Scalable Localization with Mobility Prediction for Underwater Sensor Networks

Zhong Zhou, Jun-Hong Cui and Amvrossios Bagtzoglou

UCONN CSE Technical Report: UbiNet-TR07-01

Last Update: July 2007

Abstract

Due to adverse aqueous environments, non-negligible node mobility and large network scale, localization for large-scale mobile underwater sensor networks is very challenging. In this paper, by utilizing the predictable mobility patterns of underwater objects, we propose a scheme, called Scalable Localization scheme with Mobility Prediction (SLMP), for underwater sensor networks. In SLMP, localization is performed in a hierarchical way, and the whole localization process is divided into two parts: anchor node localization and ordinary node localization. During the localization process, every node predicts its future mobility pattern according to its past known location information, and it can estimate its future location based on its predicted mobility pattern. Anchor nodes with known locations in the network will control the whole localization process in order to balance the tradeoff between localization accuracy, localization coverage and communication cost. We conduct extensive simulations, and our results show that SLMP can greatly reduce localization communication cost while maintaining relatively high localization coverage and localization accuracy.

I. Introduction

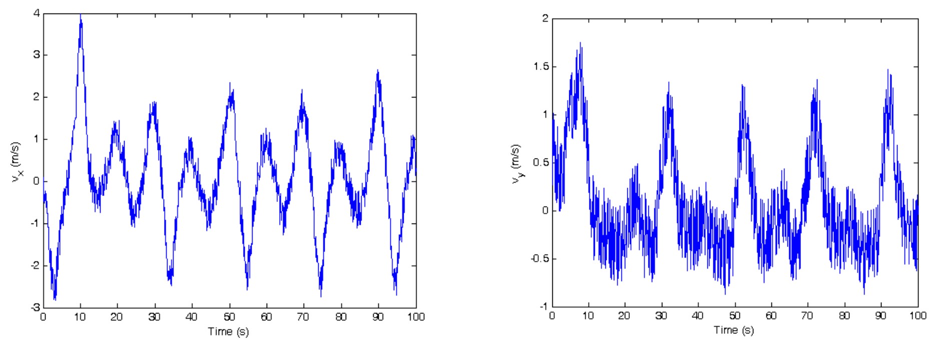
Last several years have overseen a rapidly growing interest in underwater sensor networks [1], [10], [16], [27]. One important reason is because they can offer significant advantages and benefits in a wide spectrum of aquatic applications: underwater environmental observation for scientific exploration, commercial exploitation, and coastline protection and target detection in military or terrorist events. On the other hand, underwater sensor networks probably represent one of the hardest networking environments we have ever encountered. They drastically differ from terrestrial sensor networks: i) electromagnetic waves cannot propagate over a long distance in water, thus, underwater sensor networks must rely on acoustic signals that feature large-latency, low-bandwidth, and long end-to-end delays; and ii) sensor nodes move with water currents and dispersion, which results in dynamic network configurations that need to be monitored. All these unique features cause grand challenges to the networking issues at almost every layer of the protocol stack [1], [10], [16], [27].

Localization of mobile sensor nodes is indispensable for underwater sensor networks. For example, in aquatic environment monitoring applications, localization is a must-do task in order to get useful location-aware data. Location information is also required for geo-routing, which is proved to be more scalable and efficient in mobile underwater sensor networks [35]. So far, only a limited number of schemes have been proposed for the localization service in underwater acoustic networks [2], [4], [12], [36]. These solutions are mainly designed for small-scale static networks (usually with tens of nodes or even less). However, many aquatic applications, such as marine surveillance, requires a localization solution that can scale to a large number (hundreds to thousands) of nodes. In this paper, we focus on the localization service for large-scale mobile underwater sensor networks.

Due to adverse aqueous environments, non-negligible node mobility and large network scale, localization for large-scale mobile underwater sensor networks is very challenging. Since radio does not work in water, acoustic communications have to be employed. The unique features of acoustic channels (large-latency, low-bandwidth, and long end-to-end delays) cause many constraints on the localization schemes for underwater sensor networks. Traditional multi-hop localization schemes for terrestrial sensor networks are inefficient because of their huge communication overhead. Meanwhile, underwater sensor networks are mobile networks and node locations change continuously. In such environments, most localization schemes designed for static sensor networks need to run periodically to update the location results, as will dramatically increase the communication overhead. Further, distributed localization schemes designed for small-scale underwater acoustic networks [12], [36] can not work well in large-scale underwater sensor networks due to their slow convergence speed and high communication overhead. Last, high localization coverage and low localization error are always desired for a localization scheme, while these performance requirements are specially challenging for underwater sensor networks with stringent resource limitation. In this paper, *we aim to design a scalable localization scheme with low communication overhead while good localization performance for large-scale underwater water sensor networks*.

Though the network conditions in underwater environments are extremely tough for localization (as we discussed above), some unique properties can be indeed effectively exploited. A very useful property we find is that objects underwater move with predictable patterns, though these patterns are in a large part determined by environmental factors [3], [26]. This mobility property can actually provide us an alternative for high performance localization. In this paper, we propose a scheme, called **S***calable* **L***ocalization scheme with* **M***obility* **P***rediction (***SLMP***)*, for underwater sensor networks. In SLMP, localization is performed in a hierarchical way, and the whole localization process is divided into two parts: anchor node localization and ordinary node localization. During the localization process, every node predicts its future mobility pattern according to its past known location information, and it can estimate its future location based on its predicted mobility pattern. Anchor nodes with known locations in the network will control the whole localization process in order to balance the tradeoff between localization accuracy, localization coverage and communication cost. Simulation results show that SLMP can greatly reduce localization communication cost while maintaining relatively high localization coverage and localization accuracy.

Our contributions are two-folds. First, we propose a novel localization scheme, SLMP. This scheme is scalable and can achieve a good balance between localization accuracy, localization coverage and communication cost. Second, as a case study, we analyze the mobility pattern in seashore environments and propose a simple yet effective mobility prediction algorithm. Third, we investigate the performance of SLMP through extensive simulations. Our results show that some useful technologies such as linear prediction algorithms in signal processing can be imported for network localization. We hope that our work can shed some light on this new research direction.



(a) Velocity in x-axis (b) Velocity in y-axis

Fig. 1. Mobility patterns of underwater objects in a seashore environment

The rest of this paper is organized as follows. In Section II, we will give some background knowledge on underwater object mobility, and briefly review some related work on localization. Then, in Section III, we will describe SLMP in details using seashore environments as an example. Following that, we present simulation results in Section IV. Finally, we conclude the paper in Section V.

II. Background and Related Work

A. *Mobility Characteristics of Underwater Objects*

Underwater objects are moving continuously with water currents and dispersion. Research in hydrodynamics shows that the movement of underwater objects is closely related to many environment factors such as the water current, water temperature [3], [26]. In different environments, the mobility characteristics of underwater objects are different. For example, the mobility patterns of objects near the seashore demonstrate a certain semi-periodic property because of tides; but for objects in rivers, their moving behaviors have no such property. While it is almost impossible to devise a generic mobility model for underwater objects in all environment conditions, some mobility models for underwater objects in specific environments based on hydrodynamics have been devised [3], [26]. This indicates that the movement of underwater objects is not a totally random process. Temporal and spatial correlation are inherent in such movement, as makes their mobility patterns predictable in nature.

As a case study, in this paper, we will investigate the mobility characteristics of objects in shallow seashore areas (referred to as *seashore environments* in this paper). Tidal areas are characterized by their shallowness and their strong tidal currents. The nonlinear interaction of tidal currents and bottom topography produces currents, which give nonzero contributions to the tidally averaged currents. These so-called residual currents are important for the transport and mixing properties of the tidal flow. In first approximation, the flow in tidal areas can be considered to be the superposition of a time-dependent part (the tidal flow) and a time-independent part (the residual flow). The advection problem in this type of flows, studied in a Lagrangian way, can be written in terms of perturbed 2D Hamiltonian systems. From dynamical system theory it is known that these systems can be characterized as chaotic, which is extremely relevant for mixing in fluids and its environmental consequences. Therefore, we opt to study a model/analog of a tidally-dominated estuary that exhibits complex, chaotic flow characteristics.

Fig. 1 shows the velocity of an underwater object with time in a typical seashore environment. We can clearly see that the moving speed of the object changes continuously and shows certain semi-periodic property. This is mainly due to the tides and bathymetry. The exhibited strong time-domain correlations tell us that it is possible to estimate the future value based on the measured past values with high accuracy[1].

Further, [26] and [3] show that spatial correlations exist in underwater environments, which means that the movement of one object is closely related to its nearby objects. In other words, underwater objects possess certain group movement properties.

B. Related Work on Localization

Localization has been widely explored for terrestrial sensor networks, and a significant number of schemes have been proposed [6]–[9], [11], [13], [15], [18], [20]–[24], [31], [33], [34]. Generally speaking, these schemes can be classified into two categories: range-based schemes and range-free schemes. The former covers the protocols that use absolute point-to-point distance estimates (i.e., range) or angle estimates to calculate locations while the latter makes no assumptions about the availability or validity of such range information . Although range-based protocols can provide more accurate position estimates, they need additional hardware for distance measures, which will increase the network cost. On the other hand, range-free schemes do not need additional hardware support, but can only provide coarse position estimates. In this paper, we are more interested in accurate localization. Moreover, in UWSNs, acoustic channels are naturally employed and range measurement using acoustic signals is much more accurate and cheaper compared with that in terrestrial sensor networks using radio [10], [35]. Thus, range-based schemes are potentially good choices for underwater sensor networks. However, due to the unique characteristics (such as low communication bandwidth, node mobility, and three dimensional node deployment) of underwater sensor networks, the applicability of these existing range-based schemes is unknown.

Localization for terrestrial mobile sensor networks has also been explored recently [17], [32]. In [17], the authors propose a range-free localization scheme based on a sequential Monto Carlo localization method and show that their scheme can exploit mobility to improve the localization accuracy. In [32], the authors propose predictive protocols which can control the frequency of localization based on sensor mobility behavior to reduce the energy requirements while bounding the localization error. Both of these two studies assume that sensor nodes are moving randomly and the inherent properties of object mobility patterns are not explored.

As mentioned in Section I, there are a couple of studies on localization for underwater acoustic networks [2], [4], [12], [36]. These proposals are mainly designed for small-scale static networks. For example, underwater "GPS" systems (GIB (GPS Intelligent Buoys) [4] and PARADIGM [2]) have been proposed based on surface buoys and one hop communication. Localization for sensor nodes is centrally performed at surface buoys. In [36],

[1]In this paper, we assume that objects keep constant in z (depth) axis, and the mobility pattern of objects is only related to the (x, y) axis. Thus, objects with the same (x, y) but different z move with the same mobility pattern. This is a common assumption in hydrodynamics [3], [26]

a distributed protocol is proposed for multi-hop underwater robot networks. This protocol is based on the iterative multilateral methods and is suitable for small-scale static underwater networks. For large-scale mobile underwater sensor networks, this protocol is inefficient because of the high communication cost and low convergence speed.

III. Description of SLMP

In this section, we present SLMP, a scalable localization scheme with mobility prediction for underwater sensor networks. We will first describe the network architecture, and give an overview of SLMP. We then show how SLMP works in details using the mobility patterns in seashore environments.

A. Network Architecture

To accomplish the localization task for large-scale underwater sensor networks, we propose a network architecture that comprises of three different types of nodes, as shown in Fig. 2.

- *Surface Buoys*. Surface buoys are equipped with GPS to obtain their location estimates. They serve as the "satellite nodes" in underwater localization schemes.
- *Anchor Nodes*. Anchor nodes are powerful nodes which can make direct contact with the surface buoys, and are capable of self-localization based on such contacts.
- *Ordinary Nodes*. Ordinary sensor nodes are low-complexity sensor nodes which cannot directly communicate with surface buoys – they are cheap, and they do not wish to be profligate with their energy. Typically, the ordinary sensor nodes can only connect to its local (usually one-hop) neighbors. Through local message passing among themselves and with nearby anchor nodes, the ordinary sensor nodes desire to self-localize so that they can effectively participate in the network operation.

In short, anchor nodes localize themselves via direct communication with several surface buoys (e.g., using existing underwater GPS approaches), while ordinary nodes perform self-localization through neighboring anchor nodes and/or ordinary nodes.

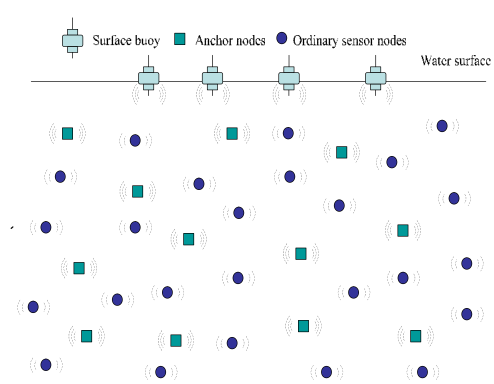


Fig. 2. Underwater sensor network architecture

In the target network, we assume that every sensor node needs to get its location periodically. We define T_1 as the period each node needs to get its location, and we call T_1 *localization period*. This is a reasonable assumption since in many applications such as environment monitoring, sensor nodes only need to report their observed data periodically to the sink. For these applications, location information is only useful at these discrete time points.

It should be noted that when the localization period T_1 is small, a relatively high communication cost will be imported into the network for the localization service. This will put a high burden on the network because of the limited bandwidth of its acoustic channels. But if we can take full advantages of the inherent predictable nature of underwater objects' mobility, it is possible to get needed location information with low communication cost and high localization accuracy. Our "SLMP" is exactly based on this motivation.

B. Overview of SLMP

SLMP adopts a hierarchical localization approach. In SLMP, the whole localization process is divided into two sub-processes: anchor node localization and ordinary node localization.

At the beginning, only several surface buoys know their locations through common GPS or by other means. Four or more buoys are needed in our system. These buoys work as the "satellites" for the whole network and anchor nodes can be localized by these surface buoys. Since anchor nodes are more powerful and can measure their locations directly from the surface buoys in every localization period, some complicated mobility prediction algorithms can also be implemented on them. For the ordinary node localization, we propose a distributed recursive range-based scheme which will be described in Section III-D in details. Since ordinary nodes are limited in computation power and memory, it is hard to implement complicated prediction algorithms on them. Fortunately, due to the group movement properties of underwater objects, an ordinary node can deduce its mobility pattern from the mobility patterns of nodes nearby.

In every localization period, an anchor node estimate its current location, according to its previous location estimations and its predicted mobility pattern. It compares this estimated location with its measured location[2]. If the Euclidean distance between its measured location and its estimated location is larger than the stipulated threshold s_t, this anchor node will judge that its current mobility pattern is not so accurate and needs to be updated. Then, it runs its mobility prediction algorithm (which will be described in Section III-C) to get a new mobility pattern. After that, it will broadcast a new localization message which contains its current location and new mobility pattern to the network. It is clear that a new localization process is initiated by anchor nodes, and it is the anchor nodes who can control the frequency of the localization message flooding.

For an ordinary node, it tries to receive any localization messages it can in the network. If it has not received any localization message for a long period (larger than some predefined threshold), it will judge that it is out of touch with other nodes and will label itself as un-localized. On the other hand, when it receives some localization messages from others, it will run its localization and mobility prediction algorithm to estimate its own location and mobility pattern (the prediction algorithm will be described later in Section III-D.

[2]It should be noted that an anchor node can communicate with surface buoys directly and can measure its current location at its will.

The Design Philosophy The guideline behind SLMP is to achieve a good balance between localization error and communication cost. With mobility prediction mechanisms, anchor nodes need not broadcast localization message in every localization period and every localized node can predict its location in the future localization period. Thus communication cost and energy consumption can be saved. However, companied with the benefits of the mobility prediction mechanisms, another problem arises: how to control the localization error throughout the network. With mobility prediction, ordinary nodes can no longer judge the accuracy of their predicted locations. Fortunately, anchor nodes can measure their locations at any time, as makes the judgement of their predictions possible. Because anchor nodes are assumed to be dispersed randomly in the network, their measurements can be viewed as samples of the prediction error of the whole network. In SLMP, we use anchor nodes to measure and control the prediction error of the whole network. Simulation results in Section IV will show that SLMP can achieve a good balance between communication cost and localization error.

C. Anchor Node Mobility Prediction

Anchor nodes can easily measure their locations directly at any localization period since they can directly communicate with the surface buoys, and they could also predict their future mobility patterns based on their past measurements.

From Fig. 1, we can observe that the moving speed of the underwater object in seashore environments changes continuously and demonstrates certain semi-periodical properties. We also notice that its waveform is very similar to that of the voice signal. It is well known that the voice signal can be closely approximated by all-pole model and thus can be well predicted by linear prediction algorithm [14], [28], [29]. Inspired by this, we adopt the linear prediction method [30] in our system for anchor node mobility prediction. It works as follows.

First, as shown in Fig. 3, we divide time into multiple *prediction windows* with length set to T_w. One window is one prediction unit. We assume that the mobility behaviors of the nodes will not change during adjacent prediction windows. This is a reasonable assumption considering the continuity and the semi-periodicity of the underlying mobility behavior. Thus we can use measured values in the previous window to predict the mobility behavior of the next window. Window length T_w should be integer times of the localization period T_1.We denote this as

$$T_w = k \times T_1. \tag{1}$$

The length of prediction window T_w is important to the performance of our prediction algorithm. If T_w is too small, the known data from the previous prediction window are not enough to capture the characteristics of the underlying mobility model and thus the prediction results are not so accurate. But if T_w is too large, it will increase the computational complexity. Further, a large T_w will not be always helpful to improve the prediction accuracy. This is because the underlying mobility model changes over one window period if T_w is too large, as will degrade the prediction performance [28]–[30]. In practice, we can choose T_w according to the application requirements and system environments.

To further improve the accuracy of our prediction, neighboring windows should be overlapped to some extent to avoid the edge effects of the windowing operation [14], [28]. In Fig. 3, T_l is used to denote the length of this overlapped window area. We set $\varsigma = \frac{T_l}{T_w}$ and it is self-evident that $0 < \varsigma < 1$. The typical value for ς is from 0.1 to 0.5 [14], [28]. In our simulations, we choose ς to be 0.2 and we find out that this value is good enough to filter out the edge effects in our applications.

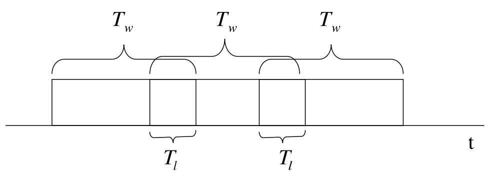


Fig. 3. Overlapped prediction windows

For every node, we use speed vector $V = [v(1), v(2), \ldots v(i), \ldots, v(k)]$ to represent its mobility behavior in every prediction window, where $v(i)$ denotes the average speed in the ith localization period. In order to predict $v(i)$, we use a linear prediction algorithm as follows:

$$v(i) = \sum\nolimits_{m=0}^{l} a_m v(i-m), \tag{2}$$

where l is the length of prediction steps, and a_m is the linear prediction model coefficient between $v(i)$ and $v(i-m)$. a_m's can be estimated by using measured location data from previous windows. In our work, we use the well known Durbin algorithm [28]–[30] to estimate them. We choose Durbin algorithm mainly because of its simplicity and low computation complexity $O(l^2)$, where l is the length of prediction steps.

In localization period i, an anchor node can measure its actual location $Loc_a(i)$ and calculate its estimated location $Loc_e(i)$ as follows.

$$Loc_e(i) = Loc_a(j) + \sum_{m=j}^{i} t_1 \times v(m), \tag{3}$$

where $Loc_a(j)$ is the measured location in localization period j. As shown in Fig. 4, j is the last localization period when this node runs its prediction algorithm and broadcasts a localization message. j is also the starting point of the current prediction window.

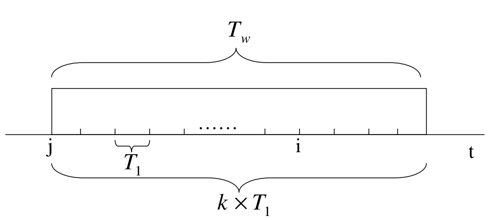


Fig. 4. One window structure

If the error between the estimated location $Loc_e(i)$ and the measured location $Loc_a(i)$ is smaller than the stipulated threshold s_t, this indicates that the predicted speed vector V works well and there is no need for further action.

Otherwise, the current active speed vector V is not so accurate. Then, this anchor node needs to rerun the mobility prediction algorithm to update its speed vector V, and broadcast its current location and predicted speed vector V in one localization message to inform the network.

The structure of an localization message is shown in Fig. 5, with each field explained as follows.

- *Node id*: The unique identification number of the message sender.
- *Time stamp*: The time when this message is sent. It is needed for some range measurement methods such as TOA (Time of Arrival) and TDOA (Time Difference of Arrival).
- *Location*: The current location of the message sender.
- *Speed vector*: The predicted speed vector for the next window.
- *Confidence value*: The confidence value of the message sender. It is used to denote the location estimation accuracy. For original anchor nodes, they are set to be 1. We will discuss how to calculate this value for ordinary nodes later in Section III-D.

Note localization messages could be sent by both anchor nodes and localized ordinary nodes. The later case will be discussed in the next section.

Node-id	Time-stamp	Location	Speed vector	Confidence value

Fig. 5. Localization message structure

D. Ordinary Node Mobility Prediction & Localization

As for ordinary nodes, because of their limited memory and computing capacity, they can not perform complicated temporal prediction algorithms. In SLMP, we take full advantages of the spatial correlation that underwater objects possess to facilitate mobility prediction.

1) Mobility Prediction: Assume we want to get the velocity $[v_x(j), v_y(j)]$ of node j, where $v_x(j)/v_y(j)$ denotes the current speed of node j in the x/y axis. If we get to know the velocities of its neighbor nodes, then we can estimate the velocity of j as follows [3], [26].

$$\begin{cases} v_x(j) = \sum_{i=1}^{m} \zeta_{ij} v_x(i) \\ v_y(j) = \sum_{i=1}^{m} \zeta_{ij} v_y(i) \end{cases}, \tag{4}$$

where m is number of neighbors. The interpolation coefficient ζ_{ij} is calculated as

$$\zeta_{ij} = \frac{\frac{1}{r_{ij}}}{\sum_{i=1}^{m} \frac{1}{r_{ij}}}, \tag{5}$$

where r_{ij} denotes the Euclidean distance between node i and node j.

2) Localization Process: For ordinary nodes, we adopt a recursive localization method. We smoothly incorporate mobility prediction into the localization process.

First, we define *reference nodes* as nodes with known locations and confidence values higher than the confidence threshold. In the *initialization phase*, all anchor nodes label themselves as reference nodes and set their confidence values to 1. All the ordinary nodes are non-localized nodes. With the advance of the localization process, more and more ordinary nodes are localized and become reference nodes. Each ordinary node maintains a reference list to record all its known reference nodes. Each reference node entry in the list includes the following information: the reference node ID; the arrival time of the latest localization message from this reference node; the reference node's location and speed vector, the distance to this reference node; and the confidence value of the reference node.

In each localization period, every node updates its known reference list. This updating process include the following operations.

- Check the arrival time of the latest localization message from every known reference node. If the distance between the current time and the last arrival time is larger than k localization periods, this reference node is too old to be useful, and should be deleted from the list.
- For every known reference node, update its location as follows:

$$Loc(i) = Loc(i-1) + T_1 \times v(i), \tag{6}$$

  where $Loc(i)$ denotes the estimated location of this reference node in localization period i and $v(i)$ is the estimated speed of this reference node in localization period i.
- Update the confidence value of every known reference node. To well reflect the network conditions, the confidence value of a known reference node should decrease with time if no new localization message has been received from that reference node. In our simulations, we update the conference value for a reference node as follows:

$$\eta = \frac{k - \frac{(t_c - t_{rev})}{T_1}}{k} \times \eta_0, \tag{7}$$

  where t_c denotes the current time; t_{rev} denotes the arrival time of the latest localization message from this reference node; and η_0 is the old confidence value obtained from the last localization message.

In a localization period, for any non-localized ordinary node, if it does not receive any localization message, it will update its current location estimation by using its previous location estimation and its predicted speed vector. If it receives a localization message from one reference node, it will update its reference list and perform new location estimations. This process includes the following operations.

- If this reference node is unknown before, a new entry will be inserted into the known reference list. Otherwise, the corresponding entry should be updated.
- If the number of its known reference nodes is equal to or larger than 4, then select 4 reference nodes with largest confidence values to do location estimation. It also performs mobility prediction according to (4), labels

itself as localized and calculates its own confidence value η. The confidence value of this node is not only related to itself, but also related to the confidence value of the chosen reference nodes. In our simulations, we calculate the confidence value as follows:

$$\eta = \frac{\sum_{i=1}^{4} \eta_i}{4}\left(1 - \frac{\delta}{\sum_{i=1}^{4}\left[(u-x_i)^2 + (v-y_i)^2 + (w-z_i)^2\right]}\right), \tag{8}$$

where (u, v, w) are the estimated coordinates of this node; (x_i, y_i, z_i) is the reference node i's location; η_i is the confidence value of reference node i; and δ is defined as follows,

$$\delta = \sum_{i} \left|(u-x_i)^2 + (v-y_i)^2 + (w-z_i)^2 - l_i^2\right|, \tag{9}$$

where l_i is the measured distance between the this node and reference node i.

- If the calculated confidence value η is larger than or equals to the confidence threshold λ, then this node labels itself as a new reference node and send out a localization message. Otherwise, this node is treated as a localized ordinary node and keep silent.

The localization process of an ordinary node is illustrated in Fig. 6.

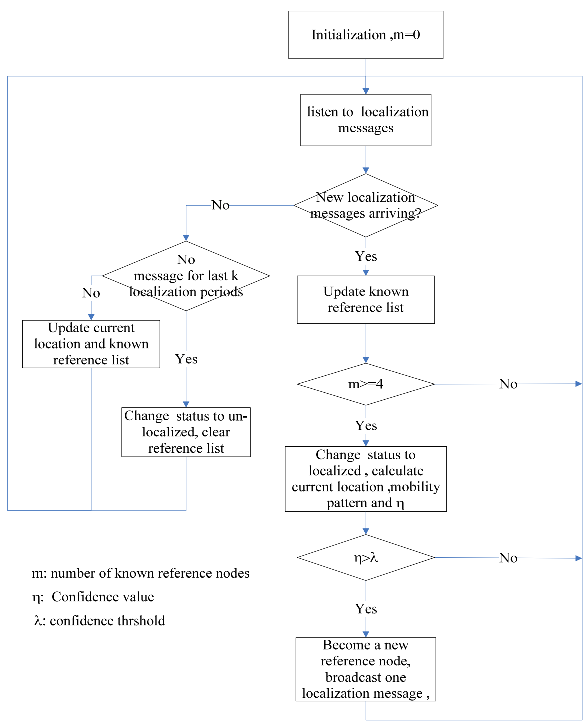


Fig. 6. Ordinary nodes localization process

IV. Simulation Results

In this section, we evaluate the performance of SLMP using simulations.

A. Simulation Settings

In our simulations, 500 sensor nodes are randomly distributed in a $100m \times 100m \times 100m$ region. We define node density as the expected number of nodes in a node's neighborhood. Hence node density is equivalent to node degree. We control the node density by changing the communication range of every node while keeping the area of deployment the same. Range (i.e., distance) measurements between nodes are assumed to follow normal distributions with real distances as mean values and standard deviations to be two percent of real distances. This is a reasonable assumptions and can be easily satisfied by existing underwater distance measurement technologies [4], [19]. 5%, 10% and 20% anchor nodes are considered in our simulations. Besides SLMP, we also simulate one localization scheme without mobility prediction for comparison. The localization process of this scheme is almost the same as that of SLMP, except that it does not involves mobility prediction.

As to the node mobility pattern, we consider the kinematic model in [5]. The current field is assumed to be a superposition of a tidal and a residual current field. The tidal field is assumed to be a spatially uniform oscillating current in one direction and the residual current field is assumed to be an infinite sequence of clockwise and anticlockwise rotating eddies. The dimensionless velocity field in the kinematical model can be approximated as

$$\begin{cases} V_x = k_1 \lambda v \sin(k_2 x) \cos(k_3 y) + k_1 \lambda \cos(2 k_1 t) + k_4 \\ V_y = -\lambda v \cos(k_2 x) \sin(k_3 y) + k_5 \end{cases} \tag{10}$$

where V_x is the speed in the X axis and V_y is the speed in the Y axis. $k_1, k_2, k_3, \lambda, v$ are variable which are closely related to environment factors such as tides and bathymetry. These parameters will change with different environments. k_4, k_5 are random variables. In our simulations, we assume k_1, k_2 to be random variables which are subject to normal distribution with π as mean values and the standard derivations to be 0.1π. k_3 is subject to normal distribution with 2π as mean value and the standard derivation to be 0.2π. λ is subject to normal distribution with 3 as the mean value and 0.3 as the standard derivation. v is subject to normal distribution with 1 as the mean value and 0.1 as the standard derivation. k_4, k_5 are random variables which are subject to normal distribution with 1 as mean values and 0.1 as standard derivations.

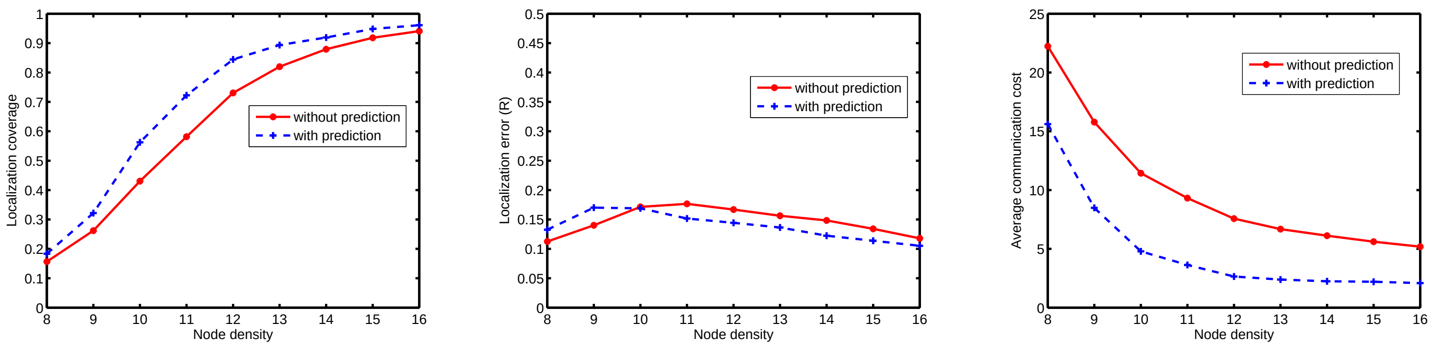
Every simulation lasts for 1100 seconds. The first 100 seconds of every simulation are used for anchor nodes' training. During this training process, anchor nodes measure their locations every T_1 seconds and get their initial mobility patterns. After this training process, anchor nodes will send out their first localization messages which contain their initial locations and mobility patterns, and then start the whole localization process.

Unless specified otherwise, we have the following parameters. Localization period T_1 is set to be 1s. The prediction error threshold of anchors s_t is set to be $0.05R$. The length of prediction steps l is set to be 15. The prediction window size T_w is set to be 60s and $\varsigma = 0.2$. The confidence threshold of ordinary node λ is set to be 0.98.

Three performance metrics are considered in our simulations: localization coverage, localization error and average communication cost. *Localization coverage* is defined as the ratio of the localizable nodes to the total nodes. *Localization error* is the average distance between the estimated positions and the real positions of all nodes. As in [9], [15], [25], for our simulations, we normalize this absolute localization error to the node's communication range R. *Average communication cost* is defined as the overall messages exchanged in the network divided by the number of localized sensor nodes.

B. Results and Analysis

1) Performance with Changing Node Density: In this set of simulations, we set the anchor percentage to be 10% and change the average node density from 8 to 16 by changing the node range R. The results for localization coverage, localization error, and communication cost are plotted in Fig. 7(a), Fig. 7(b), and Fig. 7(c) respectively.



(a) Localization coverage (b) Localization error (c) Communication cost

Fig. 7. Performance with changing node density

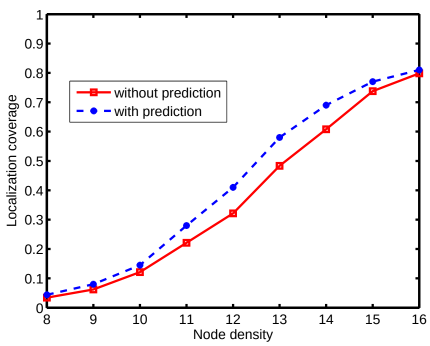
Without surprise, Fig. 7(a) shows us that the localization coverage increases monotonically with node density. Further, the localization coverage of our scheme with prediction (i.e., SLMP) is higher than that of the scheme without prediction. This is because with mobility prediction, some nodes which can not get located in current network conditions can be localized by the prediction scheme. Thus, compared with the scheme without mobility prediction which needs to redo one complete localization process in every localization period independently, our scheme can even achieve higher localization coverage.

Fig. 7(b) shows us that with the increase of node density, the average localization error of our scheme will first increase a little and then it will decrease monotonically. We think that the increase in the lower node density region is due to the rapid increase of the corresponding localization coverage. While when the node density reaches a certain point, most localizable nodes get located. With the further increase of node density, un-localized nodes will get to know more reference nodes and have more choices to calculate their locations. Thus, the localization error will reduce.

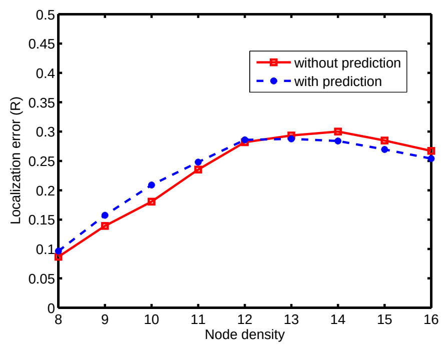
Compared with the scheme without mobility prediction, the localization error of our scheme is slightly higher in the low node density region. While when the node density is relatively high (in this simulation setting, more than 10), our scheme can achieve lower localization errors. This can be explained as follows. In the low node density region, the localization coverage is low and most localizable nodes are located with high accuracy. Mobility prediction will introduce prediction errors to the network, thus will increase the localization error a little. But with the increase of node density, more and more nodes get located. Because of the mobility prediction, our scheme will introduce more reference nodes into the network, which will contribute to the reduction of the localization error. With the increase of node density, this reduction finally outweighs the effects of prediction errors. Thus, our scheme achieve smaller localization error than the scheme without mobility prediction

Fig. 7(c) clearly shows us that with the increase of node density, the average communication cost decreases rapid first. This is because with the increase of node density, the localization coverage increases drastically. Thus the average communication cost decreases rapidly. Further, we can observe that when the node density increases to some point, for example, in Fig.7(c), when it increases to 12, the average communication cost will stabilize to some small values. Fig. 7(c) also shows us that our mobility prediction scheme can greatly reduce the communication cost. This is reasonable since in our scheme, anchor nodes need not send localization message in every localization period and thus limit the frequency of localization message flooding. Correspondingly, the overall communication cost is reduced. This is quite meaningful for underwater sensor networks with limited bandwidth and energy.

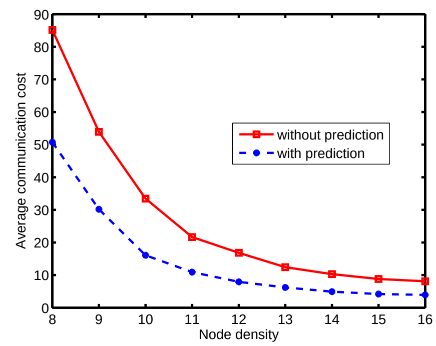
2) Performance with Changing Node Density when Anchor Node $=5\%$*:* In this set of simulations, we set anchor percentage to be 5%. We change the average node density from 8 to 16 by changing the node range R.



(a) Localization coverage



(b) Localization error



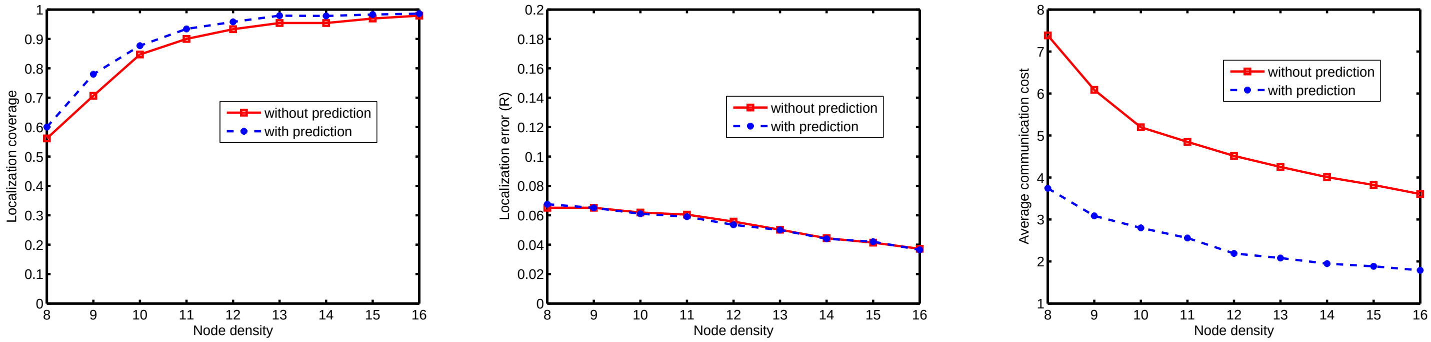
(c) communication cost

Fig. 8. Performance with changing node density when anchor node 5%

Fig. 8 shows almost the same trend as the case with 10% situation. But the cross point of our scheme with the scheme without mobility prediction changes. It is a little higher than that of 10% situation. Which means for our scheme to achieve the same or smaller localization error, a higher node degree is needed. This is reasonable since less anchor nodes mean larger average prediction errors in the network. Correspondingly, a larger node density is needed in order to counteract the prediction error.

3) Performance with Changing Node Density when Anchor Node $=20\%$*:* In this set of simulations, we set anchor percentage to be 20%. We change the average node density from 8 to 16 by changing the node range R.

Fig. 9 is a little different from the previous two cases. Here, the localization error decreases monotonically with node density. The cross point of our scheme with the scheme without mobility prediction become smaller. This is



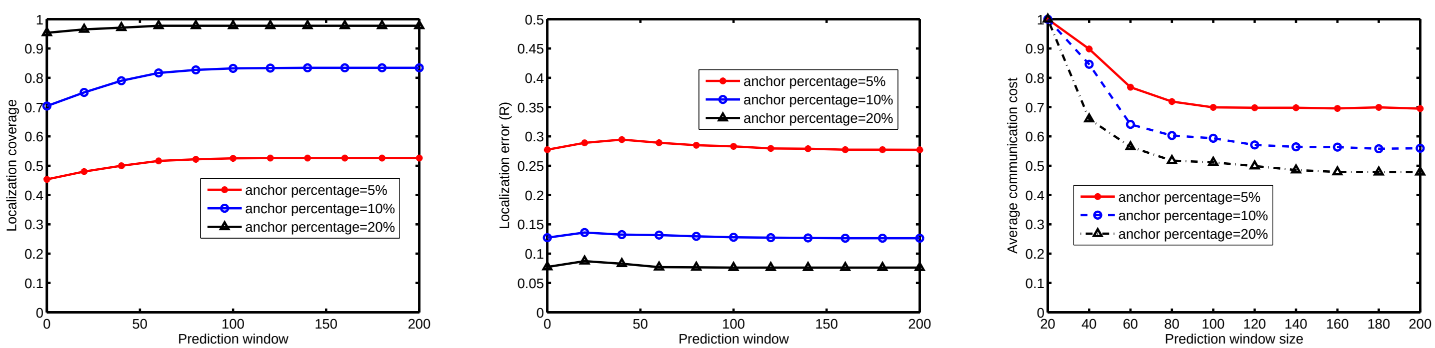
(a) Localization coverage (b) Localization error (c) communication cost

Fig. 9. Performance with changing node density when anchor node 20%

also reasonable since more anchor nodes mean smaller average prediction errors in the network. Correspondingly, a lower node density is needed in order to counteract the prediction error.

4) Performance with Changing Prediction Window T_w: In this set of simulations, node range R is fixed to be 20m and thus the average node degree is about 12. We change the prediction window T_w from 20s to 200s.

Here, in order to show the advantages of mobility prediction, we normalize the average communication cost to that of localization scheme without mobility prediction.



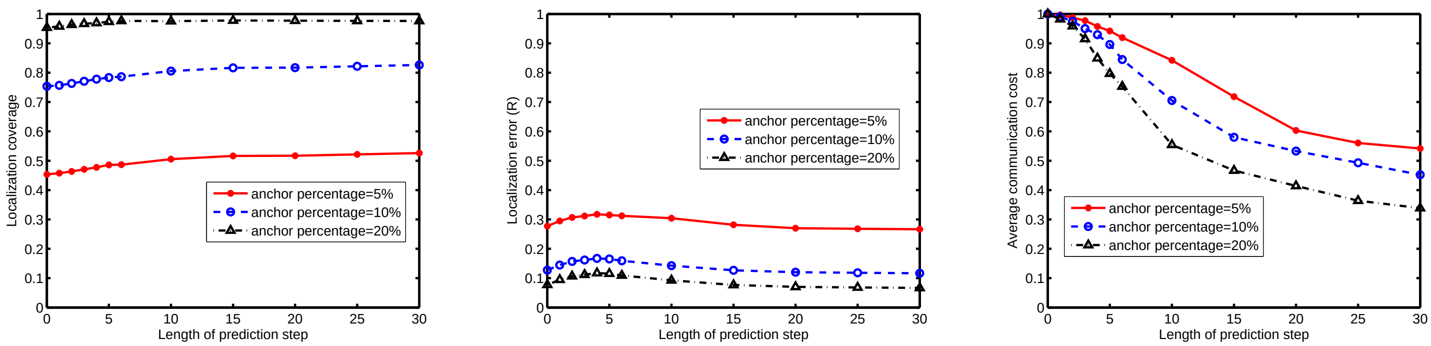
(a) Localization coverage (b) Localization error (c) Normalized communication cost

Fig. 10. Performance with changing prediction window

Fig. 10 shows that with the increase of prediction window, the localization coverage increases slightly. At the same time, the localization error decreases a little. This is because the larger the window size, the more accurate the prediction results for anchor nodes, as will cause more ordinary nodes to become reference nodes and thus the localization coverage of the whole network will increase and the localization error will decrease slightly.

As shown in Fig. 10, the normalized average communication cost decreases monotonically with the prediction window. This is reasonable since more accurate prediction results mean less traffic that anchor nodes send out. But when the prediction window reaches some large value, for example, 100s for the case of 10% percent of anchor nodes, increasing prediction window will not contribute much to the reduction of communication cost. This is because in such situations, the communication cost of the localization process is dominated by the localization messages among ordinary nodes as well as the beacon messages which are used to measure distance among nodes periodically.

5) Performance with Changing Length of Prediction Steps l: In this set of simulations, node range R is fixed to be 20m and thus the average node degree is about 12. We change the length of prediction steps l from 0 (without prediction) to 30. Here, to show the advantages of mobility prediction, we normalize average communication cost to that of localization scheme without mobility prediction.



(a) Localization coverage (b) Localization error (c) communication cost

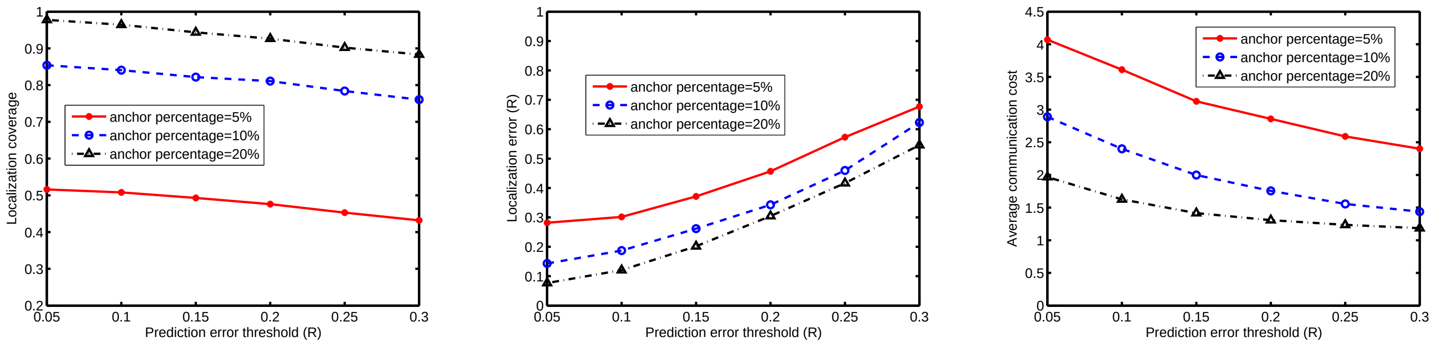
Fig. 11. Performance with changing length of prediction steps

From Fig. 11(a), we can clearly see that with the increase of prediction step, the localization coverage will increase slowly, we believe this is mainly because with the increase of prediction step, the prediction accuracy will increase correspondingly. This will make more nodes work as reference nodes and thus will increase the localization coverage to some extent.

Fig. 11(b)shows us the average localization error with prediction step. When prediction step l is small, the localization error will increase slowly. This is because when l is small, although anchor nodes have mobility prediction capacity, the prediction accuracy is limited, which will increase the average location error in the network. But with the increase of l, the average localization error will decrease because of more accurate mobility prediction of anchor nodes. We can see that when $l > 15$, the average localization error decreases below the the value of localization scheme without mobility prediction(when $l = 0$). This is because when l is relatively large, the mobility prediction is accurate enough and more reference nodes will be introduced to the network in one localization period. Subsequently, every node will get to know more reference nodes which will correspondingly decrease the localization error.

From Fig.11(c), we can see that with the increase of prediction steps, the average communication cost will decrease drastically, this is reasonable since the more the prediction step, the more accurate the prediction results of anchor nodes, and thus the less the localization message that anchor nodes send out. From Fig.11(c), we can see that when $l \geq 15$, its relative communication cost decreases as low as 0.5, which means that our simple mobility prediction will save more than $1/2$ traffic load compared with localization scheme with no mobility prediction.

6) Impact of Prediction Error Threshold s_t: In this set of simulations, the node range R is set to be $20m$. We change anchor node prediction error threshold s_t from $0.05R$ to $0.3R$. The results are plotted in Fig. 12.

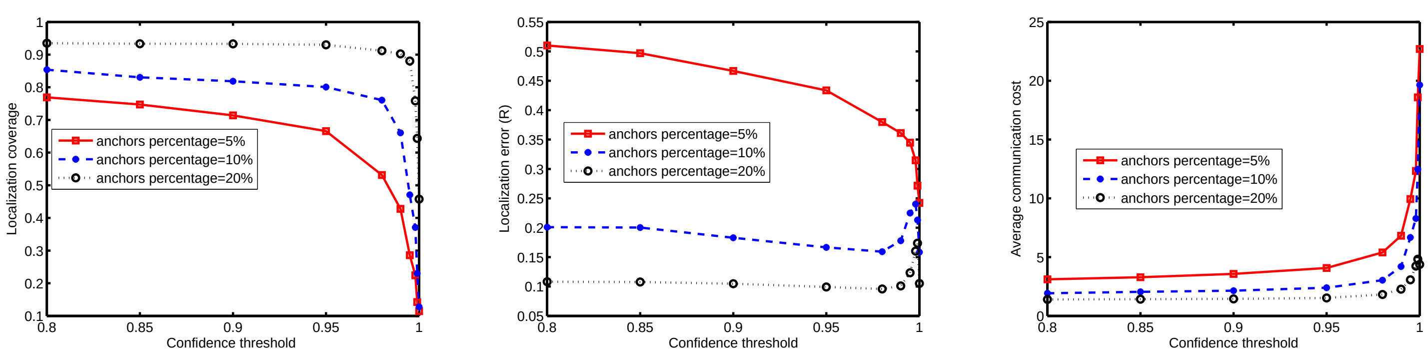


(a) Localization coverage (b) Localization error (c) Communication cost

Fig. 12. Performance with changing prediction error threshold

From Fig. 12, we can see that with the increase of prediction error threshold, the localization coverage decreases slowly, and the localization error increases correspondingly. This can be explained as follows. With the increase of error threshold, the average prediction error of anchor nodes will increase, as leads to less nodes get high accurate localization results. Correspondingly, this will reduce the available reference nodes in the network (which should be nodes with high localization accuracy), thus resulting in the decrease of the final localization coverage. On the other hand, however, as we can see from Fig. 12(c), the average communication cost decreases dramatically with the increase of prediction error threshold. This is reasonable since a large prediction error threshold will definitely reduce the localization message that the anchor nodes send out, as will directly contribute to low average communication cost.

7) Impact of Confidence Threshold λ: In this set of simulations, the node range R is set to be $20m$. We change the confidence threshold of ordinary nodes λ from 0.8 to1.



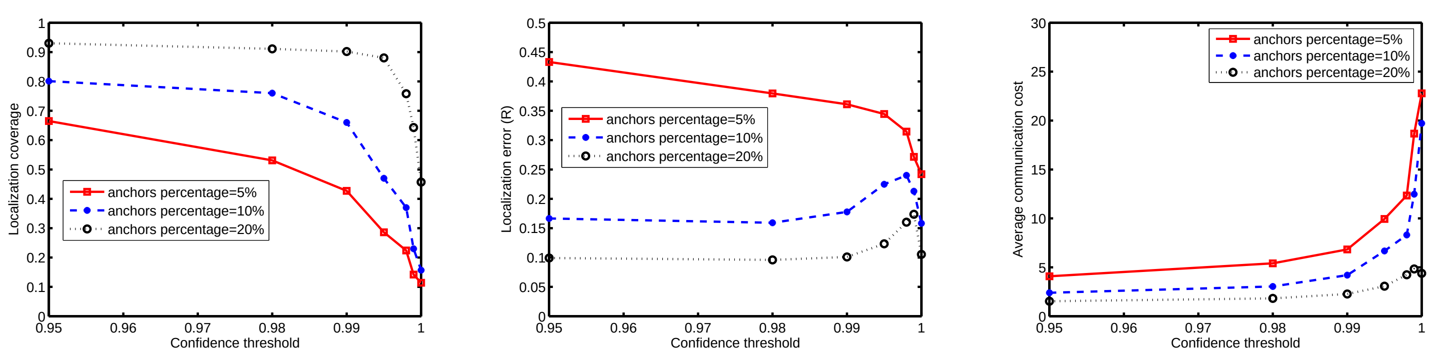
(a) Localization coverage (b) Localization error (c) communication cost

Fig. 13. Performance with changing confidence threshold λ

Fig. 13(a) shows that with the increase of confidence threshold, the average localization coverage will decrease correspondingly. There exists some critical values. Below these values, the localization coverage will not decrease much, but above these values, the localization coverage will drop abruptly. For example, as show in Fig. 13(a), when the anchor percentage is 10%, this critical value is 0.98.

Fig. 13(b) shows the relationship between the localization error and the confidence threshold. When the anchor percentage is 5%, the localization error will decrease monotonously with the confidence threshold. But when the anchor percentage is 10% and 20%, when the confidence threshold is small, the localization error will decrease slowly with it. But after some critical points, the localization error will increase to some extent and then decrease again. This can be explained as follows, with the increase of the confidence threshold, nodes with higher localization accuracy will become reference nodes during the following localization process. This will reduce the localization error. But when the confidence threshold is too high, the available reference nodes will decrease significantly, which will correspondingly increase the localization errors. Thus, after some critical values, localization errors will increase. When we continue to increase the confidence value, the available reference nodes will be further reduced and the corresponding localization coverage will decrease dramatically, and only nodes with high accuracy can be localized. Therefore, the average localization error will decrease. While when the anchor percentage is small, 5% for example, the localization coverage is small and changes relatively slow with the confidence threshold. And thus the average localization error will decrease monotonically with the increase of the confidence threshold.

Fig. 13(c) shows the average communication cost with the confidence threshold. We can observe that when the confidence threshold is small, the average communication cost increases slowly with the that of the confidence threshold. But after some critical values, the average communication cost will increase much faster. This is reasonable since the higher the confidence value, the lower the localization coverage.



(a) Localization coverage

(b) Localization error

(c) communication cost

Fig. 14. Performance with changing confidence threshold λ in fine granularity

Fig. 14 shows the details in the high confidence area (from 0.95 to 1).

8) Localization Error with α: In this set of simulation, node range R is fixed to be 20m and thus the average node degree is about 12. The prediction error threshold of anchors s_t is set to be $0.05R$, The prediction length

$l = 15$. We set $\varsigma = 0.2$ and change the prediction window size t_2 from 20s to 200s. The confidence threshold of ordinary node λ is set to be 0.98. Here, to show the advantages of mobility prediction, we normalize average communication cost to that of localization scheme without mobility prediction.

In this set of simulations, we want to further reduce the localization error. And we estimate the location Loc according to the following equation

$$Loc = \alpha \times Loc_1 + (1 - \alpha) \times Loc_2, \tag{11}$$

where Loc_1 is the current prediction location and Loc_2 is the current estimation location.

Fig. 15 shows that this simple linear formula does not help much: the average localization error only slightly increases with the decrease of α.

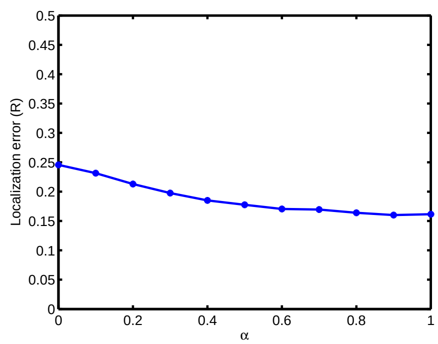


Fig. 15. Localization error with α

V. Conclusions

In this paper, we have presented SLMP, a new localization scheme with mobility prediction, for large scale underwater sensor networks. In SLMP, anchor nodes conduct linear prediction by taking advantages of the inherent temporal correlation of underwater object mobility pattern. While each ordinary sensor node predicts its location by utilizing the spatial correlation of underwater object mobility pattern, weighted-averaging its received mobilities from other nodes. Our simulation results show that SLMP can greatly reduce the communication cost while maintaining a relatively high localization coverage and localization accuracy. By simulations, we also evaluate the impact of various design parameters, such as prediction window, prediction steps, prediction error threshold and confidence threshold on the localization performance.

Future Work We plan to extend our work in two directions: 1) Explore other underwater mobility patterns and examine the applicability of our design; 2) Investigate other prediction algorithms and examine how the localization performance will be affected.

References

[1] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks (Elsevier)*, 3(3):257–279, March 2005.

[2] T. C. Austin, R. P. Stokey, and K. M. Sharp. PARADIGM: a buoy-based system for auv navigation and tracking. In *IEEE Proceedings of Oceans*, 2000.

[3] A. C. .Bagtzoglou and N. A. Chaotic behavior and pollution dispersion characteristics in engineered tidal embayments: A numerical investigation. *Journal of the American Water Resources Association*, pages 207–219, 2007.

[4] C. Bechaz and H. Thomas. GIB system: The underwater gps solution. In *Proceedings of ECUA*, May 2001.

[5] S. P. Beerens, H. Ridderinkhof, and J. Zimmerman. An analytical study of chaotic stirring in tidal areas. *Chaos, Solitons and Fractals*, 1994.

[6] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of IPSN*, pages 46–54, 2004.

[7] N. Bodhi, H. Balakrishnan, E. Demaine, and S. Teller. Mobile-assisted localization in wireless sensor networks. In *Proceedings of Infocom*, pages 172–183, March 2005.

[8] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, pages 28–34, Oct. 2000.

[9] K. K. Chintalapudi, A. Dhariwal, R. Govindan, and G. Sukhatme. Ad-hoc localization using range and sectoring. In *Proceedings of Infocom*, pages 2662–2672, March 2004.

[10] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou. Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications. *IEEE Network, Special Issue on Wireless Sensor Networking*, 20(3):12–18, 2006.

[11] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. Anderson, and P. N. Bellhumenur. Rigidity, computation and randomization in network localization. In *Proceedings of Infocom*, pages 2673–2684, April 2004.

[12] J. E. Garcia. Ad hoc positioning for sensors in underwater acoustic networks. In *IEEE Proceedings of Oceans*, 2004.

[13] D. K. Goldenberg, A. Krishnamurthy, W. C. Maness, Y. R. Yang, A. Young, A. S. Morse, A. Savvides, and B. D. O. Anderson. Network localization in partially localizable networks. In *Proceedings of Infocom*, pages 313–326, March 2005.

[14] S. Goronzy. *Robust Adaptation to Non-Native Accents in Automatic Speech Recognition*. Springer Berlin / Heidelberg, 2002.

[15] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of Mobicom*, pages 81–95, Sep 2003.

[16] J. Heidemann, Y. Li, A. Syed, J. Wills, and W. Ye. Research challenges and applications for underwater sensor networking. In *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2006.

[17] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of Mobicom*, pages 45–57, Sep 2004.

[18] X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In *Proceedings of Infocom*, pages 2652–2661, March 2004.

[19] N. H. Kussat, C. D. Chadwell, and R. Zimmerman. Absolute positioning of an autonomous underwater vehilce using GPS and acoustic measurements. *IEEE Journal of Oceanic Engineering*, 30(1):153–164, Jan 2005.

[20] H. Lim and J. C. Hou. Localization for anisotropic sensor networks. In *Proceedings of Infocom*, pages 138–149, March 2005.

[21] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of SenSys*, pages 50–61, Nov 2004.

[22] R. Nagpal, S. H, and J. Bachrach. Organizing a global coordinate system from local inforamtion on an ad hoc sensor network. In *Proceeding of IPSN*, 2003.

[23] D. Nichulescu and B. Nath. DV based positioning in ad hoc networks. *Springer, Telecommunication Systems*, pages 267–280, Oct. 2003.

[24] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AOA. In *Proceedings of Infocom*, pages 1734–1743, March 2003.

[25] D. Niculescu and B. Nathi. Ad hoc positioning system (APS). In *Proceedings of IEEE Globecom*, November 2001.

[26] A. Novikov and A. C. Bagtzoglou. Hydrodynamic model of the lower hudson river estuarine system and its application for water quality management. *Water Resource Management*, pages 257–276, 2006.

[27] J. Partan, J. Kurose, and B. N. Levine. A Survey of Practical Issues in Underwater Networks. In *Proc. of ACM International Workshop on UnderWater Networks (WUWNet)*, pages 17–24, September 2006.

[28] L. R. Rabiner and R. W. Schafer. *Digital processing of Speech Signals*. Prentice-Hall,Inc, Engelwood Cliffs, New Jersey, 1978.

[29] C. Rowden. *Speech processing*. McGraw-Hill, 1992.

[30] M. R. Schroeder. *Linear Prediction Theory: A Mathematical Basis for Adaptive Systems*. Springer-verlag, 1989.

[31] Y. Shang and W. Ruml. Improved MDS-Based Localization. In *Proceedings of Infocom*, pages 2640–2651, March 2004.

[32] S. Tilak, V. Kolar, N. B. Abu-Ghazaleh, and K.-D. Kang. Dynmaic localization protocols for mobile sensor networks. In *IEEE International Workshop on Strategies for Energy Efficiency in Ad-hoc and Sensor Networks*, April 2005.

[33] C. Wang and L. Xiao. Locating sensors in Concave environments. In *Proceedings of Infocom*, March 2006.

[34] H. Wu, C. Wang, and N.-F. Tzeng. Novel self-configurable positioning technique for multihop wireless networks. *IEEE/ACM Transaction on Networking*, pages 609–621, 2005.

[35] P. Xie, L. Lao, and J.-H. Cui. VBF: vector-based forwarding protocol for underwater sensor networks. In *To appear in Proceedings of IFIP Networking*, May 2006.

[36] Y. Zhang and L. Cheng. A distributed protocol for multi-hop underwater robot positioning. In *IEEE Proceedings of International Conference on Robotics and Biomimetics*, 2004.