Abstract—Time synchronization and localization are basic services in a sensor network system. Although they are usually tackled independently, they are closely related. In this work, we explore the joint design of synchronization and localization, in which the stratification effect of underwater medium is considered, so that the bias in the range estimates caused by assuming sound waves travel in straight lines in water environments is compensated. By combining time synchronization and localization, the accuracy of both are improved jointly. Additionally, an advanced tracking algorithm IMM (interactive multiple model) is adopted to improve the accuracy of localization in the mobile case. Furthermore, by combining both services, the number of required exchanged messages is significantly reduced, which saves on energy consumption. Simulation results show that both services are improved and benefit from this scheme.

Index Terms—UWSNs, Time synchronization, Localization.

I. INTRODUCTION

In recent years, Underwater Sensor Networks (UWSNs) has attracted significant attention [1], [2], [3]. Among the services UWSNs can provide, time synchronization and localization are very critical, because most UWSNs applications benefit from or require these two services. For instance, TDMA (Time Division Multiple Access), one of the commonly used medium access control (MAC) protocols, often requires precise synchronization among sensor nodes. Additionally, most geographic routing algorithms assume the availability of location information [4], [5].

Although localization and synchronization services are closely related, they are usually studied independently. This is mainly because localization is traditionally studied from the signal processing point of view in radio networks, and synchronization is mainly studied from protocol design point of view. However, especially in UWSNs, localization and synchronization are closely “bonded”. Since the ranging is estimated based on time of arrival (TOA) or time difference of arrivals (TDOA) in UWSNs, many localization algorithms rely on the time synchronization services. For example, in TOA, a popular localization algorithm, synchronization is a prerequisite. On the other hand, knowledge of location helps time synchronization because it can be used to estimate propagation delays. Furthermore, both localization and time synchronization require a sequence of message exchanges among the nodes. Based on these bonds relationships, we believe that localization and time synchronization could be solved jointly, with two major benefits. First, a joint strategy would save energy, since localization and synchronization can use only one set of message exchanges instead of two. This is important for energy constrained network systems like UWSNs. Second, a joint solution can help to improve the accuracy of both services.

However, to the best of our knowledge, there is no work has explored the joint design of synchronization and localization in UWSNs. Additionally, in UWSNs, all current localization algorithms assume the straight line transmission of acoustic waves. In fact, due to the sound speed variation with depth in the water environment, called “stratification effect”, the real transmission path usually bends. This will severely affect the ranging estimation, and in turn affect localization accuracy.

In this paper, we propose a joint solution for localization and synchronization, called JSL for UWSN, JSL is a four phases scheme in which, time synchronization and localization are performed at different phases. During iterations, the output of synchronization is fed back as the input of localization, and the output of localization is fed back as the input of synchronization. In this way, synchronization and localization are interleaved and can benefit each other by improving the accuracy of both. During the localization phase, unlike other algorithms that assume sound waves travel in straight lines in the water environment, JSL compensates the stratification effect when performing the underwater acoustic ranging, so that the propagation delay estimation will be significantly improved. Furthermore, to account for the mobility characteristic in UWSNs, a tracking method - called interactive multiple model (IMM), is used to predict sensor node mobility to improve the accuracy of localization.

The rest of this paper is organized as follows. We first review the design challenges in Section II and some background knowledge in Section III. Then we describe JSL in Section IV. Simulation results are presented in Section V and related work is summarized in Section VI. Finally, we offer our conclusions and future work in Section VII.

II. DESIGN CHALLENGES

In order to achieve effective time synchronization and localization in UWSNs, following four critical challenges have to be addressed.

1) Stratification Effects: Underwater acoustic localization usually relies on TOA measurements, which are converted into range estimates. However, the water medium is inhomogeneous
and the sound speed varies depending on several parameters, such as temperature, pressure and salinity. As a result, sound waves do not necessarily travel in straight lines. Ignoring this stratification effect could lead to considerable bias in the range estimates.

2) Long Propagation Delays: Any software based time synchronization approaches using message exchanges have to face several uncertainties which could affect accuracy. Those uncertainties include sending time, accessing time, transmission time, propagation time, reception time, interrupt handling, encoding time, decoding time and byte alignment time [6], [7]. In UWSNs, among these uncertainties, the propagation time is dominant due to the low propagation speed of acoustic signals. Such long propagation latencies heavily affect the accuracy of time synchronization algorithms which assume instant synchronization message reception [8], [9]. Therefore, in order to achieve more accurate time synchronization in UWSNs, estimating and compensating the long propagation delay is a must do job.

3) Sensor Node Mobility: While terrestrial sensor networks are usually static, sensor nodes in an underwater environment often have passive mobility caused by water currents or proactive mobility coming with mobile platforms, which makes localization challenging. This is because in such a situation, it is difficult, if not impossible, to estimate the real time distance between two sensor nodes, which will in turn affect localization accuracy.

The mobility also complicates time synchronization by causing continuous changes of propagation delays. However, most of the existing time synchronization schemes use half of the round trip time to calculate one way propagation delay. Due to node mobility, the propagation delays on the way to and from nodes are not necessarily identical, especially when nodes move at a high speed. Therefore, to improve the time synchronization accuracy, the sensor node mobility should be considered.

4) Energy Constraints: Underwater sensor nodes are usually powered by batteries, for which it is not easy to replenish. Therefore, the life time of a sensor node is restricted by the limited power supply. For this reason, synchronization and localization overhead needs to be carefully controlled. Synchronization or localization protocols requiring frequent message exchanges are not suitable in UWSNs. Therefore, an effective synchronization and localization algorithm should be developed with a limited message exchange overhead.

III. RANGING WITH STRATIFICATION EFFECT COMPENSATION

As we mentioned, the range estimate is not linearly proportional to the TOA measurement in UWSNs, mainly due to the inhomogeneity of water medium in terms of the pressure, salinity, and temperature, etc [10], as shown in Fig. 1. In order to avoid the bias from assuming the straight line transmission, in JSL, we adopt an improved range estimation scheme with the compensation of the above stratification effect. Following gives the details of this scheme.

![Sound velocity profile (SVP) and sound wave propagation paths](image)

In this paper, we assume that the sound velocity profile (SVP) is only depth dependent, thus, due to the axis-symmetric assumption of the medium physical parameters, it is sufficient to consider that both the sender and receiver are on a two-dimensional plane as shown in Fig. 1. Let $v(z)$ denote the SVP as a function of depth $z$, $(y_s, z_s)$ and $(y_r, z_r)$ denote the position of the sender and the receiver, respectively. We further assume that i) the SVP $v(z)$ is available; ii) the position of the sender $(0, z_s)$ is known as a priori; iii) the depth of the receiver can be roughly estimated as $z_r$ through a depth-sensor. Once the estimation of $y_r$ is obtained, the range estimate can be calculated as

$$R = \sqrt{y_r^2 + (z_r - z_s)^2}.$$ (1)

Next, we focus on the estimation of $y_r$ based on the noisy TOA measurement $\hat{T}$, the SVP $v(z)$, the sender depth $y_s$ and the receiver depth $\hat{z}_r$. It is known that the travel time for an acoustic ray along a possible path $S$ is expressed as

$$T = \int_{s} \frac{1}{v(s)} ds.$$ (2)

Define $y = f(z)$ and $f'(z) = dy/dz$. We have

$$ds = \sqrt{dz^2 + dy^2} = \sqrt{1 + f'(z)^2} dz.$$ (3)

The travel time can be equivalently written as

$$T = \int_{z_s}^{z_r} \sqrt{1 + f'(z)^2}$$

$$\frac{1}{v(z)} dz.$$ (4)

According to Fermat’s principle, the true travel path is the one which minimizes the travel time, so that $f(z)$ can be obtained via

$$\frac{d}{dz} \frac{\partial}{\partial f'} \sqrt{1 + f'(z)^2} = 0$$ (5)

which leads to

$$\frac{f'(z)}{v(z)\sqrt{1 + f'(z)^2}} = C$$ (6)
where $C$ is an integration constant. The travel path $f(z)$ can be then obtained based on

$$f'(z) = \frac{Cv(z)}{\sqrt{1 - [Cv(z)]^2}}, \quad (7)$$

where the constant $C$ can be determined based on the ToA measurement $\hat{T}$ via

$$\hat{T} = \int_{z_s}^{z_r} \frac{1}{v(z)} \frac{1}{\sqrt{1 - [Cv(z)]^2}} dz. \quad (8)$$

The depth of the receiver is therefore

$$\hat{y}_r = f(z_r) = \int_{z_s}^{z_r} f'(z) dz = \int_{z_s}^{z_r} \frac{Cv(z)}{\sqrt{1 - [Cv(z)]^2}} dz. \quad (9)$$

It has been proved in [10] that estimates $\hat{y}_r$ and $\hat{R}$ are bias-free and reach the Cramer-Rao lower bound.

IV. DESCRIPTION OF JSL

For this work, the network architecture consists of three types of sensor nodes as shown in Fig. 2 and 3. Surface buoys equipped with GPS serve as “satellite nodes”. Anchor nodes are powerful sensor nodes which can communicate with a certain number of surface buoys directly to become synchronized and localized. Both surface buoys and anchor nodes act as reference nodes, and ordinary nodes are those sensor nodes or autonomous underwater vehicles (AUVs) which intend to be synchronized and localized.

A. Overview

The procedure of JSL consists of four major phases, Message Exchange, Synchronization, Localization and Iteration, which are shown in Fig. 4. In Phase I, an ordinary sensor node acquires reference time and location information from neighboring reference nodes. In Phase II, synchronization process is performed by ordinary nodes based on the information obtained in Phase I. Phase II consists of four steps. First, a node’s rough position is estimated by using the TDOA method. After the stratification effect is compensated, the propagation delay is calculated. Next, JSL performs linear regression to synchronize the ordinary sensor node, which is followed by the update of the corresponding propagation delays. During Phase III, JSL carries out localization process based on the estimated propagation delays in Phase II. Additionally, a tracking algorithm, IMM, is used to improve localization accuracy, such that the final location estimates are combined from the estimates based on Fermat’s principle and the predicted value from IMM. Phase IV is a iteration process, the position estimated in Phase III acts as input to Phase II to replace the rough position. Then Phase II and Phase III are repeated until locations, clock skew and offset become stable.

B. Phase I: Message exchange and rough position estimation

As shown in Fig. 5, an ordinary node initiates the procedure by broadcasting a Req message to its neighboring reference nodes. The ordinary node stamps the sending time $T_1$ obtained at the MAC layer, right before the message leaves. Upon receiving the Req message, each reference node marks its local time as $t_2$. Then, after a time interval $t_r$ (waiting for the hardware
sending-receiving transition and avoiding collisions), each of the reference node sends back a Res message containing its location information, \(t_2\), and sending time \(t_3\). When receiving the Res message, the ordinary node marks its receiving time \(T_4\). Fig. 5 shows the message exchange process among the ordinary node and reference nodes. Fig. 6 demonstrates the procedure between each pair.

After receiving Res messages from all the reference nodes, the ordinary node will estimate its rough position with TDOA. We call it “rough position” because of two reasons. First, at this moment, the ordinary node is not synchronized yet, therefore the obtained TDOA is not precise. Second, without considering stratification effect, straight line transmissions are still assumed in this phase.

In JSL, ordinary nodes are clocks aiming to become synchronized with the clocks of reference nodes. Hence we have:

\[ T = \theta * t + \beta, \]  

(10)

where \(T\) stands for the measured time of the ordinary node; \(t\) is the reference time; \(\theta\) and \(\beta\) are the relative clock skew and offset, respectively.

At this stage, in order to calculate ordinary node’s rough position, we first assume the ordinary node has been synchronized. This means we assign an initial clock skew “1”, and an initial clock offset “0”. We take reference node 1 as the base node and therefore comparing with reference node “1”, the time difference for reference node “n” is:

\[ \Delta \hat{t}_{n1} = \hat{t}_n - \hat{t}_1, \quad n = 2, \ldots, N. \]  

(11)

where \(\hat{t}_1\) denotes the estimate of the TOA for base reference node “1”, \(\hat{t}_n\) denotes the estimate of the TOA for base reference node “n”, and \(\Delta \hat{t}_{n1}\) stands for the estimate of the TDOA for base reference node “n”. Therefore, the distance difference \(d_{n1} = d_n - d_1\) can be estimated as:

\[ \hat{d}_{n1} = \eta \Delta \hat{t}_{n1} \]  

(12)

where \(\eta\) denotes the sound average propagation speed. Now we define the following matrices:

\[
M = \begin{pmatrix} x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_N & y_N \end{pmatrix}, \quad D = \begin{pmatrix} -\hat{d}_{21} \\ -\hat{d}_{31} \\ \vdots \\ -\hat{d}_{N1} \end{pmatrix},
\]

\[
Q = \frac{1}{2} \begin{pmatrix} x_2^2 + y_2^2 - \hat{d}_{21}^2 \\ x_3^2 + y_3^2 - \hat{d}_{31}^2 \\ \vdots \\ x_N^2 + y_N^2 - \hat{d}_{N1}^2 \end{pmatrix}, \quad P = \begin{pmatrix} x_r \\ y_r \end{pmatrix}.
\]

With the least-squares method, the position can be estimated as:

\[
\hat{P} = d_1 M^T + M^T Q
\]

(13)

Depending on the accuracy requirement, the above message exchange process runs for several times. After a few rounds, the ordinary node will collect a set of time stamps consisting of \(T_1, t_2, t_3\) and \(T_4\) and a sequence of the reference node’s rough positions.

\[ C. \ \text{Phase II: Synchronization} \]

Synchronization phase consists of three major steps: propagation delay estimation, linear regression and propagation delay update, which are explained in the following.
1) Step I: Propagation Delay Estimation: According to Section III, in order to increase the accuracy of propagation delay estimation, it is necessary to deal with the bias associated with the straight-line propagation assumption. As shown in Fig. 7, since we know the ordinary node’s rough position, represented with $P_r$, in Fig. 7, a rough $y_r − y_s$ is easy to calculate, where $y_r$ is ordinary node’s position projected onto the Y axis, and $y_s$ is the reference node’s position on the Y axis. Therefore, with Eq. (9), by performing numerical search, a constant “$C$” can be fixed. In Eq. (8), by knowing “$C$”, the propagation delay “$T$” can be estimated with stratification effect compensated. Using this method, the ordinary node calculates the corresponding propagation delay “$T$” for each exchanged message. Thus, in the end, the ordinary node will collect a set of propagation delays that consist of $\tau_1$ and $\tau_2$.

2) Step II: Linear Regression: During this step, the ordinary node performs linear regression with Ordinary Least Square Estimation (OLSE) over the data points in Eq.(14) to estimate the draft clock skew $\theta$ and offset $\beta$, where $[\hat{t}]$ determines the index of the message exchange process, and $\tau_1$ and $\tau_2$ stand for the propagation delays for the exchanged messages.

\[
\begin{align*}
T_1 &= \theta * t_1 + \beta \\
T_4 &= \theta * t_4 + \beta \\
t_2 &= t_1 + \tau_1 \\
t_4 &= t_3 + \tau_2
\end{align*}
\]  

(15)

Thus, with a new skew $\theta$ and offset $\beta$, the propagation delay $\tau_1$ and $\tau_2$ can be updated. And since synchronization procedure is an optimization process, the propagation delays tend to be more accurate than the estimated ones in step I.

D. Phase III: Localization

Localization procedure is actually the opposite process of estimating propagation delay. After Phase II, using Eq. (8), with updated propagation delay “$T$”, the constant “$C$” can be fixed by doing numerical search. Then in Eq. (9), the $y_r − y_s$ can be estimated. Since $z_r−z_s$ is known, so “$\hat{R}$” in Fig. 7 is:

\[
R = \sqrt{(y_r - y_s)^2 + (z_r - z_s)^2}
\]

(16)

With this method, the ordinary node computes “$\hat{R}$” for each reference node to build a vector consisting of “$\hat{R}$s”:

\[
(\hat{R}_1, \hat{R}_2, \hat{R}_3, \cdots, \hat{R}_N)^T
\]

Considering TDOA, we can observe that “$R$” is just the “d” and therefore it is easy to convert the above vector to “$D$” in Eq. (13). Next, with updated “$D$”, the position “$P$” can be estimated with Eq. (13). Using this method, for each run of message exchanges, one position will be estimated, such that the ordinary node will have a sequence of updated position measurements.

However, in the mobile scenario where either ordinary nodes and reference nodes are moving, it is hard and nearly impossible to estimate the real time distance or propagation delay. Note that our localization method is based on the estimates of the distances. Therefore, some errors might be introduced. In order to reduce the effect from mobility, we adopt a tracking algorithm to predict sensor nodes’ position at next measurement time. In this work, localization estimates come from a combination of prediction and measurements.

However, given the potential of multiple moving modes of each node, a universal state equation is not possible. We hence employ an IMM estimator to accommodate the multiple maneuver patterns. A brief introduction on the IMM estimator is provided in the sequel, and a detailed description can be found in the book [11].

IMM is an adaptive estimation approach. Different from many other methods which assume a particular moving pattern of the node, the IMM filter incorporates all the possible moving patterns of the node, by running a bank of filters in parallel with each filter corresponding to one particular moving pattern. And the overall state estimate is a certain combination of these model-conditional estimates.

In this paper, we assume two moving patterns of the node:

- **uniform motion**: moving along the straight line with a constant velocity, which can be modeled by a Kalman filter.
- **maneuver**: coordinated turn with a constant turn rate and a constant speed, which can be modeled by an extended Kalman filter.

We have the two filters running in parallel, and the combination of the estimates from the two filters yields the final estimated
position of the node as shown in Fig. 8
Given that nodes’ moving pattern can change from one mode to another, the IMM estimate of the node location is computed as a weighted summation of the estimates from multiple mode-matching filters. A Markov chain transition matrix characterizing the transition probability of the node’s moving pattern from one mode to another, is introduced to update the weighting coefficients. Except that the initial condition of each filter is a combination of the model-conditioned estimates of the two filters at the preceding state, operation of each filter is identical to that of the Kalman filter or the extended Kalman filter, depending on whether the state equation under the considered mode is linear or not.

E. Phase IV: Iteration
In the iteration process, synchronization and localization phase runs multiple times to improve the accuracy as shown in Fig. 4. For each loop, synchronization will apply the updated positions estimated in localization phase in the previous run. In other words, synchronization uses results produced by localization. Meanwhile, during localization, it applies the propagation delays updated with synchronization process. That actually means, localization uses the result produced by synchronization. So with iteration, the two independent processes are jointed together and help each other to improve their accuracy. The iteration will run for certain times or until the position, clock skew, clock offset become stable.

F. Discussions
Considering the whole procedure of JSL shown in Fig. 4, message exchange is to collect sample data, say “time stamps”, and the first TDOA and the depth sensor are to launch the procedure by giving initial position values. In the iteration loop, the propagation delay is the bridge to connect time synchronization and localization. With the method introduced in Section III, the propagation delay with stratification effect compensated is estimated based on rough position (first run) or the updated position from previous localization process. With this propagation delay, the ordinary node can be synchronized by applying linear regression (here we use OLSE). And since linear regression is a optimization process, after ordinary node is synchronized, the updated propagation delay suppose to be more accurate than that before it is synchronized. Next, with the updated propagation delay, localization can also be done with a least-square solution formulated with Eq. (13). In the same way, the estimated position suppose to be more accurate than it in the previous run, so that it can be an update input to the loop to launch another run of iteration. In summary, since both localization and synchronization use least-squares solutions to optimize, the accuracy of both services will be improved jointly.

V. PERFORMANCE EVALUATION
A. Simulation Settings
The simulations are implemented in Matlab. 20 underwater sensor nodes are randomly distributed in a $200m \times 200m \times 1000m$ region, and any two sensor nodes can communicate with each other. We randomly pick one node as the ordinary node which aims to become synchronized and select another node as the reference node. Without loss of generality, the inherent clock skew of the ordinary node is predefined as $10^4 ppm$, and it remains unchanged during the time synchronization process. The clock offset of the ordinary node is initialized as 2s. In addition, the response time $t_r$ is fixed as 1s. We consider the kinematic model in [12]. Its moving speed can be described as follows:

$$
\begin{align*}
    V_x &= k_1 \lambda v \sin(kk_2x) \cos(kk_3y) + k_1 \lambda \cos(2kk_1t) + k_4 \\
    V_y &= -\mu v \cos(kk_2x) \sin(kk_3y) + k_5 \\
\end{align*}
$$

where $V_x$ represents the speed at $X$ axis, $V_y$ stands for the speed at $Y$ axis. $k_1, k_2, k_3, \lambda, v$ are variables which are closely related to environment factors such as tides and bathymetry. These parameters change in different environments. $k_4, k_5$ are random variables which are used to simulate the random factors. In our simulation, we assume $k_1, k_2$ to be random variables which are normally distributed with a mean $\pi$ and a standard deviation 0.1, $k_3$ follows a normal distribution with $2\pi$ as the mean value and the standard derivation 0.2. $\lambda$ is also normal distributed with 3 as the mean value and 0.3 as the standard derivation. $v$ also follows a 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. $k$ is a coefficient controlling the frequency of changing direction.

In this simulation, we assume that sensor nodes follow two mobility patterns: uniform motion and maneuver with coordinated turn. Therefore, an IMM-tracker with the above two modes is used. Mode 1 is used for the uniform motion with a constant velocity, while mode 2 is used for the maneuver with coordinated turn. The two modes are described by a Kalman filter and an extended Kalman filter, respectively. The transition probabilities of the two modes used in simulations are

$$
[p_{ij}] = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \quad i,j = 1, 2
$$

B. Results and Analysis
1) Performance on localization accuracy: Fig. 9 shows localization accuracy of four algorithms including JSL, SLM, without stratification compensation, SLMp VI) and TDOA which are pure localization algorithms acting as benchmarks. SLMp is a localization algorithm which adopts linear prediction schemes to reduce its overhead, and to make fair comparison, we only count the localization process with anchor nodes as reference nodes. Results disclose that JSL performs much better than the other three, and that is because it does not assume the straight line acoustic transmission. Additionally, coupling time synchronization and localization together, JSL makes them help

1In the real scenario, sensor nodes can have various mobility patterns. The IMM-tracker can be extended to have multiple mode-matching filters to address these different patterns.
each other to improve the accuracy of both. That JSL without stratification compensation is worse than SLMP indicates the importance of the stratification effect, but it is still better than TDOA since it benefits from the joint design. From Fig. 9 we can also observe that, with more reference sensor nodes, the results become better, and that is because when more reference nodes are involved, more data can be collected in the rough position estimations and synchronization procedure. Both of these will help to improve localization accuracy.

2) Performance on synchronization accuracy: Fig. 10 illustrates how error grows after time synchronization completion with different algorithms, e.g. JSL, JSL-without stratification compensation, MU-Sync and TSHL. As discussed above, as time passes, the skew causes increasing errors. Therefore, this comparison actually demonstrates different accuracies on synchronization that these three algorithms can achieve. Fig. 10 compares the performance of the three algorithms using an identical message overhead. The result shows that JSL performs better than MU-Sync and TSHL, and that is because of two reasons. First, JSL considers the range bias caused by the stratification effect, which will finally improve the accuracy of both localization and synchronization. The second is the benefit of the joint design, which is also why JSL-without stratification compensation is more precise than MU-Sync and TSHL.

3) Effect of stratification effect on synchronization: Fig. 11 shows that with the same amount of messages, JSL compensating for the stratification effect can achieve higher accuracy than that without stratification compensation in terms of time synchronization, where the errors are measured 100s after running the JSL algorithm. That is because ignoring the stratification effect would lead to a considerable bias in the range estimates. This bias will be delivered to the propagation delay estimation and eventually affects the accuracy of time synchronization. From the result we also observe that as more messages are involved, more sample data will be collected, and the estimated skew and offset will be more precise.

4) Effect of stratification effect on localization: Fig. 12 demonstrates that JSL compensating for the stratification effect can achieve higher accuracy than that without stratification compensation in terms of localization. The normalized MSE (mean square error) stands for $NMSE = E[(\hat{r} - r)^2]$, where $r = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2 + (z - \hat{z})^2}$. This is also because ignoring stratification effect would lead to a considerable bias in the range estimates, which will in turn affect localization accuracy. From the results, we can also recognize that with more iterations, localization error goes down and becomes
stably. Since during the iteration process, localization is using the result produced by synchronization, and synchronization process is using the result produced by localization. This indicates that localization and synchronization are helping each other to improve the accuracy of both services.

5) **Performance of iteration:** Fig. 13 demonstrates that with more iterations, the synchronization error which is measured 100s after running the JSL algorithm goes down and becomes stable, which is also a proof that localization and synchronization are helping each other to improve the accuracy. From the results, it also concludes that with different real skews, the synchronization accuracy does not vary a lot, which indicates that the real skew does not affect the synchronization result.

6) **Performance of IMM:** Fig. 14 shows the contribution of the tracking mechanism to localization accuracy. From the result, we can clearly see that with the help of IMM tracking, localization error is always lower than without using any tracking algorithm. Although in our simulation setting, as the coefficient “k” increases, the sensor node’s velocity changes too fast in direction, the tracking mechanism still helps.

VI. RELATED WORK

A. Related Work

Although there are significantly growing interests in UWSNs in the past several years, the research on time synchronization for UWSNs is still relatively limited.

TSHL [8] is designed for high latency networks, which can manage long propagation delays and remain energy efficient. TSHL combines one-way and two-way MAC-layer message delivery. TSHL works well for static underwater sensor networks, but it cannot handle mobile scenarios as it assumes constant propagation delays among sensor nodes.

MU-Sync [9] is proposed to synchronize nodes in a cluster based UWSNs. Although MU-Sync aims to solve sensor node mobility issue in UWSNs, it requires relatively high message overhead. Furthermore, MU-Sync uses half of the round trip time to calculate one way propagation delay, which causes extra errors, especially for fast moving situations or when ordinary nodes respond to cluster head after experiencing a long time duration.

Mobi-Sync [13] is a time synchronization scheme for mobile underwater acoustic sensor networks. Mobi-Sync distinguishes itself by considering spatial correlation among the mobility patterns of neighboring UWSNs nodes. This enables Mobi-Sync to accurately estimate the long dynamic propagation delays. However, Mobi-Sync only works for dense network as it is based on spatial correlation.

TSMU [14] is a pairwise synchronization method. It effectively utilizes Doppler effects and incorporates the relative speed of the transmitter and receiver to improve the dynamic propagation delay estimation. Additionally, Kalman filter and the calibration process are exploited to further enhance synchronization accuracy. However, TSMU only show its performance with linear kinematic model, which limits its practical applicability.

Traditionally, localization algorithms are either range-based or range-free. In this work, we focus on the range-based localization algorithms. Ranging techniques are either communication-based or connectivity-based. GPS Intelligent Buoys (GIBs) [15] and PARADIGM [16] use underwater GPS. Those to-be-localized sensor nodes need to communicate with the surface buoys, because of which, the time synchronization is a base for them in order to convert time information into range information. [17] tries to avoid time synchronization by using half of the round trip time to estimate the propagation delay. It requires a sensor node to communicate with multiple surface buoys, which may introduce a heavy load of traffic in the network. A silent positioning scheme is proposed [18] which does not depends on time synchronization. Instead, all the sensor nodes get localized by passively listening to the beacon messages exchanged among anchor nodes, so that it requires four noncoplanar anchors that can mutually hear each other.

Usually, connectivity-based method is only used when there is no direct communication between nodes and sensors, where
range estimation is estimated based on network connectivity. [19] proves that euclidean method performs the best in anisotropic topologies, but with more cost on computation and communication. In [20], it proves that with short communication range among anchor nodes, euclidean method can be adopted for 3D underwater localization. [21] relaxes this limitation by proving it is also working when the anchor nodes communication range is long. However, both [20] and [21] are suffering from heavy traffic due to using flooding mechanism. [22] introduced “SLMP”, which applies some prediction schemes to localization algorithm to reduce its overhead. 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. However, because the prediction is based on temporal and spatial correlations, the algorithm only works in dense network.

Tian et al. proposed the first localization and synchronization scheme for 3D underwater acoustic networks in [23], using atomic multilateration and iterative multilateration techniques. In this scheme, the to-be localized sensor node obtains time and localization information from the anchor nodes, which are localized on the surface of the water in order to perform time synchronization and localization. The drawbacks of this algorithm is that it ignores the clock skew when doing time synchronization, which will lead to frequent re-synchronization. Additionally, the time values and locations are calculated in the same formulas as variables and they do not improve upon each other, which makes this “joint” solution weak.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we presented JSL, a joint solution for time synchronization and localization. To our best knowledge, it is the first localization scheme which compensates the stratification effect in the underwater environment. Furthermore, synchronization and localization are closely coupled and help each other to improve the accuracy of each other. An advanced tracking algorithm IMM is adopted to further improve accuracy. Our simulation results show that JSL can achieve high accuracy for both synchronization and localization.

In the future, we plan to design a network-wide synchronization and localization scheme based on JSL. After one ordinary node is synchronized and localized, it will become a reference node to help to synchronize and localize other ordinary nodes. In this way, the scheme can be used in a large scale network.

REFERENCES