2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Relocation of Hopping Sensors

Zhiwei Cen
Google Inc.
1600 Amphitheatre Pky
Mountain View, CA 94043, USA

Matt W. Mutka
Dept. of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824, USA

Abstract— **Hopping sensors are a class of mobile sensors whose mobility design are inspired by creatures such as grasshoppers. Such sensors are able to maintain mobility in harsh terrains but may lack movement accuracy of those sensors that are powered by wheels. We examine the opportunities and challenges for utilizing the mobility of low cost hopping sensors to ensure coverage and maintain energy efficiency within a sensing field. We focus on the problem of transporting a number of hopping sensors from multiple sources to a destination. Probabilistic methods are used to contain the movement inaccuracies along the hopping course. We also consider the impact of wind under an aerodynamic setting. Two transport schemes are designed to minimize the number of hops needed while considering other constraints, such as sustaining the capability of relocating sensors within the whole network. In one scheme we use upper and lower hopping limits to apply the network mobility constraints. The other scheme uses a balancing coefficient to construct a new optimization target to meet the requirement of path optimality and network mobility dynamically. Simulation results show that both schemes work well regardless of the wind factors, while the dynamic scheme is also shown to be resilient to topological changes of the network.**

Index Terms— **Hopping sensors, Sensor relocation, Sensor networks, Energy efficiency**

I. Introduction

Large scale sensor networks call for automatic deployment and maintenance. Mobile sensors are important for facilitating sensor deployment and maintaining coverage and communication during runtime. Hopping sensors are a class of mobile sensors with a bionic mobility design that is inspired by creatures, such as grasshoppers. In order to move to a different location, a hopping sensor throws itself high and toward the destination direction. Hopping sensors are capable of maintaining mobility in terrains where wheeled mobility is not possible. Mobility powered by the concepts of hopping are widely discussed in the area of planetary exploration [1], [2]. Bergbreiter et al., proposed to use rubber bands to power small jumping sensors [3]. The work of Feddema, et al. [4] described a prototype minefield hopping robot.

In this paper, we assume that the hopping sensors are capable of adjusting the hopping direction. A fixed propelling force for hopping is also assumed. In reality, the hopping range may vary due to the physical load of the sensor, different terrain conditions, and local aerodynamic settings. In order to facilitate routing, the sensors are also assumed to have a localization capability.

Hopping enabled mobility can be used to facilitate the sensor network deployment and maintain coverage and connectivity during runtime. In the lifetime of a sensor network, it often happens that the sensors in a certain area are depleted faster than other areas. Those areas that have depleted sensors are called sensing holes or sensing wells. A well planned deployment may allocate redundant sensors in the field, thus when sensing wells are detected, sensors can be migrated from those regions that have redundant sensors (referred as suppliers or sources) to sensing wells. We consider the problem of transporting a certain number of hopping sensors from multiple sources to a detected sensing well.

To facilitate sensing well detection and the matching of sources to the well, we organize the sensor network field as a set of clusters. Quorum or broadcast based approaches can be used to match the supplier and consumer clusters. We model the hopping inaccuracy using a multivariate normal distribution. In the transporting stage we employ cascaded movement to speed the migration and argue the distance between relay clusters is crucial in determining the routing path length. We propose two schemes to minimize the total number of hops needed to fill a certain sensing well, while at the same time maintaining the relocation capability of the whole network. One scheme uses upper and lower relay edge hop limits, while the other uses a balancing coefficient to construct a new optimization target dynamically. Simulation results indicate that both algorithms are effective in balancing the requirement of path optimality and maintaining the relocation capability of the network. The dynamic algorithm is also shown to be resilient to topological changes of the network.

The paper is organized as follows. Section II briefly covers related work. In section III, we study the multi-hop landing of the sensors under a multivariate normal distribution model. Under such premises we present two route planning schemes in section IV. We evaluate the performance of the route planning schemes using Matlab simulation and the results are presented in section V. Section VI provides the conclusions.

II. Related Work

Much work has utilized the mobility of general mobile sensors to facilitate deployment, coverage maintenance and

This paper is supported in part by the National Science Foundation under grants No. CNS-0721441 and CNS-0524163.

improving energy efficiency [5], [6], [7], [8], [9], [10].

The use of hopping or flipping based sensors in sensor network deployment is also studied in [9]. However, the work in [9] does not explicitly consider the characteristics of hopping sensors we investigate in this paper, and more importantly does not build the relocation scheme based on the understanding of the movement of large number of multi-hop sensors. The work of Wang, et al. [8] also comes close to our paper in terms of the objectives and system settings. They follow a similar approach for migrating multiple wheeled sensors between a single source and destination within the sensor network. In contrast, we focus on the problem of relocating hopping based sensors, which have a different mobile and dynamical model compared with wheeled sensors. In addition, we consider the case where a single source cannot provide sufficient sensors and multiple sources are needed.

III. System Model

Hierarchical models are widely used in sensor network design. The work of Zou, et al. [11] assumed that a cluster head was available to coordinate the sensor deployment based on virtual forces. Wang, et al. [8] also used a grid based quorum method to match the source and destination grids. In our hopping sensor network model, we assume that the sensors are organized into clusters. A cluster head assumes the responsibility of evenly distributing the sensors, detecting sensor deficiency, and detecting redundant sensors within the cluster.

There are well established methods to detect redundant sensors in a sensor network based on computing Voronoi diagrams [12]. The clusters that are detected to have redundant sensors mark themselves as sources and identify themselves through the cluster head over the sensor network. The problem of detecting sensing holes or sensing wells are also studied [5], [13], [14]. Matching the sources and holes could be complicated in the presence of multiple sources and holes. The work of Chellapan, et al. reduces the matching problem in the sensor deployment stage as a multi-commodity maximum flow problem [9]. We take the approach of filling the sensing holes one by one and consider the problem of migrating hopping sensors from multiple sources to one destination.

A. Normal Distribution Model of Inaccurate Hopping

Compared with wheeled mobile sensors, hopping sensors lack the accuracy of movement. We use a multivariate normal distribution model to determine the landing accuracy of multiple sensors that hop together. Based on the model, we derive the upper bound of the number of hops needed to migrate a number of sensors to a target location. The result is also used to derive the number of sensors that can reach the consumer from the sources for a given demand when astray sensors are considered. A sensor is regarded as astray when its landing point is outside a range of the projected point.

In a two dimensional case, the landing accuracy is characterized by the displacement between the targeted location, represented by vector T, and the actual landing location represented by vector L. The displacement vector D can be represented as $D = T - L$. D is modeled by a two dimensional normal distribution with mean $(0, 0)$, standard deviation (σ, σ), and correlation ρ. Note that we used a nondiscriminating standard deviation vector for the two dimensions. The probability density function of D in terms of (x, y) can be given as

$$f(x,y) = \frac{1}{2\pi\sigma^2\sqrt{1-\rho^2}} \exp\left(-\frac{x^2+y^2-2\rho xy}{2\sigma^2(1-\rho^2)}\right). \quad (1)$$

In order to model the uncertainty of the landing location, we define an acceptable landing area as a disk S around the targeted location. As shown in Figure 1, the radius of the disk $n\sigma$ is determined by multiplying a factor n to the standard deviation σ. For the migration distance that requires one hop only, the acceptable landing area lies within the target location of the sensor. For migrations that need more than one hop, if the sensor lands within the acceptable landing area, the sensor is recharged for another hop towards the target; otherwise the sensor is deemed as astray and will seek to join a local cluster.

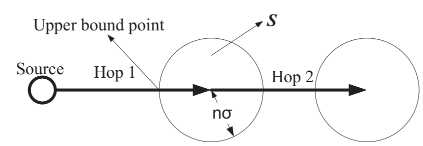

Fig. 1. Modeling the Hopping Accuracy Using Normal Distribution

The probability that the hopping sensor lands in the acceptable landing area S can be represented as

$$P(S) = \iint_S f(x,y)\,dx\,dy. \quad (2)$$

It is known that Equation (2) does not have a closed form solution. Numerical methods can be used to calculate the probability [15].

Based on the probability that the hopping sensor lands in the acceptable landing area, we derive the upper bound of the number of hops needed for a migration distance l. Assuming the distance covered by one sensor hop is r, the lower bound of the number of hops N needed to cover distance l is $N = \frac{l}{r}$. The upper bound of N, however, is determined by the situation when, for every hop, the sensor lands at the farthest point from the target and on the direct line segment between the source and the target location, as indicated in Figure 1. This is equivalent to reducing the hopping range of the sensor from r to $r - n\sigma$. Thus the upper bound of the number of hops given distance l and hopping range r can be given by

$$N_u = \frac{l}{r - n\sigma}. \quad (3)$$

Assume we know the number of sensors demanded by the target cluster and the number of sensors that are tasked to the current source is E_t. As will be discussed in section IV-A, the consumer estimates the number of astray sensors and demands some additional sensors to cover those sensors. For each hop, the number of hopping sensors will decrease due to straying. For the first hop, the number of hopping sensors is E_t. For the second hop, the number is $E_t P(S)$, etc. Thus the total number of hops needed for all sensors is

$$H = \sum_{i=0}^{N_u - 1} E_t P^i(S) = E_t \sum_{i=0}^{N_u - 1} P^i(S) = \frac{1 - P^{N_u}(S)}{1 - P(S)} E_t. \tag{4}$$

B. Aerodynamical Model of Hopping under Air Disturbance

Unlike wheeled movement, the motion of hopping is more susceptible to air disturbance. We need to estimate the impact of air disturbance so that we can adjust the hopping orientation properly when wind is detected. We also model the influence of air disturbance on the hopping range of the sensors so that we can obtain an accurate estimation of the number of hops needed to traverse a certain path.

We assume the hopping sensor has a fixed propelling impetus. In order for the sensor to have the longest horizontal hopping range, the sensor should jump towards a tilted upward direction. When there is no air disturbance, the optimal upward angle that provides the longest horizontal range is $\alpha = 45°$. Assuming a flat terrain, the horizontal distance traveled by the sensor in the air is

$$r = t|\mathbf{v_h}| = t|\mathbf{v}|\cos\alpha = \frac{|\mathbf{v}|^2 \sin 2\alpha}{g}. \tag{5}$$

We assume that the air disturbance only influences the horizontal velocity of the sensor. If the velocity of the wind in the horizontal direction is $\mathbf{v_w}$, the real velocity of the hopping sensor in the horizontal direction is $\mathbf{v_{hreal}} = \mathbf{v_h} + \mathbf{v_w}$. If the direction of the targeted location is given in $\mathbf{v_{hreal}}$, the hopping direction of the sensor can be given as

$$\mathbf{v_h} = \mathbf{v_{hreal}} - \mathbf{v_w}. \tag{6}$$

Under the influence of air disturbance, the number of hops needed for a certain distance will also change. When there is no wind, the hopping range of the sensor can be given in Equation (5). Since the wind only influences the horizontal velocity of the sensor, the flying time of the sensor remains the same when there is air disturbance. Thus the hopping range of the sensor with wind can be represented as

$$r_w = t|\mathbf{v_{hreal}}|. \tag{7}$$

In practice, we use the proportion of the hopping range under air disturbance to the normal hopping range to facilitate calculation of the number of hops needed. According to Equation (5) and Equation (7), we have

$$\frac{r_w}{r} = \frac{|\mathbf{v_{hreal}}|}{|\mathbf{v_h}|}. \tag{8}$$

Assuming the number of hops needed for a trip of length l under the normal condition is N, the number of hops needed under air disturbance N_w can be obtained through

$$\frac{N_w}{N} = \frac{\frac{l}{r_w}}{\frac{l}{r}} = \frac{r}{r_w} = \frac{|\mathbf{v_h}|}{|\mathbf{v_{hreal}}|} = \frac{|\mathbf{v_h}|}{|\mathbf{v_h} + \mathbf{v_w}|}. \tag{9}$$

In conclusion, when the air disturbance generates a horizontal velocity of $\mathbf{v_w}$, the hopping direction of the sensor can be given using Equation (6), while the number of hops needed for a fixed range compared with the normal case can be given using Equation (9). When considering the upper bound of the number of hops needed (denoted as N_{wu} and N_u), Equation (9) is modified to

$$\frac{N_{wu}}{N_u} = \frac{\frac{l}{r_w - n\sigma}}{\frac{l}{r - n\sigma}} = \frac{r - n\sigma}{r_w - n\sigma} = \frac{|\mathbf{v_h}|t - n\sigma}{|\mathbf{v_{hreal}}|t - n\sigma}. \tag{10}$$

Similar to Equation (4), the upper bound of the total number of hops for all sensors $H_w$with air disturbance can be given by

$$H_w = \frac{1 - P^{N_{wu}}(S)}{1 - P(S)} E_t, \tag{11}$$

where E_t is the number of sensors that are tasked to the current source.

In order to obtain the wind information needed by the model, anemometers are deployed within the sensor network. The model implied in Equation (11) is only applicable to the situations where the wind flow has a predictable and relative stable nature. For settings where the air flow follows random changes, a pre-deployment can be used to measure the air flow impacts.

IV. Route Planning in Hopping Sensor Migrations

Route planning involves migrating the required number of sensors from multiple sources to the destination cluster. The process includes identifying supplying and consuming sensor clusters, matching those clusters and selecting the optimal route to migrate the sensors.

A. Matching of the Consumer and Suppliers

Assume that for supplier i ($i = 1 \cdots M$), the number of sensors it can provide is p_i and the distance between supplier i and the consumer is l_i, the number of sensors that can reach the consumer can be estimated as

$$E_{est} = \sum_{i=1}^{M} p_i P^{\frac{l_i}{r - n\sigma}}(S). \tag{12}$$

In Equation (12), $P(S)$ is the rate of the sensors that land in the acceptable range, as defined in Equation (2). $\frac{l_i}{r-n\sigma}$ is the number of hops needed between the consumer and supplier i, which is similar to Equation (3). When considering the wind factor, the number of hops estimation can be done with the aid of Equation (8). It is noted that the number of hops estimated is based on the assumption that the sensors migrate from the supplier to the consumer directly, without using relay clusters. The concept of relaying is explained in section IV-B. When relay clusters are used, the number of hops needed might be larger. In the latter case, an augmenting factor should be applied to Equation (12).

Assuming the number of sensors the consumer needs is d, we compare d and E_{est}. If $E_{est} \geq d$ the suppliers can satisfy the consumer and a matching process can be concluded. Otherwise the consumer has to wait for additional suppliers.

B. Finding the Optimal Migration Path

In this section we discuss the algorithms to migrate the hopping sensors from multiple suppliers to the consumer cluster. We assume the routing planning process is performed at the supplier clusters. Each cluster has a topological view of the network.

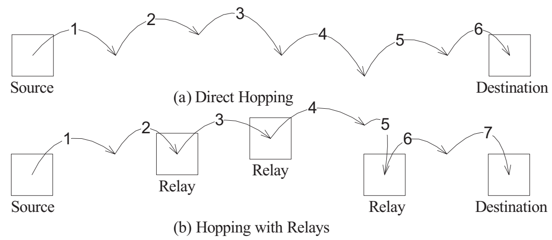

Fig. 2. Two Hopping Strategies

Two possible migration strategies can be used. The first strategy is to migrate the sensors directly from the supplier to the consumer, as shown in Figure 2(a). The second strategy, however, uses intermediate clusters as relay clusters. Sensors can hop simultaneously starting from the supplier and the relay clusters, as illustrated by Figure 2(b). Although sensors can be assumed to be able to make multiple hops, the number of hops one sensor can make (denoted as k_{hard}) is not unlimited before refueling or recharging. For example, the prototype proposed by Feddema, et al [4] has a limit of 100 hops before refueling.

Direct hopping can always guarantee a straight route but hopping with relays cannot. Furthermore, direct hopping gives sensors more freedom to adjust their hopping direction on the fly if the landing point is off the route, which will ultimately save the number of hops. For example, in Figure 2(b), hop number 5 and 6 could have been combined into one hop if the relay is not used. Thus it is desirable to use straight hopping for relatively short migration paths but using relay hopping for longer paths. For relay hopping, the tradeoffs require us to set a limit on both the maximum and minimum distance between the relay clusters. We set the upper limit k to maintain the relocation ability of the network. A lower limit j is set to avoid unnecessary hops introduced by excessive number of relays.

We use the topology and availability information of the clusters to construct a graph. The vertices set is composed of the clusters. Initially, all the available clusters, including the suppliers and the consumer, are connected with edges. The weight of the edges are set to be the estimated number of hops needed, as indicated in Equations (3) and (10). Our objective is to find an optimal path to migrate the sensors from the suppliers to the consumer. In the first route planning algorithm, we first apply the upper and lower limits of the edge weights by deleting the edges whose weights fall outside the range of $[j, k]$. Then Dijkstra algorithm is called for the sources to find the shortest paths from the suppliers. If the shortest path still cannot be found for some suppliers, we conclude the supplier is separated from the consumer and the consumer is informed to search for new suppliers.

The algorithm stated above tries to maintain even mobility consumption among clusters and minimal number of hops using manually set limits. However, for the clusters whose edges are within the limits, the mobility consumption and number of hops of the edges can still vary from edge to edge. In order to migrate the sensors from the suppliers to the destination optimally, and at the same time maintain even mobility throughout the clusters, we modify the path searching algorithm by adding an additional adjusting process. In the adjusting process, we try to minimize a fraction of the sum of the weights along the path and a fraction of the difference of the maximum and minimum weights of the edges along the path.

The second algorithm is described in Algorithm 1. The algorithm takes the graph G, edge weight set W, source node set S, destination node t and the hard limit k_{hard} as inputs. For each source node, the algorithm first runs the Dijkstra algorithm to get a shortest path solution where the only constraint is the hard limit. After that, the solution is refined through several iterations. The minimization target becomes a fraction of the sum of the weights along the path and a fraction of the difference of the maximum and minimum weight of the edges along the path, which is expressed as $(1-\gamma)w_{sum} + \gamma(w_{max} - w_{min})$ as shown in line 11. Here $\gamma\,(0 \leq \gamma \leq 1)$ is a coefficient to determine the fraction of the difference of the maximum and minimum weight of the edges to be considered in the algorithm. The algorithm terminates when it cannot find a better solution given the constraints and the last known good solution is taken as the final solution.

V. Performance Evaluations

In order to validate the proposed models and algorithms, we simulated a hopping sensor network environment under Matlab. A topology generator is used to generate the simulation environment in a $1000m \times 1000m$ rectangular area. The simulation environment can be regarded as a sensor network where some clusters are available to serve as relays, while others are not. The available clusters, as well as the source and destination clusters, are plotted. Two clusters are connected if the number of hops needed to migrate between them is less than k_{hard}.

The parameters regarding hopping sensor dynamics and sensor distribution are shown in Table I. We derive the hopping range, initial velocity and hops capacity based on the Feddema prototype minefield hopping sensor model [4]. We assume zero correlation of the X and Y direction in the landing model.

We first evaluated the performance of Algorithm 1 where the path optimality and remaining network mobility are controlled through the setting of soft hop limits. Considering the fact that most clusters may be involved in sensing tasks and not available as relays, the simulation topology is generated as a sparse graph. In a sparse environment, it is safe to use a small lower hop limit j without jeopardizing the optimality of the path since the degree of each node is small.

Algorithm 1 MinHopsExt(G, W, S, t, k_{hard})

$p' \leftarrow \emptyset, (w'_{sum}, w'_{max}, w'_{min}) \leftarrow (+\infty, +\infty, 0)$
$W' \leftarrow$ Delete edges whose weights are larger than k_{hard} in W
for all s in S **do**
  $(G_1, W'_1) \leftarrow (G, W')$
  loop
    $p \leftarrow$ Dijkstra(G_1, W'_1, s, t)
    if $p \neq \emptyset$ **then**
      $w_{sum} \leftarrow$ Get the sum of edge weights using p
      $(e_{max}, e_{min}) \leftarrow$ Get the edges with maximum and minimum weight using p
      $(w_{max}, w_{min}) \leftarrow$ Get the weights of (e_{max}, e_{min})
      if $(1-\gamma)w_{sum} + \gamma(w_{max} - w_{min}) < (1-\gamma)w'_{sum} + \gamma(w'_{max} - w'_{min})$ **then**
        $(w'_{sum}, w'_{max}, w'_{min}) \leftarrow (w_{sum}, w_{max}, w_{min})$
        $p' \leftarrow p$
        Delete edges e_{max} and e_{min} in W'_1 and G_1
      else
        Use the solution in p' and break loop
      end if
    else
      Use the solution in p' and break loop
    end if
  end loop
end for

Parameter	Value
Number of Total Hops Capable without Refueling (k_{hard})	100
Hops Capable per Sensor Initially	100
Magnitude of Initial Horizontal Velocity ($\lvert \mathbf{v_h} \rvert$)	$7m/s$
Hopping Range (r)	$3m$
Sensors per Cluster Initially Deployed	200
Acceptable Landing Area Radius ($n\sigma$)	$0.6m$
Acceleration due to Gravity (g)	$9.8m/s^2$

TABLE I
PARAMETERS USED IN HOPPING SENSOR SIMULATION

In the first simulation we fixed the lower limit j to be 1 and evaluated the number of hops needed to migrate a group of sensors and the mobility capacity of the network after the migration when the upper hop limit k changes (Figure 3).

The path optimality is measured using the average number of hops consumed by each migrated sensor (referred as $\bar{H}_m$, Figure 3(a)). In the simulation, a group of 10 sensors ($E_t = 10$) are requested by the consumer. Thus $E_t\bar{H}_m$ is the total number of hops needed in the migration. The remaining mobility of the network is measured using two metrics (Figure 3(b) and (c)), the standard deviation (STD) of the number of remaining hops per sensor throughout the network, and the minimum of the number of remaining hops per sensor throughout the network.

In the simulation we evaluated the metrics under three circumstances, the normal condition, the condition with a positive wind force and the condition with a negative wind force. In the simulated topology, a positive force is modeled as a wind velocity that is $45°$ counterclockwise of the positive X axis. Likewise, a negative force is modeled as a wind velocity that is $225°$ counterclockwise of the positive X axis. In either case, the magnitude of the wind velocity $\lvert \mathbf{v_w} \rvert$ is set to be $1/10$ of the magnitude of the initial horizontal hopping velocity $\lvert \mathbf{v_h} \rvert$.

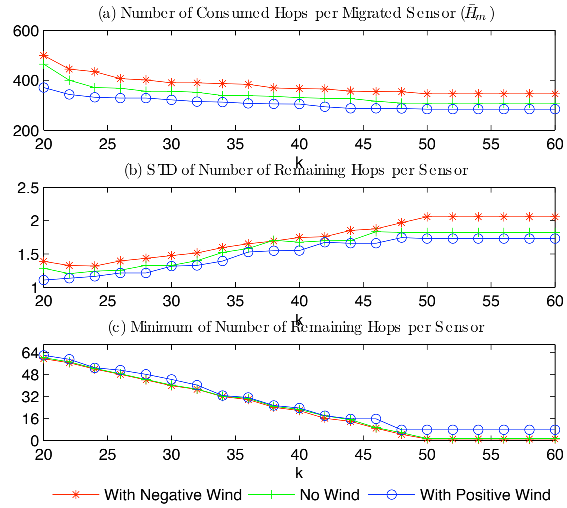

Fig. 3. Number of Consumed Hops per Migrated Sensor and Mobility Metrics Vs. Upper Soft Limit k

The simulation results of Figure 3 indicates that the value of the upper soft limit k has a significant influence on the path optimality and remaining network mobility. Wind factors also slightly influence the results, but not as dramatic as k. The influence of wind factors on the minimum of the number of remaining hops per sensor (Figure 3(c)) is even negligible, although we can see that the case with positive wind has higher number of remaining hops. This is because that the impact of k is so significant that a larger range of Y axis has diminished the slight differences of the three cases.

Larger k will yield better paths with fewer hops, but the STD of the number of remaining hops per sensor is also larger, which indicates imbalance of the mobility distribution throughout the network. The number of remaining hops per sensor also confirms the trend with a steep curve. When k is over a certain value (50 in this case), the parameters will not change since the constraints of the topology is reached. When k is too small, however, a path might not be found due to the topology constraints.

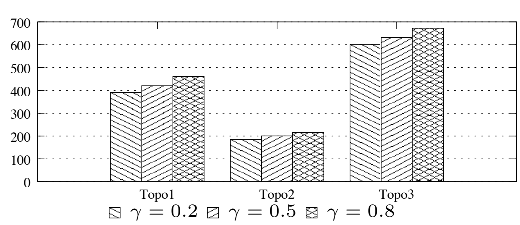

Fig. 4. Number of Consumed Hops per Migrated Sensor ($\bar{H}_m$) under Different Topologies and γ

In the second simulation we evaluated the performance of Algorithm 1, where a dynamic approach is used to balance the need of path optimality and network mobility distribution. We use scaling coefficient γ to determine the optimization target (as shown in Line 11 of Algorithm 1). In this simulation we evaluated the effect of the γ coefficient over the average number of hops consumed in a path and the mobility metrics of the network after the migration is performed.

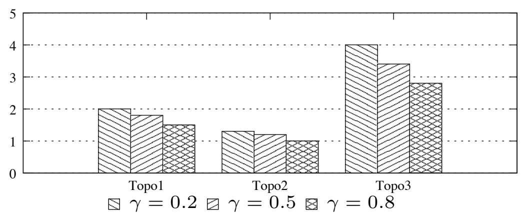

Fig. 5. STD of Number of Remaining Hops per Sensor under Different Topologies and γ

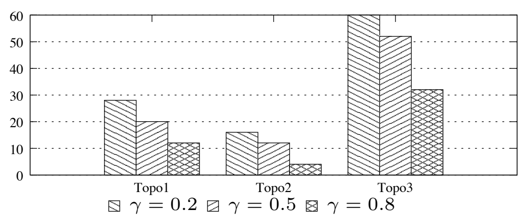

Fig. 6. Minimum of Number of Remaining Hops per Sensor under Different Topologies and γ

Unlike the method with upper and lower hopping limits, we argue that the dynamic method is more resilient to topological changes of the network. To validate this, we simulated the algorithm over three different topologies, namely Topo1 (the same as the topology used in the first simulation), Topo2, and Topo3, as indicated in the Figures 4, 5 and 6. The results show that under different topologies, the coefficient has comparable balancing influences. The results also show that under different γ values, the minimum of the number of remaining hops per sensor changes more dramatically compared with the STD of the number of remaining hops per sensor. This indicates that the minimum of the number of remaining hops per sensor is a more sensitive mobility metric. The same observation is also drawn from the first simulation, as we see a sharper curve in Figure 3(c) than Figure 3(b).

VI. CONCLUSIONS

Hopping sensors are more adaptable to harsh terrains compared with wheeled mobile sensors. We studied the multi-hop landing of the sensors under a multivariate normal distribution model and obtained an upper bound of the number of hops needed given the physical distance of the source and destination. The bound is further refined by considering the aerodynamics of air disturbances. Under such premises we focused on the problem of migrating a number of hopping sensors from multiple source clusters to a destination in a sensor network. We argue that cascaded relay hopping can speed up the migration but frequent relays may introduce unnecessary hops. We devised two algorithms to find the best migration plan. One uses upper and lower relay edge hop limits, while the other uses a balancing coefficient to construct a new optimization target dynamically. The two schemes are simulated using the physical and dynamic parameters of the Feddema prototype minefield hopping robots [4]. Simulation results indicated that both algorithms are effective in balancing the requirement of path optimality and maintaining the relocation capability of the network. The dynamic algorithm is also shown to be resilient to topological changes of the network.

REFERENCES

[1] M. Confente, C. Cosma, P. Fiorini, and J. Burdick, "Planetary Exploration Using Hopping Robots," in *7th ESA Workshop on Advanced Space Technologies for Robotics and Automation 'ASTRA 2002'*, ESTEC, Noordwijk, The Netherlands, 2002.

[2] E. Hale, N. Schara, J. Burdick, and P. Fiorini, "A minimally actuated hopping rover for exploration of celestialbodies," in *Proceedings of IEEE International Conference on Robotics and Automation, 2000. ICRA '00*, San Francisco, CA, USA, 2000, pp. 420–427.

[3] S. Bergbreiter and K. Pister, "Design of an Autonomous Jumping Microrobot," in *Proceedings of IEEE International Conference on Robotics and Automation, 2007. ICRA '07*, 2007.

[4] J. T. Feddema and D. Schoenwald, "Decentralized control of cooperative robotic vehicles," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 852–864, 2002.

[5] G. Wang, G. Cao, and T. L. Porta, "Movement-Assisted Sensor Deployment," in *IEEE INFOCOM*, 2004.

[6] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, 2002.

[7] J. Wu and S. Yang, "SMART: A Scan-Based Movement-Assisted Sensor Deployment Method in Wireless Sensor Networks," in *IEEE INFOCOM*, 2005.

[8] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks," in *IEEE INFOCOM*, 2005.

[9] S. Chellappan, X. Bai, B. Ma, and D. Xuan, "Sensor Networks Deployment Using Flip-Based Sensors," in *Proceedings of 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2005)*, Washington, DC, USA, 2005.

[10] J. Friedman, D. C. Lee, I. Tsigkogiannis, S. Wong, D. Chao, D. Levin, W. J. Kaiser, and M. B. Srivastava, "RAGOBOT: A New Platform for Wireless Mobile Sensor Networks," in *Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2005)*, 2005.

[11] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization in Distributed Sensor Networks," *ACM Transaction on Embedded Computing Systems*, vol. 3, pp. 61–91, 2004.

[12] B. Carbunar, A. Grama, J. Vitek, and O. Carbunar, "Redundancy and Coverage Detection in Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 94–128, 2006.

[13] R. Ghrist and A. Muhammad, "Coverage and Hole-detection in Sensor Networks via Homology," in *The Fourth International Symposium on Information Processing in Sensor Networks (IPSN'05)*, UCLA, Los Angeles, California, USA, 2005.

[14] Q. Fang, J. Gao, and L. J. Guibas, "Locating and Bypassing Routing Holes in Sensor Networks," in *INFOCOM 2004*, 2004.

[15] A. Genz, "Numerical Computation of Multivariate Normal Probabilities," *J. Comp. Graph. Stat.*, 1992.