Motion-Aware Self-Localization for Underwater Networks

Diba Mirza
diba@ucsd.edu

Curt Schurgers
curts@ece.ucsd.edu

Electrical and Computer Engineering Department
University of California, San Diego.

ABSTRACT

A myriad of ocean processes affect life on the planet and are a source of intrigue to oceanographers and scientists. Understanding these processes and their interactions with currents requires collection of relevant data. A network of mobile platforms can be used to learn the correlation of processes in space and over time. To do this, data samples collected by nodes have to be annotated with location information. Given limited access to Global Positioning Systems underwater, collaborative self-localization schemes applied periodically are well-suited for this purpose. However, the specific nature of the underwater acoustic environment introduces significant error during network self-localization due to the combined effect of large latencies in communication and node mobility. We propose a method to account for these effects thus significantly improving the accuracy of position estimates.

Categories and Subject Descriptors

C.2 [Computer-Communication Networks]: C.2.3 Network Operations **C.4 [Performance of Systems]**

General Terms: Algorithms, Performance.

Keywords: Underwater Networks, Localization, Mobility, Latency, Acoustic Networks.

1. INTRODUCTION

The oceans are home to abundant physical and biological processes that are in constant flux, with far reaching impacts on the global climate and life on earth. However, as many aspects of the oceans remain unexplored, scientists and oceanographers are continuously seeking to further their understanding of oceanic phenomena. Collecting accurate and relevant data is crucial to this end. Recently, researchers have begun exploring sensing platforms that no longer consist of a single device, but a collection of underwater sensors and vehicles, operating in a coordinated and networked manner [12].

As the ocean is a dynamic environment, most of these sensing platforms exhibit some form of mobility. Examples are guided AUVs, gliders and drifters, but even anchored sensors are subjected to motions induced by tides and currents. Because of this mobility, networked sensing platforms require a location-finding service, as data samples need to be annotated with position information for meaningful interpretation. Because GPS does not work underwater, new localization techniques need to be developed for these oceanic sensing platforms. Specifically, in a networked setting, collaborative techniques allow devices to *self-localize* [10], i.e. use local measurements and a few surface beacons, to find their positions within the underwater network. The process has to be repeated each time the positions of nodes in the network need to be found, as network topology is constantly varying due to the ocean dynamics. At each such localization time T_{loc}, self-localization operates on inter-node distance estimates. When equipped with an acoustic modem, nodes can estimate these distances via time-of-flight measurements, a process referred to as *ranging*.

Now, the intrinsic mobility of the underwater environment creates some very specific challenges with respect to ranging, which are illustrated using Figure 1. Ideally, all range estimates should be acquired at the target localization time T_{loc}. However, because communication occurs over a shared channel, medium access control (MAC) has to ensure that excessive collisions are avoided. As a result, the gathering of ranging information actually occurs over a short time epoch T around the target localization time. The problem is that mobility causes the node positions to change significantly during epoch T. This is illustrated in Figure 1 by showing the positions of three nodes, at the three time instances at which the range for each pair is estimated. We observe that ranging (and topology) becomes ambiguous and non-consistent, which results in error in position estimates obtained from self-localization.

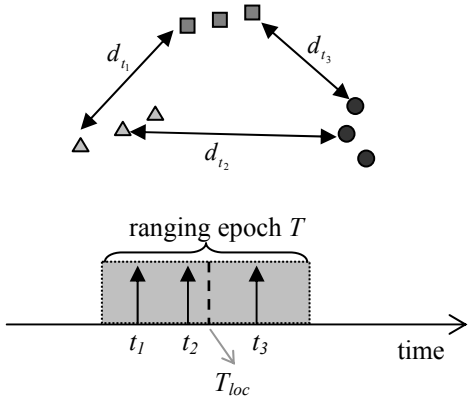

Figure 1: Position uncertainty due to measurement delay

WUWNet'08, September 15, 2008, San Francisco, California, USA.

In traditional terrestrial systems, this effect hardly ever comes into play. However, as we will illustrate shortly, in underwater networks, it can be very significant and the reason is the very nature of the underwater communication environment: low data rate, long propagation delays and inherent mobility. The goal of this paper is to devise a novel localization scheme that limits the resulting error on self-localization performance.

2. PROBLEM DESCRIPTION

Communication underwater has long been known to be challenging. Acoustic channels are characterized by long propagation delays and acoustic modems can achieve relatively low data rates [12]. For example, the micro-modem developed by WHOI [2] transmits at 80 bps, and even short data packets take around a second or more to complete. With speed of sound underwater being around 1500 m/s, propagation delays can also be in the order of a second. As such, to avoid collisions in time, MAC protocols end up spacing competing transmissions over multiple tens to hundreds of seconds. For example, with packets of 10 bytes and distances of 500 m, CSMA backoff needs to be around 300s to limit collisions to less than 5% in moderately dense networks[†]. This means that the ranging epoch T is in the order of 100 seconds. On the other hand, current speeds vary between 0.1 to 1 m/s [19], while relative speeds of guided AUVs may even exceed this [18]. As a result, displacement in node positions during localization can range from a few tens to up to a hundred meters.

This displacement (essentially a ranging error due to ambiguity) is much larger than the intrinsic ranging errors of the system. We performed ranging experiments with two WHOI micro-modems in Mission Bay, San Diego [13]. When stationary, the ranging error is only a few meters, and consistent across various scenarios. Therefore, and as we will show in the results section of this paper, mobility causes significant degradation of self-localization performance in underwater networks, if traditional techniques are utilized. These traditional collaborative self-localization techniques are primarily designed for static networks and do not explicitly account for displacement of nodes during the ranging epoch [6][7][8][9][10]. Conversely, most tracking solutions for mobile robots only track devices individually with respect to anchors [14][15]. While inter-robot distance measurements have been used to improve tracking performance [3], they do not operate in a setting where many nodes only have other to-be-localized nodes as neighbors with distance measurements being the only information available for tracking. In fact, a key difference in localizing nodes underwater compared to tracking terrestrial mobile nodes is that the medium in which nodes are moving has its own unknown motion. Therefore any velocity information that nodes obtain independently is only relative to that of currents, providing little information about the motion of the system in an absolute sense, while terrestrial mobile nodes can often obtain velocity information that aids in tracking.

[†] Although CDMA could allow simultaneous transmissions, it is difficult to assign orthogonal codes in a mobile network with low overhead. Also it does not allow sending and receiving at the same time (and therefore does not allow concurrent ranging either). We assume a CSMA or TDMA style MAC protocol, as is common for most acoustic modems.

There are ongoing efforts towards more dense systems, built on cheaper short range modems [11]. The problem of node mobility during the ranging epoch is especially relevant to such short range systems since errors are large enough to skew our estimate of the network topology. Even if nodes move, it is important to have consistent position estimates across sampled data, as these sensor networks are data-centric (i.e. the correlation between data samples is important, not the identity of the specific device that collected the data; as such tracking individual devices is not important).

Our goal is to devise a collaborative localization scheme that effectively compensates for node motion within the ranging epoch, around the specific localization time of interest T_{loc}. Our scheme is targeted towards situations where localization information is only needed once the mission has finished. This is reasonable since in many applications, data gathered by underwater instruments is analyzed offline by scientists. It is only then that annotation of position information to data samples is needed. As such, while submerged, the networked devices only collect range information. The actual self-localization algorithm itself can operate post-mission, where ample computation resources are available. We will leverage this in the current version of our scheme, which is specifically targeted towards this subset of applications.

Although applicable to the more general class of mobile underwater networks, our solution was developed also with a specific system in mind. This system, depicted in Figure 2, consists of a networked swarm of autonomous sensor-equipped drifters. A prototype system was described in our earlier work [1]. Our goal is to collaboratively localize nodes in such a network in the presence of mobility and delay in obtaining inter-node distance measurements. In the next section, we formally define the localization problem and introduce a solution strategy.

Figure 2: Networked swarm of underwater drifters

3. SOLUTION STRATEGY

We will mathematically formulate the problem of localizing nodes from non-concurrent distance measurements. As described earlier, nodes perform ranging with their neighbors during the localization epoch, T. We denote the set of measurements taken between node pairs at any time t as d_t. The collection of all distance measurements obtained in the interval $(0,T)$ is denoted by $\{d_t\}_{t\in(0,T)}$. Given these measurements, our objective is to determine the maximum-a-posteriori (MAP) estimate for the position of each

node i at a target localization time, T_{loc} which we will denote by $\theta_{i,Tloc}$. The MAP estimate is formally given by equation (1).

$$\theta^*_{i,T_{loc}} = \arg\max\, p(\theta_{i,T_{loc}} \mid \{d_t\}_{t\in(0,T)}) \quad (1)$$

To compute this estimate, we need the probability distribution of each node's position given all distance measurements. If measurements were made with beacons alone, the problem would simplify to independently tracking each node. However, since this is not the case, we have to take into account that node positions over the localization interval are correlated. This correlation is induced by inter-node distance constraints and can be expressed by the joint distribution of node positions in the interval (0, T). The position distribution of each node at T_{loc} is derived from the joint distribution, equation (2). In order to describe the joint distribution over finite number of variables, we will annotate distance measurements in discrete time. The joint distribution is then defined over node positions at discrete instances in the interval (0, T).

$$p(\theta_{i,T_{loc}} \mid \{d_t\}_{t=0}^{T}) = \int_{\Theta} p(\Theta, \theta_{i,T_{loc}} \mid \{d_t\}_{t=0}^{T}).d\Theta \quad (2)$$

Where, $\Theta = \left\{ \{\theta_{k,t}\}_{t=0,k\in U}^{T} \setminus \{\theta_{i,T_{loc}}\} \right\}$

U : Set of all unknown nodes

A direct approach to evaluating equation (2) is infeasible. However, the general problem of computing the distribution of individual variables from a global function defined over many variables is frequently encountered in coding theory. Solutions for particular instances of this problem (i.e. particular structures of the joint distribution) have been previously proposed under different names. Some examples are the Forward /Backward algorithm or BCJR, iterative turbo decoding and decoding of LDPC codes. However, it has been shown that all these algorithms and many others (Pearls belief propagation and even Kalman filters) are all instances of a single generic message passing algorithm, the sum-product algorithm that operates on a 'factor-graph'[4][5].

Factor graphs offer a graphical way of representing any generic global function, often a joint probability distribution, in terms of simpler local functions that depend only on a subset of variables. The sum product algorithm then exploits these simple relations to efficiently and simultaneously compute the marginal distributions of all variables via iterative message passing. For us, this is particularly interesting because factor graphs offer a simple way to capture the complex structure of the localization problem, allowing all distance measurements to be considered simultaneously for estimation. Further, the sum-product algorithm can efficiently perform the actual estimation. A brief overview of the sum-product algorithm is given in Appendix A. The reader is referred to excellent tutorials by Kschischang et al [4] and Loeliger [5] for an in-depth review of the algorithm and its applications to various problems. In subsequent sections we will demonstrate how the sum-product algorithm can be used to solve our specific localization problem.

4. COMPUTING POSITION DISTRIBUTIONS

We begin by obtaining a graphical representation for the joint distribution of node positions. We model the motion of nodes in time as Markovian i.e. the position of a node at any time *t,* given its position at all previous times only depends on its most recent position. By applying the chain rule, the joint distribution can be expressed as a product of simpler functions, equation (3).

$$p(\{\theta_{k,t}\}_{t=0,k\in U}^{T} \mid \{d_t\}_{t\in(0,T)}) \sim \Gamma.\Psi.\prod_{i\in U} p(\theta_{i,0})$$
$$\Gamma = \prod_{t=1}^{T}\left(\prod_{(i,j)\in L_t} p(d_{ij}^{t} \mid \theta_{i,t}, \theta_{j,t}) \prod_{i\in U} p(\theta_{i,t} \mid \theta_{i,t-1}) \right) \quad (3)$$
$$\Psi = \prod_{(i,j)\in L_0} p(d_{i,j}^{0} \mid \theta_{i,0}, \theta_{j,0})$$

Where, L_t is the set of node pairs that obtained a measurement at time t.

The equivalent factor graph for equation (3) is given by Figure 3. The position of any node i at time t is symbolized by a circle and we will here on refer to it as the *state-variable* $\Theta_{i,t}$. Functions that relate state-variables are represented as square nodes. A link exists between a state-variable and a function node if the state-variable is an argument of the function. The progression of node positions in time is shown in the horizontal dimension. The vertical dimension indicates this time progression for each unknown node.

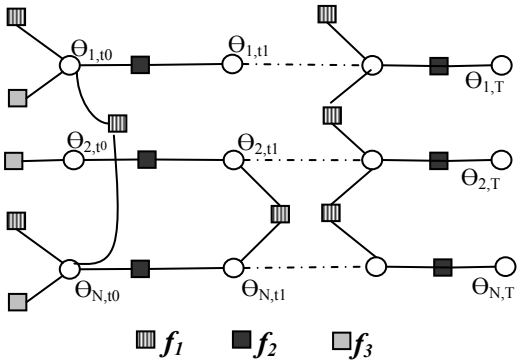

Figure 3: Factor-graph representation of the localization problem

The probabilistic model that describes the evolution of a node's position at each time step is given by function-nodes of type f_2. We model node speeds to be uniformly distributed between (0, v_{max}), equation (4).

$$f_2(\theta_{i,t_{k-1}}, \theta_{i,t_k}) = p(\theta_{i,t_k} \mid \theta_{i,t_{k-1}}) = \left\{ \begin{array}{ll} \dfrac{1}{\pi(v_{\max}\delta_t)^2}, & \left\|\theta_{i,t_{k-1}} - \theta_{i,t_k}\right\| \prec v_{\max}\delta_t \\ 0, & \left\|\theta_{i,t_{k-1}} - \theta_{i,t_k}\right\| \geq v_{\max}\delta_t \end{array} \right\} \quad (4)$$

Where, δ_t is the granularity at which time is discretized.

The likelihood of a distance measurement between a node pair is given by functions of type f_1. If a measurement is obtained with a beacon the corresponding function-node is single-ended i.e. it has a link only to the unknown state-variable as shown in Figure 3. Our experiments with the WHOI micro-modem [13] suggest that a zero mean Gaussian model is well suited for ranging error, equation (5). However, our solution can accommodate any other model.

$$f_1(\theta_{i,t}, \theta_{j,t}) = p(d_{ij}^t \mid \theta_{i,t}, \theta_{j,t}) \sim N\left(\left\| (\theta_{i,t}, z_{i,t})^T - (\theta_{j,t}, z_{j,t})^T \right\|, \sigma \right) \quad (5)$$

Where, $z_{i,t}$ is the depth at which node i is at time t.
σ is the standard deviation of ranging error.

Further, we may know that the position of unknown nodes is restricted to a finite region, ζ around the beacons. Function-nodes of type f_3 describe the distribution for node locations over this region. If no additional information is known, we assume that node positions can be anywhere in ζ with equal probability, equation (6)

$$f_3(\theta_{i,0}) = p(\theta_{i,0}) = \begin{Bmatrix} \frac{1}{A_\zeta}, \theta_{i,0} \in \zeta \\ 0, \theta_{i,0} \notin \zeta \end{Bmatrix} \quad (6)$$

Where, A_ζ is the area of the region ζ.

Now that the factor-graph model for the problem has been obtained, the sum-product algorithm can be applied to compute the distributions of state-variables via iterative message passing. Next, we will present the actual messages generated by nodes and then describe their physical interpretation for the localization problem.

4.1 Position Estimation via Message Passing

During the operation of the sum-product algorithm, messages are generated by state-variables and function nodes. Let $\mu_{x\text{-}f}(x)$ denote the message sent from a state-variable, $\boldsymbol{x}$ to a function node f (X), where X is the set of arguments of f. Let $\mu_{f\text{-}x}(x)$ be the message sent from a function node to a state-variable. Also, let $n(w)$ denote the set of neighbors of a given node w on the graph. Messages are computed by each node as per the following rule [4].

$$\mu_{f-x}(x) = \sum_{\sim\{x\}} \left(f(X) \prod_{y \in n(f)\setminus\{x\}} \mu_{y-f}(y) \right) \quad (7)$$

$$\mu_{x-f}(x) = \prod_{h \in n(x)\setminus\{f\}} \mu_{h-x}(x) \quad (8)$$

The algorithm operates as follows. Per iteration, nodes compute outgoing messages on all their links based on the (latest) messages that had arrived on those links in a previous iteration. Since the factor-graph representation of our problem is cyclic, nodes initiate message-passing by assuming that a unit message has arrived on each of their links. Further the algorithm has to iterate a number of times to converge. The estimate of a state-variable's distribution is the product of all its incoming messages. To obtain a physical interpretation of the operation of the sum-product algorithm, it is sufficient to examine a message passed from one state-variable to another via a function node as shown in Figure 4. In step 1, the message, $\mu_{x\text{-}f}(x)$ is sent from state-variable $\boldsymbol{x}$ to function-node f with $f \in \{f_1, f_2\}$. This message is an estimate of the probability mass distribution of $\boldsymbol{x}$. Next, in step 2, node $\boldsymbol{f}$ computes an estimate of the distribution of the position of $\boldsymbol{y}$ based on the likelihood of any information that relates $\boldsymbol{x}$ and $\boldsymbol{y}$. This information could be a distance measurement or a model of how a node's position evolves in time.

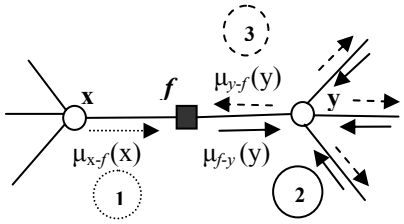

Figure 4: Messages passed between two state-nodes

We observe in step 2, that a number of messages simultaneously arrive at $\boldsymbol{y}$, each being an individual estimate of its position-distribution. Node $\boldsymbol{y}$ intersects all these individual distributions to obtain an estimate of its distribution. This is the product step of the algorithm. In step 3, $\boldsymbol{y}$ sends out the most recent estimate of its distribution to all its neighbors. Therefore, information obtained from distance measurements made at different instances travels back and forth (in time and space), modifying the distributions of node positions along the way. One of the main challenges in applying the above solution to the problem of network localization is an efficient representation for the distribution of node positions. As such the sum-product algorithm operates on discrete state-variables. Prior to localization there is a large uncertainty (up to a few kms) in node locations. Therefore, fine-grained discretization would result in unrealistically long messages while a nominal message size gives rise to coarse and inaccurate estimates. Therefore, we need an efficient way to represent position distributions over a large region without losing information.

4.2 Efficient representation of position distributions

As discussed in the previous section, each run of the sum-product algorithm gives an estimate of position-distributions. We store distributions as a set of weights defined over grids. Thus, the length of messages generated by the sum-product algorithm is directly proportional to the number grids used for representing each distribution. In order to obtain fine-grained estimates with nominal message size, we start with coarse representations (large grids) and iteratively shrink the region over which distributions are defined based on the likelihood of node locations (as obtained from the sum-product algorithm). Per iteration, we select the smallest area in which a node resides with high probability. This region is then re-discretized keeping the number of grids (hence the message size) constant. The sum-product algorithm is used in the next iteration to compute new likelihood estimates. The algorithm terminates when the change in grid size is less than a set threshold or when a target granularity is reached for all state-

variables. We further propose methods for efficient and fast operation of the algorithm.

Since representations for node distributions are not modified during the operation of the sum-product algorithm, mappings defined by function nodes can be pre-computed at the end of each run of the algorithm. We had defined function nodes in equations (4), (5) and (6) on continuous random variables. We must modify these mappings for grid representations. Since grids can be of large size, the position of nodes cannot be approximated as the grid-center which is commonly done in grid-based representations[22]. Function-nodes introduced earlier are redefined for grid representations in Appendix B. If we estimate node-positions for all time instances in the localization interval T and the granularity of time discretization is δ_t, the number of state-variables grows with finer discretization as per equation (9).

$$N_s = N.\frac{T}{\delta_t} \tag{9}$$

Where, N is the number of nodes with unknown positions.

However, the only node-states that are informative are at the target localization time and instances when a distance measurement was obtained. By including node states only for such instances, the number of state-variables in the factor-graph no longer follows equation (9). Instead it tends to a constant as δ_t goes to zero because the number of distance measurements is finite. By maintaining only necessary states fewer messages are generated. In addition, the maximum number of iterations required for convergence of the sum-product algorithm depends on the width of the factor-graph, defined as the shortest path of maximum length. Reducing state-variables also reduces the graph-width and significantly drops the run-time of the algorithm. Next, we present simulation results when our proposed solution strategy is used for localization.

5. RESULTS

To evaluate the performance of our localization scheme, we performed simulations in Parsec [20]. All simulation parameters are summarized in Table 1. We deployed nodes over a 3D region where they move with current streams of equal thickness. The velocity of these streams varies with depth which is a commonly assumed model [21]. The speed of each layer is independently chosen between 0 and v_{max}.

Distance estimates are obtained from broadcast transmissions. However prior to transmitting, nodes choose a random back off between 0 and $T_{backoff}$ to avoid collisions. Four surface beacons are used to localize a network of 15 nodes. The transmission range is R with some variation. We estimate the position of nodes in 2D since nodes know their depth from pressure measurements.

Figure 5 shows a view of the network when looking from above and elucidates the problem we are trying to address. In Figure 5, the movement of nodes during the time all distance measurements are obtained is shown by blue dots. The actual positions of unknown nodes at a target localization time are represented by stars. Beacon positions at that time are shown as triangles. We observe that nodes are displaced between 10m to 180m during localization which is considerably large compared to their transmission range.

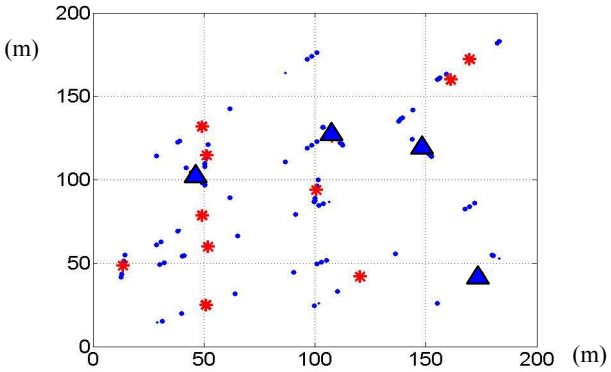

Figure 5: Movement in node positions during localization

For the above scenario we used our proposed method to localize nodes at a target localization time. However, to compare the performance of our scheme we also estimate node positions using a robust self-localization algorithm, Multi-dimensional Scaling (MDS) [6]. To enhance the performance of MDS we chose for each node pair distance measurements that were closest to the target time.

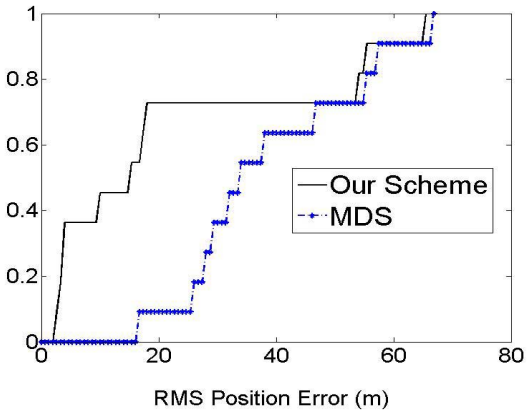

Figure 6: Cumulative error distribution of nodes

Figure 6 shows the cumulative error distribution for position estimates using our scheme and MDS. Our proposed scheme localized 70 % of nodes with error lower than the minimum error obtained from MDS. The actual distributions of node positions are shown in Figure 7, with the estimate for each node indicated by circle. In Figure 6 we observed that around 30% of the nodes had a much larger error compared to the rest of the network when our scheme was used. This is because three of the eleven unknown nodes have multimodal distributions due to insufficient measurements. The distribution of these nodes is spread out over a large region as shown in Figure 7. We used only 36 grids to represent the distribution of each node. The effect of allowing variable grid sizes can also be observed in Figure 7 where fine granularity is achieved for nodes that are well constrained.

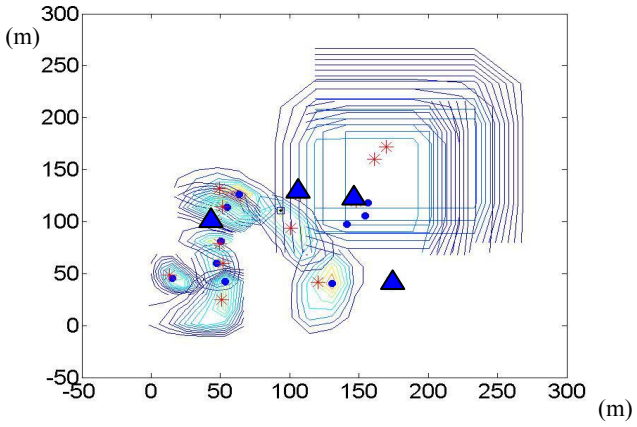

Figure 7: Probability distributions of node positions

Table 1: Simulation Parameters

Max Current speed, v_{max}	60 cm/s
Thickness of current layers	10 m
Max Depth , D	100 m
Transmission range, R	100 m
Number of nodes, N	15
Number of beacons, N_B	4
Area of deployment	300m x 300 m
MAC Back-off, $T_{backoff}$	300 s
Number of grids per node	36

6. RELATED WORK

Graphical inference has been previously used to localize a network of static nodes [17]. Here non-parametric belief propagation is used for estimating node positions. We use the general framework of factor graphs to solve the extended problem where nodes are mobile. The sum-product algorithm is equivalent in operation to a number of estimation methods including belief propagation. In a mobile setting collaborative robot localization has been proposed [3]. Here nodes use proximity information with neighbors to improve their position estimates. The proposed method is aimed at real-time localization and incorporates each measurement sequentially in time. It does not have a mechanism to use measurements to improve past estimates. Further node distributions are assumed independent leading to inaccuracies in estimates as reported. This scheme is based on an earlier work in global positioning of robots that assumes a Markov model for the propagation of positions in time [16]. As such our present scheme is also based on a Markov model for node motion, however, due to the generic framework that factor graphs provide other motion models can be incorporated. A method to incorporate non-simultaneous measurements for tracking the path of a mobile robot moving through a fixed beacon field has been proposed by Corke et al. [15]. Underwater nodes are individually tracked by beacons. We extend this to the case where nodes have to be localized based on measurements with other unknown nodes.

7. CONCLUSIONS

We have considered collaborative localization of a network of mobile nodes in the presence of a few surface beacons. We show that a specific problem arises while applying self-localization algorithms to estimate node positions underwater. Due to the high latency in acoustic communication and low data rates of acoustic modems, nodes have to back off for a substantial time before they can transmit, introducing a large delay in obtaining distance-measurements. The combined effect of this delay and node mobility causes node positions to change during localization resulting in erroneous estimates. Our proposed method shows significant improvement in localization performance by taking into account these effects.

ACKNOWLEDGEMENTS

This work was supported in part by the NSF under award number ECCS-0622005. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of NSF. We would like to thank our collaborator Jules Jaffe at Scripps for his support and Zeinab Taghavi for her insightful inputs.

REFERENCES

[1] J. Jaffe, C. Schurgers, “Sensor networks of freely drifting autonomous underwater explorers,” WUWNET’06, Los Angeles, CA, pp. 93-96, Sept 2006.

[2] Woods Hole Oceanographic Institution, Acoustics Communications group, “WHOI Micro-modem,” http://acomms.whoi.edu/micromodem/.

[3] D. Fox, W. Burgard, H. Kruppa, S. Thrun, “A Probabilistic Approach to Collaborative Multi-Robot Localization”, Autonomous Robots, 8(3):325–344, 2000.

[4] F.R. Kschischang, B.J. Frey, H.A. Loeliger,”Factor graphs and the sum-product algorithm,” Information Theory, IEEE Transactions on, Vol.47, No. 2. (2001), pp. 498-519.

[5] H.A. Loeliger,”An introduction to factor graphs”, Signal Processing Magazine, IEEE, Vol. 21, No. 1. (2004), pp. 28-41.

[6] Y. Shang, W. Ruml, Y. Zhang, M. Fromherz, “Localization from Mere Connectivity,” ACM MobiHoc, June 2003.

[7] P. Biswas, Y. Ye, “Semidefinite programming for ad hoc wireless sensor network localization,” IPSN’04, Berkeley, CA, Apr. 2004.

[8] L. Doherty, K. S. J. Pister, L. E. Ghaoui, “Convex position estimation in wireless sensor networks,”INFOCOM 2001, Anchorage, AK, Apr. 2001.

[9] D. Goldenberg, P. Bihler, M. Cao, J. Fang, B. Anderson, A. Morse, Y. Yang, “Localization in Sparse Networks using Sweeps,” ACM MOBICOM, Los Angeles, CA, Sept 2006.

[10] V. Chandrasekhar, W.K.G Seah, Y.S. Choo, H.V. Ee, “Localization in Underwater Sensor Networks -- Survey and Challenges,”WUWNET’06, Los Angeles, CA, pp. 33-40, Sept 2006.

[11] J. Wills, W. Ye, and J. Heidemann, "Low-power acoustic modem for dense underwater sensor networks," WUWNET'06, Los Angeles, CA , Sept 2006.

[12] J. Heidemann, Y. Li, A. Syed, J Wills and W. Ye, "Research Challenges and Applications for Underwater Sensor Networking", Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC2006), 2006

[13] Diba Mirza, Curt Schurgers, "Energy-Efficient Ranging for Post-Facto Self-Localization in Mobile Underwater Networks", submitted to IEEE JSAC Special Issue on Underwater Wireless Communications and Networks, March 2008.

[14] J.J. Leonard, H.F. Durrant-Whyte ,"Mobile robot localization by tracking geometric beacons", IEEE Transactions on Robotics and Automation, Vol(7) pp 376-382, Jun 1991

[15] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, I. Vasilescu, "Experiments with Underwater Robot Localization and Tracking", In Proc. of 2007 IEEE International Conference on Robotics and Automation,pp 4556-4561, April 2007.

[16] D. Fox. Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Naviagation. PhD thesis, Dept. of Computer Science, University of Bonn, Germany, December 1998.

[17] A. Ihler, J. Fisher, R. Moses, A. Willsky.," Nonparametric belief propagation for self-calibration in sensor networks" Information Processing in Sensor Networks (IPSN) 2004, pp. 225- 233, April 2004.

[18] T. Chance, A. Kleiner, J. Northcutt, "The Autonomous Underwater Vehicle (AUV): A Cost-Effective Alternative to Deep-Towed Technology," Integrated Coastal Zone Management, 6th Edition, pp. 65-69.

[19] Speed of Ocean Currents, The Physics Factbook, http://hypertextbook.com/facts/2002/EugeneStatnikov.shtml

[20] PARSEC, parallel simulation environment for complex systems, http://pcl.cs.ucla.edu/projects/parsec/

[21] D. Pompili, T. Melodia, I. F. Akyildiz, "Deployment Analysis in Underwater Acoustic Wireless Sensor Networks," in Proc. of ACM International Workshop on UnderWater Networks (WUWNet 06), Los Angeles, CA, pp 48-55, September 2006.

[22] W. Burgard, D. Fox, D. Hennig, T. Schmidt, "Position tracking with position probability grids", Proceedings of the First Euromicro Workshop on Advanced Mobile Robot, Oct. 1996.

APPENDIX A

Summary of the sum-product algorithm [4]

The sum-product algorithm is the solution to the following generic problem. Given a global function $g(x_1, x_2, x_3,\ldots, x_n)$, where each x_i takes values in the discrete domain A_i, the sum-product algorithm computes simultaneously and efficiently the *summary* for each x_i where the summary function is defined in equation (A.1)

$$\sum_{\sim x_1} g(x_1, x_2, x_3) = \sum_{x_2 \in A_2} \sum_{x_3 \in A_3} g(x_1, x_2, x_3) \tag{A.1}$$

When the global function is a joint distribution, the algorithm naturally computes the marginal distributions for each x_i. To do this, it exploits the way the global function *g()* factorizes. For an example function given by equation (A.2), the equivalent factor graph is given in Figure 8.

$$g(x_1, x_2, x_3, x_4) = f_A(x_1, x_2) f_B(x_3, x_4) f_C(x_1, x_4) \tag{A.2}$$

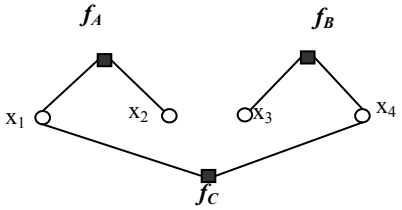

Figure 8: An example factor-graph

APPENDIX B

Defining function nodes for grid representations:

The position of a node, *i* at time *t* is denoted by $\Theta_{i,t}$. Its domain is the collection of grid-centers and the grid size, denoted by $G_{i,t,k}=(X_{i,t,k}, Y_{i,t,k}, a_{i,t}),k=1,....S,$ Where. *S* is the number of grids used to represent distributions.

For any two state-variables, $\Theta_{i,t}$ and $\Theta_{j,t}$, each belonging to grids $G_{i,t,k}$ and $G_{j,t,m}$ respectively , the function $\boldsymbol{f_1}(\Theta_{i,t}, \Theta_{j,t})$ is computed as the likelihood of a distance measurement, d_{ij} averaged for all points in grids $G_{i,t,k}$ and $G_{j,t,m}$. Starting with the definition of $\boldsymbol{f_1}$ in equation (5) this is given by equation (B.1)

$$f_1(\theta_{i,t}, \theta_{j,t}) = p(d_{ij}^t \mid \theta_{i,t} = G_{i,t,k}, \theta_{j,t} = G_{j,t,m})$$

$$= \frac{\int_{x_i}\int_{y_i}\int_{x_j}\int_{y_j} p(d_{ij}^t \mid |x_i - x_j|, |y_i - y_j|, z_{ij}) dx_i dy_i dx_j dy_j}{(a_{i,t}^2 a_{j,t}^2)}$$

$$x_i \in \left(X_{i,t,k} - \frac{a_{i,t}}{2}, X_{i,t,k} + \frac{a_{i,t}}{2}\right), x_j \in \left(X_{j,t,m} - \frac{a_{j,t}}{2}, X_{j,t,m} + \frac{a_{j,t}}{2}\right)$$

$$y_i \in \left(Y_{i,t,k} - \frac{a_{i,t}}{2}, Y_{i,t,k} + \frac{a_{i,t}}{2}\right), y_j \in \left(Y_{j,t,m} - \frac{a_{j,t}}{2}, Y_{j,t,m} + \frac{a_{j,t}}{2}\right)$$

$$z_{ij} = z_{i,t} - z_{j,t} \tag{B.1}$$

Where $z_{i,t}$, $z_{j,t}$ are the depths of node i and node j respectively at time t.

Equation (B.1) when further simplified yields equation (B.2).

$$f_1(\theta_{i,t},\theta_{j,t}) = p(d_{ij}^t \mid \theta_{i,t} = G_{i,t,k}, \theta_{j,t} = G_{j,t,m})$$

$$= \frac{\int_{-\alpha}^{\alpha}\int_{-\alpha}^{\alpha} p(d_{ij}^t \mid (x - X_{ik,jm}^t, y - Y_{ik,jm}^t, z_{ij}) U(x,\alpha,\beta,\gamma) U(y,\alpha,\beta,\gamma) dx dy}{(a_{i,t}^2 a_{j,t}^2)}.$$

$$a_{i,t} \neq 0, a_{j,t} \neq 0; \alpha = (a_{i,t} + a_{j,t})/2; \beta = \left|a_{i,t} - a_{j,t}\right|/2; \gamma = \min(a_{i,t}, a_{j,t})$$

$$X_{ik,jm}^t = X_{i,t,k} - X_{j,t,m}, Y_{ik,jm}^t = Y_{i,t,k} - Y_{j,t,m}$$

$$\text{Where, } U(x,\alpha,\beta,\gamma) = \begin{Bmatrix} \alpha - |x|, \beta < |x| \le \alpha \\ \gamma, \quad 0 < |x| \le \beta \end{Bmatrix} \quad \text{(B.2)}$$

When a measurement is made between an unknown node, $\Theta_{i,t}$ and a beacon whose position is $Q_{b,t}=(X_{b,t},Y_{b,t},Z_{b,t})$, functions of type $\boldsymbol{f_1}$ are computed as per equation (B.3)

$$f_1(\theta_{i,t}) = p(d_{ij}^t \mid \theta_{i,t} = G_{i,t,k}, Q_{b,t})$$

$$= \frac{1}{a_{i,t}^2} \int_{x_i}\int_{y_i} p(d_{ij}^t \mid \left|x_i - X_{b,t}\right|, \left|y_i - Y_{b,t}\right|, z_{ib}) dx_i dy_i,$$

$$x_i \in \left(X_{i,t,k} - \frac{a_{i,t}}{2}, X_{i,t,k} + \frac{a_{i,t}}{2}\right), y_i \in \left(Y_{i,t,k} - \frac{a_{i,t}}{2}, Y_{i,t,k} + \frac{a_{i,t}}{2}\right) \quad \text{(B.3)}$$

$$z_{ib} = z_{i,t} - z_{b,t}$$

We also observe that the likelihood function is the same for different orientations of a node pair as long as their relative orientation is the same. Therefore, we compute probabilities for unique orientations alone.

Functions of type $\boldsymbol{f_2}$ compute the transition probabilities, $P(\Theta_{i,t1}, \Theta_{i,t2})$. Node speeds are modeled to be uniformly distributed between 0 and v_{max}. Let state-variables $\Theta_{i,t1}$ and $\Theta_{i,t2}$ belong to grids $G_{i,t1,k}$ and $G_{i,t2,m}$ respectively. We compute approximate transition probabilities as follows. If $a_{i,t1} < a_{i,t2}$, expand each side of the grid $G_{i,t1,k}$, by $v_{max.}|t_2-t_1|$ to obtain the region $Q_{i,t1,k}$. If $Q_{i,t1,k} \cap G_{i,t2,m} \neq \phi$, $P(\Theta_{i,t1} = G_{i,t1,k}, \Theta_{i,t2} = G_{i,t2,m}) = 1/M$. Where, M is the total number of grids belonging to $\Theta_{i,t2}$ that have a non-empty intersection with $Q_{i,t1,k}$ else $P(\Theta_{i,t1} = G_{i,t1,k}, \Theta_{i,t2} = G_{i,t2,m}) = 0$.