Motion Estimation of Underwater Platforms Using Impulse Responses From the Seafloor

JOSÉ E. ALMANZA-MEDINA, (Member, IEEE), BENJAMIN HENSON, (Member, IEEE), LU SHEN, (Member, IEEE), AND YURIY ZAKHAROV, (Senior Member, IEEE)

Department of Electronic Engineering, University of York, YO10 5DD York, U.K.

Corresponding author: José E. Almanza-Medina (jeam502@york.ac.uk)

The work of José E. Almanza-Medina was supported by the Mexican National Council of Science and Technology (CONACyT). The work of Benjamin Henson, Lu Shen, and Yuriy Zakharov was supported in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/R003297/1 and Grant EP/V009591/1.

ABSTRACT

Autonomous underwater vehicles require accurate navigation. Techniques such as image registration using consecutive acoustic images from a sonar have shown promising results for this task. The implementation of such techniques using sonar images augmented with deep learning (DL) networks demonstrate high navigation accuracy; this is possible even with highly compressed images. The sonar images are estimates of sampled in time (with a ping period) magnitudes of channel impulse responses representing the underwater acoustic environment. More information about the environment is contained in (almost) continuous in time estimates of the channel impulse responses. Such estimates can be obtained using full-duplex technology. Rather than using sonar images, this paper investigates the use of channel impulse response estimates for underwater platform motion estimation. The proposed system uses a single projector and a small number of receiving transducers installed on the moving platform. A DL network is used to estimate the motion in two degrees of freedom (forward/backward and sideways), using two or more consecutive impulse response estimates as the input. To train the DL network, a specially designed simulator is used to model the underwater acoustic environment, populated with multiple objects spread on the seafloor. The proposed technique can significantly reduce the acoustic hardware and processing complexity of the DL network and obtain a higher accuracy of motion estimation, compared with techniques based on the processing of sonar images, e.g., the error achieved with the technique proposed in this paper is 1.7% of the maximum platform displacement, compared to 4% achieved with a technique using sonar images. The navigation accuracy is further illustrated by examples of estimation of complex trajectories.

INDEX TERMS

Deep learning, full-duplex, impulse response, motion estimation, sonar, underwater navigation.

I. INTRODUCTION

Almost three quarters of the Earth’s surface is covered by water but only a small part of the underwater environment has been explored [1], [2], [3]. Nowadays, underwater exploration missions rely on the use of autonomous platforms whose navigation system is crucial for their accurate operation [4], [5]. The task of the navigation system is to obtain the platform location and velocity as it moves through the body of water [5]. The navigation can be based on a variety of motion estimation techniques by integrating the motion estimates into the underwater platform trajectory with a navigation algorithm [6]. In this paper, we address the motion estimation.

A. RELATED WORK

Underwater motion estimation techniques and their corresponding navigation techniques can be divided into three types [6], as described below. However, combinations of the techniques are possible.

1) Techniques based on beacons: This type of navigation requires the use of nodes at fixed positions that send signals (acoustic beacons) used by the underwater platform to calculate its position. The long baseline (LBL),
short baseline (SBL) and ultra short baseline (USBL) systems fall in this type of navigation [7], [8]. The major constraints of these techniques are that calibration can be problematic [9] and that navigation is only possible in a limited geographical area covered by the beacons [10], [11].

2) Inertial and dead reckoning systems: An inertial navigation system (INS) uses accelerometers and gyroscopes to estimate the platform position, velocity and attitude [12]. However, an INS suffers from an unavoidable drift which can be compensated by the fusion with other sensors such as the Doppler velocity log (DVL) [13], [14]. A DVL uses a bottom-referenced acoustic navigation method that consists in transmitting narrow beams from three or four sensors in the direction of the seafloor at different angles and the platform velocity is calculated from the measured Doppler effect. To produce the narrow beams, high frequencies are required and therefore the acoustic signals are highly attenuated, limiting its usage to shallow waters [15]. A correlation velocity log (CVL) is less constrained in this respect [16]. A CVL has a looking down single wide-beam projector and an array of hydrophones to receive the backscattered signals from the seafloor. This allows the use of lower frequencies with less signal attenuation [16].

3) Techniques based on environment referencing: These techniques use sensors for extracting information from the platform surrounding environment to estimate the position and attitude and can be divided into visual and acoustic. The visual odometry uses images from optical cameras to estimate the underwater platform motion and can provide simultaneous localization and mapping [17], [18]. Optical methods have the limitation of visibility, even in perfect conditions they only work up to around 20 m of water [19]. Acoustic methods can overcome this limitation and they can operate further than 100 m [19]. In [20] and [21], pairs of consecutive sonar images are used to estimate the motion of an underwater platform in three degrees of freedom (DoF) by using a deterministic image registration method. In [22], the method from [21] is improved by using deep learning (DL) networks. In [23], sonar images from multiple sensors are compressed into vectors and grouped together to be used as the input of the DL network, thus reducing the DL network complexity, while keeping the high motion estimation accuracy.

B. CONTRIBUTION

In this work, we combine features from the different mentioned techniques and propose a novel technique for motion estimation, using a DL network and considering the motion on a plane parallel to the seafloor. Similar to the CVL, we use a single transmitter that sends acoustic signals to the seafloor and several (up to four in our simulated scenarios) receivers (hydrophones) placed around the transmitter to listen to the reflected signals.

One of the main differences between the proposed technique and a CVL is that the CVL works by transmitting two short ping pulses that are reflected from the seafloor and the echoes are measured at the receivers [16], [24], while the proposed technique uses a continuous signal transmission and the receivers are continuously listening at the same time. This allows continuous measurement of the channel impulse response while systems like the CVL have a ping period for each measurement, which limits amount of information used for the motion estimation. The simultaneous transmission/listening operation is possible to implement with full-duplex techniques such as those recently proposed in [25] and [26]. Another difference is that a CVL measures the correlation between responses of the two pings [24], whereas our technique exploits the complex-valued channel impulse response variation in time and uses a DL network to estimate the motion. Finally, a CVL requires an array with a large number of receivers [16], [27] while the proposed technique can achieve fairly accurate motion estimation with only two receivers and high accuracy with four receivers. Compared to the techniques based on environment referencing, such as in [20], [21], [22], and [23], where sonar images represent estimates of the channel impulse response magnitudes varying in time, and thus ignoring the phase information, the proposed method is based on the whole information of the channel impulse response and thus is capable of providing a better motion estimation performance. The proposed technique is based on acoustic impulse response estimates with a much higher time resolution and carrying more information than the classical sonar images.

The motion is estimated by a DL network, which has as its input two or more consecutive in time complex-valued baseband estimates of the impulse responses between the transmitter and hydrophones. We do not propose a new DL architecture, but reuse the PoseNet architecture [28] previously used for motion estimation using sonar images, which has shown good results compared to other DL networks. In this paper, we however adjust the parameters of the PoseNet to guarantee a high motion estimation performance with impulse responses as input. Note that a large amount of labeled data is required when training a DL network for regression tasks [29]. To solve this problem, we use synthetic data generated by a computer simulator, similar to the approach used in [22] and [23]. The simulator can provide large data sets with the ground truth required for supervised learning.

In summary, we propose a motion estimation method, enabled by full-duplex operation, and based on DL analysis of time variation of complex-valued channel impulse responses. Compared to known techniques based on sonar images represented by channel impulse response magnitudes, the proposed method is based on the whole information of the channel impulse response and thus is capable of providing a better motion estimation performance. The technique
presented in this paper is validated using simulated data from multiple trajectories, and it shows a high accuracy in estimation of the trajectories.

The structure of the rest of this paper is as follows. In Section II, the simulator, steps to generate channel impulse responses and corresponding data sets are described. In Section III, the DL network for motion estimation is described. Section IV presents results of motion estimation experiments for numerous sets of system parameters. A comparison with methods that use sonar images is also presented. Finally, conclusions are given in Section V.

II. SIMULATOR TO GENERATE IMPULSE RESPONSES OF THE ACOUSTIC ENVIRONMENT

This section describes the simulator capable of generating channel impulse responses for complex underwater acoustic environments. These channel impulse responses are used for training the DL network for motion estimation and for evaluation of the proposed method.

Simulators presented in the literature are capable of modeling underwater environments, platforms and sensors. For example, the simulator [30] uses a combination of OpenSceneGraph and Robot Operating System (ROS) to generate realistic underwater scenarios, platforms and sensors; however, none of them are an acoustic sensor. The work [31] presents a ray-tracing simulator for the design and evaluation of sonars. The simulator is implemented in Unreal Engine to display the simulated environment and ROS to manipulate the platforms. The sonar simulator presented in [32] is capable of simulating sonars that can be used for validating image registration techniques. The simulated sonar sensors and the underwater environment are implemented in Unity [33]. In [32], the simulator uses ray-tracing to generate the sonar images. The simulated sonar sends multiple rays according to its field of view (FoV). The rays in the FoV are equally separated in the aperture and elevation angles. When a ray hits an object, the sonar measures the distance to the hit point and the angle of incidence on the surface of the object and stores these values along with a reflectivity value of the hit surface and the orientation and position coordinates of the sensor in the simulated scenario at this instant. Then, MATLAB [34] uses the data from Unity to generate sonar images. To generate sonar images along a trajectory, the simulator uses a hop-and-generate process, i.e., the simulated sensor collects data to generate an image in a position of the simulated environment, then the sensor “hops” to another position in accordance with the trajectory and collects data to generate another image and so on. These sonar images are sets of magnitudes of the channel impulse response.

The aim of this work is to estimate the motion of a platform using relatively simple acoustic sensors (one projector and two - four hydrophones) and at the same time reduce the estimation complexity. The use of impulse responses allows us to achieve both these aims. However, a simulator that can produce the channel impulse responses with a high precision is required. The work [35] presents an underwater channel simulator that uses the BELLHOP ray-tracing model [36] to simulate an acoustic underwater channel with multipath propagation. A method to model signal transmissions in underwater acoustic communications is presented in [37], where the motion of the transmitter and the receiver is taken into account. This method reduces the model complexity with an approximation of the time-varying channel impulse response by sampling the trajectory and using local splines for the approximation. The work [38] complements the work [37] by reducing even further the simulation time. This is obtained with the use of baseband processing for modelling the signal transmission which allows a lower sampling rate compared to the passband processing.

For this work, it is required to model the channel with a precision enough to recover the phases of the channel impulse response as opposed to sonar images, where only magnitudes are required. Another requirement is that the simulator must be capable of simulating multiple underwater objects to have a high variety of data for training and validation of DL networks. Also, the simulator must generate simulated scenarios in a three-dimensional space. Therefore, the simulator from [32] is modified to generate impulse responses. The ray-tracing in this simulator provides a precision high enough for the signal phase to be taken into account and the Unity platform allows complicated 3D underwater environments to be generated.

The steps to generate the impulse responses are summarized in Figure 1 and they are described in the next subsections.

A. COLLECTING DATA IN A SINGLE POSITION IN THE SIMULATED ENVIRONMENT

We assume that the transmitter and receiver are at different positions on the platform and they have (maybe different) predefined FoVs. A set of rays uniformly sampling the transmitter FoVs are emitted from the transmitter position. When a ray hits an object, it bounces towards the receiver and the following measurements are recorded:
B. COMPUTATION OF THE PROPAGATION DELAYS FOR A STATIC PLATFORM

The measurements obtained for every ray in a position of the platform are used to generate the corresponding impulse response. The rays outside the FoV of the receiver are discarded by analyzing the arrival angles. For each of the remaining rays, the distance from the transmitter to the hit point to the receiver is calculated as

$$\delta_0 = ||P-A||,$$

(1)

and the distance from the hit point to the receiver is calculated as

$$\delta_1 = ||B-P||,$$

(2)

where $B = [b_x, b_y, b_z]$ is the position of the receiver on the platform. Considering that a receiver is placed at point $B$ and it has an offset $E = [e_x, e_y, 0]$ from the transmitter position, then

$$B = [a_x + e_x, a_y + e_y, a_z].$$

(3)

The propagation time that a ray takes to travel over the distance $\delta_0 + \delta_1$, from the transmitter to the receiver, assuming that the platform is static is given by

$$\tau_P = \tau_0 + \tau_1 = \frac{\delta_0}{c_{uw}} + \frac{\delta_1}{c_{uw}},$$

(4)

where $c_{uw}$ is the speed of sound underwater, considered to be 1500 m/s. The calculated times for each ray are saved into a vector of propagation delays $D$.

In addition, a vector $I$ is defined with ray magnitudes computed as the product $J_iG_i$.

C. MOTION COMPENSATION

The propagation delay $\tau_P$ in (4) is calculated assuming that the platform is static during the travel of the rays. However, for a more accurate representation of the impulse response, the delay computation should take into account that the platform is moving at a certain velocity $V = [V_x, V_y]$. The diagram in Figure 3 shows the motion of the platform and how it affects the propagation delay of a ray.

The displacement of the receiver ($\delta_{xy} = [\delta_x, \delta_y]$) at velocity $V$ during the time that the ray travels can be approximately calculated as

$$\delta_{xy} = (\tau_0 + \tau_1)V.$$

(5)

The propagation delay should be calculated at the displaced position of the receiver, which is

$$B' = [a_x + e_x + \delta_x, a_y + e_y + \delta_y, a_z].$$

(6)

The distance from the hit point to the displaced position of the receiver is given by

$$\delta_2 = ||B'-P||,$$

(7)

and the time to travel the distance $\delta_2$ is computed as

$$\tau_2 = \frac{\delta_2}{c_{uw}},$$

(8)

then the propagation time along the ray taking into account that the platform is moving is

$$\tau'_P = \tau_0 + \tau_2.$$  

(9)

The delay $\tau'_P$ in (9) is a more accurate ray propagation delay compared to the delay $\tau_P$ computed in (4). However, this computation can be further refined by repeating the steps (5) to (9) as follows. The propagation delay of the ray is adjusted once again by taking into account the updated propagation delay...
time $\tau'_{P}$. The receiver moves to a new position $B''$ during this time. The distance from the hit point to the receiver at $B''$ and the time that a ray travels this distance are

$$\delta_3 = ||B'' - P||,$$

$$\tau_3 = \frac{\delta_3}{c_{uw}},$$

where

$$B'' = [a_x + e_x + \delta'_x, a_y + e_y + \delta'_y, a_z],$$

$$\delta'_{x,y} = [(\tau_0 + \tau_2)V].$$

The propagation time along the ray taking into account that the platform is moving is refined as

$$\tau''_{P} = \tau_0 + \tau_3.$$  

This process can be iterated further, but for this work, the delay is compensated only twice given that further adjustment of the delay is negligible for the delay accuracy required. Finally, the vector $D$ is populated by the updated propagation delays $\tau''_{P}$ and used together with the vector $I$ to generate the impulse response.

**D. IMPULSE RESPONSE GENERATION**

The delay $D$ and magnitude $I$ of all rays (multipath components) are used for computing the channel impulse response. The maximum propagation delay is $\tau_{\text{max}} = \max(D)$. The carrier frequency is $f_c$, the frequency bandwidth of interest is $F_d$, thus the frequency interval of interest is $[f_c - F_d/2, f_c + F_d/2]$.

First, the channel frequency response is computed and then converted into the time domain impulse response ($h$) using the inverse Fast Fourier Transform (IFFT). The frequency response $H_f(k)$ at a frequency $f_k$ is computed as:

$$H_f(k) = \begin{cases} 
1, & f_k \in [-c, c] \\
\frac{1}{2} \left[ 1 + \cos \left( \frac{2\pi|f_k|}{\alpha F_d} \right) - \frac{(1-\alpha)F_d}{2} \right], & f_k \in [-a, -c) \cup (a, c], \\
0, & f_k \in [-b, -a] \cup (a, b) 
\end{cases}$$

where $c = (1-\alpha)F_d/2$. The frequency response is multiplied by $X_R(k)$ to guarantee a smooth transition at the edges of the frequency bandwidth as:

$$H(k) = H_f(k)X_R(k), \quad f_k \in [-b, b].$$

![FIGURE 3. Diagram that explains the delay compensation due to the platform motion. The point $B$ moves with the platform while the ray travels from the hit point to the receiver.](image)

![FIGURE 4. The raised-cosine spectrum with roll-off factor $\alpha$.](image)
E. GENERATION OF DATA SETS OF IMPULSE RESPONSES FOR TRAINING AND VALIDATION OF DL NETWORKS

Impulse responses for multiple receivers are generated and grouped together. It is assumed that a single transmitter is used to illuminate the underwater environment, then the difference in the impulse responses will depend on the positions of the receivers on the platform.

To generate a data set, groups of impulse responses from \( M \) receivers are packed together. The real and imaginary parts of the impulse responses (Re(h) and Im(h)) from all the receivers are obtained. Compared to the techniques that use sonar images, where only magnitudes are obtained by the sonar, the advantage of using the complex values (real and imaginary parts) is that fast phase variations can be detected, which is an indication of motion.

A matrix is made by putting together the real and imaginary parts of impulse responses generated in \( Q \) consecutive positions of the platform. Each column of a matrix is a real or imaginary part of an impulse response. Therefore, each matrix has \( 2MQ \) columns. The order of the columns is defined as follows: (i) the real parts of the impulse responses from the first receiver at the \( Q \) platform positions; (ii) imaginary parts of the impulse responses from the first receiver at the \( Q \) platform positions; (iii) the same order (real parts and then imaginary parts) for the second receiver; (iv) and so on for the rest of the receivers.

The maximum magnitude value is computed from the impulse responses of the \( M \) receivers in the \( Q \) consecutive platform positions whose real and imaginary parts are used to make the matrix. Then the matrix is normalized by dividing its elements by the maximum magnitude value. Every normalized matrix is associated with a label corresponding to the displacement between the first and the last positions. The displacement is denoted by \( \Delta = [\Delta_x, \Delta_y] \) (motion in forward/backwards and sideways direction, respectively). Finally, a data set is generated with multiple matrices and their associated labels.

III. DL NETWORK

Numerous DL networks for motion estimation using sonar images are evaluated in [22] where the PoseNet [28] showed the best performance. In this work, the PoseNet is adapted for motion estimation using the impulse responses. The whole network is shown in Figure 6. The input of the network is a matrix with the real and imaginary parts of impulse responses obtained by \( M \) receivers in \( Q \) consecutive platform positions. The matrix is split into matrix blocks. Each matrix block corresponds to the data from one receiver. For every matrix block, the DL network contains a PoseNet, which is a series of convolutional layers as shown in Figure 6b. The number of convolutional layers varies according to the need to optimize the network for each parameter configuration used to generate the impulse responses. It was observed that 9 convolutional layers was the maximum needed for the parameter configurations investigated in this work. All the convolutional layers with the exception of the last one use a ReLU activation function and batch normalization. An average pooling layer with an averaging window of size 4 is connected to the last convolutional layer. The dropout regularization method is applied at the output of the first ReLU and to the average layer.

Parameters of the convolutional layers are presented in Table 1. Experiments showed better results in reducing overfitting when using dropout in the first and last layers only. The last layers of each PoseNet are concatenated together and connected to a fully connected layer and finally to an output regression layer that generates the motion estimates for the two DoF. As in [22] and [23], the Mean Squared Error (MSE) is used as the loss function of the regression layer as follows:

\[
\mathcal{L} = \frac{1}{2SR} \sum_{k=1}^{S} ||\hat{\Delta}_k - \Delta_k||^2,
\]

where \( S \) is the number of mini-batches, \( k \) is the index that refers to the training samples in the mini-batch and \( R \) is the number of parameters to estimate, which is set to 2 for this work and \( \hat{\Delta} = [\hat{\Delta}_x, \hat{\Delta}_y] \) are the estimates in forward/backwards and sideways directions.

IV. EXPERIMENTS AND RESULTS

This section presents results of numerical experiments for one transmitter and four receivers on a simulated platform. The transmitter is surrounded by the receivers, each 10 cm from the transmitter as shown in Figure 7. The FoVs of the receivers are a narrow beam looking to the front, left, back and right of the platform for the receivers 1 to 4, respectively, as shown in Figure 8. The use of four receivers is the most investigated case in this work but the use of only two receivers (forward and right looking) is also considered.

A. SIMULATED ENVIRONMENT

The simulated environment consists of a 50 m by 50 m flat seafloor, divided into a grid of cells (2048 \( \times \) 2048). Each cell has a reflectivity defined as a realization of random value
with a uniform distribution between 0 and 1. The environment is populated by multiple objects placed on the seafloor. There are two types of objects: ellipses and rectangles. The objects are almost flat (1 mm height) while the other two dimensions (width and length) are realizations of random values uniformly distributed between 0.2 m and 1 m. The low height value is chosen to keep a simple model but later the height of the objects is increased to investigate how the motion estimates behave with taller objects. Every object has a reflectivity, which is a random value with a uniform distribution between 0 and 1. Twenty different simulated scenarios are generated to have a high variety of impulse responses for training and validation of DL networks. The difference between the scenarios is in the number and position of the objects on the seafloor. The number of objects and their position are randomly generated for each scenario. In each scenario there are between 2000 and 5000 rectangles and between 2000 and 5000 ellipses. An example of a simulated scenario with the underwater platform and the FoV of one of the receivers is shown in Figure 9.

B. TRANSMITTER RECEIVERS CONFIGURATIONS

For this work, two DoF are considered, corresponding to the forward/backward and sideways motions of the platform at a constant height from the seafloor of 10 m. The platform velocity is a random value with a uniform probability distribution in the range from $-2$ m/s to 2 m/s for each DoF and a time interval between consecutive impulse response estimates ($\tau_r$) is set to 50 ms. Therefore, the maximum speed in a particular direction is approximately 2.8 m/s. The value $\tau_r$ approximately matches to the ping period in a model of a
TABLE 1. Parameters of the PoseNet convolutional layers. The output size per channel is calculated considering an impulse response length of 800 taps. The output size per channel changes according to parameter configurations investigated in this work.

| Layer | Kernel | Channels | Stride | Output size per channel | ReLU / Dropout |
|-------|--------|----------|--------|--------------------------|----------------|
| Conv1 | 7×4    | 32       | 2×1    | 400×4                    | Yes / Yes      |
| Conv2 | 5×4    | 64       | 2×1    | 200×4                    | Yes / No       |
| Conv3 | 3×4    | 128      | 2×1    | 100×4                    | Yes / No       |
| Conv4 | 3×4    | 256      | 2×1    | 50×4                     | Yes / No       |
| Conv5 | 3×4    | 512      | 2×1    | 25×4                     | Yes / No       |
| Conv6 | 3×4    | 512      | 2×1    | 13×4                     | Yes / No       |
| Conv7 | 3×4    | 512      | 2×1    | 7×4                      | Yes / No       |
| Conv8 | 3×4    | 512      | 2×1    | 4×8                      | Yes / No       |
| Conv9 | 3×4    | 512      | 2×1    | 2×4                      | No / No        |
| Average| 4×4    | 512      | 1×1    | 2×4                      | No / Yes       |

DIDSON 300 sonar [39] which has been used for investigation of the motion estimation in works [22] and [23].

Data sets of impulse responses are generated for various parameter sets (configurations) of the transmitter and receivers. The configurations differ in the signal parameters and the FoV size and shape. The investigated configurations are the following:

- **Two receivers:** This configuration uses two \((M = 2)\) receivers (forward and side-looking) and \(Q = 2\). The DL network is modified to have only two series of convolutional layers shown in Figure 6b. This is the only configuration with two receivers, in the other configurations, four \((M = 4)\) receivers are used.

- **Standard configuration:** This is considered as the basis configuration that uses the data from four \((M = 4)\) receivers and \(Q = 2\). The other configurations are obtained by changing one or a few parameters of this configuration. The impulse responses are generated by considering a FoV of 10° by 50° for the receivers, a bandwidth of 4.8 kHz, and a carrier frequency of 80 kHz.

- **Narrow FoVs:** This configuration uses a narrow beam of 2° by 50° in each receiver. The same frequency parameters as in the standard configuration are used.

- **Other carrier frequencies:** Four data sets are generated with the impulse responses by changing the carrier frequency \(f_c\) to 40 kHz, 60 kHz, 100 kHz and 160 kHz. The bandwidth is 4.8 kHz.

- **Higher frequency bandwidth:** The bandwidth is set to 9.6 kHz and 19.2 kHz. The carrier frequency is 80 kHz.

- **Circular FoVs:** This configuration investigates the use of a FoV that focuses on a spot on the seafloor. It consists in a circular FoV of 10° by 10° in each receiver. It looks at a pitch angle of 45° as shown in Figure 10. Two data sets are generated with the bandwidth of 4.8 kHz and 19.2 kHz. The carrier frequency is 80 kHz.

- **Objects of higher height:** For this configuration, the objects on the seafloor are replaced with objects of 10 cm height rather than 1 mm to investigate how it can affect the motion estimation.

- **Groups of ten impulse responses:** The platform velocity is the same, but the receivers generate ten \((Q = 10)\) impulse responses in the same time interval of 50 ms rather than only two \((Q = 2)\) impulse responses. Groups with the ten impulse responses are packed together as a training sample. The carrier frequency is set to 80 kHz, 160 kHz or 320 kHz.

For each configuration, a data set of 3000 samples (matrices with real and imaginary parts of the impulse responses from the receivers, associated with the displacement labels) is generated and divided into training and validation sets, 80% and 20% of samples, respectively.

A DL network is trained and validated for each data set. When training the network, the Adam optimization algorithm is used [40]. The mini-batch size \((S)\) is set to 4. The learning rate at the start of the training is set to \(5 \times 10^{-5}\) and it is reduced by 20% every 20 epochs. The training stops when the validation loss converges. The root mean square error (RMSE) is used to evaluate the estimation performance for each DoF.
FIGURE 8: Receivers FoV seen from the side, top and back view of the platform.

FIGURE 9: Simulated scenario showing the underwater platform with the FoV of one of the receivers and the almost flat objects on the seafloor. The seafloor is divided in tiny cells whose colors are defined by the reflectivity in the range from 0 to 1, where black color corresponds to 0 and the totally brown color corresponds to 1 (Left side bar). The colors of the objects are defined by the reflectivity in the range from 0 to 1, where black color corresponds to 0 and the totally red color corresponds to 1 (Right side bar).

An example of the training and validation curves obtained during training is shown in Figure 11. The curves show the RMSE (which is the square root of the loss function $L$) against training iterations. The curves were obtained when training the DL network with parameters shown in Table 1 for the configuration with a FoV of 10° by 50° for the receivers, a bandwidth of 19.2 kHz, and a carrier frequency of 80 kHz (see Table 2 below). Thus it can be seen that that the DL model does not suffer from the overfitting problem. Similar observations have been witnessed for the other scenarios considered in this paper.

Table 2 shows results of the validation and computing time per estimate for each of the configurations. The time required per estimate is shown for a standard PC with i5-6500 CPU @3.20GHz processor, 8.0 GB of RAM and without a GPU. The number of layers of the PoseNet used for every configuration are also specified in Table 2. This number was observed to be the most suitable for each configuration and adding more layers showed no further improvement. For each configuration, all the convolutional layers with the exception of the last one have a ReLU activation function and batch normalization.

From the results in Table 2 the following observations are made:

- The best estimation accuracy is achieved when using the groups of ten ($Q = 10$) impulse responses. This is due to the highest amount of information given to the network for the estimation. The greatest constraint of training a network with this technique is the time to generate the training sets, since it requires 5 times more impulse responses per training sample compared to the standard configuration. The time per estimate is comparable to most of the other configurations with the same frequency bandwidth.

- The increase of the frequency bandwidth to 9.6 kHz and further to 19.2 kHz significantly improves the estimation accuracy. The impulse responses with higher bandwidth contain more information, which improves the estimates. However, this results in a longer time interval to compute the estimate.

- The increase in the number of receivers from two to four improves the motion estimation accuracy, but even with only two receivers accurate estimates can be obtained.
This is an important option if the platform is small, and therefore it would not be possible to accommodate many receivers.

- The narrow FoV of $2^\circ \times 50^\circ$ provides better estimates than the wider FoV of $10^\circ \times 50^\circ$.
- With the frequency bandwidth 4.8 kHz, the circular FoV of $10^\circ \times 10^\circ$ shows an increase in the RMSE compared to using the FoV of $10^\circ \times 50^\circ$. This is due to a significant reduction in the elevation angle, i.e., a smaller part of the seafloor is involved in the motion estimation. However, with the frequency bandwidth of 19.2 kHz, the reduction in the FoV less affects the performance compared to the frequency bandwidth of 4.8 kHz.
- In the case of using pairs of impulse responses, the different values of the carrier frequency show the best estimation accuracy for frequencies nearer to 80 kHz. The network parameters were modified to improve the performance for the other carrier frequency values. However, the estimation error obtained is not as low as the estimation error for 80 kHz.
- In the case of using groups of 10 impulse responses, the best estimation accuracy is for higher carrier frequencies.
- The use of higher carrier frequencies and higher bandwidth require more layers in the PoseNet to improve the estimation accuracy compared to lower frequencies. However, from the point of view of implementation, higher frequencies allow the equipment to be smaller compared to the use of lower frequencies.
- The use of 10 cm height objects on the seafloor does not affect the estimation performance much compared to the case of 1 mm height.
- In general, the motion estimation in forward/backward and sideways directions show a similar accuracy. This is because of the symmetry between both types of motion.
- It can be seen that the lower the bandwidth, the lower is the computation time, which is due to the size of the network. For a higher bandwidth the network is required to be more complex and have more layers and internal parameters. For most of the configurations, the computation time per estimate is lower than 50 ms. This makes them suitable for real-time applications considering 50 ms as the measurement interval.

C. COMPARISON WITH OTHER TECHNIQUES FOR MOTION ESTIMATION

The accuracy of three known techniques for motion estimation using sonar images are compared with the accuracy of the technique presented in this paper. The three techniques can estimate the motion of a platform in 3 DoF (forward/backward and sideways motion and yaw rotation). However, for the comparison only the motion in forward/backward and sideways is considered. The first technique is a deterministic algorithm for attitude and trajectory estimation [21]. It uses sequences of sonar images and estimates pixel displacements between two sonar images. The second technique performs motion estimation using a DL network trained with sonar images generated by a simulator [22]. The images are concatenated in pairs of consecutive images and the concatenated image is the input of the DL network. The third technique [23] converts sonar images into vectors by adding up all the pixels in each row of the image. The images for training and validation are generated in a simulator. Then the vectors are concatenated and input to a DL network to perform the motion estimation. The accuracy of these three techniques is measured by the RMSE and validated using simulated data. The simulated environment consists of an underwater scenario with rocks of different sizes and reflectivity coefficients covering the seafloor. The simulated sonar is DIDSON 300 for the first two techniques.

For the comparison, two proposed configurations with the best estimation performance are used: (i) pairs of impulse responses with $F_d$ of 19.2 kHz, $f_c$ of 80 kHz and FoV of $10^\circ \times 50^\circ$ and (ii) groups of ten impulse responses with $F_d$ of 4.8 kHz, $f_c$ of 80 kHz and FoV of $10^\circ \times 50^\circ$. The RMSE for motion estimation of all the techniques are shown in Table 3. It can be seen that the proposed technique significantly improves the estimation performance compared to the known techniques. Furthermore, the proposed technique could reduce the implementation cost, given that it only requires the use of one transducer and a few hydrophones, rather than an imaging sonar.

D. TRAJECTORY RECONSTRUCTION USING SIMULATED DATA

The DL network trained with the data set of 19.2 kHz is used to estimate the displacements in simulated trajectories and recover the trajectories. The impulse responses obtained in the trajectories are grouped into pairs of consecutive positions of the platform. Each pair of impulse responses is given to the trained DL network to estimate the displacement between these two positions. The platform trajectory is recovered using these displacement estimates.
The points of a trajectory are represented as \( x_i \) and \( y_i \), where \( i \) is an index that refers to the position number in the trajectory. The initial position \( x_1, y_1 \) of the estimated trajectory is set to zero. The other points of the trajectory are calculated by adding the displacement estimate in each DoF:

\[
x_{i+1} = x_i + \hat{x}_i, \\
y_{i+1} = y_i + \hat{y}_i,
\]

where \( \hat{x}_i \) and \( \hat{y}_i \) are displacement estimates in the forward/backward and sideways directions, respectively.

Six trajectories are generated in the simulator as shown in Figure 12. The description of each trajectory is as follows:

- **Trajectory 1** (Figure 12a): It has a square shape with starting and finishing points at the same place (loop trajectory). The platform moves in all four directions at different time intervals (forward, right, backward, left).

- **Trajectory 2** (Figure 12b): It has forward and right motion in a stair shape. The trajectory consists of 400 positions with different speeds the same as in trajectory 1 but in different directions.

- **Trajectory 3** (Figure 12c): The platform moves in a zigzag shape. The trajectory consists of 340 positions. The sequence of displacements consists of 80 displacements to the backward at a speed of 1 m/s and 50 displacements at a speed of 1.5 m/s and 50 displacements at a speed of 0.5 m/s, the last motion (backwards) has 50 displacements at a speed of 1.5 m/s and 50 displacements at a speed of 0.5 m/s.

The size of each side of the square is 5 m. The trajectory is built with 400 positions (399 displacements). The first motion (left) consists of 100 displacements at a speed of 1 m/s, the second motion (forward) has 100 displacements at a speed of 1 m/s, the third motion (right) consists of 50 displacements at a speed of 1.5 m/s and 50 displacements at a speed of 0.5 m/s, the last motion (backwards) has 50 displacements at a speed of 1.5 m/s and 50 displacements at a speed of 0.5 m/s.

### TABLE 2. Validation errors of DL networks trained for different configurations.

| Bandwidth \((F_d)\) | Number of receivers \((M)\) | Carrier frequency \((f_c)\) | FoV | PoseNet layers | RMSE (mm) | Time per estimate (ms) |
|------------------|----------------|----------------|-----|---------------|------------|---------------------|
|                  |                |                |     |               | Fwd/bwd | Sideways |                  |
| 4.8 kHz          | 2              | 80 kHz         | 10° × 50° | 7     | 5.9          | 6.0        | 7.8              |
|                  | 4              | 40 kHz         | 10° × 50° | 7     | 10.5         | 8.8        | 13.8             |
|                  |                | 60 kHz         | 10° × 50° | 7     | 6.2          | 6.3        | 38.0             |
|                  |                | 80 kHz         | 10° × 50° | 7     | 4.2          | 5.1        | 45.1             |
|                  |                |                | 2° × 50°  | 7     | 2.3          | 2.7        | 44.1             |
|                  |                |                | Circular | 7     | 15.2         | 11.0       | 8.4              |
|                  |                |                | 10° × 10° |       |             |            |                  |
| 100 kHz          | 4              | 80 kHz         | 10° × 50° | 7     | 7.6          | 7.7        | 22.6             |
| 160 kHz          |                | 10° × 50°      | 8     | 8.84         | 9.05       | 11.9             |
| 9.6 kHz          | 4              | 80 kHz         | 10° × 50° | 8     | 3.8          | 3.9        | 116.6            |
| 19.2 kHz         | 4              | 80 kHz         | 10° × 50° | 9     | 2.4          | 1.9        | 237.5            |
|                  |                |                | Circular | 9     | 3.5          | 2.0        | 389.4            |
|                  |                |                | 10° × 10° |       |             |            |                  |

### TABLE 3. RMSE for motion estimation in forward/backward and sideways directions for known and proposed techniques.

| Technique | RMSE (mm) | Maximum displacement (mm) | RMSE / maximum displacement |
|-----------|-----------|---------------------------|-----------------------------|
|           | Fwd/bwd  | Sideways | Fwd/bwd | Sideways |
| [21]      | 8.39      | 1.29     | 20     | 42%       | 6.5%      |
| [22]      | 2.70      | 0.87     | 20     | 13.5%     | 4.4%      |
| [23]      | 0.80      | 1.20     | 20     | 4.0%      | 6.0%      |
| Impulse responses method \((F_d = 19.2 kHz)\) | 2.37 | 1.90 | 100 | 2.4% | 1.9% |
| Impulse responses method (groups of ten) | 1.70 | 1.80 | 100 | 1.7% | 1.8% |
displacements at 1 m/s, 40 forward displacements at 0.5 m/s, 50 left displacements at 1.5 m/s, 40 backward displacements at 0.5 m/s, 50 left displacements at 0.5 m/s, and 30 forward displacements at 1.5 m/s.

- Trajectory 4 (Figure 12d): It has motion in all four directions to make a circle shape. The radius of the circle is 5 m. The trajectory is made of 400 positions while the platform moves at a speed of approximately 1.5 m/s.
- Trajectory 5 (Figure 12e): It is of a sinusoid shape combining forward motion while moving sideways. The trajectory consists of 400 positions. The platform changes the speed across the trajectory, with maximum speed of approximately 2 m/s and minimum speed of approximately 0.6 m/s.
- Trajectory 6 (Figure 12f): It has motions in forward and right directions at the same time to make a combined diagonal motion. The trajectory is made of 200 positions at a constant speed of approximately 1.4 m/s (1 m/s forward speed and 1 m/s left motion speed).

Comparing the ground truth and estimated trajectories, it can be seen that the estimated trajectories accurately follow the ground truth with a small error. The RMSE for each of the estimated trajectories in two DoF can be seen in Table 4 as well as the final error magnitude (difference between the point where the estimated trajectory should end and the point where it actually does). The RMSE is in the range from 1.5 mm to 3.2 mm for all the trajectories and the highest error magnitude is 31 cm, which is a high accuracy considering that the total travelled distances are higher than 14 m.

| Trajectory | RMSE (mm) | Error magnitude (cm) |
|------------|-----------|----------------------|
|            | Fwd/bwd   | Sideways             |
| 1          | 2.08      | 3.16                 | 31        |
| 2          | 3.26      | 2.60                 | 19        |
| 3          | 3.35      | 2.93                 | 10        |
| 4          | 2.08      | 3.30                 | 15        |
| 5          | 2.08      | 3.16                 | 19        |
| 6          | 2.18      | 1.51                 | 14        |

The error in the loop trajectories (1 and 4) is also measured according to how far from the initial point is the ending point. For trajectory 1 (square shape), the error is 31 cm and for trajectory 4 (circle shape) the error is 15 cm. The total distance travelled by the platform over these two trajectories is 20 m and about 31.42 m, respectively, which makes the final error of about 1.6% and 0.5% of the total travelled distance, respectively.

V. CONCLUSION

In this paper, we developed a DL method for motion estimation of an underwater platform that uses the impulse responses obtained for (two or four) receivers mounted on the platform. A model to generate the impulse responses in an underwater simulator is presented and used to produce different data sets to train the DL network. The data sets differ from each other in the parameters used to generate the impulse responses such as the bandwidth and carrier frequency. When generating the impulse responses, a delay compensation technique is applied before calculating the impulse responses, to consider the Doppler effect due to the platform motion.

The motion estimation techniques are validated with simulated data. The proposed technique shows a similar or higher accuracy than techniques that use sonar images rather than impulse responses. Furthermore, the use of impulse responses can reduce significantly the cost of implementation compared to the use of sonar images.

Six simulated trajectories are presented and fully estimated with the impulse responses method, showing high estimation accuracy and almost replicating the ground truth trajectory. For two loop trajectories, where the starting point and the ending point in the underwater scenario are the same, low errors of 1.6% and 0.5% are achieved.

In this paper, the motion estimation based on the channel impulse responses from the seafloor has been investigated only for two DoF. The estimation for three DoF (including the yaw motion) is the next step of the investigation. In our future work, we will also optimize configuration parameters of the sensors, such as the distance and orientation relative to the transmitter and the FoV to further improve the estimation accuracy. The important question is also how to integrate these motion estimates into a trajectory estimate; in this paper, we only used a simple integration algorithm to illustrate the accuracy of the motion estimates. Finally, we are also planning to conduct field experiments to validate the proposed technique.

REFERENCES

[1] C.-C. Kao, Y.-S. Lin, G.-D. Wu, and C.-J. Huang, “A comprehensive study on the internet of underwater things: Applications, challenges, and channel models,” Sensors, vol. 17, no. 7, p. 1477, Jun. 2017.
[2] Z. Jin, Q. Zhao, and Y. Luo, “Routing void prediction and repairing in AUV-assisted underwater acoustic sensor networks,” IEEE Access, vol. 8, pp. 54200–54212, 2020.
[3] R. Nwosu and J. Peter, “A review on underwater acoustic sensor network,” Austral. J. Sci. Technol., vol. 5, no. 4, pp. 678–685, 2021.
[4] C. Meek and Z. Song, “Fast autonomous underwater exploration using a hybrid focus model with semantic representation,” in Proc. OCEANS, Oct. 2019, pp. 1–6.
[5] I. Klein and R. Diamant, “Dead reckoning for trajectory estimation of underwater drifters under water currents,” J. Mar. Sci. Eng., vol. 8, no. 3, p. 205, May 2020.
[6] L. Paull, S. Saeedi, M. Seto, and H. Li, “AUV navigation and localization: A review,” IEEE J. Ocean. Eng., vol. 39, no. 1, pp. 131–149, Jan. 2013.
[7] Y. Xu, W. Liu, X. Ding, P. Lv, C. Feng, B. He, and T. Yan, “USBL positioning system based adaptive Kalman filter in AUV,” in Proc. OCEANS, May 2018, pp. 1–4.
[8] Y. Yang, Y. Xiao, and T. Li, “A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability,” IEEE Commun. Surveys Tuts., vol. 23, no. 2, pp. 815–841, 2021.
[9] Y. Cong, C. Gu, T. Zhang, and Y. Gao, “Underwater robot sensing technology: A survey,” Fundam. Res., vol. 1, no. 3, pp. 337–345, May 2021.
[10] W. Yan, W. Chen, and R. Cui, “Moving long baseline positioning algorithm with uncertain sound speed,” J. Mech. Sci. Technol., vol. 29, no. 9, pp. 3995–4002, Sep. 2015.
[11] L. N. P. Pasnani and M. L. Seto, “Underwater gravity-based navigation for long-range AUV missions,” Maritime Eng. J., no. 94, pp. 18–21, 2020.
[12] H. Zhao, Z. Wang, S. Qiu, Y. Shen, L. Zhang, K. Tang, and G. Fortino, “Heading drift reduction for foot-mounted inertial navigation system via multi-sensor fusion and dual-gait analysis,” IEEE Sensors J., vol. 19, no. 19, pp. 8514–8521, Oct. 2018.
Z. Xu, M. Haroutunian, A. J. Murphy, J. Neasham, and R. Norman, “Design considerations,” BELLHOP3D User Guide, vol. 20, no. 23, pp. 14329–14336, Dec. 2020.

P. Boltryk, M. Hill, A. Keary, B. Phillips, H. Robinson, and P. White, “An ultrasonic transducer array for velocity measurement in underwater vehicles,” Ultrasonics, vol. 42, nos. 1–9, pp. 473–478, Apr. 2004.

T. E. Blanford, D. C. Brown, and R. J. Meyer, Jr., “Design considerations for a compact correlation velocity log,” in Proc. Meetings Acoust., vol. 33, no. 1, May 2018, Art. no. 007003.

B. Teixeira, H. Silva, A. Matos, and E. Silva, “Deep learning for underwater visual odometry estimation,” IEEE Access, vol. 8, pp. 44667–44701, 2020.

Z. Xu, M. Haroutunian, A. J. Murphy, J. Neasham, and R. Norman, “An integrated visual odometry system for underwater vehicles,” IEEE J. Ocean. Eng., vol. 46, no. 3, pp. 848–863, Jul. 2021.

F. Ferreira, D. Machado, G. Ferri, S. Dugelay, and J. Potter, “Underwater optical and acoustic imaging: A time for fusion? A brief overview of the state-of-the-art,” in Proc. OCEANS, Monterey, CA, USA, Sep. 2016, pp. 1–6.

N. Hurtós, D. Ribas, X. Cufí, Y. Petillot, and J. Salvi, “Fourier-based registration for robust forward-looking sonar mosaicing in low-visibility underwater environments,” J. Field Robot., vol. 32, no. 1, pp. 123–151, Jan. 2015.

B. T. Henson and Y. V. Zakharov, “Attitude-trajectory estimation for forward-looking multibeam sonar based on acoustic image registration,” IEEE J. Ocean. Eng., vol. 44, no. 3, pp. 753–766, Jul. 2018.

J. E. Almanza-Medina, B. Henson, and Y. V. Zakharov, “Deep learning architectures for navigation using forward looking sonar images,” IEEE Access, vol. 9, pp. 33880–33896, 2021.

J. E. Almanza-Medina, B. Henson, and Y. V. Zakharov, “Sonar FoV segmentation for motion estimation using DL networks,” IEEE Access, vol. 10, pp. 25591–25604, 2022.

A. Keary, M. Hill, P. White, and H. Robinson, “Simulation of the correlation velocity log using a computer based acoustic model,” in Proc. Int. Symp. Unmanned Untethered Submersible Technol., 1999, pp. 446–454.

L. Shen, B. Henson, Y. Zakharov, and P. Mitchell, “Two-stage self-interference cancellation for full-duplex underwater acoustic systems,” in Proc. OCEANS, Jun. 2019.

L. Shen, B. Henson, Y. Zakharov, and P. Mitchell, “Digital self-interference cancellation for full-duplex underwater acoustic systems,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 67, no. 1, pp. 192–196, Jan. 2019.

P. J. Boltryk, M. Hill, A. C. Keary, and P. R. White, “Surface fitting for improving the resolution of peak estimation on a sparsely sampled two-dimensional surface with high levels of variance, for an acoustic velocity log,” Meas. Sci. Technol., vol. 15, no. 3, p. 581, 2004.

Z. Yin and J. Shi, “GeoNet: Unsupervised learning of dense depth, optical and acoustic imaging: A time for fusion? A brief overview of the state-of-the-art,” in Proc. OCEANS, Monterey, CA, USA, Sep. 2016, pp. 1–6.

T. A. Le, A. G. Baydin, R. Zinkov, and F. Wood, “Using synthetic data to improve the resolution of peak estimation on a sparsely sampled two-dimensional surface with high levels of variance, for an acoustic velocity log,” Meas. Sci. Technol., vol. 15, no. 3, p. 581, 2004.

M. Prats, J. Perez, J. J. Fernandez, and P. J. Sanz, “An open source tool for improving the resolution of peak estimation on a sparsely sampled two-dimensional surface with high levels of variance, for an acoustic velocity log,” Meas. Sci. Technol., vol. 15, no. 3, p. 581, 2004.

C. Liu, Y. V. Zakharov, and T. Chen, “Doubly selective underwater acoustic channel model for a moving transmitter/receiver,” IEEE Trans. Veh. Technol., vol. 61, no. 3, pp. 938–950, Mar. 2012.

B. Henson, J. Li, Y. V. Zakharov, and C. Liu, “Waymark baseband underwater acoustic propagation model,” in Proc. Underwater Commun. Netw. (UComms), Sep. 2014, pp. 1–5.

DIDSON 300 Standard Version Specifications. Sound Metrics Corp. Accessed: Jul. 8, 2021. [Online]. Available: http://www.soundmetrics.com/products/DIDSON-Sonars/DIDSON-300/DIDSON-300-Standard-Version Specifications

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv.1412.6980.