$\therefore \frac{\partial\theta_i}{\partial y_i} = \begin{vmatrix} x_1 & \theta_1 & 1 \\ x_2 & \theta_2 & 1 \\ x_3 & \theta_3 & 1 \end{vmatrix}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}^{-1}$$

Let $A = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$, $B = \begin{vmatrix} y_1 & \theta_1 & 1 \\ y_2 & \theta_2 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix}$ and $C = \begin{vmatrix} x_1 & \theta_1 & 1 \\ x_2 & \theta_2 & 1 \\ x_3 & \theta_3 & 1 \end{vmatrix}$

$\therefore \dfrac{\partial \theta_i}{\partial x_i} = -BA^{-1}$ and $\dfrac{\partial \theta_i}{\partial y_i} = CA^{-1}$

Substituting the above into the gradient equation, we have our objective function as:

$$\nabla \theta_i = \begin{bmatrix} \dfrac{\partial \theta_i}{\partial x_i} & \dfrac{\partial \theta_i}{\partial y_i} \end{bmatrix} = A^{-1}[-B \quad C]^T$$

Let $u_{goal} = A^{-1}[-B \quad C]$.

$$\therefore \nabla \theta_i = u_{goal} \tag{C2}$$

Therefore, simply stated, the equation requires the node to search for a point in the triangular region bounded by its three neighbors such that the gradient at that point is equal to the mean gradient taken over the triangular region. Recall that for the 1D case, the goal is given by,

$$\frac{d\theta_i}{dx_i} = \frac{\theta_2 - \theta_1}{x_2 - x_1} \tag{C3}$$

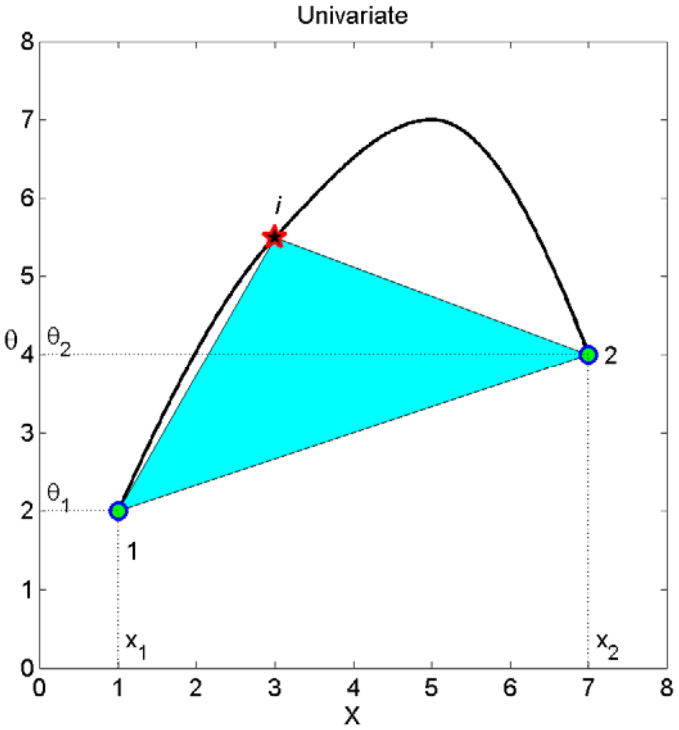


Figure C1: Optimal control goal for the intermediate node in a 1D scenario

Figure C1 and C2 illustrate the similarity between the expression for the 1D and 2D case. As explained, the RHS of equation (C2) and (C3) is actually mean gradient. An analogy is obtaining the mean speed of a vehicle from a distance-time graph. The speed of the vehicle is the gradient of the distance-time graph. The mean speed of a vehicle after travelling from point 1 to point 2 is the ratio of the total distance over the total time. Similarly, in Figure C1, the mean temperature gradient over the two reference point 1 and 2 is the ratio of the total temperature range over the total distance. Another interpretation from Figure C1 is that the mean gradient is the ratio of the line segment bounded by point 1 and 2 projected on the θ-axis over that of the *x*-axis. Similarly, in equation (C2), *A*, *B* and *C* is the areas of the triangle (green colour) bounded by point 1, 2 and 3 projected onto the *xy*, *x*θ and *y*θ planes. The areas are positive when taken anti-clockwise w.r.t. the 3 reference points. The mean partial gradients along the two orthogonal *x* and *y* directions are therefore the ratios of the areas of the respective projected triangles as shown in Figure C2.

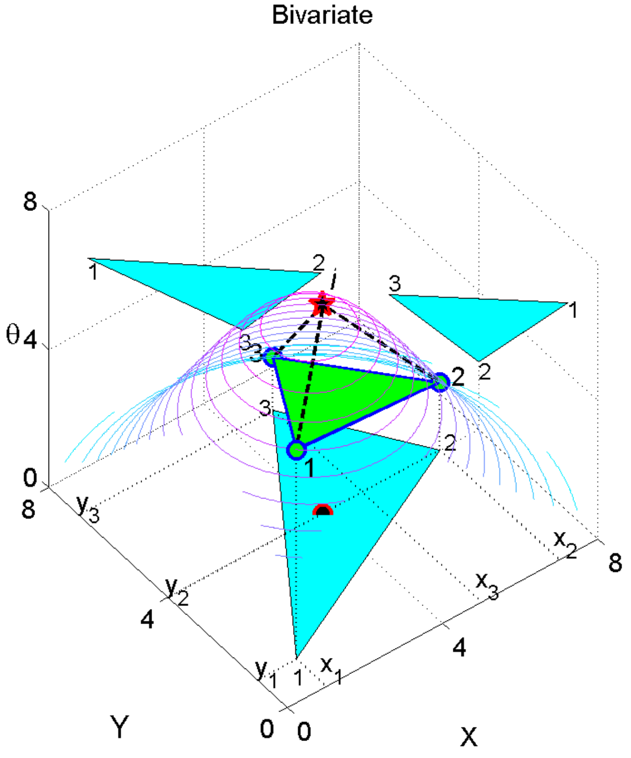


Figure C2: Optimal control goal for the intermediate node in a 2D scenario

Since, we are looking at a continuous temperature distribution, an appropriate class of search methods will be the descent method. Newton search method is the search method in this class that offers a rapid second order of convergence. The rapid convergence is due to the use of more information (first and second order derivatives) in the search method. The assumption in this method is that we have a-priori perfect knowledge of the distribution. We will look at the modifications required for the case where we do not have a priori perfect knowledge of the distribution. A discussion of the Newton Search Method can be found in [5].

From Equation (C1), the volume is given by

$$
\begin{aligned}
V = \frac{1}{6} & x_i\left[y_1(\theta_2-\theta_3)-y_2(\theta_1-\theta_3)+y_3(\theta_1-\theta_2)\right] \\
& -x_1\left[y_i(\theta_2-\theta_3)-y_2(\theta_i-\theta_3)+y_3(\theta_i-\theta_2)\right] \\
& +x_2\left[y_i(\theta_1-\theta_3)-y_1(\theta_i-\theta_3)+y_3(\theta_i-\theta_1)\right] \\
& -x_3\left[y_i(\theta_1-\theta_2)-y_1(\theta_i-\theta_2)+y_2(\theta_i-\theta_1)\right]
\end{aligned}
$$

Ignoring the constant term, $\frac{1}{6}$, we let

$$
\begin{aligned}
V' = & x_i\left[y_1(\theta_2-\theta_3)-y_2(\theta_1-\theta_3)+y_3(\theta_1-\theta_2)\right] \\
& -x_1\left[y_i(\theta_2-\theta_3)-y_2(\theta_i-\theta_3)+y_3(\theta_i-\theta_2)\right] \\
& +x_2\left[y_i(\theta_1-\theta_3)-y_1(\theta_i-\theta_3)+y_3(\theta_i-\theta_1)\right] \\
& -x_3\left[y_i(\theta_1-\theta_2)-y_1(\theta_i-\theta_2)+y_2(\theta_i-\theta_1)\right]
\end{aligned}
$$

The idea of Newton's method is to maximize V' w.r.t. $p_i = (x_i, y_i)$ by maximizing the quadratic approximation of V' about $p_i = (x_i, y_i)$. This maximization is performed at each iteration of the algorithm. Specifically, using a second-order Taylor series expansion around the point $p_i = (x_i, y_i)$, we may write:

$$\Delta V'(p_i) = V'(p_i^{(k+1)}) - V'(p_i^{(k)}) \approx g^T(k)\Delta p_i^{(k+1)} + \frac{1}{2}(\Delta p_i^{(k)})^T H(k)(\Delta p_i^{(k)}) \quad \text{(C4)}$$

where ,

$$g(k) = \nabla V'(k) = \begin{bmatrix} \dfrac{\partial V'(k)}{\partial x_i} & \dfrac{\partial V'(k)}{\partial y_i} \end{bmatrix}^T$$

$$H(k) = \nabla^2 V'(k) = \begin{bmatrix} \dfrac{\partial^2 V'(k)}{\partial x_i^2} & \dfrac{\partial^2 V'(k)}{\partial x_i \partial y_i} \\ \dfrac{\partial^2 V'(k)}{\partial y_i \partial x_i} & \dfrac{\partial^2 V'(k)}{\partial y_i^2} \end{bmatrix}$$

For a continuous function, $\dfrac{\partial^2 V'(k)}{\partial x \partial y} = \dfrac{\partial^2 V'(k)}{\partial y \partial x}$, therefore, the Hessian, *H(k)* is symmetric. The change $\Delta V'(p_i)$ is minimized or maximized when $\nabla\left[\Delta V'(p_i)\right] = \underline{0}$. Therefore, differentiating equation (C4) w.r.t Δp_i to obtain $\nabla\left[\Delta V'(p_i)\right]$, we have,

$$\nabla\left[\Delta V'(p_i)\right] = g(k) + H(k)\Delta p_i^{(k)} = \underline{0} \Rightarrow \Delta p_i^{(k)} = -H^{-1}(k)g(k) \qquad \text{(C5)}$$

$$\therefore p_i^{(k+1)} = p_i^{(k)} + \Delta p_i^{(k)} = p_i^{(k)} - H^{-1}(k)g(k) \qquad \text{(C6)}$$

We will need to compute the first and second partial derivatives of $V'(k)$ and substitute them into equation (C4).

$$\frac{\partial V'}{\partial x_i} = \left[y_1(\theta_2-\theta_3)-y_2(\theta_1-\theta_3)+y_3(\theta_1-\theta_2)\right]$$

$$-x_1\left[-y_2\frac{\partial\theta_i}{\partial x_i}+y_3\frac{\partial\theta_i}{\partial x_i}\right]+x_2\left[-y_1\frac{\partial\theta_i}{\partial x_i}+y_3\frac{\partial\theta_i}{\partial x_i}\right]-x_3\left[-y_1\frac{\partial\theta_i}{\partial x_i}+y_2\frac{\partial\theta_i}{\partial x_i}\right]$$

$$\Rightarrow\frac{\partial V'}{\partial x_i}=\left[y_1(\theta_2-\theta_3)-y_2(\theta_1-\theta_3)+y_3(\theta_1-\theta_2)\right]$$
$$+\frac{\partial\theta_i}{\partial x_i}\left[-x_1(-y_2+y_3)+x_2(-y_1+y_3)-x_3(-y_1+y_2)\right]$$

$$\Rightarrow\frac{\partial V'}{\partial x_i}=\left[y_1(\theta_2-\theta_3)-y_2(\theta_1-\theta_3)+y_3(\theta_1-\theta_2)\right]$$
$$+\frac{\partial\theta_i}{\partial x_i}\left[x_1(y_2-y_3)-x_2(y_1-y_3)+x_3(y_1-y_2)\right]$$

$$\Rightarrow\frac{\partial V'}{\partial x_i}=\begin{vmatrix}y_1&\theta_1&1\\y_2&\theta_2&1\\y_3&\theta_3&1\end{vmatrix}+\frac{\partial\theta_i}{\partial x_i}\begin{vmatrix}x_1&y_1&1\\x_2&y_2&1\\x_3&y_3&1\end{vmatrix}\qquad\text{(C7)}$$

$$\frac{\partial V'}{\partial y_i}=-x_1\left[(\theta_2-\theta_3)-y_2\frac{\partial\theta_i}{\partial y_i}+y_3\frac{\partial\theta_i}{\partial y_i}\right]$$
$$+x_2\left[(\theta_1-\theta_3)-y_1\frac{\partial\theta_i}{\partial y_i}+y_3\frac{\partial\theta_i}{\partial y_i}\right]-x_3\left[(\theta_1-\theta_2)-y_1\frac{\partial\theta_i}{\partial y_i}+y_2\frac{\partial\theta_i}{\partial y_i}\right]$$

$$\Rightarrow\frac{\partial V'}{\partial y_i}=-x_1(\theta_2-\theta_3)-x_1(-y_2+y_3)\frac{\partial\theta_i}{\partial y_i}$$
$$+x_2(\theta_1-\theta_3)+x_2(-y_1+y_3)\frac{\partial\theta_i}{\partial y_i}-x_3(\theta_1-\theta_2)-x_3(-y_1+y_2)\frac{\partial\theta_i}{\partial y_i}$$

$$\Rightarrow\frac{\partial V'}{\partial y_i}=\frac{\partial\theta_i}{\partial y_i}\left[-x_1(-y_2+y_3)+x_2(-y_1+y_3)-x_3(-y_1+y_2)\right]$$
$$-\left[x_1(\theta_2-\theta_3)-x_2(\theta_1-\theta_3)+x_3(\theta_1-\theta_2)\right]$$

$$\Rightarrow\frac{\partial V'}{\partial y_i}=\frac{\partial\theta_i}{\partial y_i}\left[x_1(y_2-y_3)-x_2(y_1-y_3)+x_3(y_1-y_2)\right]$$
$$-\left[x_1(\theta_2-\theta_3)-x_2(\theta_1-\theta_3)+x_3(\theta_1-\theta_2)\right]$$

$$\Rightarrow\frac{\partial V'}{\partial y_i}=\frac{\partial\theta_i}{\partial y_i}\begin{vmatrix}x_1&y_1&1\\x_2&y_2&1\\x_3&y_3&1\end{vmatrix}-\begin{vmatrix}x_1&\theta_1&1\\x_2&\theta_2&1\\x_3&\theta_3&1\end{vmatrix}\qquad\text{(C8)}$$

Also,

$$\frac{\partial^2 V'}{\partial x_i^2}=\frac{\partial^2\theta_i}{\partial x_i^2}\left[x_1(y_2-y_3)-x_2(y_1-y_3)+x_3(y_1-y_2)\right]$$

$$\Rightarrow \frac{\partial^2 V'}{\partial x_i^2} = \frac{\partial^2 \theta_i}{\partial x_i^2}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \tag{C9}$$

$$\frac{\partial^2 V'}{\partial y_i^2} = \frac{\partial^2 \theta_i}{\partial y_i^2}\left[x_1(y_2 - y_3) - x_2(y_1 - y_3) + x_3(y_1 - y_2)\right]$$

$$\Rightarrow \frac{\partial^2 V'}{\partial y_i^2} = \frac{\partial^2 \theta_i}{\partial y_i^2}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \tag{C10}$$

$$\frac{\partial^2 V'}{\partial x_i \partial y_i} = \frac{\partial^2 \theta_i}{\partial x_i \partial y_i}\left[x_1(y_2 - y_3) - x_2(y_1 - y_3) + x_3(y_1 - y_2)\right]$$

$$\Rightarrow \frac{\partial^2 V'}{\partial x_i \partial y_i} = \frac{\partial^2 \theta_i}{\partial x_i \partial y_i}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \tag{C11}$$

$$\frac{\partial^2 V'}{\partial y_i \partial x_i} = \frac{\partial^2 \theta_i}{\partial y_i \partial x_i}\left[x_1(y_2 - y_3) - x_2(y_1 - y_3) + x_3(y_1 - y_2)\right]$$

$$\Rightarrow \frac{\partial^2 V'}{\partial y_i \partial x_i} = \frac{\partial^2 \theta_i}{\partial y_i \partial x_i}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = \frac{\partial^2 V'}{\partial x_i \partial y_i} \tag{C12}$$

Note that: $A = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$, $B = \begin{vmatrix} y_1 & \theta_1 & 1 \\ y_2 & \theta_2 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix}$ and $C = \begin{vmatrix} x_1 & \theta_1 & 1 \\ x_2 & \theta_2 & 1 \\ x_3 & \theta_3 & 1 \end{vmatrix}$, substituting equation (C7)– (C12) into (C5) and (C6), we have,

$$g(k) = \nabla V'(k) = \begin{bmatrix} B + A\frac{\partial \theta_i}{\partial x_i} & -C + A\frac{\partial \theta_i}{\partial y_i} \end{bmatrix}^T$$

$$\Rightarrow g(k) = \begin{bmatrix} B & -C \end{bmatrix}^T + A\begin{bmatrix} \frac{\partial \theta_i}{\partial x_i} & \frac{\partial \theta_i}{\partial y_i} \end{bmatrix}^T$$

$$\Rightarrow g(k) = \begin{bmatrix} B & -C \end{bmatrix}^T + A\nabla\theta_i \tag{C13}$$

$$H(k) = \nabla^2 V'(k) = \begin{bmatrix} \frac{\partial^2 V'(k)}{\partial x_i^2} & \frac{\partial^2 V'(k)}{\partial x_i \partial y_i} \\ \frac{\partial^2 V'(k)}{\partial y_i \partial x_i} & \frac{\partial^2 V'(k)}{\partial y_i^2} \end{bmatrix} = A\begin{bmatrix} \frac{\partial^2 \theta_i}{\partial x_i^2} & \frac{\partial^2 \theta_i}{\partial x_i \partial y_i} \\ \frac{\partial^2 \theta_i}{\partial y_i \partial x_i} & \frac{\partial^2 \theta_i}{\partial y_i^2} \end{bmatrix}$$

$$\Rightarrow H(k) = A\nabla^2\theta_i \qquad \text{(C14)}$$

The form presented in equation (C5) is the form used in Newton Search Method. We write equation (C5) in the standard control equation form for the purpose of stabilization.

$$\begin{aligned}\Delta p_i^k &= -H^{-1}(k)g(k)\\ &= (A\nabla^2\theta_i)^{-1}\left\{[-B \quad C]^T - A\nabla\theta_i\right\}\\ &= (\nabla^2\theta_i)^{-1}\left\{A^{-1}[-B \quad C]^T - A\nabla\theta_i\right\}\\ &= K_u\left\{u_{goal} - \nabla\theta_i\right\}\end{aligned} \qquad \text{(C15)}$$

Where $K_u = (\nabla^2\theta_i)^{-1}$ is the control gain.

$\nabla\theta_i$ and $\nabla^2\theta_i$ are the temperature partial derivatives about p_i and can be obtained from the measurement data. Therefore we need to evaluate first and second partial derivatives of θ_i, however, we do not have any a-priori information on the temperature distribution except that θ_i is smooth and continuous. Let $\|\bullet\|$ be the norm of a vector. To obtain an approximation of θ_i, we perform a local interpolation of the measurements available at the four points: p_i, p_1, p_2 and p_3 using the Radial Basis Function Interpolation and derive the approximate partial derivatives.
Let $\varphi(\|p - p_h\|) = \exp\left(-\|p - p_h\|^2 / 2\sigma^2\right)$, where $h = i$, 1, 2, and 3. Then,

$$\theta(p) \approx \sum_{h=i,h=1}^{3} w_h\varphi(\|p - p_h\|) \qquad \text{(C16)}$$

where, the weight vector, $\underline{w} = [w_i \quad w_1 \quad w_2 \quad w_3]^T$ is obtained from the interpolation matrix, Φ as follows:

$$\underline{w} = \Phi^{-1}\underline{\theta} \qquad \text{(C17)}$$

where,

$$\Phi = \begin{bmatrix} \varphi_{ii} & \varphi_{i1} & \varphi_{i2} & \varphi_{i3}\\ \varphi_{1i} & \varphi_{11} & \varphi_{12} & \varphi_{13}\\ \varphi_{2i} & \varphi_{21} & \varphi_{22} & \varphi_{23}\\ \varphi_{3i} & \varphi_{31} & \varphi_{32} & \varphi_{33} \end{bmatrix}$$

$\varphi_{lh} = \varphi(\|p_l - p_h\|)$, $l = i$, 1, 2, and 3.
$\underline{\theta} = [\theta_i \quad \theta_1 \quad \theta_2 \quad \theta_3]^T$

We can now, obtain the gradients from equation (C16)

$$\varphi(\|p - p_h\|) = \exp\left(-\|p - p_h\|^2 / 2\sigma^2\right) = \exp\left(-[(x - x_h)^2 + (y - y_h)^2] / 2\sigma^2\right)$$

$$\Rightarrow \varphi(\|p-p_h\|) = \exp\left(-\frac{(x-x_h)^2}{2\sigma^2}-\frac{(y-y_h)^2}{2\sigma^2}\right) = E_1 \exp\left(-\frac{(x-x_h)^2}{2\sigma^2}\right)$$

where $E_1 = \exp\left(-\frac{(y-y_h)^2}{2\sigma^2}\right)$ $E_2 = \exp\left(-\frac{(x-x_h)^2}{2\sigma^2}\right)$

$$\therefore \frac{\partial\varphi(\|p-p_h\|)}{\partial x} = E_1 \exp\left(-\frac{(x-x_h)^2}{2\sigma^2}\right)\times-\frac{2(x-x_h)}{2\sigma^2} = -\frac{(x-x_h)}{\sigma^2}\varphi(\|p-p_h\|)$$

$$\therefore \frac{\partial\varphi(\|p-p_h\|)}{\partial y} = E_2 \exp\left(-\frac{(y-y_h)^2}{2\sigma^2}\right)\times-\frac{2(y-y_h)}{2\sigma^2} = -\frac{(y-y_h)}{\sigma^2}\varphi(\|p-p_h\|)$$

$$\begin{aligned}&\therefore \nabla\varphi(\|p-p_h\|) = -\sigma^{-2}\varphi(\|p-p_h\|)\cdot\left[(x-x_h) \quad (y-y_h)\right]^T \\ &= d'\varphi(\|p-p_h\|)\end{aligned} \qquad \text{(C18)}$$

where $d' = -\sigma^{-2}\left[(x-x_h) \quad (y-y_h)\right]^T$

$$\nabla\theta(p) \approx -\sigma^{-2}\left[\sum_{h=i,h=1}^{3} w_h(x-x_h)\varphi(\|p-p_h\|) \quad \sum_{h=i,h=1}^{3} w_h(y-y_h)\varphi(\|p-p_h\|)\right]^T \qquad \text{(C19)}$$

Also,

$$\begin{aligned}\frac{\partial^2\varphi(\|p-p_h\|)}{\partial x^2} &= \frac{\partial}{\partial x}\left[-\frac{(x-x_h)}{\sigma^2}\varphi(\|p-p_h\|)\right] \\ &= \frac{\partial}{\partial x}\left[-\frac{(x-x_h)}{\sigma^2}\right]\times\varphi(\|p-p_h\|)+\frac{\partial}{\partial x}\left[\varphi(\|p-p_h\|)\right]\times-\frac{(x-x_h)}{\sigma^2} \\ &= -\sigma^{-2}\varphi(\|p-p_h\|)+-\frac{(x-x_h)}{\sigma^2}\varphi(\|p-p_h\|)\times-\frac{(x-x_h)}{\sigma^2} \\ &= \varphi(\|p-p_h\|)\left\{-\sigma^{-2}+\sigma^{-4}(x-x_h)^2\right\} \\ &= \sigma^{-4}\varphi(\|p-p_h\|)\left\{(x-x_h)^2-\sigma^2\right\}\end{aligned}$$

$$\begin{aligned}\frac{\partial^2\varphi(\|p-p_h\|)}{\partial y^2} &= \frac{\partial}{\partial y}\left[-\frac{(y-y_h)}{\sigma^2}\varphi(\|p-p_h\|)\right] \\ &= \frac{\partial}{\partial y}\left[-\frac{(y-y_h)}{\sigma^2}\right]\times\varphi(\|p-p_h\|)+\frac{\partial}{\partial y}\left[\varphi(\|p-p_h\|)\right]\times-\frac{(y-y_h)}{\sigma^2} \\ &= -\sigma^{-2}\varphi(\|p-p_h\|)+-\frac{(y-y_h)}{\sigma^2}\varphi(\|p-p_h\|)\times-\frac{(y-y_h)}{\sigma^2} \\ &= \varphi(\|p-p_h\|)\left\{-\sigma^{-2}+\sigma^{-4}(y-y_h)^2\right\} \\ &= \sigma^{-4}\varphi(\|p-p_h\|)\left\{(y-y_h)^2-\sigma^2\right\}\end{aligned}$$

$$\frac{\partial^2\varphi(\|p-p_h\|)}{\partial y\partial x}=\frac{\partial}{\partial y}\left[-\frac{(x-x_h)}{\sigma^2}\varphi(\|p-p_h\|)\right]$$
$$=-\frac{(x-x_h)}{\sigma^2}\frac{\partial}{\partial y}\left[\varphi(\|p-p_h\|)\right]$$
$$=-\frac{(x-x_h)}{\sigma^2}\times-\frac{(y-y_h)}{\sigma^2}\varphi(\|p-p_h\|)$$
$$=\sigma^{-4}(x-x_h)(y-y_h)\varphi(\|p-p_h\|)$$

$$\frac{\partial^2\varphi(\|p-p_h\|)}{\partial x\partial y}=\frac{\partial}{\partial x}\left[-\frac{(y-y_h)}{\sigma^2}\varphi(\|p-p_h\|)\right]$$
$$=-\frac{(y-y_h)}{\sigma^2}\frac{\partial}{\partial x}\left[\varphi(\|p-p_h\|)\right]$$
$$=-\frac{(y-y_h)}{\sigma^2}\times-\frac{(x-x_h)}{\sigma^2}\varphi(\|p-p_h\|)$$
$$=\sigma^{-4}(x-x_h)(y-y_h)\varphi(\|p-p_h\|)$$

$$\therefore\nabla^2\varphi(\|p-p_h\|)=\sigma^{-4}\varphi(\|p-p_h\|)\cdot\begin{bmatrix}(x-x_h)^2-\sigma^2 & (x-x_h)(y-y_h)\\(x-x_h)(y-y_h) & (y-y_h)^2-\sigma^2\end{bmatrix}$$
$$=(d'd'^T-\sigma^{-2}I_2)\varphi(\|p-p_h\|)$$
$$=d'\varphi(\|p-p_h\|) \qquad \text{(C20)}$$

where, $d''=(d'd'^T-\sigma^{-2}I_2)$ and I_2 is a 2 by 2 identity matrix.

$$\nabla^2\theta(p)\approx\sigma^{-4}\begin{bmatrix}\sum_{h=i,h=1}^{3}w_h\left[(x-x_h)^2-\sigma^2\right]\varphi(\|p-p_h\|) & \sum_{h=i,h=1}^{3}w_h(x-x_h)(y-y_h)\varphi(\|p-p_h\|)\\ \sum_{h=i,h=1}^{3}w_h(x-x_h)(y-y_h)\varphi(\|p-p_h\|) & \sum_{h=i,h=1}^{3}w_h\left[(y-y_h)^2-\sigma^2\right]\varphi(\|p-p_h\|)\end{bmatrix}$$
(C21)

Therefore, at p_i,

$$\theta_i=\theta(p_i)\approx\sum_{h=i,h=1}^{3}w_h\varphi(\|p_i-p_h\|) \qquad \text{(C22)}$$

$$\nabla\theta_i\approx-\sigma^{-2}\left[\sum_{h=1}^{3}w_h(x_i-x_h)\varphi(\|p_i-p_h\|)\quad\sum_{h=1}^{3}w_h(y_i-y_h)\varphi(\|p_i-p_h\|)\right]^T \qquad \text{(C23)}$$

$$\nabla^2\theta_i\approx\sigma^{-4}\begin{bmatrix}\sum_{h=i,h=1}^{3}w_h\left[(x_i-x_h)^2-\sigma^2\right]\varphi(\|p_i-p_h\|) & \sum_{h=1}^{3}w_h(x_i-x_h)(y_i-y_h)\varphi(\|p_i-p_h\|)\\ \sum_{h=1}^{3}w_h(x_i-x_h)(y_i-y_h)\varphi(\|p_i-p_h\|) & \sum_{h=i,h=1}^{3}w_h\left[(y_i-y_h)^2-\sigma^2\right]\varphi(\|p_i-p_h\|)\end{bmatrix}$$
(C24)

Let $d'_{ih}=-\sigma^{-2}\left[(x_i-x_h)\quad(y_i-y_h)\right]^T$ and $d''_{ih}=(d'_{ih}d'^{\,T}_{ih}-\sigma^{-2}I_2)$. Equation (C22) to (C24) can be expressed in compact matrix form as,

$$\theta_i \approx \sum_{h=i,h=1}^{3} w_i \varphi_{ih} \tag{C25}$$

$$\nabla\theta_i \approx \sum_{h=i,h=1}^{3} w_i d'_{ih} \varphi_{ih} \tag{C26}$$

$$\nabla^2\theta_i \approx \sum_{h=i,h=1}^{3} w_i d''_{ih} \varphi_{ih} \tag{C27}$$

Before we discuss the method to set the steering direction for the node, we would briefly digress on the three basic strategies in statistical decision theory when there is a diversity of redundant unreliable choices that is used to make a decision: Equal Ratio Combining (ECG), “most reliable choice” and Maximal Ratio Combining (MRC). Our objective is to make a choice in manner that minimizes the cost of the decision error. The cost is assumed to be quantifiable. It is also assumed that we could measure the reliability of a choice, usually through a metric.

In ECG strategy, we use the average of all choices to determine our decision. This strategy is best when there is no good measure for reliability or it is difficult to obtain the measure in the current situation. It is sometimes known as the “diversification of risk”.

In “most reliable choice” strategy, if a measure of reliability is available, we would choose the most reliable choice. This strategy works best when there are too many unreliable choices and one of the choices is overwhelmingly reliable than the myriad rest. The obvious risk is that the burden of making a correct decision is placed solely on one choice.

In MRC strategy, this strategy is most appropriate when there is a group of reliable choices and the standard deviation is not too large among them. In this strategy, we use a weighted average of the choices. The average is weighted such that the most reliable choice is given the highest weight. It is considered the best of both strategies above because it provides diversification of risks as well as minimization of the cost of decision error due to the use of unreliable choice.

In our problem, a good measure of reliability is the magnitude of the mean temperature gradient along the path leading to one of the neighbors. This is because it indicates the likelihood that there are minimum or maximum points in that direction. We would use this to help the node decides on the best direction to move. First, we define the mean directional derivative from point, p_i to p_j as follows:

$$D_{ij} = \frac{\theta_j - \theta_i}{\left\| p_j - p_i \right\|} u_{ij} \tag{C28}$$

where u_{ij} is the unit steering directional vector pointing from point i to point j, given as:

$$u_{ij} = \frac{p_j - p_i}{\left\| p_j - p_i \right\|} \tag{C29}$$

The mean directional derivative D_{ij} describes the average temperature change when a point moves along the straight path from point i to j.

To perform the orientation stabilization, we use the mean directional derivative as a criterion to choose a steering direction. Let D_{st} and u_{st} be the directional derivative and the unit directional vector for the steering node respectively. The new change in position with orientation stabilization obtained by the positive projection of the original change in position onto u_{st} is as follows:

$$\Delta p_{st,i} = \left| u_{st}^{T} \Delta p_i \right| u_{st} \tag{C30}$$

Where

$$u_{st} = D_{st} \left\| D_{st} \right\|^{-1} \tag{C31}$$

Here, we describe two approaches use in our simulation to set a steering direction.

Electing a leading node:

In this approach, we “elect” one of the neighboring nodes to be the leading node. Figure C3 shows three vectors radiating from the intermediate node *i*. They represent the 3 possible steering directions for node *i*. The criterion for choosing the leading node is the neighboring node which has the largest magnitude of directional derivative pointing from node *i* to it. That is, the most reliable node.

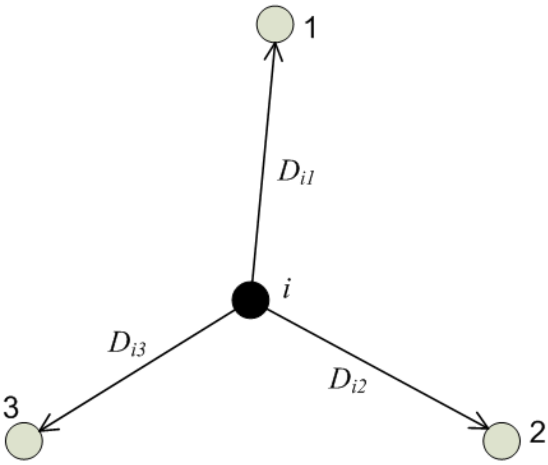


Figure C3: Directional derivatives

Therefore,

$$D_{st} = \max \left\{ D_{ij} \right\}_{j=1}^{3} \tag{C32}$$

Maximal Ratio approach:

The above approach has the risky effect that the burden of leading node *i*, is placed squarely on one node. If the difference in magnitudes between the directional derivative of the leading node and the other nodes is not large, the above method is not the best approach. Since all of them are capable of setting the steering direction albeit to varying degrees, it is better to use the weighted combination of the three directional derivatives to minimize the decision error, especially in the context of unreliable noisy information which we will investigate in the future. For optimal decision, we use MRC (Maximal Ratio Combining) as follows:

$$D_{st} = \sum_{j=1}^{3} \left\| D_{ij} \right\|^2 u_{ij} \tag{C33}$$

Note that this approach still places more bias or weight on a leading node (that is, the node with the largest magnitude in the directional derivative), because of the squares in the weightings. However, the method reduces the risk of setting the wrong steering direction by diversifying the role of setting the steering direction over three neighbors. In fact, if one of the neighbors has an overwhelmingly large directional derivative, this approach is approximately similar to the earlier approach in equation (C32).

Appendix D: Stability Analysis of Optimization

We now examine the stability of the optimization phase of our algorithm.

Fundamentally, for our distributed algorithm, the variation of temperature as observed by a sensor *i* can be decomposed into the following:

$$\partial\theta_i\left(p_i, p_{ref}, a\right) = \partial p_i \frac{\partial\theta_i}{\partial p_i} + \partial p_{ref} \frac{\partial\theta_i}{\partial p_{ref}} + \partial a \frac{\partial\theta_i}{\partial a} \qquad \text{(D1)}$$

The right hand side of Equation (D1) states that the variation of the temperature observed by sensor *i*, is actually the superposition or sum of the variations of temperature represented by the three terms. The first term states that the variation is due to the movement of sensor *i*. The second term states that the variation is due to the movement of the source. The third term states that the variation is due to the varying intensity of the source. (D1) sums up the challenges of the sensor at the individual ground level in a realistic scenario. The temperature change observed by a sensor is actually the aggregate of the three possible abovementioned changes.

The stability of the controlled system under perturbations means its ability to restore back the equilibrium state under environmental perturbations. In our context, the equilibrium state is the state where all the sensors are locked into the optimal stationary positions. In this context, the first factor at RHS is zero since the sensor is not moving. Therefore, there are two types of environmental perturbations that will disturb the system equilibrium state: variation in intensity of the sources of the hotspots and movement of the sources of the hotspots. The first perturbation is trivial because variation in intensity is a form of mathematical scaling where all points in the hotspots are symmetrically scaled by factor. The second perturbation may disturb the stability of the network.

We will first examine visually the stability behavior of the sensors under our original algorithm when the hotspot moves in figure 1 prior to our actual detailed analysis in section 3.3. This examination will set the context and rationale behind the new design of our algorithm. First, we assume that there is no variation of intensity. Hence, when the sensors observe a change in temperature, it is caused by the movement of the hotspot. We ignore the problem of acquiring information and information processing for the sensor and assume that it is always able to obtain good information to compute the gradient and hessian for the Newtonian equation. Let *k* be the time step. $p_i^{(k)}$ be the position of the sensor *i* at time step *k*. We assume that sensor *i* attempts to reach the local maximum for clarity of presentation and without loss of generality as it is one of the optimal positions.

In the topmost plot, the hotspot is initially stationary during the optimization. So at initial step k = 0, sensor *i* is at location A of the hotspot, at k = 1, it moves to location B, finally at k =3, it converges to location D which is the optimal position. We have assumed that all the locations of the hotspot fall within the speed limit of the sensor so that it can reach the location in one time step.

In the middle plot, we show the sensor settles at location D at k=3.

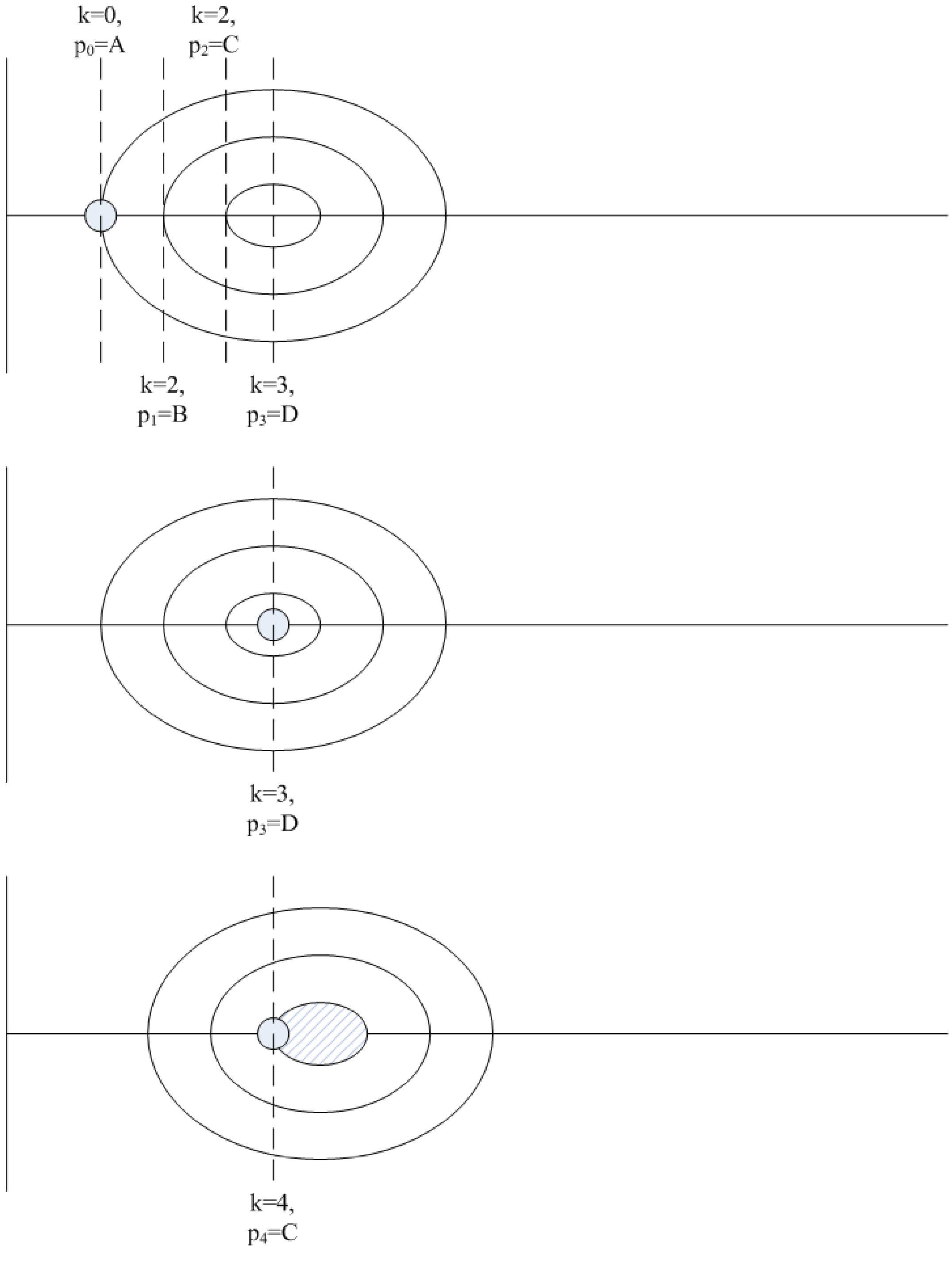


Figure D1: Optimization Scenario prior and after movement of the hotspot

In the last plot, the hotspot starts to move at a constant velocity to the right of the plot such that at k=4, the sensor is back at location C of the hotspot. We assume that the hotspot stops at k=4 and cross-examine the first plot. Notice that the effect of the moving hotpot in this scenario is to rollback the sensor by one iteration step back to k=2. As the intensity of the source does not change, at location C of the hotspot at k =4 in the second plot, the sensor experiences the same condition as location C of the hotspot at k=2 in the first plot. Therefore, it will take exactly one iteration step for the sensor to reach the optimal location D again. In fact, as long as the sensor is inside the shaded region, the sensor will be able to restore back to the original position. Finally, we see in this illustration that the absolute movement of the sensor and the hotspot is not important, what matters is the difference between the positions of the sensor and the hotspot. That is, the relative movements.

Back to the last plot again, if the hotspot stops moving, the sensor will be able to catch up the hotspot in the next time step. Therefore, there is one time step lag in response. Now if the hotspot is moving at a constant velocity in a straight line, by the time the sensor reach the location at k =5, the hotspot will have already moved again such that the sensor ended up at location C of the hotspot again. In other words, in the worst case, when the hotspot continuously moves, there is a constant time lag and hence a separation with the hotspot. For such a scenario to be stable, a sufficient condition is that the separation is bounded and does not increase with time.

To illustrate an unbounded case, we consider the case where the hotspot is moving much faster than the previous discussion such that at k=4, the sensor now ends at location B. On cross-examination with the first plot, the effect is similar to rolling back the sensor by two iteration steps to time k=1. Therefore, under similar assumption discuss previously, the sensor will now require two iteration steps to restore back to location D if the hotspot stops moving at k=4. We now examine the general effect when the hotspot moves at a constant speed. Let T_{step} be the time step. Let the separation between location D of the hotspot and the current position of the sensor be d_k, in units of time step. Let the speed the hotspot be v_h, in units of time step. Similarly the speed travelled by the sensor, v_s is also given in time step.

Therefore, $v_s = 1$ (This is always true)

Let $v_h = n$

At time k = 0, the sensor detects a change of temperature in the environment, by then the hotspot has already moved, so the separation is:

$$\therefore d_0 = T_{step} \times v_h = nT_{step}$$

At time k = 1,

$$\therefore d_1 = T_{step} \times v_h - T_{step} \times v_s + d_0 = nT_{step} - T_{step} + nT_{step} = (2n-1)T_{step}$$

At time k = 2,

$$\therefore d_2 = T_{step} \times v_h - T_{step} \times v_s + d_1 = nT_{step} - T_{step} + (2n-1)T_{step} = (3n-2)T_{step}$$

At time k,

$$\therefore d_k = [(k+1)n - k]T_{step}$$

Consider the first scenario, where the speed of the hotspot is n =1,
$\therefore d_k = T_{step}$ which is stable and bounded when the iteration increases.

However, if n =2,
$\therefore d_k = [k+2]T_{step}$,

The separation is unbounded and increases when k increases because the sensor is unable to catch up.

Therefore, a sufficient condition for stability is to prove that there exist a convergence region (indicated as the shaded region) such that sensor can converge back after been disturbed by small movement in the sources in one iteration step.

The objective is to show that an arbitrary sensor *i* is stable to movement in sources at steady state once it has locked into its optimal position. Let $\theta[p_i(t) - p_{opt}(t)]$ represents the temperature distribution over the entire terrain w.r.t. $p_{opt}(t)$. $p_i(t)$ is the position of sensor *i* at time *t*. *t* is measured when the sensors are in optimal positions. $p_{opt}(t)$ is an optimal position on the distribution that sensor *i* occupied at time $t = 0$. Therefore, the separation: $\lambda(t) = |p_i(t) - p_{opt}(t)|$ at $t = 0$ is 0. If the distribution moves continuously, $\lambda(t) \geq 0$ when $t > 0$. For stability, we are to prove that $\exists \rho > 0$ such that (s.t.) $\lambda(t) < \rho \ \forall t$. Recall that the objective of sensor *i* in our optimization is to locate a position in a region enclosed by the three surrounding neighbors *j*, where $j = 1, 2, 3$ s.t. the volume of the tetrahedron with the four vertices: (x_i, y_i, θ_i) and (x_j, y_j, θ_j) is maximum. The actual volume, *V* is given as:

$$V = \frac{1}{3!}\begin{vmatrix} x_i & y_i & \theta_i & 1 \\ x_1 & y_1 & \theta_1 & 1 \\ x_2 & y_2 & \theta_2 & 1 \\ x_3 & y_3 & \theta_3 & 1 \end{vmatrix} = \frac{1}{3!}V' \qquad \text{(D2a)}$$

Ignoring the factorial and expanding the determinant, *V'* directly using the Taylor's expansion about $p_i(t)$, we have,

$$V' = V_i + \nabla V_i\,(\Delta p_{opt})^{\mathrm{T}} + 0.5(\Delta p_{opt})\nabla^2 V_i\,(\Delta p_{opt})^{\mathrm{T}} + O(\Delta p_{opt}) \qquad \text{(D2b)}$$

$O(\Delta p_{opt})$ is the sum of the higher order terms. V_i, ∇V_i and $\nabla^2 V_i$ are evaluated at $t = 0$ when $p_i(t) = p_{opt}(t)$. Particularly, ∇V_i and $\nabla^2 V_i$ are evaluated by differentiating *V'* in (D2a) w.r.t. $p_i(t)$ in appendix C reproduced here as:

$$\nabla V_i = [B \ \ (-C)] + A\nabla\theta_i \qquad \text{(D3a)}$$

$$\nabla^2 V_i = A\nabla^2\theta_i \qquad \text{(D3b)}$$

Where, *A*, *B*, *C* are given in appendix C reproduced here as:

$$A = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix},\ B = \begin{vmatrix} y_1 & \theta_1 & 1 \\ y_2 & \theta_2 & 1 \\ y_3 & \theta_3 & 1 \end{vmatrix} \text{ and } C = \begin{vmatrix} x_1 & \theta_1 & 1 \\ x_2 & \theta_2 & 1 \\ x_3 & \theta_3 & 1 \end{vmatrix}$$

At $t > 0$, due to the movement of the sources, there is a change in both $p_{opt}(t)$ and the temperature measured at the stationary sensors: θ_i, θ_j. Therefore, there is a change in *V'* as given by (D2a-b). Consider the movement, $\Delta p_{opt} = \rho$ to be sufficiently small so that $O(\Delta p_{opt})$ is negligible and can be ignored. The objective of sensor *i* is to move Δp_{opt} so $p_i(t) = p_{opt}(t)$ once again. Since at $p_{opt}(t)$, *V'* is maximum, we differentiate (D2b) w.r.t. $\Delta p_{opt}(t)$,

$\partial V' / \partial (\Delta p_{opt}) = \nabla V_i + (\Delta p_{opt}) \nabla^2 V_i$

Setting $\partial V' / \partial (-\Delta p_{opt}) = [0\ 0]$,

$$\Delta p_{opt} = -\nabla V_i \, (\nabla^2 V_i)^{-1} \tag{D4a}$$

Substituting (D3a-b) into (D4a), we have,

$$\Delta p_{opt} = \{A^{-1}[\,(-B)\ \ C] - \nabla \theta_i\} \nabla^2 \theta_i \tag{D4b}$$

Comparing with (C15) in appendix C reproduced here as:

$\Delta p_i^{(k)} = [u_{\mathrm{goal}} - \nabla \theta_i^{(k)}]\, K_u$

And note that $u_{goal} = A^{-1}[-B \quad C]$ as defined in appendix C. Under the condition that ρ is small s.t. $O(\Delta p_{opt})$ is negligible and consequently, (D2b) is quadratic, restoring sensor i to its optimal position given by (D4a-b) is equivalent to executing our algorithm for one step. As Taylor's series exists for a continuous distribution, there always exists a stable region centered at $p_{opt}(t)$ with radius, ρ.

Appendix E: Stability Analysis of Tracking Mechanism

We now examine the stability of the tracking mechanism. Let the maximum number of communications hops in the networks be N_{hops}. The two costs are measurement delay, T_θ and communication delay, T_{comm}. T_θ is determined from the specifications of the thermometer. A fast electronic thermometer has a delay that is less than 1s. T_{comm} is given as: $T_{comm} = P \div S_{data}$ where P is the packet length in bits and S_{data} is the the data throughput per node. S_{data} [P2][P3][93] is affected by channel conditions such as: noise, fading, shadowing, the type of MAC protocols, data traffic load, maximum communication rate, etc. In the tracking algorithm (figure 4.12, chapter 4), any member that first locates the hotpot can respond immediately after a delay of T_θ. In the worst case, the last member responses with a delay of T_0 given as: $T_0 = T_\theta + N_{hops}T_{comm}$. Let the maximum speed of the hotspot and the sensor be: V_h, V_s. Let $D^{(k)}$ be the separation between the center of the cluster and the hotspot at k^{th} iteration. Let $T^{(k)}$ be the delay at the k^{th} iteration. Let U be a random variable uniformly distributed at the interval $[-0.5D_s\ 0.5D_s]$, where D_s is the maximum separation of the sensors adjacent to the cluster-head. U represents the uncertainty due to the possibility that the hotspot is at the blind spot at the k^{th} step. In the worst case scenario where the hotspot moves continuously at constant speed, V_h and direction, $D^{(k)}$ is derived by induction as follows:

At $k = 0$, the hotspot starts to move, the delay in the first response is T_0. Due to this delay, by the time the sensor starts to move, the hotspot would have already moved:

$$D^{(0)} = T_0 V_h + U$$

Note that we have examined the worst case by assuming that the hotspot continues to move in the same direction. The assumption here is that there is at least some coverage around the region to detect the approximate location of the hotspot. To close up, the sensor moves at maximum speed, V_s.

$$T^{(1)} = \frac{D_0}{V_s} = \frac{V_h}{V_s} T_0$$

At $k = 1$, one of the members measures and detects movement, it informs others, due to this total delay, in the worst case, the hotspot would have moved.

$$D^{(1)} = (D^{(0)} V_s^{-1} + T_0) V_h + U$$

Therefore to close up again, we have,

$$T^{(2)} = \frac{D^{(1)}}{V_s} = \left(T^{(1)} + T_0\right)\frac{V_h}{V_s} = \left[\frac{V_h}{V_s}T_0 + T_0\right]\frac{V_h}{V_s}$$

$$\Rightarrow T^{(2)} = T_0\left[\left(\frac{V_h}{V_s}\right)^2 + \left(\frac{V_h}{V_s}\right)\right]$$

Similarly at k = 2,

$$D^{(2)} = \left(T^{(2)} + T_0\right)V_h + U$$

$$T^{(3)} = \frac{D^{(2)}}{V_s} = \left(T^{(2)} + T_0\right)\frac{V_h}{V_s} = \left\{T_0\left[\left(\frac{V_h}{V_s}\right)^2 + \left(\frac{V_h}{V_s}\right)\right] + T_0\right\}\frac{V_h}{V_s}$$

$$T^{(3)} = \left[\left(\frac{V_h}{V_s}\right)^3 + \left(\frac{V_h}{V_s}\right)^2 + \frac{V_h}{V_s}\right]T_0$$

$$\therefore T^{(k)} < \left[\left(\frac{V_h}{V_s}\right)^k + \ldots + \left(\frac{V_h}{V_s}\right)^2 + \left(\frac{V_h}{V_s}\right)^1\right]T_0$$

Let $r = V_h V_s^{-1}$. The RHS is recognized to be the sum of Geometric Progression.

$$\therefore T^{(k)} = \frac{T_0 r\left(1 - r^{k+1}\right)}{1 - r}$$

$$\therefore D^{(k)} = \left(T^{(k)} + T_0\right)V_h + U = \left[\frac{T_0 r\left(1 - r^{k+1}\right)}{1 - r} + T_0\right]V_h + U$$

Simplifying,

$$\therefore D^{(k)} = \left[\frac{r\left(1 - r^{k+1}\right)}{1 - r} + 1\right]T_0 V_h + U = \left[\frac{r\left(1 - r^{k+1}\right) + 1 - r}{1 - r}\right]T_0 V_h + U$$

$$= \left[\frac{\left(r - r^{k+2}\right) + 1 - r}{1 - r}\right]T_0 V_h + U = \frac{T_0 V_h\left(1 - r^{k+2}\right)}{1 - r} + U$$

$$\therefore D^{(k)} = \frac{a V_h\left(1 - r^{k+2}\right)}{1 - r} + U$$

$$\Rightarrow D^{(k)} = T_0 V_h\,[1 - (V_h V_s^{-1})^{k+2}] \div (1 - V_h V_s^{-1}) + U \qquad \text{(E1a)}$$

Equation (E1a) is obtained by summing the geometric progression terms. Taking expectation, we obtain the maximum separation, $\sigma_{\max}$ by letting $V_h V_s^{-1} < 1$ and $k \to \infty$,

$$\therefore \mathrm{E}[D^{(k)}] < \sigma_{\max} = T_0 V_h \div (1 - V_h V_s^{-1}) \qquad \text{(E1b)}$$

Therefore, from (E1b), the tracking is stable as long as $V_s > V_h$ because the separation is bounded by $\sigma_{\max}$.

REFERENCE

Reference

1. Y. U. Cao, A. S. Fukunaga and A. B. Kahng, “Cooperative mobile robotics: antecedents and directions,” *Autonomous Robots*, Mar 1997, vol. 4, no. 1, pp. 7-27.
2. L. E. Parker, “Current State of the art in distributed autonomous mobile robotics,” *Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems*, Knoxville, Tennessee, USA, Oct 4-6, 2000.
3. V. Srinivasan, P. Nuggehalli, C. F. Chiasserini and R. R. Rao, “Cooperation in Wireless Ad Hoc Networks,” *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom’03)*, San Francisco, CA, USA, Mar 30-Apr 3, 2003.
4. D. Grundel, R. Murphey and P. M. Pardalos, *Theory and Algorithms for Cooperative Systems, Series on Computers and Operations Research – Volume 4*, World Scientific Publishing Co. Pte. Ltd., 2004.
5. D. P. Bertsekas, *Dynamic Programming and Optimal Control – Volume 1*, Athena Scientific, Belmont, Massachusetts, 1995.
6. M. D. Canon, C. D. Cullum Jr and E. Polak, *Theory of Optimal Control and Mathematical Programming*, McGraw-Hill Series in Systems Science, McGraw-Hill, Inc., 1970.
7. M. Zuniga and B. Krishnamachari, “Optimal Transmission Radius for flooding in Large Scale Sensor Networks,” *Workshop on Mobile and Wireless Networks, MWN 2003*, held in conjunction with the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS), Providence, Rhode Island, USA, May 19-22, 2003.
8. W. K. G. Seah, H. X. Tan, Z. Liu and M. H. Ang Jr., “Multiple-UUV Approach for Enhancing Connectivity in Underwater Ad-hoc Sensor Networks,” *Proceedings of the MTS/IEEE OCEANS 2005*, Washington D.C., USA, Sep 18-23, 2005.
9. W. P. Chen, J. C. Hou and S. Lui, “Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks,” *IEEE Transactions on Mobile Computing*, Jul 2004, vol. 3, no. 3, pp. 258-271.
10. H. Zhang and J. C. Hou, “Maintaining sensing coverage and connectivity in large sensor networks,” *NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, Fort Lauderdale, FL, USA, Feb 2004 (invited paper).
11. N. Li and J. C. Hou, “Localized topology control algorithms for heterogeneous wireless networks,” *IEEE/ACM Transactions on Networking*, Dec 2005, vol. 13, no. 6, pp. 1313-1324.
12. W. Wang, V. Srinivasan and K. C. Chua, “Using mobile relays to prolong the lifetime of wireless sensor networks,” *Proceedings of the 11th Annual international Conference on Mobile Computing and Networking (MobiCom’05)*, Cologne, Germany, Aug 28-Sep 02, 2005.
13. L. L. Dai and V. W. S. Chan, “Proactive Topology Reinforcement of Wireless Networks,” *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2007)*, Hong Kong, Mar 11-15, 2007.

14. Y. Shang, W. Ruml, Y. Zhang and M. Fromherz. “Localization from Connectivity in Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems*, Nov 2004, vol. 15, no. 11, pp. 961-974.
15. J. T. Feddema, C. Lewis and D. A. Schoewald, “Decentralized Control of Cooperative Robotic Vehicles: Theory and Application,” *IEEE Transactions on Robotics and Automation*, Oct 2002, vol. 18, no. 5, pp. 852-864.
16. A. Howard, M. J. Matarić and G. S. Sukhatme, “An Incremental Self-Deployment Algorithm for Mobile Sensor Networks,” *Autonomous Robots*, Sep 2002, vol. 13, no.2, pp. 113-126.
17. J. D. Tan, O. M. Lozano, N. Xi and W. H. Sheng, “Multiple Vehicle Systems for Sensor Network Area Coverage,” *Proceedings of the 5th World Congress on Intelligent Control and Automation*, Hangzhou, P.R. China, Jun 15-19, 2004.
18. J. Cortes, S. Martinez, T. Karatas and F. Bullo, “Coverage Control for Mobile Sensing Networks,” *IEEE Transactions on Robotics and Automation*, Apr 2004, vol. 20, no. 2, pp. 243-255.
19. H. Choset and K. Nagatani, “Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization,” *IEEE Transactions on Robotics and Automation*, Apr 2001, vol. 17, no. 2, pp. 125-137.
20. B.R Bellur, M.G. Lewis and F.L. Templin, “An Ad-Hoc Network for Teams of Autonomous Vehicles,” *Proceedings of First Annual Symposium on Autonomous Intelligence Networks and Systems*, University of California, Los Angeles, California, USA, May 8-9, 2002.
21. S. Haykin, *Neural Networks – A Comprehensive Foundation*, Prentice-Hall, Inc., 2nd Edition, 1999.
22. M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2003.
23. J.G. Proakis, *Digital Communications*, 4th Edition, McGraw-Hill.
24. A. Muqattash and M. Krunz, “CDMA-based MAC protocol for wireless ad hoc networks,” *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing (MobiHoc '03)*, Annapolis, Maryland, USA, Jun 1-3, 2003.
25. S. Xu and T. Saadawi, “Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?” *IEEE Communications Magazine*, Jun 2001, vol. 39, no. 6, pp. 130-137.
26. L. Kleinrock and F. A. Tobagi, “Packet Switching in Radio Channels: Part II – the Hidden Terminal Problem in Carrier Sensing Multiple Access and the Busy Tone Solution,” *IEEE Transactions on Communications*, Jun 1975, vol. COM-23, no. 12, pp. 1417-1433.
27. “*IEEE 802.11b 1999: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*,” published by IEEE, approved 16 September 1999, reaffirmed 12 June 2003.
28. S. S. Basu and A. Chaudhuri, “Self-Adaptive Topology Management for Mobile Ad-Hoc Network,” *The Institution of Engineers (India) Journal-ET*, Jul 2003, vol. 84, pp. 7-13.
29. M. Krunz, A. Muqattash and Sung-Ju Lee, “Transmission power control in wireless ad hoc networks: challenges, solutions and open issues,” *IEEE Network*, Sep-Oct 2004, vol. 18, no. 5, pp. 8-14.

30. C. E. Perkins and P. Bhagwat, "DSDV Routing over Multihop Wireless Network of Mobile Computers," *Ad Hoc Networking*, Addison Wesley, 2nd Edition, 2001, pp. 53-74.
31. D. B. Johnson, D. A. Maltz and J. Broch, "DSR – The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," *Ad Hoc Networking*, Addison Wesley, 2nd Edition, 2001, pp. 139-172.
32. C. E. Perkins and E.M. Royer, "The Ad-Hoc On-Demand Distance Vector Protocol," *Ad Hoc Networking*, Addison Wesley, 2nd Edition, 2001, pp. 173-219.
33. Z.J. Haas and M.R. Pearlman, "ZRP - A Hybrid Framework for Routing in Ad-Hoc Networks," *Ad Hoc Networking*, Addison Wesley, 2nd Edition, 2001, pp. 221-253.
34. M. S. Corson and V. Park, "Link Reversal Routing," *Ad Hoc Networking*, Addison Wesley, 2nd Edition, 2001, pp. 255-298.
35. J. J. Garcia-Luna-Aceves, "Bandwidth-Efficient Link-State Routing in Wireless Networks," *Ad Hoc Networking*, Addison Wesley, 2nd Edition, 2001, pp. 323-350.
36. Y.B. Ko and N.H. Vaidya, "Location-Aided Routing (LAR) in mobile ad hoc networks," *Wireless Networks*, 2000, vol. 6, no. 4, pp. 307-321.
37. RFC 3636, *Optimized Link State Routing Protocol (OLSR)*. T. Clausen, Ed., P. Jacquet, Ed.. October 2003. (Status: EXPERIMENTAL)
38. H. Hassanein and A. Zhou, "Routing with Load Balancing in Wireless Ad-Hoc Networks," *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation Of Wireless and Mobile Systems*, Rome, Italy, Jul 16-21, 2001.
39. B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom '00)*, Boston, Massachusetts, USA, Aug 6-11, 2000.
40. P. H. Hsiao, "Geographical Region Summary Service for Geographical Routing," *ACM SIGMOBILE Mobile Computing and Communications Review, Best poster papers from MOBIHOC 2001*, vol. 5, no. 4, Oct 2001, pp. 25-39.
41. X. Hong and M. Gerla, "Dynamic Group Discovery and Routing in Ad Hoc Networks," *Proceedings of the 1st Annual Mediterranean Ad Hoc Networking Workshop (Med-hoc-Net 2002)*, Sardegna, Italy, Sep 4-6, 2002.
42. R. G. Ogier, "Efficient Routing Protocols for Packet-Radio Networks Based on Tree-Sharing," *Proceedings of the 6th IEEE International Workshop on Mobile Multimedia Communications (MOMUC'99)*, San Diego, CA, USA, Nov 17-19, 1999.
43. S. Basagni, I. Chlamtac, V. R. Syrotiuk and B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, Dallas, Texas, USA, Oct 25-30, 1998.
44. S. Basagni, I. Chlamtac and V. R. Syrotiuk, "Dynamic Source Routing for Ad Hoc Networks Using the Global Positioning System," *Proceedings of the IEEE Wireless Communications and Networking Conference 1999 (WCNC'99)*, New Orleans, LA, USA, Sep 21-24, 1999.
45. R. Vincent, P. Berry, A. Agno, C. Ortiz and D. Wilkins, "Teambotica: a robotic framework for integrated teaming, tasking, networking, and control," *Proceedings of the Autonomous Agents and Multiagent Systems Conference*, Melbourne, Australia, Jul 14-18, 2003.
46. K.J. O'Hara, "Active routing for mobile robot teams," Technical report, College of Computing, Georgia Institute of Technology, Nov 2002.

47. H.G. Nguyen, H.R. Everett, N. Manouk and A. Verma, "Autonomous Mobile Communication Relays," *SPIE Proceedings 4715: Unmanned Ground Vehicle Technology IV,* Orlando, FL, USA, Apr 1-5, 2002.
48. H.G. Nguyen, N. Farrington and N. Pezeshkian, "Maintaining Communication Link for Tactical Ground Robots," *Proceedings of the AUVSI Unmanned Systems North America 2004,* Anaheim, CA, USA, Aug 3-5, 2004.
49. Q. Li and D. Rus, "Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00)*, Boston, Massachusetts, USA, Aug 6-11, 2000.
50. W. Zhao and M.H. Ammar, "Message Ferrying: Proactive Routing in Highly-partitioned Wireless Ad Hoc Networks," *Proceedings of the 9th IEEE Workshop on Future Trends of Distributed Computing Systems*, San Juan, Puerto Rico, May 28-30, 2003.
51. J. Redi and B. Welsh, "Energy Conservation for Tactical Robot Networks," *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Atlantic City, New Jersey, USA, Oct 31- Nov 3, 1999.
52. D. Goyal and J. Caffery, Jr., "Partitioning Avoidance in Mobile Ad Hoc Networks Using Network Survivability Concepts," *Proceedings of the 7th International Symposium on Computers and Communications (ISCC'02)*, Taormina/Giardini Naxos, Italy, Jul 1-4, 2002.
53. G.J. Pottie and W.J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, May 2000, vol. 43, no. 5, pp. 51-58.
54. F. Xue and P.R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, Mar 2004, vol. 10, no. 2, pp. 169-181.
55. H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Transactions on Communications*, Mar 1984, vol. 32, no. 3, pp. 246-257.
56. Y.-C. Cheng and T. G. Robertazzi, "Critical Connectivity Phenomena in Multihop Radio Models," *IEEE Transactions on Communications*, Jul 1989, vol. 37, no. 7, pp. 770-777.
57. T.K. Philips, S. S. Panwar and A. N. Tantawi, "Connectivity Properties of a Packet Radio Network Model," *IEEE Transactions on Information Theory*, Sep 1989, vol. 35, no. 5, pp. 1044-1047.
58. M. D. Penrose, "On k-connectivity for a geometric random graph," *Random Structures and Algorithms*, vol. 15, no. 2, Sep 1999, pp. 145-164.
59. C. Bettstetter, "On the Minimum Node Degree and Connectivity of a Wireless Multihop Network," *Proceedings of the 9th ACM Annual international Conference on Mobile Computing and Networking* (*MobiCom '03)*, Lausanne, Switzerland, Jun 9-11, 2003.
60. L. Kleinrock and J. Silvester, "Optimum Transmission Radii for Packet Radio Networks or Why six is a magic number," *IEEE National Telecommunications Conference*, Dec 1978, pp. 4.3.1-4.3.5.
61. N. Li and J. C. Hou, "Topology control in heterogeneous wireless networks: problems and solutions," *Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*, Hong Kong, Mar 7-11, 2004.
62. W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *Proceedings of the*

33rd Annual Hawaii International Conference on System Sciences, Wailea Maui, Hawaii, USA, Jan 4-7, 2000.

63. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, Feb 2003, vol. 11, no. 1, pp. 2-16.
64. C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *Proceedings of the 6th ACM Annual international Conference on Mobile Computing and Networking (MobiCom '00)*, Boston, Massachusetts, USA, Aug 06-11, 2000.
65. N. Li and J. C. Hou, "Improving Connectivity of Wireless Ad Hoc Networks," *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, San Diego, CA, USA, Jul 17-21, 2005.
66. J.A. Costa, N. Patwari and A.O. Hero III, "Achieving high-accuracy distributed localization in sensor networks", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, USA, Mar 18-23, 2005.
67. Y. Shang, W. Ruml, Y. Zhang and M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, Nov 2004, vol. 15, no. 11, pp. 961-974.
68. S. Meguerdichian, S. Slijepcevic, V. Karayan and M. Potkonjak, "Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure," *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '01)*, Long Beach, CA, USA, Oct 4-5, 2001.
69. P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, Mar 2000, vol. 46, no. 2, pp. 388-404.
70. F. Xue, L. L. Xie and P. R. Kumar, "The transport capacity of wireless networks over fading channels," *IEEE Transactions on Information Theory*, Mar 2005, vol. 51, no. 3, pp. 834-847.
71. P. R. Kumar, "Towards an information theory of large networks: An achievable rate region," *IEEE Transactions on Information Theory*, Aug 2003, vol. 49, no. 8, pp. 1877-1894.
72. O. Leveque and I.E. Telatar, "Information-theoretic upper bounds on the capacity of large extended ad hoc wireless networks," *IEEE Transactions on Information Theory*, Mar 2005, vol. 51, no. 3, pp. 858 - 865.
73. M. Grossglauser and D.N.C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, Aug 2002, vol. 10, no. 4, pp. 477-486.
74. S. A. Jafar, "Too Much Mobility Limits the Capacity of Wireless Ad Hoc Networks," *IEEE Transactions on Information Theory*, Nov 2005, vol. 51, no. 11, pp. 3954-3965.
75. Y. Chen, Q. A. Zeng and D. P. Agrawal, "Performance of MAC Protocol in Ad Hoc Networks," *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'03)*, Orlando, Florida, USA, Jan 19-23, 2003.
76. T. Camp, J. Boleng and V. Davies, "A Survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, Sep 2002, vol. 2, no. 5, pp. 483-502.

77. E.M. Royer, P.M.M. Smith and L.E. Moser, "An analysis of the optimum node density for Ad hoc Mobile Networks," *Proceedings of the IEEE International Conference on Communications (ICC 2001)*, Helsinki, Finland, Jun 11-14, 2001.
78. RFC 3561, *Ad hoc On-Demand Distance Vector (AODV) Routing*. C. Perkins, E. Belding-Royer, S. Das. July 2003.(Status: EXPERIMENTAL)
79. D. Kotz, C. Newport and C. Elliot, "The mistaken axioms of wireless-network research," Dartmouth College, Computer Science Technical Report TR2003-467, July 18, 2003.
80. A. Cerpa, J. Wong, L. Kuang, M. Potkonjak and D. Estrin, "Statistical Model of Lossy Links in Wireless Sensor Networks," *Proceedings of the 4th International Conference on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, California, USA, Apr 25-27, 2005.
81. W. Kiess and M. Mauve, "A Survey on Real-World Implementations of Mobile Ad-Hoc Networks," *Ad Hoc Networks*, Apr 2007, vol. 5, no. 3, pp. 324-339.
82. L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part I – Carrier Sensing Multiple Access Modes and Their Throughput Delay Characteristics," *IEEE Transactions on Communications*, Dec 1975, vol. COM-23, no. 12, pp. 1400-1416.
83. L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part IV – Stability Considerations and Dynamic Control in Carrier Sensing Multiple Access," *IEEE Transactions on Communications*, Oct 1977, vol. COM-25, no. 12, pp. 1103-1120.
84. F. Tobagi and V. Hunt, "Performance Analysis of Carrier Sensing Multiple Access with Collision Detection," *Computer Networks*, 1980, vol. 4, pp. 245-259.
85. Y. C. Tay and K. C. Chua, "A Capacity Analysis for the IEEE 802.11 MAC Protocol," *ACM/Baltzer Wireless Networks*, Mar 2001, vol. 7, no. 2, pp. 159-171.
86. F. Hwang and D. S. Du, *Computing in Euclidean Geometry*, World Scientific, 1995.
87. G. W. Zobrist and C. Y. Ho, *Intelligent Systems and Robotics* , CRC Press, 2000.
88. L. M. Bishop, *Essential Mathematics for Games and Interactive Applications: Programmer's Guide*, Elsevier, 2004.
89. C. M. Zierhofer, "Numerical Adaptive Delta Modulation – A technique for digital Signal Representation in ASIC Applications," *Proceedings of IEEE 6th Mediterranean Electrotechnical Conference*, LJubljana, Slovenia, May 22-24, 1991.
90. Gutmann, T. Weigel and B. Nebel, "Fast, accurate, and robust self-localization in the robocup environment," *Proceedings of the 3rd Robot World Cup Soccer Games and Conferences held in conjunction with 16th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, Jul 31-Aug 6, 1999.
91. B. Bradie, *A Friendly Introduction to Numerical Analysis*. Pearson Prentice Hall, International Edition, 2006.
92. E. Kreyszig, *Advanced Engineering Mathematics*. John Wiley & Sons, 7th Edition, 1993.
93. I. Rhee, A. Warrier, M. Aia and J. Min, "ZMAC:a Hybrid. MAC for Wireless Sensor Networks," *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*, San Diego, California, USA, Nov 2-4, 2005.
94. A. Barberis, L. Barboni and M. Valle, "Evaluating Energy Consumption in Wireless Sensor Networks Applications," *Proceedings of the 10th Euromicro*

Conference on Digital System Design Architectures, Methods and Tools (DSD 2007), Lübeck in Germany, Aug 29-31, 2007.

95. Z. Zhou, J. H. Cui and S. Zhou, "Localization for Large-Scale Underwater Sensor Networks," *Proceedings of the 6th International IFIP-TC6 Networking Conference*, Atlanta, GA, USA, May 14-18, 2007.
96. H. C. Perle and B. Rechberger, "Throughput Analysis of a Decentralized RLAN Based on Asynchronous DS/CDMA using Random Periodic Spreading Sequences," *Wireless Personal Communications*, Aug 1998, vol.**7,** no. 2-3, pp. 197-211.
97. B. Liu, P. Brass, O. Dousse, P. Nain and D. Towsley, "Mobility improves coverage of sensor networks," *Proceedings of the 6th ACM international Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05)*, Urbana-Champaign, IL, USA, May 25-27, 2005.
98. X.L. Wu, L. Shu, J. Wang, J.S. Cho and S.Y. Lee, "Energy-efficient Deployment of Mobile Sensor Networks by PSO," *Proceedings of the 8th Asia Pacific Web Conference International Workshops: XRA, IWSN, MEGA, and ICSE*, Harbin, China, Jan 16-18, 2006.
99. G.L. Wang, G.H. Cao and T. L. Porta, "Proxy-based sensor deployment for mobile sensor networks," *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '04)*, Fort Lauderdale, Florida, USA, Oct, 25-27, 2004.
100. G.L. Wang, G.H. Cao, T.L. Porta and W.S. Zhang, "Sensor Relocation in Mobile Sensor Networks," *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, Miami, FL, USA, Mar 13-17, 2005.
101. I.I. Hussein and D.M. Stipanovic, "Effective Coverage Control for Mobile Sensor Networks With Guaranteed Collision Avoidance," *IEEE Transactions on Control Systems Technology*, Feb 2004, vol. 15, no. 4, pp. 642-657.
102. X.L. Wu, L. Shu, M. Meng, J.S. Cho and S.Y. Lee, "Potential Field Based Hierarchical Structure for Mobile Sensor Network Deployment," *Proceedings of the American Control Conference (ACC '07)*, New York City, USA, Jul, 9-13, 2007.
103. S. Poduri and G. S. Sukhatme, "Constrained Coverage for Mobile Sensor Networks," *IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, Apr 26-May 1, 2004.
104. Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, Feb 2004, vol 3, no. 1, pp. 61-91.
105. J.M. Chen, S.J. Li and Y.X. Sun, "Novel Deployment Schemes for Mobile Sensor Networks," *Sensors*, Nov 2007, vol. 7, pp. 2907-2919.
106. X.L. Wu, L. Shu, M. Meng, J.S. Cho and S.Y. Lee, "Coverage-Driven Self-Deployment for Cluster Based Mobile Sensor Networks," *Proceedings of the 6th IEEE International Conference on Computer and Information Technology (CIT '06)*, Seoul, Korea, Sep 20-22, 2006.
107. P. Ogren, E. Fiorelli and N. E. Leonard, "Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment," *ACM Transactions on Embedded Computing Systems*, Feb 2004, vol. 3, no. 1, pp. 61-91.
108. C.H. Caicedo-N and M. Zefran, "Balancing Sensing and Coverage in Mobile Sensor Networks: Aggregation Based Approach," *Proceedings of the IEEE*

International Conference on Robotics and Automation, Workshop on Collective Behaviors inspired by Biological and Biochemical Systems, Rome, Italy, Apr 10-14, 2007.

This page is intentionally

left blank.