Integrated Mobile and Static Sensing for Target Tracking

Oliver Kosut, Andrey Turovsky, John Sun,
Matthew Ezovski, and Lang Tong
School of Electrical and Computer Engineering
Cornell University, Ithaca, NY 14853
Email: {oek2,at329,jzs3,gme8,lt35}@cornell.edu

Gene Whipps
Army Research Laboratory
Email: gwhipps@arl.army.mil

***Abstract*— Networks of low-power, low-cost, and widely distributed wireless sensor nodes are being envisioned and developed for many military applications including surveillance and localization. However, due to energy and communication constraints, combinations of static nodes (such as unattended ground sensors (UGS)) and mobile nodes (such as robotic ground vehicles (RGV) and unmanned aerial vehicles (UAV)) are being used to expand or fill in coverage areas and reduce energy costs.**

A multi-chaser target tracking scenario is developed where mobile nodes, or chasers, use information provided by static nodes to infer the position of an intruder in the sensor grid. In large-scale sensor networks with mobile access, the communication range of static nodes may be limited by terrain and signal transmission energy. A drop-box static sensor architecture is developed to aid data propagation in such limited range networks. Chasers will "drop off" previously collected information about the target and retrieve new information stored in the static sensors. An algorithm is presented which takes advantage of this particular architecture. It is used to demonstrate that the drop-box nature of the static nodes improves performance.

I. Introduction

Wireless sensor networks are an always-growing area of interest in signal processing and wireless communications. Current research areas include optimal routing [1], medium access control [2], energy efficiency, data fusion, and consensus problems [3], among many others. Mobile agents continue to play an increasing role in sensor networks and require architectural support.

Sensor networks have applications in many industries such as health care, transportation, safety, security, and military. For example, sensor networks have been used to monitor oil pipelines, reduce aircraft weight by eliminating heavy wiring harnesses, and, in some cities, help police monitor and respond to criminal activity. There are a number of applications within the military alone. The United States Army envisions transforming from large, individual platforms to more agile, networked systems of systems. Components of those systems include networked unmanned ground sensors, ground vehicles, and aerial vehicles. Such systems can aid in a variety of military actions, peacekeeping operations, and humanitarian relief. These unmanned systems can provide valuable intelligence, surveillance, and reconnaissance to soldiers and up the chain of command from a safe stand-off distance, whether in combat or disaster area support. Other military applications mirror those in the commercial worlds, such as inventory control and logistical support.

There are many constraints to consider in unmanned systems, including but not limited to size, mission lifetime, and, of course, cost. These constraints typically limit sensing modalities to passive ones (e.g., acoustic, infrared, magnetic, seismic, and visual). The sensing range of such sensors is limited to 10s to 100s of meters, although in some specific applications and under ideal conditions, ranges can be on the order of 1000 meters. In urban environments, sensing ranges can be less than 100 meters. The specific mission may only require short-range sensing.

Due to the lack of accessible wireline infrastructure, unmanned systems must be powered through a combination of batteries, solar power, and power scavenging. The required mission lifetime of some of today's static, unmanned ground sensors has been more than 30 days. Mobile systems tend to have significantly shorter mission lifetimes due to the additional power needed for locomotion. However, mobile systems can be recharged or refueled once they have returned from a mission.

These constraints suggest that it would be costly and impractical to have a large number of high power, high cost mobile units patrolling a field of interest. We consider a more efficient architecture, in which a small number of these mobile units accomplish some task by employing a much larger network of stationary ground sensors that have long lifetime but short range and low power. These static nodes have such low communication range that they cannot interact directly with other static nodes, only nearby mobile nodes. As such, the static nodes function not as network relays, but as drop-boxes where mobile nodes can deposit information for other mobile nodes to retrieve. We explore the potential of this network architecture, in particular with regard to target tracking, and demonstrate that the drop-box memory of the static nodes improves performance.

The problem of tracking a moving object through a static sensor field has been addressed using a variety of distributed and centralized solutions for different sensing modalities [4]. Most of these approaches focus on static topologies, and also on propagating tracking information to either a single node or to all of the static nodes. In addition, static nodes often are able to communicate with each other, allowing for constant flow of information throughout the sensor network. The introduction of mobile agents with limited communications ability for tracking and propagating data is a major focus of this paper. [5] presents the idea of mobile nodes "dropping" data at points in a field for use by other agents; we apply this idea, but in the context of tracking in a field of static sensors, as a variation to the Sensor Network with Mobile Access (SENMA) Testbed [6].

We consider a scenario in which an intruder to be tracked moves through the sensor field. Static nodes take range-to-target readings, which are distributed through the network as the mobile nodes pick up the data and bring it to other static nodes, which then send it on to further mobile nodes, and so on. The goal is for all the mobile nodes to accumulate enough readings in order to estimate the target's motion and then intercept it. Thus, if one mobile node determines the target's location, it is not necessarily the best strategy for it to track down the target, it must distribute its information through the network to assist other mobile nodes in meeting it as well.

Our main contribution is a distributed algorithm to solve this problem designed to take advantage of the drop-box nature of the static nodes. In this algorithm, each mobile node executes three modes of operation. In the first mode, it has no information about the target's position, so it wanders the field looking to receive a measurement from a static node. After it collects one, it goes into mode 2, in which it aims to distribute information through the network to help other mobile nodes. Once it discovers that either all the other mobile nodes are informed, or the information is sufficiently spread through the network, it moves into mode 3. In this final mode, it uses its data to estimate the target's location and then drive toward it. We demonstrate that the cooperation between mobile nodes in this algorithm—mode 2, in particular—improves the performance. We do this primarily with simulation, but the algorithm has also been implemented on a testbed with small robots acting as the mobile nodes and acoustic communication. The testbed is described in detail below.

The rest of the paper is organized as follows. In Section II, we give a formal description of the tracking problem. In Section III, we describe our algorithm. In Section IV, we present some simulation results showing demonstrating algorithm performance. In Section V, we describe the features and methodology of the testbed. Finally, in Section VI we conclude.

II. Problem Statement

The field is made up of a large number n_{static} of static nodes, a smaller number n_{mobile} of mobile nodes, and a single target. Each static node i has a fixed location x_i somewhere in the field, known to all other nodes. In principle, these locations could be arbitrary, though in our simulations and testbed they are arranged in a grid with distance between adjacent nodes d_{grid}. At the start of the run, the mobile nodes are placed randomly on the field and the target enters the field from a random direction. The goal is for the mobile nodes to find and meet the target with the help of the static nodes. All the nodes execute a purely distributed algorithm, so a given node may base its decisions only on knowledge it has received itself, and communication between nodes is entirely short range as described below. We seek to minimize the time it takes for every mobile node to come within a radius r_{goal} of the target. Thus it is not enough for a single mobile node to discover the whereabouts of the target and track it down, it must also somehow share its information with the other mobile nodes. Since a given mobile node has no initial knowledge about the location of the others, it should execute an algorithm that informs potentially far-away nodes without wasting time if the nodes are close.

We consider a turn-based scenario. We use this discrete time model because it is easier to implement both in simulation and on the testbed. Additionally, the turn-based scenario is more realistic, assuming communication constraints on the nodes. That is, we want to approximate a finite bandwidth communication channel. Even though we allow unlimited communication in our model, the turn-based scenario requires that a pair of nodes can only exchange information once per turn, so exchanging several messages back and forth or relaying messages through many nodes takes multiple turns. Finally, we choose fairly small step sizes so that the behavior closely resembles that of continuous time.

Each turn is made up of the following events:

1) Static nodes take measurements.
2) Mobile nodes communicate to static nodes.
3) Static nodes communicate to mobile nodes.
4) Mobile nodes and the target move.

Each of these events is described in more detail below.

We assume that each static node is equipped with a range sensor. Every time step, if the target is within a radius r_{sensor} of the static node, it is informed of the range to the target without error. Note that since a static node receives no specific information about the target's position if it is out of range, the only static nodes that will initially have any measurements are those that are close to the target's path through the field. However, the drop-box concept allows mobile nodes to pick up measurements from certain static nodes and deliver them to other static nodes, causing measurements to be relayed across the network.

Communication occurs only between mobile nodes and static nodes. We assume that static nodes are very low power devices, so two static nodes may not communicate directly to each other. In principle, two mobile nodes could communicate directly to each other, but we do not allow that, mainly for simplicity. In any case, if two mobile nodes were close enough to communicate, they would also both be close enough a static node to communicate via that node, so the direct mobile-to-mobile communication would be unnecessary.

We assume that a mobile node and a static node may communicate exactly when they are within a distance r_{comm} of each other. If they can communicate, then they may send an arbitrary amount of information. We restrict communication to mobile nodes first and then static nodes to simulate bandlimited communication, as discussed above. Thus, even if several nodes form a connected component via the communication range, they cannot all communicate directly, only over several time steps as the information is relayed. The shared information is not necessarily confined to relayed range measurements taken by static nodes. It may, for example, include information about the state of other nodes, or what the mobile node plans to do next. We also assume that all mobile and static nodes have an arbitrarily large memory, so they need never discard any information they receive.

TABLE I
SUMMARY OF PARAMETERS USED TO SPECIFY THE TRACKING PROBLEM

Symbol	Description	Value
n_{static}	Number of static nodes	20
n_{mobile}	Number of mobile nodes	3
d_{grid}	Distance between adjacent static nodes in grid	4 ft
r_{sensor}	Sensor range	4 ft
r_{comm}	Communication range	4 ft
s_{mobile}	Maximum distance traveled by mobile node each turn	1 ft
s_{target}	Distance traveled by target each turn	0.5 ft

Every turn, mobile nodes may travel up to a distance s_{mobile} in any direction. They may base their decision on where to move on any information they have received from static nodes. We also assume that mobile nodes know their exact current location.

Once the target enters the field, it moves a fixed distance s_{target} in its predetermined direction every time step. We assume that the mobile nodes know the value of s_{target}. This allows the target's future position to be easily predicted. We use this deterministic linear motion model because we wish to think of the target not as an adversary attempting to avoid capture by the mobile nodes, but rather as a neutral stranger who was wandered into the field and is observed by the nodes. Thus the target neither specifically makes it easy for the mobile nodes to catch it nor does it move intentionally to make tracking difficult.

All of the problem parameters are summarized in Table I, including the values that we use in our testbed and simulations. The values given are nominal—they are used in the testbed, but can be changed in simulation (especially n_{static}, as described in Section IV).

III. TRACKING ALGORITHM

Our tracking algorithm is based on the observation that it takes a very small number of measurements to accurately predict the target's position. This is primarily because the target's motion is entirely deterministic given its initial position and direction of motion. In addition, even a single measurement, while not enough to predict the target's position, does let a mobile node know the general area where the target was at a certain time. Thus, a mobile node possessing just one measurement can move toward the static node that originally took that measurement, and will with certainty find more measurements which will allow it to determine the target's exact position. Thus, the algorithm is designed to make sure as many nodes as possible possess at least one measurement.

Let i be a node, either mobile or static. For each other node j, node i keeps a bit b_{ij} representing whether node i believes node j has at least one measurement. These bits, in addition to the measurements themselves, are primarily what static and mobile nodes remember and share with one another. Whenever a static node takes a measurement or a mobile node receives a measurement from a static node, it sets its own bit. Whenever a mobile node and a static node share information, they perform binary ORs on each of their bits. In this manner, information about the state of other possibly-distant nodes is distributed through the network.

The algorithm is such that each mobile node is in one of three modes at any given time. A mobile node cannot move to a previous mode once it enters a later one. The three modes are summarized here, then described in detail below:

1) A mobile node is in this mode if it has received no measurements from static nodes. The mobile node wanders the field until it receives a measurement.
2) A mobile node i is in this mode if it possesses a measurement, but $b_{ij} = 0$ for some other mobile node j. The mobile node attempts to disperse information throughout the field by moving toward static nodes j for which $b_{ij} = 0$.
3) A mobile node i is in this mode if $b_{ij} = 1$ for all mobile nodes j. It uses its measurements to estimate the trajectory of the target, then moves to meet the target in the shortest possible time.

A. Mode 1: Wandering the Field

In mode 1, the goal of the mobile node is to come within communication range of a static node holding a measurement. Since a node in mode 1 has no information about where the target is, except that it has not come close to any static nodes with which the mobile node has communicated, it should go on a trajectory that takes it near as many static nodes as possible as quickly as possible. At the same time, if two mobile nodes start near one another, their trajectories should not be too similar, as that would repeat effort. Furthermore, there could be other mobile nodes that have moved on to mode 2, dispersing information throughout the field, so the strategy should interact with the strategy for mode 2 such that a node in mode 1 comes across the deposited information by one of the other sensors.

It may be possible to design a complex strategy that takes into account all these requirements, perhaps by dividing the field into regions in an ad hoc manner, such that each node is assigned one region to explore. However, we choose a simpler approach, leaving the design of a better strategy open for future work. In our algorithm, each mobile sensor chooses an initial random direction to move. It continues to move in that direction until it comes near the edge of the field, off which it "bounces." That is, it flips its direction of motion so as to go back into the field. This takes the nodes on trajectories that bring them near many static nodes, and it is likely that different mobile nodes will follow different trajectories.

The downside of this approach is that sometimes the random choices of initial direction cause the mobile nodes not to come close to the target for awhile, or to move together instead of apart, but we have found that it works fairly well despite its simplicity.

B. Mode 2: Dispersing Information

In mode 2, a mobile node holds a measurement, but suspects that at least one other mobile node does not. The goal is to employ the drop-box architecture by dropping information off at static nodes throughout the field so that other mobile nodes will pick them up. Note that even though a mobile node in mode 2 is attempting to relay information to other mobile nodes, it knows nothing about where those mobile nodes are, so it focuses on delivering information to the static nodes instead.

The method is to move toward the nearest static node j such that $b_{ij} = 0$, and there are no other static nodes j' with $b_{ij'} = 1$ where the distance between nodes j and j' is no more than r_{near}. We impose the second requirement because it is unnecessary to inform every single static node, since the mobile nodes that we are trying to inform—the ones still in mode 1—will be moving around the field. Thus, we need only inform a scattering of static nodes to make sure that the uninformed mobile nodes will come near one holding a measurement. An important parameter then is the necessary separation r_{near}. This parameter must be large enough that a node in mode 2 does not waste time visiting too many static nodes, but not so large that there are not enough informed static nodes for a mobile node in mode 1 to quickly come across one. It will also be based on the size of the field, since a smaller fraction of the static nodes need be informed in a larger field. We find that a good values for r_{near} are between a third and a half of the size of the field.

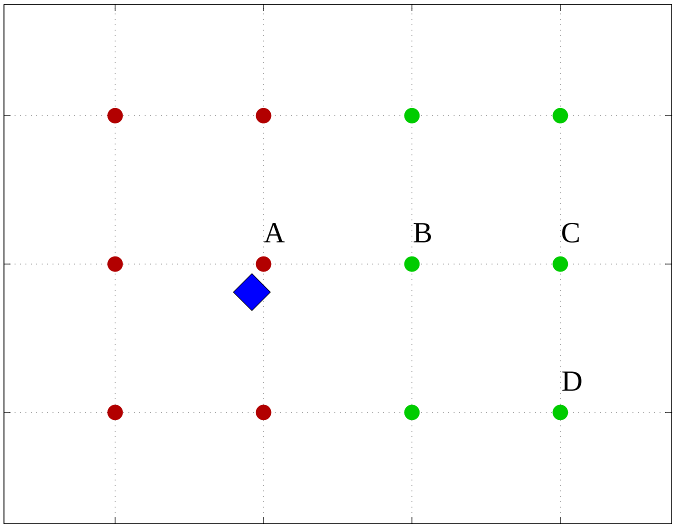

Fig. 1. Example of a mobile node's behavior in mode 2. The diamond is the mobile node. The small circles are the static nodes. Dark static nodes are ones that the mobile node believes possess a measurement, where the light ones are not. If $r_{\text{near}} = d_{\text{grid}}$. The mobile node would move in the direction node C until coming into communication range of node B, when node B would become informed and the mobile node would then move toward node D.

Figure 1 illustrates an example of mode 2 behavior. Suppose $r_{\text{near}} = d_{\text{grid}}$. Thus, two static nodes are within necessary separation range if they are adjacent in the grid. In particular, node A, which is informed, is close enough to node B, so node B cannot be the goal even though it is uninformed. The nearest static node to the mobile node further than r_{near} from an informed node is node C, which will be the mobile node's goal. However, before the mobile node gets within communication range of static node C, it will come within communication range of static node B. When it does, static node B will become informed, which means that static node C will no longer be greater that r_{near} away from all informed static nodes, so the mobile node will change its goal to node D. This sort of behavior is typical—a mobile node in mode 2 never actually reaches its goal, it will always communicate with a closer static node first, causing its goal to change. However, moving toward further away nodes causes the mobile node's path to be more direct.

C. *Mode 3: Meeting the Target*

In mode 3, the mobile node now knows that the other mobile nodes have at least one measurement, so it must perform the final task of estimating the target's location and then meeting up with it. Since we have assumed zero error in the sensors, this is a relatively straightforward problem. However, the fact that we use range measurements presents some minor issues, as the target's position cannot always be determined exactly given a set of measurements.

To find the target, a mobile node needs to estimate its initial position as well as its angle of motion. Depending on the number and source of the measurements that the node possesses, it may not be able to solve for these parameters. Since there are three parameters, if it has fewer than three measurements, it cannot reduce the possible solutions to a finite set. In addition, if all the measurements are from the same sensor, or are all from the same time step, there are again an infinite number of possibilities. In any of these cases, the mobile node moves toward the mean position of the static nodes from which it has measurements, in the hopes that it will find more measurements with which to solve for the target's position.

Furthermore, due to the nonlinear nature of range measurements, there are many cases where the number of solutions for the parameters given the measurements is finite but greater than one. For example, with two measurements from one static node and one measurement from another, there are four possible solutions. In such a case, we use lack of measurements to pare down the solution set. That is, if the mobile node has communicated with a static node and learned that it did not receive a measurement at a particular time step, that means that the distance to the target must have more than r_{sensor}. This potentially allows us to reduce the number of solutions. If after this reduction there is still more than one solution, the mobile node finds the nearest static node that will be able to distinguish the remaining possible solutions, then moves toward this static node until within communication range.

Once the mobile node is able to get a unique solution for the target's initial position and angle of motion, it projects the target's course. Based on the relative speeds of the target and the mobile node, it moves toward the location where it will meet the target in the shortest possible time.

IV. Simulation Results

Figure 2 shows the mean number of steps to find the target over 100 simulation runs each of two different algorithm methods with a variety of different field sizes. The first method is the tracking algorithm exactly as described above. The second is meant as a benchmark with which to compare the performance of the full algorithm. It is the same algorithm, except without any cooperation between mobile nodes—that is, mobile

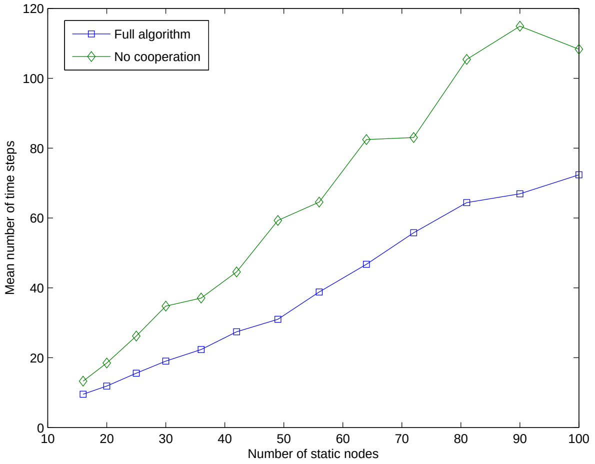

Fig. 2. Mean time to find target for both the full algorithm and a version without cooperation between mobile nodes over a variety of field sizes.

nodes do not transmit information they have collected to static nodes, and they go directly from mode 1 to mode 3 once they collect a measurement. In effect, each mobile node operates independently, collecting information just from the static nodes as if the other mobile nodes were not there at all. The field sizes range from 16 to 100 static nodes. They are arranged in a grid, always either square or nearly square, so the only values of n_{static} used are of the form n^2 or $n(n+1)$ for integers n. All other parameters were set to the values given in Table I.

V. Testbed Overview

We use the SENMA testbed [6], specifically developed for testing networked sensor algorithms, to receive a performance metric in a real, nondeterministic environment. Due to the relatively small size of the testbed, long-range wireless communication between static and mobile sensors is modeled using acoustic signals. There are several advantages to using acoustic communication, including a similar propagation characteristic to RF with much higher attenuation to suit a smaller environment. Experiments have also shown signal fading due to multiple wall reflections to be similar to that of RF. Figure 3 shows a photograph of the testbed in operation.

A Dell Optiplex PC serves as the computational backbone of the testbed and controls the overall flow of a simulation by using the Wi-fi Local Area Network (WLAN). All computation-intensive work is transferred to this PC to be processed. A computer vision system is used to retrieve absolute vehicle positions in the field. The computer processes image data from two Lumenera cameras to find specially colored squares of paper to determine the absolute positions of all mobile sensors and intruders. The intruder position is used to simulate sensor data and the position of the mobile sensors is used in the control algorithms. Twenty Dell Axim X50 PDAs are used as the static sensors, due to their wireless (internal wireless card) and acoustic (microphone and speaker) communication capabilities. These PDAs, arranged in a 4x5 grid, are suspended from the ceiling to prevent collisions with mobile nodes. Three Acroname Palm Pilot Robot Kits (PPRK) are used as mobile nodes. Their unique three-wheel triangular design gives them the ability to turn in place and their flat structure provides space for a PDA, which serves as a wireless communication relay between the onboard microcontroller and PC. The Mega32 microcontroller allows for precise control of the servo motors that control the wheels. This allows us to accurately move the vehicles throughout the field. A fourth PPRK is used as an intruder. It is independent of the rest of the system and information about it gained by mobile sensors comes from the static sensor grid.

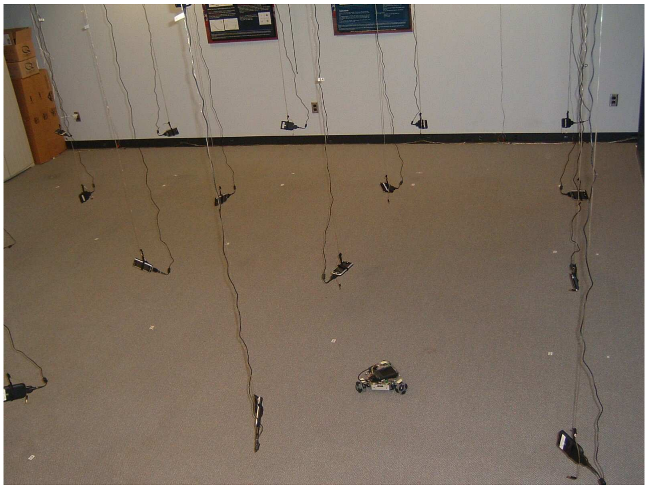

Fig. 3. The SENMA testbed in operation. Static nodes hang from the ceiling while a mobile node drives below them.

VI. Conclusion

We make no claims that the benchmark scenario with no cooperation described in Section IV is the optimal strategy for this problem for mobile nodes acting on their own, but we believe that it still allows a good comparison with the full algorithm. Thus the fact that the tracking algorithm with cooperation performs demonstrably better, especially for larger fields, illustrates that the dropbox use of these static nodes—even though they are low power, low communication range, potentially unreliable devices—can improve the performance of a team of mobile robots.

There are many ways in which the tracking algorithm could be improved, some of which have been discussed above. In particular, mode 1 (wandering the field) and mode 2 (dispersing information) are currently fairly simplistic algorithms meant to approximate the right strategy. Much more complicated algorithms could doubtless be designed that allow the mobile nodes to more efficiently work together to first survey the field for the target, and then inform each other of a sighting as quickly as possible. However, we believe that the overall three mode structure is the correct framework with which to solve this problem.

References

[1] K. Akkaya and M. Younis, "A Survey of Routing Protocols in Wireless Sensor Networks," in *Elsevier Ad Hoc Network Journal*, vol. 3/3, pp. 325–349, 2005.

[2] D. P. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, 1992.

[3] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, Vol.49, Iss.9, pp. 1520–1533, Sept. 2004.

[4] T. Pham and H. C. Papadopoulos, "Distributed tracking in AD-HOC sensor networks," *IEEE/SP 13th Workshop on Statistical Signal Processing*, pp. 1226–1231, July 2005.

[5] J. Friedman, et al. "Ragobot: A New Hardware Platform for Research in Wireless Mobile Sensor Networks," in *Information Processing in Sensor Networks*, April 2005.

[6] Y. Sung, Z. Yang, and L. Tong, "Multitone acoustic sensor network with mobile access: An experimental testbed," in *Proc. of MILCOM*, (Monteray, CA), Oct. 2004.