Energy-Efficient Localization in Networks of Underwater Drifters

Diba Mirza, Curt Schurgers
Department of Electrical and Computer Engineering
University of California, San Diego, CA.
diba@ucsd.edu, curts@ece.ucsd.edu

ABSTRACT

In their quest to study the intricate underwater ecosystems, oceanographic scientists need spatially-rich data that is collected within the moving reference frame created by the oceans' currents. To this end, we are developing a system of networked underwater ocean explorers that drift freely with the currents. However, effective interpretation of collected sensor data hinges on knowing the positions of these drifters while submerged. Due to their uncontrollable motion, positions have to be tracked in time. Unlike in static networks, position estimation therefore represents a recurring cost, which should be minimized for reasons of limited on-board energy supplies. In this paper, we present a low-overhead scheme that is able to trade accuracy for energy savings by cleverly selecting the specific links that are required for the self-localization algorithm. Our technique is also agile enough to quickly adapt to on-demand changes in accuracy requirements, and cuts the position estimation costs by 40% or more under various ocean current dynamics.

Categories and Subject Descriptors

C.2 [Computer-Communication Networks]: C.2.3 Network Operations **C.3 [Special Purpose and Application Based Systems]:** Real-time and Embedded Systems.

General Terms: Algorithms, Performance.

Keywords: Underwater Networks, Localization, Energy Efficiency, Acoustic Networks.

1. INTRODUCTION

The ongoing shift from traditional stand-alone oceanographic systems to distributed underwater sensor networks greatly expands our knowledge of the fundamental biological and physical processes occurring within the oceans. The most common examples of such distributed systems are networks of bottom-anchored instruments and/or actively propelled autonomous underwater vehicles (AUVs). However, another type of system, which is being welcomed with great excitement by the oceanographic community, is that of *networked underwater drifters* [1].

WUWNet'07, September 14, 2007, Montréal, Québec, Canada.

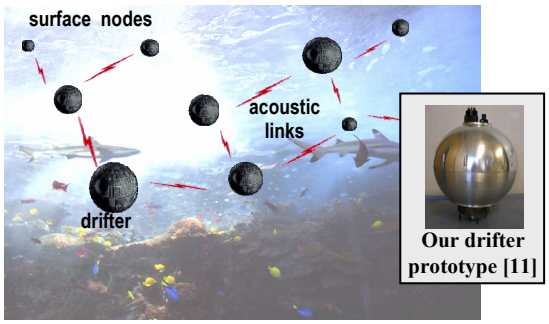

Figure 1. Network of freely drifting autonomous underwater explorers (conceptual, not to scale)

These drifters are devices that possess buoyancy control to change their depth, but are otherwise drifting freely with the underwater currents. As such, they offer oceanographers a powerful tool to study phenomena within their own Lagrangian frame of reference, i.e. within the dynamics of the ocean environment itself. We will be able to learn about ocean organisms that use current motion to migrate (according to current hypotheses), or help predict the spreading of patching such as oil spills. To provide the required temporal and spatial extend of information, such drifters will have to be deployed as true swarms on a mission of collective data gathering. Figure 1 illustrates conceptually the system we envision: a networked autonomous swarm of freely floating underwater drifters [11]. They communicate with each other through acoustic links using an acoustic modem. A picture of our current drifter prototype, which has been tested standalone [11], is shown as well.

Common to all distributed data collection systems, information gathered by each drifter in the swarm is useful only if we also know *where* it was collected. While traditional methods such as GPS are unavailable underwater, collaborative self-localization algorithms can provide the required position estimates [2], [9], [9]. These algorithms essentially triangulate devices with respect to each other or beacons (i.e. surface buoys in our case), based on distance estimates which in our case can be obtained through acoustic ranging using the acoustic modem [3].

Furthermore, in our system, each drifter in the swarm is at the mercy of the underwater currents, and locations therefore constantly change in an uncontrollable way. To track these changes, we have to rely on periodically refreshing the position estimates. As this represents a recurring cost, there is an increased impetus to minimize the overhead of each localization event (compared to quasi-static networks). Our drifters are constrained in energy resources (batteries), and *minimizing the energy consumption*

of the repeated self-localization is of primary importance and the main goal of this paper.

As we further improve our prototypes for size and price, we eventually envision relatively large swarms of drifters. Collecting ranging information for each pair of devices will be prohibitive. Instead, in this paper, we propose a method to cleverly restrict the ranging to only specific pairs of devices, thereby significantly lowering the associated cost. This is possible by matching the resulting self-localization accuracy to the user-defined application demands instead of blindly going for maximum accuracy.

2. PROBLEM SETUP

Before detailing the individual elements of our solution, we will describe the overall system setup and high-level solution strategy. Consider the scenario depicted in Figure 2, with drifters floating with the ocean currents (only three out of a larger swarm are drawn to not overload the picture). Each drifter is shown at different points in time $\{t_i\}$. The goal of the application is to collect sensor data and to annotate it post-mission with the location where it was collected. The only thing that is required during the mission is that the devices update a select set of pair-wise distance estimates at times $\{t_i\}$. Post-mission, a self-localization algorithm calculates the position of all drifters at these times, and these are in turn fed into a tracking algorithm to estimate the paths followed (as sensor data could have been collected continuously).

The times $\{t_i\}$ are pre-determined by the tracking algorithm and serve as an input to our algorithm. In addition, the desired accuracy of each position estimate is determined by the application and is also an input to our algorithm. This accuracy requirement could be data and/or time dependent as explained earlier. It is represented by the small shaded disk around the black dot representing the drifter in Figure 2. Note that the physical dimensions (e.g. achievable accuracy, transmission range, etc.) are drawn roughly to scale in this figure (see more in section 7). Our goal is to find the suitable set of links (i.e. pairs of devices) to collect the distance estimates for. In the upcoming sections, we will propose an algorithm that minimizes the size of this set, while taking into account the desired accuracy requirements. However, as the drifters move independently, the suitability of this set of links will vary over time. Therefore, the link selection will have to be repeated from time to time. Luckily, we will show that the set of links selected by our algorithm can tolerate significant changes in relative device positions. It is only when major changes in the topology occur, that the link selection needs to be updated.

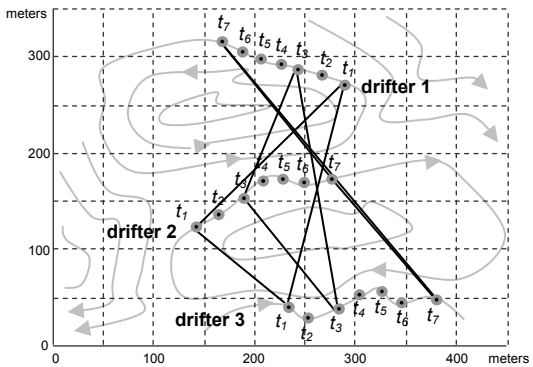

Figure 2. Network application setup

We refer to these times as $\{T_j\}$. We will show in section 7.2 that $(T_j - T_{j-1}) >> (t_i - t_{i-1})$, and range estimates therefore can be gathered for a significant number of time steps $\{t_i\}$ for a fixed collection of links, and our selection algorithm has to run at a much larger timescale $\{T_j\}$. The solid lines in Figure 2 illustrate how changes to the overall topology occur over larger timescales despite small changes in relative device locations at each time step. Figure 3 summarizes our system setup.

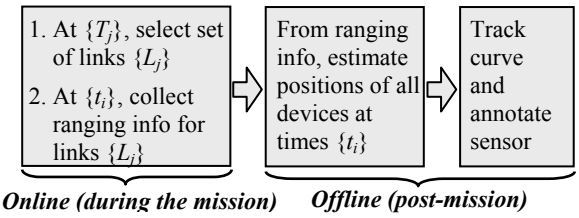

Figure 3. System setup

3. ESTIMATION COST vs ACCURACY

At a high level, we study the localization of nodes in a 3-D multi-hop underwater network. A subset of nodes resides on the surface (buoys) and has absolute position information available via GPS. The location of these *beacon* nodes is restricted to the surface since radio waves are highly attenuated underwater. Range estimates are obtained from time-of-arrival (TOA) measurements gathered via acoustic signaling [3].

As a first step in our energy-efficient localization problem, we need a measure of how estimation accuracy trades with estimation cost as it forms the basis in selecting the best set of links at times $\{T_j\}$. A well-accepted measure of localization performance that is frequently used in literature is the Cramer Rao Bound (CRB) on the accuracy of nodes' position estimates [5] which is known to be achievable by Maximum Likelihood (ML) estimators. Since we perform post-mission localization, we can apply powerful and more computationally intensive estimators, such as true ML, that can achieve the CRB.

We use a standard procedure [6] to compute the CRB on the position estimates of *n* unknowns in the network when *m* beacons are present. The mathematical formulation of the localization problem and the derived CRB is presented in Appendix A. The key point is that given the relative position of nodes and the variance of range measurements, the RMS error in the position estimates of the unknowns, $\{\sigma_i\}_{i=1}^{n}$ can be computed as a function of a binary connection matrix, *I* which we will refer to as the *link-selection policy*:

$$I(i,j) = \begin{Bmatrix} 1, & \text{nodes } i,j \text{ perform ranging} \\ 0, & \text{otherwise} \end{Bmatrix} \forall i,j \in [1,...,(n+m)] \quad (3.1)$$

TOA measurements require two-way signaling which is energy consuming due to high transmit power required for underwater communication [11], [15]. The total energy consumed thus directly relates to the number of measurements. Therefore, we define the cost, *C* as the total number of pair-wise range measurements made every localization event and compute it from the connection matrix as in eq 3.2:

$$C(I) = \sum_{i=1}^{n+m} \sum_{j>i}^{n+m} I(i,j) \tag{3.2}$$

Above we have quantified how estimation accuracy trades with estimation cost. Since the position estimation error of nodes that reside on the boundary of the network is often much larger than that of nodes in the interior, application defined error requirements are less likely to be satisfied by all nodes in the network. Therefore, we refine our definition of localization performance for the network as follows: A target error threshold $\sigma^{\max} = \{\sigma_i^{\max}\}_{i=1}^{n}$ is met by a link-selection policy *I* with cost *C,* if the RMS error of a fraction α of nodes is below the specified threshold for a suitably chosen α. The equivalent optimization problem is given by:

$$I^{opt} = \arg\min_{I} C(I) \tag{3.3}$$

s.t. the constraints:

(i) $I(i,j) = 0 \;\; if \;\; I_0(i,j) = 0 \;\; \forall (i,j) \in [1,...(n+m)]$

Where I_0 is the connection matrix when the maximum number of measurements is made given a finite acoustic transmission range.

(ii) $O(A) \geq \alpha.n \qquad A = \{i : (\sigma_i^{\max} - \sigma_i(I)) \geq 0, i \in [1,...n]\}$.

Next, we determine the optimum link-selection policy for a target localization performance when the error requirement of nodes does not change with time.

4. LINK SELECTION FOR FIXED REQUIREMENTS

We solve the optimization problem given by eq (3.3), as follows. Let the set of unknowns be *U*, and the set of beacons *B*. Initially, all links are used and we can calculate the corresponding CRB. Unknowns with error below the specified threshold are given by Ω_0:

$$\Omega_0 = \{i : (\sigma_i^{\max} - \sigma_i(I_0)) \geq 0, i \in U\} \tag{4.1}$$

If $O(\Omega_0) < \alpha.n$, then the specified thresholds are infeasible even if all links are used, and we set the cost $C_0 = \infty$. On the other hand, if the thresholds are achievable, we find a way of removing links between node pairs until no more links can be reduced. The set of nodes that can initially be considered for link removal are nodes for which the error is above the threshold and which have at least one neighbor that is either known or has error above the specified threshold. We construct this set, P_0 as:

$$P_0 = \{i : (\Omega_0 \cup B) \cap N_{I_0}(i) \neq \phi, \mathrm{O}(N_{I_0}(i)) > 3, i \in \Omega_0\} \tag{4.2}$$

Where, $N_I(i) = \{u : I(i,u) = 1\}$. At each iteration *k*, we find the optimal link, (i_{opt}, j_{opt}) to remove from the link-selection policy I_k (obtained in the previous iteration), until $P_k = \phi$. The sum of the variances in all nodes monotonically increases with the number of range measurements [12]. A selected link (to remove) is *optimal* if it *minimizes the total increase in the variance of nodes and the maximum increase occurs at the node with the maximum error allowance.* Applying these optimality conditions per iteration, we obtain (i_{opt}, j_{opt}) in eq (4.3) and eq (4.4) respectively using a one dimensional search in each case.

$$i_{opt} = \arg\max_{i \in P_k} (\sigma_i^{\max} - \sigma_i(I_k)) \tag{4.3}$$

$$j_{opt} = \begin{Bmatrix} \phi, & \text{if} \quad \inf_{j \in J} \zeta(j) = \infty \\ \arg\min_{j \in J} \zeta(j), & \text{otherwise} \end{Bmatrix} \tag{4.4}$$

Where, $J = \{(P_k \cup B) \cap N_{I_k}(i_{opt})\}$

$$\zeta(j) = \begin{Bmatrix} \infty, \text{ if } O(\{q : (\sigma_q^{\max} - \sigma_q(I_k^j)) \geq 0, q \in U\}) \leq \alpha.n \\ \sum_{q \in U} \{\sigma_q^2(I_k^j) - \sigma_q^2(I_k)\}, \qquad \text{otherwise} \end{Bmatrix}$$

$$I_k^j = I_k \text{ where } I_k^j(i_{opt}, j), I_k^j(j, i_{opt}) = 0$$

At the end of iteration *k* we set the link-selection policy, $I_{k+1} = I_k$ and additionally remove the link that resulted in minimum overall increase in error:

$$I_{k+1}(i_{opt}, j_{opt}), I_{k+1}(j_{opt}, i_{opt}) = 0 \text{ , if } j_{opt} \neq \phi \tag{4.5}$$

If $j_{opt} = \phi, P_k = P_k - \{i_{opt},\}$The updated set of nodes that can be considered for link reduction in the next iteration, P_{k+1} are obtained in eq (4.6) when the nodes that have crossed the specified threshold are removed from P_k:

$$\Omega_{k+1} = \{i : (\sigma_i^{\max} - \sigma_i(I_{k+1})) \geq 0, i \in P_k\} \tag{4.6}$$

$$P_{k+1} = \{i : (\Omega_{k+1} \cup B) \cap N_{I_{k+1}}(i) \neq \phi, \mathrm{O}(N_{I_{k+1}}(i)) > 3, i \in \Omega_{k+1}\} \tag{4.7}$$

Since a distributed solution would require transmission of large covariance matrices, we choose a centralized solution where inter-node distance measurements are used to estimate the relative position of nodes, compute the CRB and hence determine the optimal link-selection scheme. This approach results in a smaller overhead.

5. ADAPTING TO CHANGING REQUIREMENTS

Consider now a scenario where the accuracy requirements can change over time. In this case, we would have to repeat the link selection policy of Section 4 every time the requirement of one node changes (not just at times $\{T_j\}$; see more in section 7.3). The drawback is the global communication overhead incurred with every change in requirement.

An alternative option is to pre-calculate the policies for all possible accuracy requirements. Suppose these requirements are limited to a set of *K* possibilities. Nodes that are in close proximity typically would have similar accuracy requirements such that these can then be grouped together, giving rise to *L* groups. In this case, pre-computation would involve finding L^K different policies. However, if the accuracy of one group changes, all nodes in the network still need to be informed to activate the appropriate policy, resulting in global communication again. In the next subsections, we develop a solution such that nodes need to communicate locally with their one-hop neighbors only, if a group wants to switch policies. An additional advantage is that the number of pre-calculated policies is reduced to *K.L*.

Our solution strategy hinges on the fact that the position error of nodes is primarily determined by links to nodes closer to

beacons when beacons reside in a restricted region on the boundary of the network. This situation is unique to the underwater case where all beacons are on the surface. We partition the network into L distinct sets depending on their proximity to surface beacons. We then propose a (recursive) method for computing the CRB for the network where the position estimation error of a set of nodes is computed based on that of its *neighbor- set* (or 1-hop neighbors closest to beacons) and the link-selection policy among them alone. This allows us to de-couple the link-selection policy of a set of nodes from that of other sets further away from beacons.

We can then optimize the link-selection policy for a given set to achieve each of K target error requirements. Further, the associated error requirements from the neighbor-set are obtained. Therefore, should the error requirement of a group of nodes change; they can switch to the appropriate policy and communicate to their neighbor-set the desired error required from them; all of which can be achieved via local communication.

5.1 Recursive Computation of the CRB

In general, the CRB for a group of nodes in a multi-hop network cannot be computed in isolation from other nodes[1]. However, we show in lemma 1 that the estimation error of a set of unknown nodes can be computed based on that of their 1-hop neighbors only and the chosen links among them. This forms the basis of the recursive method for computing the CRB.

***Lemma 1*:** Consider a network of $n+k$ nodes. A new node U is added to the network and performs ranging with k of its neighbors, denoted by the set B. The remaining nodes are part of set A. If the CRB of nodes in set B, prior to the addition of U, is Σ_B, then the CRB of node U, Σ_U can be computed in terms of Σ_B alone provided Σ_B is bounded and invertible, given by:

$$\Sigma'_B = (\Sigma_B^{-1} + \Lambda_{BU} - F_{BU} F_{UU}^{-1} F_{BU}^T)^{-1} \quad (5.1)$$

$$\Sigma_U = F_{UU}^{-1} + F_{UU}^{-1} F_{BU}^T \Sigma'_B F_{BU} F_{UU}^{-1} \quad (5.2)$$

The proof of lemma 1, definition of the matrices F_{UU}, F_{BU} ,Λ_{BU} and extension of the result for the case where U is a group of nodes are given in Appendix B. Briefly, these matrices depend on the relative position and the link-selection policy among the nodes U and B alone. Cleary, not all unknown nodes can compute their error simultaneously using eq (5.1) and eq (5.2), as the error of their neighbors is not available. However, if we can find an appropriate ordering in which nodes are added to the network (i.e. the appropriate definitions of U and B), the above recursive calculations can be used for computing the error of all nodes in the network. This proposed scheme is presented as part of the pseudo code in Figure 4.

[1] When beacons are placed isotropically in the network, the CRB can be computed by considering one and two hop neighbors [7]. However, this is not possible in our underwater scenario where beacons are always on the surface.

1. **Denote the 'level' of a node k as $R(k)$. Denote all beacons by set S.**
2. **$R(k) = 0, \forall k \in S$**
3. **For any unknown node u:**
   $R(u) = \min_{j \in N(u)} R(j) + 1$
4. **Compute the CRB at the i^{th} recursion as:**
   $U = \{k : R(k) = i, k \notin S\}$,
   $B = \{k : R(k) = i - 1\}$
   If, $B \subseteq S$, **{Set** $\Sigma'_B = 0$**}**
   Else, {Compute Σ'_B **from (5.1) and store it as the final estimate of CRB of B}**
5. **Compute** Σ_U **from equation (5.2)**

Figure 4. Pseudo code for recursive CRB computation

The key here is defining a 'level' for each node in the network and operating the recursion per such level. Surface beacons are defined as being level 0. A node that can hear at least one beacon is level 1. In general, a node is of level l if the lowest level of any of its neighbors is l-1. Figure 5 further illustrates this assignment of levels (where the number inside the circle denotes the node's level).

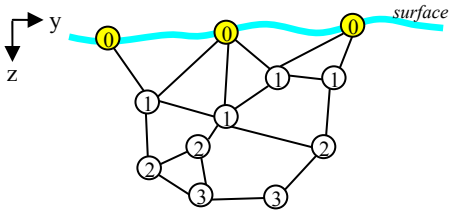

Figure 5. An example network with 3 levels

5.2 Adaptive Link Selection Algorithm

As shown by the recursive computation of the CRB, the position accuracy of nodes at level l depends on the link-selection policy among nodes of level l and l-1 and on the position estimation error of nodes of level l-1. The link selection policy of rest of the network is abstracted as the position error of nodes of level l-1. Therefore, for nodes of level l to meet a desired error threshold, they need to determine the required error threshold of nodes at a previous level and their optimum link policy with these nodes. We determine how nodes at some level l can select their optimum link policy with nodes at l-1 and the minimum accuracy requirement from nodes of level l-1. This is given by the pseudo code in Figure 6, when there are K possible target accuracy requirements.

1. Define error thresholds: $\{\sigma_1^{\max},....,\sigma_K^{\max}\}$.
2. For each level l, determine:
   a) Nodes of 'level' l: $U = \{u_1,...u_n\}$
      Nodes of 'level' l-1: $B = \{b_1,...b_m\}$
   b) K optimum ranging policies $\{I_1^l,...,I_K^l\}$ as follows:

for $p = 1:K$

for $q = 1:K$

$$I_{p,q}^{opt} = \arg\min_{I}\left[\sum_{i=1}^{n}\sum_{j=1}^{m} I(u_i,b_j) + \sum_{i=1}^{n}\sum_{j>i}^{n} I(u_i,u_j)\right]$$

s.t. the constraint:

$$O\left(\{u_i : \sigma_{u_i} \le \sigma_p^{\max}\}\right) \ge \alpha.n,$$

$$\sigma_{u_i} = sqrt\left([\Sigma_U]_{2.i-1,2.i-1} + [\Sigma_U]_{2.i,2.i}\right), i=\{1,..n\}$$

Compute Σ_U from eq (5.1), eq(5.2) using Connection Matrix I for the set $\{U,B\}$ and setting $\Sigma_B = \Sigma_B^q$. If l=1, set $\Sigma_B' = 0$ and compute Σ_U from eq (5.2)

Above optimization is solved as in Section 4.

$$C^l(p,q) = \sum_{i=1}^{n}\sum_{j=1}^{m} I_{p,q}^{opt}(u_i,b_j) + \sum_{i=1}^{n}\sum_{j>i}^{n} I_{p,q}^{opt}(u_i,u_j) + \min(C^{l-1}(q,:))$$

end

$$q^{opt} = \arg\min_{q} C^l(p,q)$$

When error threshold for nodes of level l is $\sigma_p^{\max}$,

- Optimum Link Policy of level l is $I_p^l = I_{p,q^{opt}}^{opt}$
- Accuracy requirement from level l-1 is $\sigma_{q^{opt}}^{\max}$

$\Sigma_U^p = \Sigma_U$, where Σ_U is computed from eq(5.1) & eq (5.2) using Connection Matrix I_p^l and setting $\Sigma_B = \Sigma_B^{q^{opt}}$

end

Figure 6. Pseudo code for Optimal Link Selection

6. ERROR TOLERANCE AND TOPOLOGY CHANGE

In the previous section, we have presented the link selection algorithm itself. As nodes drift with the ocean currents, the usefulness of the selected set is expected to degrade over time. As a result, the link selection has to be repeated occasionally.

To study the degradation of the link set over time, we can obtain a useful result from looking at a geometric conditioning parameter popularly known as the Geometric Dilution of Precision (GDOP) [4]. The GDOP is the normalized RMS error of the position estimate of a node when all its neighbors have perfect position information, and can be computed from the CRB [10]. Invariance of the GDOP to a certain parameter means that position estimation error is not affected much by that parameter.

Consider the case where N beacons and the unknown are projected onto the x-y plane. All beacons reside within a sector of angle Θ centered at the unknown node. The expected value of $(GDOP)^2$, denoted by parameter G is given by equation (6.1). We have provided the derivation online [10].

$$E[G] \cong \frac{4}{\eta.N} \qquad \eta = 1 - \frac{\sin^2(\Theta)}{\Theta^2} \qquad (6.1)$$

We observe that $\eta \cong 1$ for $\Theta \ge 2\pi/3$. In other words, G remains more or less unaffected by change in relative positions of nodes beyond this point. This suggests that only large changes in topology will have a noticeable on the localization accuracy. As such, the link selection policy only needs to be adjusted once such significant topology changes occur, which will be at relatively large timescales. This effect will be studied further via simulations in section 7.2.

7. PERFORMANCE EVALUATION

7.1 Spatial Gains

In this section, we will evaluate the performance of our scheme. As a first step, we look at the basic performance of the link selection algorithm of section 5 in terms of spatial gains alone: i.e. what is the reduction in number of links needed for ranging.

We simulate a network where nodes are distributed in an ocean column of depth 900m and radius 600m. The transmission range of nodes is 300m and the density of surface nodes is 35 per km^2. Results are averaged over 100 realizations of the network and shown for three network densities, which are defined as the average number of neighbors of a node λ = 6, 10 and 14. Figure 7 plots the spatial gain versus the target accuracy requirement, assumed same for all nodes. As can be observed, considerable gains are achieved if the accuracy requirement is 3 to 4 times the ranging error.

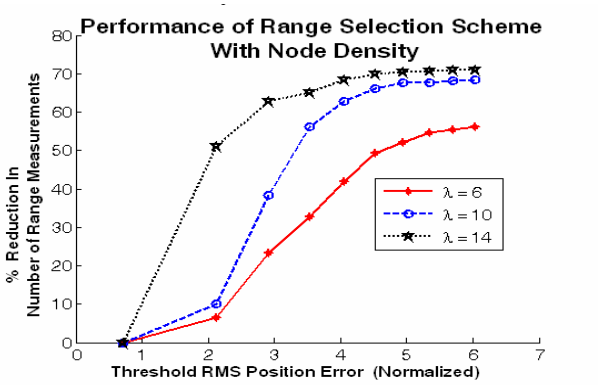

Figure 7. Performance versus node density.

7.2 Temporal Behavior of Link Selection

Next we investigate how often the link selection needs to be repeated, by evaluating the length of time for which the localization error of nodes remains within a desired accuracy threshold under the influence of currents. We have simulated currents based on the commonly-used model in which the water moves in different layers. The current at each depth were chosen randomly with speeds between 0 and 10 cm/s and direction

between 0 and 2π. Two cases were considered. In one case, we assume current velocities at each depth are constant in time. As such, node pairs tend to drift away from each other. In the second case, current speed and direction changes every 30 minutes at each layer. The transmission range is 600m (close to typical values achieved by less expensive underwater modems [15]). Though nodes suffer significant relative displacements, they do not move out of each others range with high probability in this case.

We simulated a network of density $\lambda = 15$; surface beacon density of 35/km^2, range variance of 1 and a target accuracy threshold of 4m. Nodes perform ranging every 10 minutes (specified by the tracking algorithm, see Section 2). The set of links used for ranging are found at time = 0 and remain fixed. Although the transmission range is 600m, to avoid selected node pairs from rapidly drifting out of range, we assume a smaller transmission range when performing link selection (we chose a margin of 360m). This scenario is visualized in Figure 2 (Section 2), drawn roughly to scale. We have evaluated the percentage of nodes with error less than the threshold for a period of 80 minutes. The performance results averaged over 100 realizations and are shown in Figure 8. In this case, the number of range measurements is reduced by 66% (not shown). The following interesting result can be observed in Figure 8. In the case of time-invariant currents, the performance significantly deteriorates after 1 hour. This is precisely the time it takes for nodes to move out of each others range if drifting at the average current speed of 5 cm/s. Prior to nodes going out of range, the percentage of nodes within threshold is as high as 90%. This is also consistent with case of time variant currents, when nodes hardly ever move out of range.

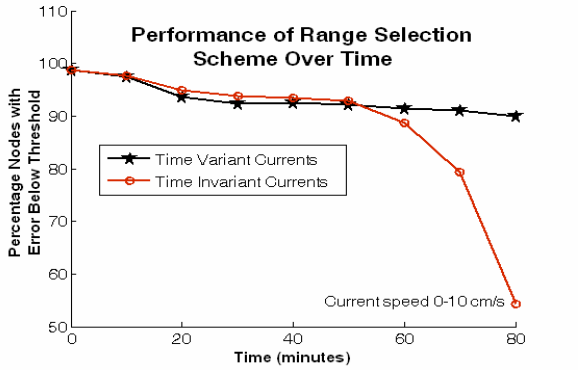

Figure 8. Performance of link-selection scheme in time

To dig a little deeper, we look at the error in the position estimates of all nodes at four different times $\{t_i\}$, as shown in Figure 9. We observe that the error truly remains close to that at time = 0, despite the fact that nodes suffer an average displacement of around 30m every localization period (t_i-t_{i-1}). This confirms our observations from Section 6, that the set of selected links is relatively insensitive to the relative motions of the nodes. It is only when large topology changes occur (such as nodes moving out of range), that the link selection needs to be repeated.

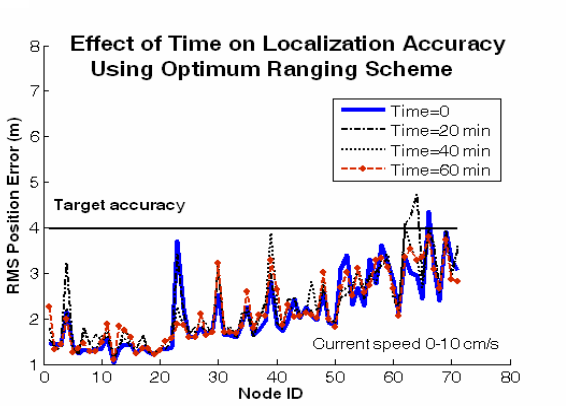

Figure 9. Position estimation error of nodes over time under dynamic current conditions

Protocol Overhead:

When the links selected for ranging have to be revised, nodes send their range measurements to a central location. The revised link set is then communicated back to nodes. We will evaluate this overhead for the above described network. Each node can combine its range measurements in a single packet. The total number of transmissions for round-trip communication to a central node, a maximum of L hops away is given by eq (7.1)

$$N_{Tx} = 2.\sum_{i=1}^{L} i.N_i \qquad (7.1)$$

Where N_i is the number of nodes i hops away from the central node. In the above network nodes are distributed in a water column of height 700m and radius 480m.We consider a central node located approximately in the middle of the network on the surface. The transmission range of nodes is 600m. Therefore, nodes are a maximum of 2 hops away. From geometrical considerations and given node density we compute N_1 and N_2 and hence the total number of transmissions from eq (7.1). The total number of range measurements per localization event when the all possible links are used is obtained from simulations. Under non ideal current conditions (where nodes move out of communication range), new policies have to be determined every hour. Localization is done every 10 min. We obtain an overall of 40% percentage reduction in the number of transmissions when protocol overhead is considered.

7.3 Adaptive Accuracy Requirements

To demonstrate that nodes can adapt to changing error requirements in a localized manner, we simulated a 4-level network. The maximum allowable normalized position error variance is initially set to 170. The ranging policy allows nodes at each level to set the required error threshold of nodes at a previous level as shown in Figure 10 (a). Further, when an event is detected (at hop 2) and this requirement changes, nodes select a new policy locally without informing the rest of the network as shown in Figure 10 (b). This example illustrates how adaptive requirements can be handled in our system.

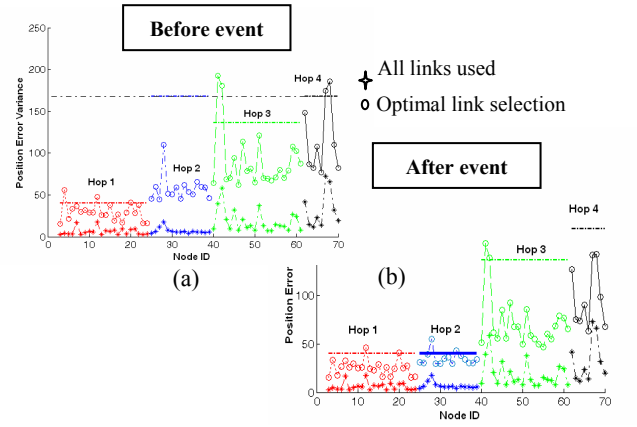

Figure 10. Adapting link selection policy

8. RELATED WORK

We distinguish our method of recursively computing the CRB from that by Hero et al. developed for image processing applications [13], [14]. There, iterative methods are proposed for efficiently computing the bound for a subset p of n unknowns when $p<<n$. At each iteration, the complete FIM obtained for n unknowns is used. Applied to our problem, this would result in a computationally efficient way to determine the CRB of a set of nodes, given the link-selection policy of the entire network. However, it does not allow us to decouple the link-selection policy of nodes at different higher levels.

Energy-efficient localization with specified accuracy, obtained from the CRB has been proposed for terrestrial wireless sensor networks. Here energy is minimized by determining the minimum set of beacons to activate every localization period so that a near uniform beacon configuration is achieved [16]. In target tracking applications energy is saved by querying only a subset of nodes for detailed tracking [17]. Given that transmit power for underwater acoustic signaling dominates idle power consumptions we consider minimizing the number of links used for ranging as opposed to allowing nodes to *sleep*. Further uniform beacon placement is not possible in the underwater case since beacons are restricted to the surface.

9. CONCLUSIONS

We have an approach to track a swarm of drifters in an energy efficient manner by minimizing the required number of range measurements. We have illustrated and given a formal intuition that link selection only needs to be repeated at time intervals when significant topology changes occur, which is less frequent than re-localization is required. The protocol overhead in general depends on current dynamics. We show a 40% reduction in the number of links needed for ranging when protocol overhead is taken into account, for non ideal current conditions (where significant topology change occurs with time). How to adaptively determine when the link selection should be repeated is part of future work.

10. REFERENCES

[1] C. Colgan. Underwater Laser Show. *Explorations, Scripps Institution of Oceanography,* Vol.12, No.4, pp.20-27, Spring 2006.

[2] C. Savaese, J. Rabaey, K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. *UXENIX Technical Annual Conference*, pp.317-328, Monterey, CA, 2002.

[3] D. Thomson, S. Elson. New generation acoustic positioning systems. pp.1312-1318, *In Proc. of Oceans'02,* 2002.

[4] M. Spirito. On the Accuracy of Cellular Mobile Station Location Estimation. *IEEE Trans. Vehicular Tech,* Vol. 50, No. 3, May 2001.

[5] S. Qicai, S. Kyperountas, N. Correal, F. Niu. Performance analysis of relative location estimation for multihop wireless sensor networks. *IEEE Journal in Selected Areas in Communications,* Vol.23, No.4, pp. 830-838, 2005.

[6] S. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory.* Vol 1, Prentice-Hall, 1993.

[7] C. Chang, A. Sahai. Estimation bounds for localization. *In proc of SECON'04,* pp.415-424, Oct. 2004.

[8] D. Niculescu, B. Nath, "Localized Positioning in Ad hoc Networks," Elsevier journal of AdHoc Networks and in SNPA 2003, May 2003.

[9] Y. Ohta, M. Sugano, M. Murata. Autonomous localization method in wireless sensor networks. *Pervasive Computing and Communications Workshops,* pp. 379 – 384, March 2005

[10] D. Mirza, C. Schurgers. Analyzing the Effect of 3D Geometric Parameters on Localization Accuracy for Underwater Sensor Networks. http://wuwnet.engr.uconn.edu/program.htm.

[11] J. Jaffe, C. Schurgers. Sensor networks of freely drifting autonomous underwater explorers. *In Proc. of WUWNET'06,* Los Angeles, CA, pp. 93-96, Sept 2006.

[12] N. Patwari, A. Hero, M. Perkins, N. Correal, R. O'Dea. Relative location estimation in wireless sensor networks. *IEEE Trans. Signal Processing,* Aug., 2003.

[13] A. Hero, J. Fessler. A recursive algorithm for computing CR-type bounds on estimator covariance. *IEEE Transactions on Information Theory*, July 1994.

[14] A. Hero, M. Usman, A. C. Sauve, J. Fessler. Recursive algorithms for computing the Cramer Rao bound. *IEEE Transactions on Signal Processing,* Sept 1996.

[15] J. Parta, J. Kurose, B. Levine. A Survey of Practical Issues in Underwater Networks. *In Proc. of WUWNET'06,* Los Angeles, CA, pp. 93-96, Sept 2006.

[16] H. Feng, R. Yuan, C. Mu . An Energy-Efficient Localization Scheme with Specified Lower Bound for Wireless Sensor Networks. *IEEE Conference on Computer and Information Technology (CIT),* 2006.

[17] Y. Zou, K. Chakrabarty. Target Localization Based on Energy Considerations in Distributed Sensor Networks. *1st IEEE Intl. Workshop on Sensor Network Protocols & Applications (SNPA)*, Anchorage, 2003.

Appendix A

P_i: Position of the node i in 3D given by $[x_i, y_i, z_i]^T$.

$\mathbf{Q}_A$: Position of set $A = \{a_1, \ldots a_n\}$, given by $[x_{a_1}, y_{a_1}, \ldots x_{a_n}, y_{a_n}]^T$.

Σ_A: CRB on the covariance of position estimates of set A

σ_i: RMS error in the position estimate of node i

Cramer Rao Bound for 3D underwater networks

Consider a network with n unknown nodes and m beacons, represented as sets $A = \{1,\ldots,n\}$ and $B = \{n+1,\ldots, n+m\}$ respectively. The vector of unknown positions is given by $\mathbf{Q}_A$, since nodes know their depth from pressure sensors. TOA measurements between any node pair (i, j) if within range, are modeled as i.i.d. Gaussian random variables [3] with mean equal to $\| P_i - P_j \| / c$ (where c is the speed of sound in water) and variance μ_T^2. $N(i) = \{j$: node j is in communication range of node $i\}$. I is a connection matrix defined in eq (3.1). The Cramer Rao Bound Σ_A is calculated as the inverse of the Fisher Information Matrix (FIM) F_A, which is computed using the standard procedure [6]:

$$\Sigma_A = (F_A)^{-1}$$

$$F_A = \begin{bmatrix} f(1,1) & \ldots\ldots & f(1,n) \\ . & . & . \\ . & . & . \\ f(n,1) & \ldots\ldots & f(n,n) \end{bmatrix} \quad \text{(A.1)}$$

$f(i, j)$ given by:

$$f(i,j) = \frac{1}{c^2 \mu_T^2} \begin{bmatrix} \sum_{q \in N(i)} I(q,i).\beta_{x,i,q}^2 & \sum_{q \in N(i)} I(q,i).\beta_{x,i,q}.\beta_{y,i,q} \\ \sum_{q \in N(i)} I(q,i).\beta_{x,i,q}.\beta_{y,i,q} & \sum_{q \in N(i)} I(q,i).\beta_{y,i,q}^2 \end{bmatrix}, i = j \quad \text{(A.2)}$$

$$f(i,j) = \frac{-I(i,j)}{c^2 \mu_T^2} \begin{bmatrix} \beta_{x,i,j}^2 & \beta_{x,i,j}.\beta_{y,i,j} \\ \beta_{x,i,j}.\beta_{y,i,j} & \beta_{y,i,j}^2 \end{bmatrix}, i \neq j \quad \text{(A.3)}$$

$$\beta_{x,i,j} = \frac{|x_i - x_j|}{\| P_i - P_j \|}, \quad \beta_{y,i,j} = \frac{|y_i - y_j|}{\| P_i - P_j \|} \quad \text{(A.4)}$$

Appendix B

Proof of Lemma1: Denote the set of nodes prior to the addition of U as $C = \{A, B\}$. $A = \{a_1, \ldots a_n\}$, $B = \{b_1, \ldots b_k\}$ and $U = \{u_1\}$

The FIM, $\mathbf{F}_C$, prior to the addition of U can be computed from eqs. (A.1)-(A.4). We partition $\mathbf{F}_C$ as in eq (B.1).

$$F_C = \begin{bmatrix} (F_{AA})_{2n \times 2n} & (F_{AB})_{2n \times 2k} \\ (F_{AB}^T)_{2k \times 2n} & (F_{BB})_{2k \times 2k} \end{bmatrix} \quad \text{(B.1)}$$

The sub matrices F_{AA}, F_{AB} and F_{BB} contain the elements from eqs. (A.1) pertaining to the node sets identified by the respective sub indices. $\mathbf{F}_C$ is assumed invertible, i.e. the chosen neighbors have bounded error prior to making a connection with the new node, U. Let Σ_B be the inverse of the Schur complement [6] of F_{AA} in eq. (B.1) and is given by:

$$\Sigma_B = (F_{BB} - F_{AB}^T (F_{AA})^{-1} F_{AB})^{-1} \quad \text{(B.2)}$$

The set of nodes after the addition of U is $D = \{A, B, U\}$. We compute the updated FIM for the set D, $\mathbf{F}_D$ from eqs. (A.1)-(A.4) and partition it as:

$$\mathbf{F}_D = \left[\begin{array}{cc:c} (F_{AA})_{2n \times 2n} & (F_{AB})_{2n \times 2k} & \mathbf{0} \\ (F_{AB}^T)_{2k \times 2n} & (F'_{BB})_{2k \times 2k} & (F_{BU})_{2k \times 2} \\ \hdashline \mathbf{0} & (F_{BU}^T)_{2 \times 2k} & (F_{UU})_{2 \times 2} \end{array}\right] \quad \text{(B.3)}$$

In this equation $F'_{BB} = F_{BB} + \Lambda_{BU}$ (B.4)

The matrices Λ_{BU} and F_{BU} and F_{UU} are derived directly from eqs. (A.1)-(A.3). Their expression is presented a bit further as eqs. (B.8), (B.9) and (B.10) respectively with p=1.From eq. (B.3), we calculate Σ_U via the Schur complement (as before) as

$$\Sigma_U = (F_{UU} - F_{BU}^T (F'_{BB} - F_{AB}^T F_{AA}^{-1} F_{AB})^{-1} F_{BU})^{-1} \quad \text{(B.5)}$$

For $k > 1$ and U not collinear with all nodes in B, we can apply the Matrix Inversion Lemma to (B.5), and use equations (B.2) and (B.4) to obtain Σ_U in terms of Σ_B:

$$\Sigma'_B = (\Sigma_B^{-1} + \Lambda_{BU} - F_{BU} F_{UU}^{-1} F_{BU}^T)^{-1} \quad \text{(B.6)}$$

$$\Sigma_U = F_{UU}^{-1} + F_{UU}^{-1} F_{BU}^T \Sigma'_B F_{BU} F_{UU}^{-1} \quad \text{(B.7)}$$

We can extend the above proof to a scenario where p nodes are added to the network, with $p \geq 1$. Here $U = \{u_1, \ldots u_p\}$ and B is the set of nodes with which ranging is done by any of the nodes in U. Following a similar procedure, we find that Σ_U is again given by eq (B.6) and eq (B.7) where,

$$F_{BU} = \begin{bmatrix} f(b_1,u_1) & \ldots. & f(b_1,u_p) \\ . & \ldots. & . \\ f(b_k,u_1) & \ldots. & f(b_k,u_p) \end{bmatrix} \quad \text{(B.8)}$$

$$\Lambda_{BU} = -1.diag\,(\Lambda_1, \ldots. \Lambda_k) \quad \text{(B.9)}$$

$$\Lambda_i = \sum_{j=1}^{p} f(b_i, u_j)$$

$$F_{UU} = \begin{bmatrix} f(u_1,u_1) & \ldots. & f(u_1,u_p) \\ . & \ldots. & . \\ f(u_p,u_1) & \ldots. & f(u_p,u_p) \end{bmatrix} \quad \text{(B.10)}$$

Where, *f(,)* is defined by eq (A.2) and eq (A.3)