IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 12, 2020, accepted March 13, 2020, date of publication March 20, 2020, date of current version April 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982272

Navnet: AUV Navigation Through Deep Sequential Learning

XIN ZHANG, BO HE, (Member, IEEE), GUANGLIANG LI, (Member, IEEE), XIAOKAI MU, YING ZHOU, AND TANJI MANG

College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China

Corresponding authors: Bo He (bhe@ouc.edu.cn) and Guangliang Li (guangliangli@ouc.edu.cn)

This work was supported in part by the Marine S&T Fund of Shandong Province for Pilot National Laboratory for Marine Science and Technology (Qingdao) under Grant 2018SDKJ0102-7, in part by the National Key Research and Development Program of China under GRANT 2016YFC0301400, in part by the National Natural Science Foundation of China under Grant 51379198 and Grant 51809246, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2018QF003.

ABSTRACT Achieving accurate navigation and localization is crucial for Autonomous Underwater Vehicle (AUV). Traditional navigation algorithms, such as Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF), require the system model and measurement model for state estimation to obtain the AUV position. However, this may introduce modeling errors and state estimation errors which will affect the final precision of AUV navigation system to a certain extent. To avoid these problems, in this paper, we proposed a deep framework — NavNet — by taking AUV navigation as a deep sequential learning problem. Firstly, the proposed NavNet can take raw sensor data at different frequencies as input, which benefits from the sequential learning capability of Recurrent Neural Network (RNN). Secondly, NavNet takes advantage of a simplified attention mechanism and Fully Connected (FC) layers to output AUV displacements per unit time, which accomplishes low-frequency AUV navigation by accumulation of it. More importantly, there is no need for the model building and state estimation with NavNet, which avoids the import of relevant errors. We compare the performance of NavNet to EKF and UKF using collected data by running Sailfish in the sea. Experimental results show that NavNet has an excellent performance in terms of both the navigation accuracy and fault tolerance. In addition, a reliable fusion strategy of NavNet and conventional method is applied to achieve high-frequency AUV navigation. The experimental results show that the proposed architecture can be a reliable supplement to limit the error growth of conventional algorithms.

INDEX TERMS Autonomous underwater vehicle, navigation, extended Kalman filter, unscented Kalman filter, sequential learning.

I. INTRODUCTION

In recent years, Autonomous Underwater Vehicles (AUVs) play a crucial role in a tremendous variety of missions, such as oceanographic surveys, marine data acquisition, submarine rescue, and underwater equipment maintenance [1], [2]. In order to accomplish these missions safely and gather the valid oceanic data, accurate navigation and localization is especially vital for AUVs. However, different from the situations of the outdoor robots [3] and aerial robots [4], underwater navigation and localization are very challenging tasks due to the limitation of the information transmission method and transmission distance. For example, Global Positioning System (GPS) is not available when the vehicle operates underwater. At the same time, the complicated underwater environment also makes it difficult to perform navigation.

Even with so many difficulties and challenges, AUV navigation and localization techniques have made considerable progress over the past few decades. The primary AUV navigation and localization approaches are divided into three categories: acoustic navigation, geophysical navigation and inertial navigation [5], [6]. By measuring the Time of Flight (TOF) of signals from acoustic beacons or modems, acoustic navigation techniques have been applied to perform AUV navigation. However, acoustic beacons need to be installed in advance, which limits the operating area to a specific scope [7]. Moreover, the cost of the installation and maintenance for acoustic beacons can be hardly affordable for developing small-scale shallow sea AUVs. Recent development of acoustic modems has made the stationary beacons

The associate editor coordinating the review of this manuscript and approving it for publication was Hui Xie.

unnecessary and reduced the cost to a certain degree [8]. Meanwhile, the mission range is extended because of the mobility of the beacons during the missions. Nevertheless, the acoustic modem technology also suffers from the disadvantages of acoustic propagation: the small bandwidth, high latency and unreliability. Moreover, additional assistance from manned surface support or autonomous surface crafts is needed, which incurs extra operational expenditure.

Geophysical navigation techniques [9] depend on sonar sensors or optical sensors to detect and identify the environmental information. However, the lack of existing submarine feature maps and the challenging underwater feature recognition tasks emerge as obstacles for the utilization of these methodologies. Therefore, inertial navigation [10] has become the most popular mean of shallow sea AUV navigation and localization. Inertial navigation integrates the rotation rate measurements from gyroscopes to obtain the vehicle attitude. Simultaneously, the observations from accelerometers are integrated twice to acquire the vehicle position relative to a known initial location. However, since the positions are acquired by continuous integration of measurements from accelerometers with respect to time, Inertial Navigation Systems (INS) are susceptible to error accumulation, leading to a position drift. In order to maintain a long-term high navigation accuracy, Doppler Velocity Log (DVL) is equipped on AUV to collect the bottom-track speed. By combining INS and DVL measurements, a degree of improvement for the issue of drift will be generated. Nonetheless, high-precision INS is too expensive as well as oversized for small-scale AUVs.

With the rapid development of the low-cost Micro Electro-Mechanical Systems (MEMS) inertial sensors, Attitude and Heading Reference Systems (AHRS) made up of several MEMS inertial sensors have become popular choices for a part of small-scale shallow sea AUVs, which can be integrated with DVL to perform AUV navigation [11]–[13]. During a specific mission, the data collected by different sensors need to be processed together to acquire an optimal estimate of the AUV position. By constructing the system model and measurement model, a wide variety of state estimation techniques are employed to derive the estimate of the AUV position. At present, one of the most commonly used state estimation techniques for AUV navigation is Kalman Filter (KF) [14]. KF is is an optimal Bayesian estimator assuming that the system is Markovian, linear and its uncertainties are subject to Gaussian distribution. In fact, the built system model of AUV navigation is always nonlinear, which dissatisfies the hypothesis requirements of KF. In order to overcome these limitations, Extended Kalman Filter (EKF) [15], [16] and Unscented Kalman Filter (UKF) [17], [18] are applied as the expansion form to approximate the nonlinear processes. The former linearizes the nonlinear model by first-order Taylor series and the latter utilizes statistical linearization to approximate the nonlinear function. In addition, by incorporating the non-Gaussian distributions and nonlinear models, Particle Filter (PF) was also proposed in [19], [20], and it has good performance in AUV navigation. The state estimation techniques mentioned above represent valid and fundamental means to achieve an accurate estimate of the vehicle position, which are widely used in the field of AUV navigation.

Simultaneous Localization and Mapping (SLAM) technique which expands from the land robotics to the AUVs enables them to build a map of the surroundings while concurrently localizing itself in this environment. The wildly used filter-based SLAM techniques rely on the previously discussed state estimation algorithms, which yield many new navigation ways for the underwater domain. As a well-known approach for SLAM in underwater environments, EKF-SLAM [16], [21] has been validated to be efficacious under the actual underwater environment. However, it is not suitable for large maps because of the high computational complexity. To reduce the computational complexity, FastSLAM that is based on PF was proposed. FastSLAM shows both the poses of the vehicle and features of the map with particles and is suitable for the real-time application of SLAM in large-scale environments [22], [23]. It should be stressed that the state estimation techniques mentioned above have contributed greatly to the development of AUV navigation. However, all of these state estimation techniques will introduce nonlinear errors, linear truncation errors, or approximation errors to a certain extent. Meanwhile, in the underwater environment, the established navigation models cannot exactly describe the complicated motion of AUV, which will import modeling errors to some degree.

In the last few years, Deep Learning (DL) has received significant attention in many fields. For instance, Convolutional Neural Network (CNN) makes a huge impact on image classification [24] and object detection [25]. Recurrent Neural Network (RNN) is a deep neural network designed to handle sequential learning, which is widely used in speech recognition [26] and machine translation [27]. Inspired by research on DL, many solutions based on DL has been proposed to perform the autonomous navigation. For example, Mohanty *et al.* [28] studied the problem of monocular Visual Odometry (VO) based on a CNN framework, instead of the usual methods through feature detection and tracking. Their work was tested in both known and unknown environments. The experimental results showed that the proposed framework worked well in the known environment, however, the effect is not very good in the unknown environment. Wang *et al.* [29] presented a novel end-to-end framework for monocular VO by utilizing deep Recurrent Convolutional Neural Network (RCNN). The deep RCNN model was trained in an end-to-end manner, which deduced poses directly from a sequence of raw RGB images without adopting any conventional VO procedures. Their framework was tested and verified on the public dataset, KITTI VO dataset [30]. Then they proposed an End-to-end, Sequence-to-sequence Probabilistic Visual Odometry (ESP-VO) framework for the monocular VO based on deep RCNNs [31]. The uncertainty is derived through the

architecture without any extra computation. In addition, Li *et al.* [32] came up with a UnDeepVO framework to estimate the poses of a monocular camera and the depth of its view by applying unsupervised deep neural networks. Clark *et al.* [33] put forward a VINet to motion estimation using visual and inertial sensors for Visual Inertial Odometry (VIO) which integrated the data at an intermediate feature-representation level. In their work, both the Inertial Measurement Unit (IMU) data and monocular RGB images are taken as input for the proposed CNN-RNN model to estimate the poses. Turan *et al.* [34] proposed an VO approach based on RCNN for endoscopic capsule robots. They evaluated their method on a pig stomach dataset and their experimental results show that the framework can realize high translational and rotational accuracies for different types of endoscopic capsule robot trajectories. However, research on navigation with DL mainly focus on the visual navigation or visual-inertial navigation field for outdoor robots and aerial robots, and only very few of them were applied in the maritime engineering field [35], [36].

Inspired by the successful research on DL in the field of vision navigation, in this paper, we propose to use sequential learning to perform AUV navigation. Specifically, different from the regular methods that work on dealing with state estimation errors, in this paper, we propose NavNet, a deep framework based on deep neural networks for AUV navigation. Since the raw sensor values are taken as input for the deep framework to get the AUV position without using a specific system model, the inevitable errors introduced by the modeling process can be prevented which results in an improvement on the the accuracy of AUV navigation. There are three main contributions in the paper: 1) A deep framework named NavNet is proposed to achieve AUV navigation without building the regular system models; 2) We verified the proposed framework on AUV with real experimental data and show an improvement in the accuracy of AUV navigation and better fault tolerance compared to the traditional methods; 3) An integration strategy of both the conventional method and the deep framework to fulfill high-frequency navigation is introduced. Specifically, in the strategy, outputs of the proposed framework are combined with the estimated position of the traditional algorithms to acquire approximated AUV position, which can be regarded as virtual position measurements for AUV. Experimental results show that the proposed sequential learning method can be a reliable supplement to the classic algorithms.

The rest of this paper is organized as follows. Section II highlights the basic structure of our independently developed AUV platform — Sailfish 210 AUV. Section III provides a detailed description of the traditional methods for AUV navigation. Both the architecture of NavNet and the fusion strategy are elucidated in Section IV. Section V presents the analysis and discussion on the experimental results. Finally, the conclusion is drawn in Section VI.

II. AUV PLATFORM

In this section, we describe the experimental AUV platform. Fig. 1 shows our independently developed AUV platform — Sailfish 210. Sailfish 210 is a small-scale AUV, which is mainly deployed in shallow sea region. The diameter and length of Sailfish 210 are respectively 0.21 m and 2.3 m. It can operate continuously for ten hours and its tonnage is 72 kg. In addition, the maximum cruising speed is 5 knot. The Sailfish 210 is equipped with many sensors to perform a variety of tasks, such as autonomous navigation, motion control, and path planning. A single board computer is equipped on Sailfish 210 and the Central Processing Unit (CPU) is two cores with 777 Mhz main frequency. In addition, two groups of rudders and one propeller are fitted at the rear of Sailfish 210.

FIGURE 1. Sailfish 210 AUV.

The basic structure of Sailfish 210 is depicted in Fig. 2. It can be seen clearly that the equipped sensors related to the navigation system include AHRS, DVL, GPS, and Intelligent Pressure Sensor (IPS). In the following part, we will introduce these sensors in detail respectively.

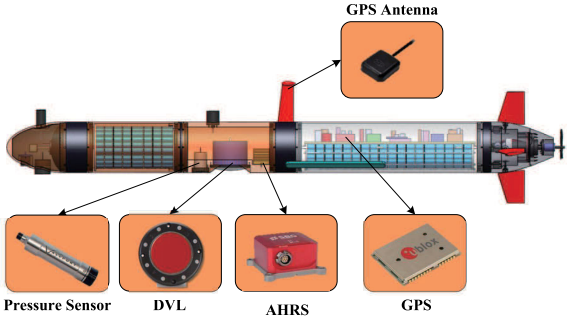


FIGURE 2. Basic structure of Sailfish 210 AUV.

A u-blox GPS is equipped on Sailfish 210 to get the ground truth during surfacing. The concrete specifications of the selected GPS are shown in Table 1. It provides the position information in the form of latitude and longitude. However, since the GPS signals are easy to encounter rapid attenuation through the water, GPS is not available in the underwater environments.

Sailfish 210 is equipped with an AHRS unit which is an attitude sensor to perceive the attitude of the vehicle.

TABLE 1. The concrete specifications of GPS equipped on AUV.

Sensor Type		u-blox LEA-M8T
Horizontal Position Accuracy	Autonomous	2.5 m
	SBAS	2.0 m
Velocity Accuracy		0.05 m/s
Heading Accuracy		0.3 degrees
Max Navigation Update Rate		5 Hz

TABLE 2. The concrete specifications of AHRS equipped on AUV.

Sensor Type		SBG Systems Ellipse-A
Bias In-run Instability	Accelerometers	20 μg
	Gyroscopes	8 deg/h
Roll, Pitch		0.2° RMS
Heading		1° RMS
Rate		Up to 200 Hz

The concrete specifications are shown in Table 2. A typical AHRS unit contains a triaxial gyroscope, a triaxial accelerometer and a triaxial magnetometer. Therefore, it can measure the angles, the triaxial accelerations, and the triaxial angular velocities of the vehicle. The orientation of the vehicle can be obtained by integrating the angular rates measured by gyroscope. Obviously, the integrals will lead to drifts during the processes of angles estimation. In addition, both the direct measurements of the gyroscope and the magnetometer will have a influence on the accuracy of long-term heading. Furthermore, the gravity vector measured by the accelerometer can affect the stability of long-term pitch and roll.

A downward DVL is installed on Sailfish 210, which can acquire the velocity of the vehicle by emitting acoustic pulses and receiving the reflected pulses. When the vehicle is close enough to the seafloor, DVL measures the Doppler shift from the reflected pulses and then determines the speed vector of the vehicle with respect to the seabed. In general, a DVL typically has four or more transceiver units. However, in Sailfish 210, a DVL with a phased array transducer is set up to obtain the forward, starboard and downward velocities relative to the body frame. Compared to the four beams sensors, the phased array DVL eliminates the effect of the sound velocity distinction of the seawater between different areas on frequency measurement. Otherwise, it will introduce errors in velocity measurement. The specifications of the selected phased array DVL are shown in Table 3.

TABLE 3. The concrete specifications of DVL equipped on AUV.

Sensor Type	TRDI DVL 600 kHz
Maximum Altitude	89 m
Minimum Altitude	2 m
Velocity Range	$\pm$16 m/s
Base Bottom Track Long Term Accuracy	$\pm$1.15% $\pm$0.2 cm/s
Ping Rate	12 Hz max
Resolution	0.1 cm/s

Sailfish 210 is also armed with an IPS and the Valeport miniIPS is actually used. In practice, the IPS is able to obtain an accurate determination of absolute depth based on the properties of seawater. As a consequence, AUV navigation and localization can be converted into a two-dimensional (2D) situation, and the 2D position of the vehicle can be obtained by integrating the data from AHRS and DVL. The specifications of the chosen Valeport miniIPS are illustrated in Table 4.

TABLE 4. The concrete specifications of IPS equipped on AUV.

Sensor Type	Valeport miniIPS
Range	5, 10, 30, 50, 100, 300 or 600 Bar
Resolution	0.001% range
Accuracy	$\pm$0.01% range
Response Time	1 milliseconds

III. REVIEW OF CONVENTIONAL METHOD FOR AUV NAVIGATION

In this section, we will review conventional method for AUV navigation. As shown in Fig. 3, after getting the raw sensor data, the system state can be estimated using the state estimation techniques. Meanwhile, the system model and measurement model are usually indispensable prerequisites for the state estimation process, which has been proven to be a valid and fundamental mean to achieve an accurate estimate of the AUV position. In the rest of this section, each part of the process will be introduced respectively.

A. MODEL BUILDING

Fig. 4 illustrates the motion of Sailfish in a simplified form. Here, the navigation coordinate system is the standard north-east-down coordinate system and *NOE* denotes the top view in a 2D case. Moreover, the body-fixed frame is defined as forward-starboard-down coordinate system and its top view is shown as $\eta G \xi$.

The state vector which reflects the status of the navigation system is composed of position, attitude, and velocity of the vehicle. Here, the specific state vector at time k is $X_k = [x, y, \varphi, v_x, v_y, a_x, a_y, w_z]^T$, where x and y denote the AUV position in the north-east coordinate system, φ is the yaw of AUV, v_x and v_y are the forward and starboard bottom-track velocity in the local coordinate system, a_x and a_y are the corresponding accelerations of the velocity v_x and v_y respectively, w_z stands for the angular velocity of yaw.

Based on the kinematic equations, the system model which is used to predict the evolvement from time $k-1$ to time k can be defined as (1) and (2), shown at the bottom of the next page:

$$X_k = f(X_{k-1}, m_{k-1}). \tag{1}$$

Here, $m = [m_x, m_y, m_\varphi, m_{v_x}, m_{v_y}, m_{a_x}, m_{a_y}, m_{w_z}]^T$ signifies the white Gaussian noises with zero mean. Meanwhile, the covariance of m is the system noise matrix Q which is shown as (3), at the bottom of the next page [37].

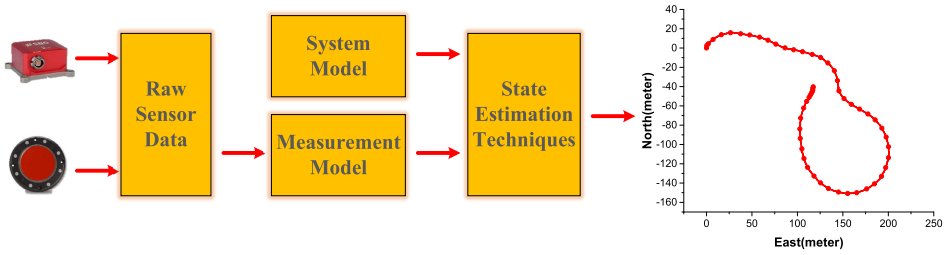


FIGURE 3. Pipeline of conventional method for AUV navigation.

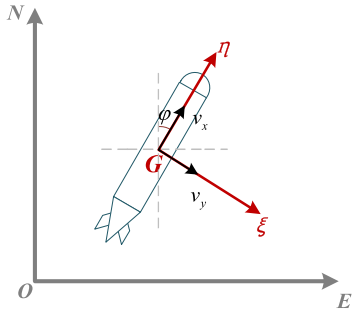


FIGURE 4. Coordinate systems in a 2D situation.

As previously discussed, several navigation sensors are equipped on the Sailfish which enable it directly to get some observations relevant to the state vector. Therefore, the measurement model is chosen as a linear form, which is described as:

$$Z_k = H_k X_{k|k-1} + n_k, \tag{4}$$

where Z represents the observation vector. For Sailfish, at time k it can be described as $Z_k = [\varphi, v_x, v_y, a_x, a_y, w_z]^T$. Moreover, n denotes the white Gaussian noises with zero mean, which is illustrated as $n = [n_\varphi, n_{v_x}, n_{v_y}, n_{a_x}, n_{a_y}, n_{w_z}]^T$. As mentioned, the corresponding observation matrix H can be written as:

$$H = [\,0_{6\times 2} \quad I_{6\times 6}\,]. \tag{5}$$

Meanwhile, the covariance of n is the measurement noise matrix R, which is expressed as:

$$R = \begin{bmatrix} \sigma^2_{n_\varphi} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{n_{v_x}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{n_{v_y}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{n_{a_x}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{n_{a_y}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{n_{w_z}} \end{bmatrix}. \tag{6}$$

In reality, because the AUV motion in the underwater environment is complex, the built models can only approximate this process. However, with the constructed system model and measurement model, various state estimation techniques can be used to achieve a comparatively accurate estimate of the AUV position.

$$\begin{bmatrix} x \\ y \\ \varphi \\ v_x \\ v_y \\ a_x \\ a_y \\ w_z \end{bmatrix}_k = \begin{bmatrix} x + (v_x t + \frac{1}{2}a_x t^2)\cos(\varphi) - (v_y t + \frac{1}{2}a_y t^2)sin(\varphi) + m_x \\ y + (v_x t + \frac{1}{2}a_x t^2)sin(\varphi) + (v_y t + \frac{1}{2}a_y t^2)\cos(\varphi) + m_y \\ \varphi + w_z t + m_\varphi \\ v_x + a_x t + m_{v_x} \\ v_y + a_y t + m_{v_y} \\ a_x + m_{a_x} \\ a_y + m_{a_y} \\ w_z + m_{w_z} \end{bmatrix}_{k-1} \tag{2}$$

$$Q = \begin{bmatrix} \sigma^2_{m_x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{m_y} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{m_\varphi} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{m_{v_x}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{m_{v_y}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{m_{a_x}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{m_{a_y}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{m_{w_z}} \end{bmatrix} \tag{3}$$

B. CONVENTIONAL STATE ESTIMATION TECHNIQUES

EKF is the most widely used and valid state estimation technique in AUV navigation. It is a nonlinear extension of KF and is more suitable for the real situations. The flow diagram of EKF is shown in Fig. 5 [38].

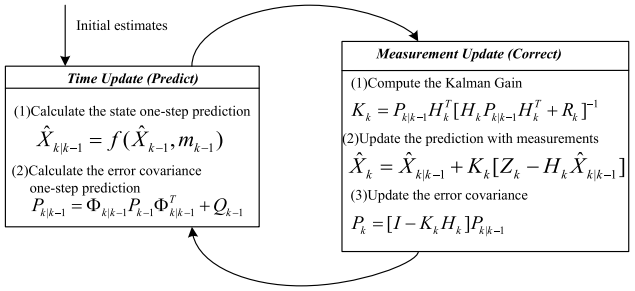


FIGURE 5. The flow diagram of EKF.

By using the first order of Taylor expansion of the nonlinear system model, EKF transfers the nonlinear problem into a linear form. The specific procedure can be divided into two parts: the predicted process and the update process. In the first place, according to the established system model, the one-step state prediction can be obtained. Upon receiving the observations, the update process will renovate the state estimate based on the distinction between the prediction and observation. Then the final state estimate can be acquired.

EKF is widely used in the actual AUV missions due to its feasibility and practicality. Nevertheless, since EKF ignores the high order terms of Taylor series, it will perform poorly and get low accuracy in a highly nonlinear situation.

As an improvement of EKF, UKF performs in a similar predict-update cycle. The difference between the EKF and UKF is that the UKF utilizes the Unscented Transform (UT) to get the statistical linearization rather than making use of linear truncation. Symmetrical sampling is used for UT to choose a set of sigma points. Then taking advantage of these selected points, a nonlinear function approximation which contains the higher order terms of the Taylor series can be obtained. Since the UKF retains the higher order terms which is ignored by the EKF, it can achieve a more accurate estimation of the state. However, some limitations such as approximate errors still exist with UKF. The specific flow chart of UKF is shown in Fig. 6 [39].

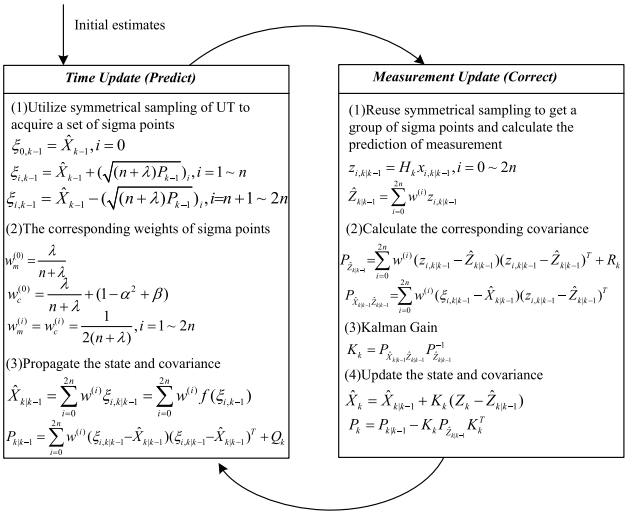


FIGURE 6. The flow diagram of UKF.

IV. AUV DEEP NAVIGATION FRAMEWORK

In order to avoid the possible errors imported by the modeling and estimating processes mentioned above, in this section, a deep sequential learning framework based on the deep neural networks — NavNet, is proposed for AUV navigation. The specific details of the proposed framework will be described later.

A. ARCHITECTURE OF THE PROPOSED DEEP FRAMEWORK

At present, numerous frameworks based on deep neural networks have achieved great success in many fields. For instance, both the AlexNet [40] and ResNet [41] play essential roles in the field of computer vision. Different from those computer vision problems which mainly focus on image processing, the AUV navigation and localization tasks aim to get the AUV position through processing the raw sensor data collected over time. Therefore, existing deep neural network frameworks are not suitable for AUV navigation and we propose a new deep framework — NavNet.

The overall architecture of NavNet is shown in Fig. 7. It consists of a multilayer RNNs module, a simplified attention module and a Fully Connected (FC) layers module. Based on the discussion of the traditional methods for AUV navigation, it is believed that the received data from navigation sensors can reflect all the possible situations of AUV position in the ocean. Therefore, in NavNet, the inputs are the collected AUV raw sensor data which are relevant to navigation. As described earlier, in the real world applications for Sailfish, the major related navigation sensors are AHRS and DVL. Moreover, all these sensor data can be regarded as time-series data to be processed. Hence, utilizing the sequential learning ability of deep RNNs, the raw sensor data of adjacent unit time are taken as input. Here, the per unit time is defined as one second. However, in consideration of the different principles of different sensors, the corresponding output frequencies differ from one another. For instance, for Sailfish, the AHRS is configured to be 10 Hz while DVL is only 2 Hz. Therefore, in order to deal with the multi-frequent input data, the proposed framework adopts two RNNs at different rates to perform time-series learning. More specifically, one RNN takes the AHRS data as input, which is a 9-dimensional vector which includes yaw, pitch, roll of the vehicle, the triaxial acceleration with respect to the body frame, and the corresponding triaxial angular velocity. Furthermore, the other RNN model takes the DVL data as input, which contains the 3-dimensional vector of the velocities: the forward, starboard and downward

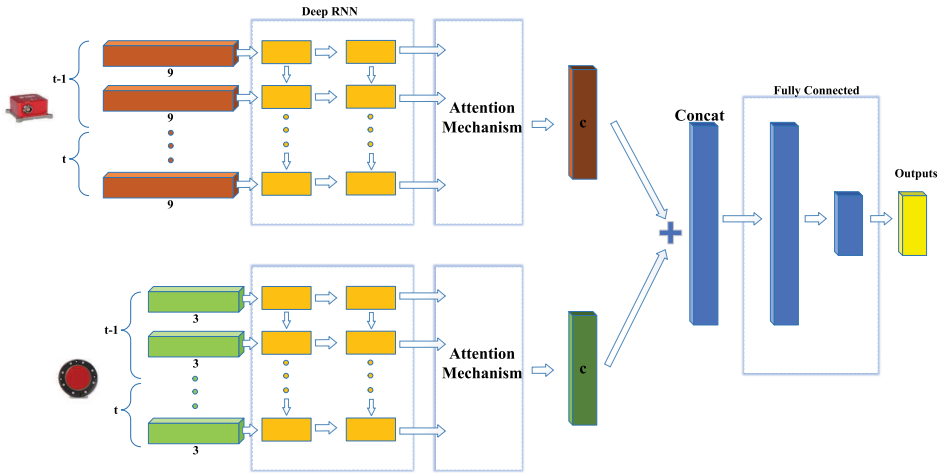


FIGURE 7. Whole architecture of NavNet.

velocity relative to the vehicle coordinate system. In order to capture the long-term time dependence between the input time-sequential data better, a simplified attention module [42] is deployed to process the outputs of RNNs. Then the connected vector which concatenates those two different context vectors from attention models is fed into FC layers and the outputs of the framework can be obtained. The output of the framework is a 2-dimensional vector and the element of each dimension illustrates the AUV displacement during one second in north-east coordinate system. The concrete description of each module will be introduced in the next part.

B. DESCRIPTION OF DIFFERENT PARTS FOR NavNet

1) RECURRENT NEURAL NETWORK

As mentioned above, for the proposed NavNet, the collected raw sensor data are considered as time-sequential data which are suitable for RNNs modelling. A typical unfolded RNN framework is illustrated in Fig. 8.

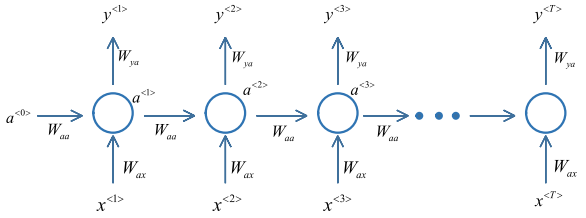


FIGURE 8. A typical unfolded RNN framework.

As shown above, the unfolded RNN can be identified as a deep feedforward network. For an input sequence with a length of T, each item of it which is described as $x^{<t>}$ will be processed in sequence. Meanwhile, a state vector $a^{<t>}$ is maintained in the hidden units, which contains all the information of the past. Therefore, RNN has the ability to keep the relation between the current item and the previous items. Then the output $y^{<t>}$ which relies on the information of all the previous moments can be obtained. It is important to note that all the layers share the same weights for RNN, that is to say, the same parameters W_{ax}, W_{aa} and W_{ya} are used in each time step. The forward propagation at time t is described as follows [43]:

$$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \tag{7}$$

$$y^{<t>} = h(W_{ya}a^{<t>} + b_y), \tag{8}$$

where b denotes the bias vectors, g and h respectively represent the nonlinear activation functions which generally use Rectified Linear Unit ($ReLU$) and $tanh$ function. Nevertheless, typical RNN will face lots of challenges in the practical applications. For example, it is difficult to memorize all the information of the sequences, especially for the very long ones. Moreover, since the deep feedforward network is trained with the Backpropagation Through Time (BPTT), there might be problems such as gradients explosion or vanishment with RNN after many time steps [44].

In order to avoid the problems of typical RNN, in our proposed framework, Long Short-Term Memory (LSTM) [45] is utilized to maintain the long-term dependencies. With the special hidden units, LSTM is capable of memorizing the inputs for a long time. The basic architecture of unfolded LSTM is depicted in Fig. 9.

By introducing the memory gates and memory cells, the framework can determine whether the previous hidden states will be reserved to update the current state or not. In Fig. 9, $c^{<t>}$ denotes the memory cell, $x^{<t>}$ represent the inputs, and $a^{<t-1>}$ are the previous hidden states. The specific computational process at time t is according to (9) to (14) [45]:

$$\tilde{c}^{<t>} = \tanh(w_c[a^{<t-1>}, x^{<t>}] + b_c) \tag{9}$$

$$\Gamma_u = \sigma(w_u[a^{<t-1>}, x^{<t>}] + b_u) \tag{10}$$

$$\Gamma_f = \sigma(w_f[a^{<t-1>}, x^{<t>}] + b_f) \tag{11}$$

$$\Gamma_o = \sigma(w_o[a^{<t-1>}, x^{<t>}] + b_o) \tag{12}$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>} \tag{13}$$

$$a^{<t>} = \Gamma_o * tanh(c^{<t>}), \tag{14}$$

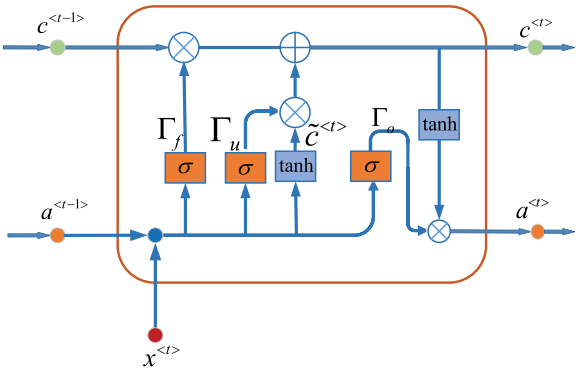


FIGURE 9. Basic architecture of unfolded LSTM.

where Γ_f, Γ_u and Γ_o respectively represent the forget gate, input gate and output gate which play important roles in LSTM. According to the characteristics of the sigmoid function, the maximum value of Γ_f is one and its minimum is limited to zero. If Γ_f equals to zero, it means that all the information of the inputs and the previous hidden states will be forgotten. On the other hand, if it is one, the whole information will be retained to perform the following computation. Otherwise, if Γ_f is a value between zero and one, the information will be memorized at the corresponding ratio. Then, Γ_u determines the information that needs to be updated. Afterwards, combining the input modulation $\tilde{c}^{<t>}$ with the previous memory cell $c^{<t-1>}$, the current memory cell $c^{<t>}$ can be obtained. In the end, Γ_o is integrated with $c^{<t>}$ to acquire the current hidden state $a^{<t>}$, and the results will be passed to the next time step.

In order to model accurately, as exhibited in Fig. 7, we stack two LSTM layers to increase the depth of the network. In the stacked network, the hidden state of an LSTM is fed into another one as input. In order to reach better final accuracy of the proposed model and save the calculating resources, the number of each hidden layer neurons of LSTM is set to be 100 according to the parameter tuning process.

2) SIMPLIFIED ATTENTION MECHANISM

In order to capture the long-term time dependence between the input time-sequential data better, a simplified attention mechanism [42] is applied to NavNet. The schematic diagram is shown in Fig. 10.

As shown in Fig. 10, $a^{<t>}$ represents the hidden state in each moment which is from the LSTM. All the hidden states are fed into a learnable function $g(a^{<t>})$, and then calculated with the *Softmax* function, the corresponding weightings α_t can be obtained. The specific formula is [42]:

$$e_t = g(a^{<t>}), \tag{15}$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{t=1}^{T} \exp(e_t)}, \tag{16}$$

where T denotes the length of input sequence. Furthermore, in the proposed framework, *tanh* function is selected as the learnable function $g(\cdot)$. After that, a context vector c is able

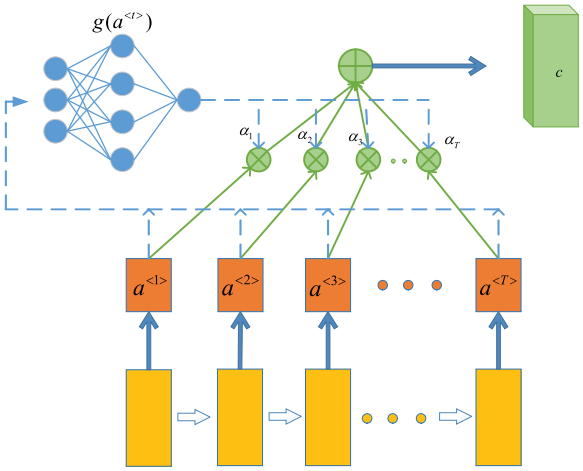


FIGURE 10. Schematic diagram of a simplified attention mechanism.

to be acquired with the weighted mean of the hidden states. The computation of c can be described as (17):

$$c = \sum_{t=1}^{T} \alpha_t a^{<t>}. \tag{17}$$

In a way, c can be seen as an adaptive weighted average of the hidden states, which is able to better memorize the information of a long-term input sequence. In the proposed framework, two context vectors from different sensor data can be concatenated to get a concat vector, and then this vector is fed into the FC layers to execute the subsequent calculation.

3) COST FUNCTION AND OPTIMIZER

At the end of the proposed NavNet, two FC layers are generated to gain a converged output. Then, the output of the framework is fed into a Euclidean loss layer to calculate an L2 Loss. Adam optimizer [46] proposed by combining the AdaGrad [47] and RMSProp [48], has the advantages of simple implementation, efficient computation and needs little memory requirements. Utilizing the estimates of the first and second moments of the gradients, it is able to calculate the individual adaptive learning rates for different parameters. In addition, the hyperparameters only need little tuning and it is verified that the default settings of the hyperparameters is useful for dealing with the majority of the machine learning problems. Therefore, in view of the outstanding performance as well as the prominent advantages of the Adam optimizer, it is employed in the proposed framework. Moreover, the default hyperparameters recommended by [46] are used for NavNet, and it means that the starting learning rate is 0.001, the exponential decay rates β_1, β_2 are 0.9 and 0.999, respectively.

4) THE ROBUSTNESS OF NavNet

It is worth mentioning that in the process of developing the deep framework, the robustness of the model is also taken into account. It is known that the sensor data collection is

susceptible to noises, which will affect the accuracy of the subsequent procedures. For Sailfish, there are the same problems with data collection by the equipped navigation sensors, AHRS and DVL.

In the ideal situation, since the AUV motion is steady during the mission, it is believed that the sensor data vary continuously without sudden change. Nevertheless, in fact, there are a lot of uncertainties in the sensor data collection process. In view of the differences in sensor principles and collection frequency between different sensors, the errors from DVL are more likely to have huge impacts on the performance of AUV navigation. For instance, the data collection of DVL is in a low frequency, which means that a wrong measurement will affect the position accuracy in a relatively long period of time and have a great impact on the final result.

In the proposed NavNet, the raw sensor data are regarded as time-series to perform the subsequent sequential learning. In order to improve the fault tolerance and enhance the robustness of the system, both the previous unit time observation and current unit time observation of the raw sensor data are considered as the input of the framework. As aforesaid, the adopted LSTMs have the ability to perform the learning in sequence, which signifies that the input measurements of all time steps will influence the output of current unit time. In conventional approaches, once there is an exception with the collected raw sensor data of a specific time step, it is bound to bring influence on the corresponding results. However, if a similar situation happened with the proposed methods, the sensor data of the adjacent time steps also make a difference to the final results, which play the role of a filter for the raw sensor data. Furthermore, the introduced attention mechanism further enhances the dependencies between different time steps and improves the ability of fault tolerance.

C. FUSION OF TRADITIONAL METHOD AND DEEP FRAMEWORK

Since the proposed NavNet can realize the low-frequency AUV navigation with the rate of only 1 Hz, we implement a fusion strategy of the conventional method and deep framework in this section. We take EKF as an example to describe the fusion strategy because of its high computational efficiency which is more suitable for real-time AUV navigation.

Because of the nonlinear error brought by EKF as well as the inherent characteristics of sensors, there is a drift in the estimation of AUV position existing over time. In general, this problem can be mitigated by resurfacing and using GPS to get a corrected position in the process of AUV missions. However, it is impractical for AUV to frequently resurface when it is carrying out various missions in the complex underwater environment.

Through the above mentioned analysis of the proposed NavNet, the deep framework outputs the corresponding displacements of the AUV during per unit time. As shown in Fig. 11, $(\delta x, \delta y)$ represent the displacements of the corresponding time interval. During the same time interval, different from other algorithms for state estimation, n times

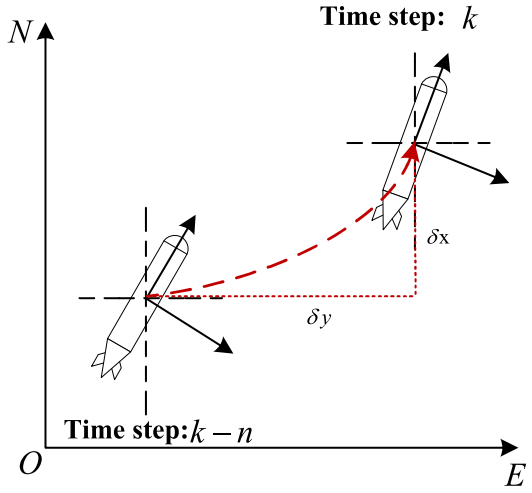


FIGURE 11. The displacements of the corresponding time interval.

iterations have already been performed by EKF-based algorithms. As previously mentioned, EKF-based algorithms perform the cycles of prediction and update to get the AUV position. In the update process, the sensor measurements are used to calculate the residual and correct the predict state. In section III-A we mentioned that the observation vector of EKF-based algorithms consists of angles, velocities and accelerations, and all these data are measured by the equipped AHRS and DVL. Since there is no usable extra position sensor in AUV, the position is not taken into account in the observation vector.

As shown in Fig. 11, $(\delta x, \delta y)$ from NavNet can express the displacements of the unit time interval. These two values can be combined with the estimated position state of EKF at n time step ago to obtain a approximated AUV position at time step k. The concrete accumulation process is shown as:

$$x = \hat{X}_{k-n}(1) + \delta x \tag{18}$$

$$y = \hat{X}_{k-n}(2) + \delta y \tag{19}$$

Here, consistent with the foregoing, $\hat{X}_{k-n}$ represents the estimated system state of $k - n$ time step and the first two items respectively denotes the position in the north and east orientation. x and y signify the estimated AUV position at time step k which can be regarded as the measurements of a virtual position sensor. Therefore, at the corresponding time step k, the observation vector can be augmented by AUV position, which is shown as:

$$Z_k = [x, y, \varphi, v_x, v_y, a_x, a_y, w_z]^T \tag{20}$$

The representations of specific quantities are the same as above. Meanwhile, as shown in (21), the observation matrix needs to be augmented accordingly to perform the correct stage.

$$H = I_{8\times 8} \tag{21}$$

During the time step $k - n$ to k, the standard EKF is performed to maintain a high-frequency AUV navigation.

Through introducing the virtual position measurements, the drifts caused by conventional methods and sensor characteristics can be reduced without equipping with any additional sensors. Combined with conventional methods, the low-frequency displacements of NavNet are used to limit the error growth of the traditional algorithm while the high-frequency navigation can be guaranteed at the same time. The concrete realizing process is described in Algorithm 1. This strategy does not require any additional sensors or rising to the surface, which will become a feasible and reliable method for AUV missions.

Algorithm 1 A Fusion Strategy of Intelligent Navigation Framework With EKF

Initialize $\hat{X}_0, P_0$;
for $k = 1, \ldots$ **do**
  {Time update}
  %%%Propagate the state and covariance%%%
  $\hat{X}_{k|k-1} = f(\hat{X}_{k-1}) + w_k$;
  $P_{k|k-1} = \Phi_{k|k-1} P_{k-1} \Phi_{k|k-1}^T + Q_{k-1}$;
  {Measurement update}
  if Deep framework outputs a result **then**
    %%%Integration with the output of deep navigation framework%%%
    $x = \hat{X}_{k-n}(1) + \delta x$;
    $y = \hat{X}_{k-n}(2) + \delta y$;
    %%%Observation vector%%%
    $Z_k = [x, y, \varphi, v_x, v_y, a_x, a_y, w_z]^T$;
    %%%Observation matrix%%%
    $H = I_{8\times 8}$;
    %%%Update the state and covariance%%%
    $K_k = P_{k|k-1} H_k^T [H_k P_{k|k-1} H_k^T + R_k]^{-1}$;
    $\hat{X}_k = \hat{X}_{k|k-1} + K_k [Z_k - H_k \hat{X}_{k|k-1}]$;
    $P_k = [I - K_k H_k] P_{k|k-1}$;
  else
    %%%Observation vector%%%
    $Z_k = [\varphi, v_x, v_y, a_x, a_y, w_z]^T$;
    %%%Observation matrix%%%
    $H = [\, 0_{6\times 2} \; I_{6\times 6} \,]$;
    %%%Update the state and covariance%%%
    $K_k = P_{k|k-1} H_k^T [H_k P_{k|k-1} H_k^T + R_k]^{-1}$;
    $\hat{X}_k = \hat{X}_{k|k-1} + K_k [Z_k - H_k \hat{X}_{k|k-1}]$;
    $P_k = [I - K_k H_k] P_{k|k-1}$;
  end if
end for

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present and discuss the experimental results.

A. DATASET

We tested our proposed algorithm with Sailfish 210 in an experiment in the sea. The data is collected at Nanjiang wharf, Qingdao in April and May 2019 as the dataset to perform the training and testing process. The recovery of Sailfish 210 is shown in Fig. 12. Since the GPS is available when AUV is carrying out tasks on the surface of the water, we collected the exact location of AUV with GPS as the ground truth. Meanwhile, the displacement obtained by GPS during per unit time is used as the training label to calculate the loss with the output of proposed framework.

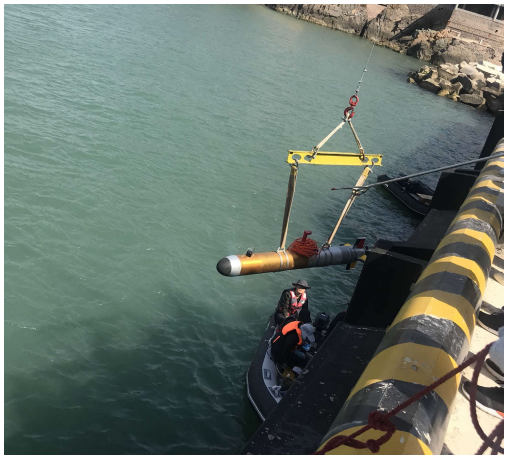


FIGURE 12. The recovery of Sailfish 210 at Nanjiang wharf.

The data are divided into three parts: training set, validation dataset, and testing set. Different trajectories of AUV missions are segmented to different lengths to generate several datasets, which include multiple circles, triangles, rectangles, pentagons, straight lines with different orientations and other mission tracks. After pruning the data, 5356 samples in total remained and are used as the training set to train the deep framework. The validation dataset is made up of multiple whole single trajectories. Moreover, several groups of trajectories representing the typical real AUV missions, such as single circle or single pentagon trajectories, are selected as the testing set. We will describe in detail in the following section.

B. EXPERIMENTAL RESULTS

1) RESULTS OF NavNet

In order to verify the effectiveness of the proposed deep framework, we chose four cases represented by four groups of datasets for the typical trajectories of the real AUV missions. In addition, in order to use the GPS trajectories as ground truth to verify the accuracy of the proposed algorithm, the AUV on-surface data are used to test the model. The estimated trajectory of the proposed method — NavNet, is compared to that of EKF, UKF and the ground truth, as shown in Fig. 13, 14, 15 and 16. Meanwhile, the single-step position error of each case which is calculated by comparing the trajectories of NavNet, EKF and UKF with GPS trajectories is also shown in the figures. The boxplots of the single-step position error are also shown for comparison of different algorithms. And the overall position Root Mean Square Error (RMSE) of four cases is shown in Table 5.

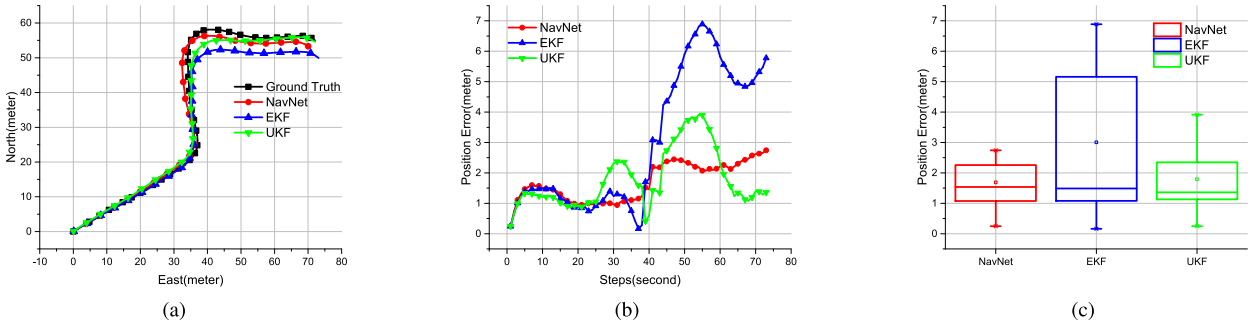


FIGURE 13. The motion trajectory results and position error of Case 1.

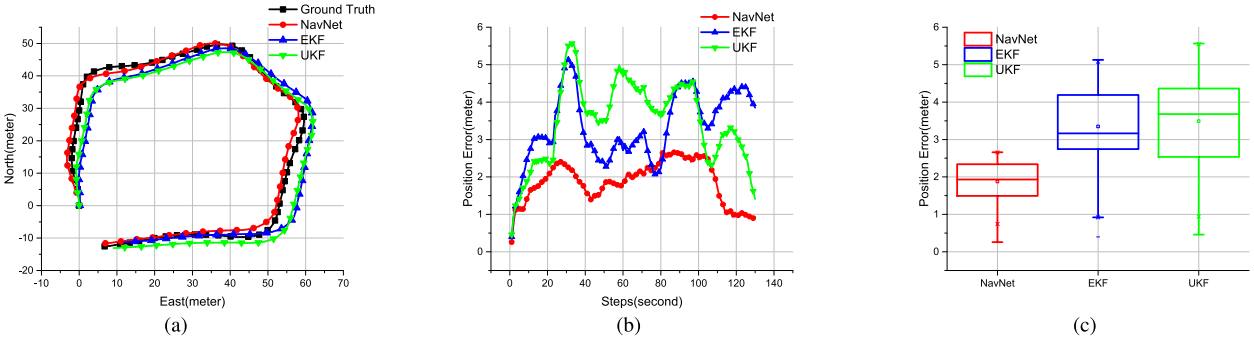


FIGURE 14. The motion trajectory results and position error of Case 2.

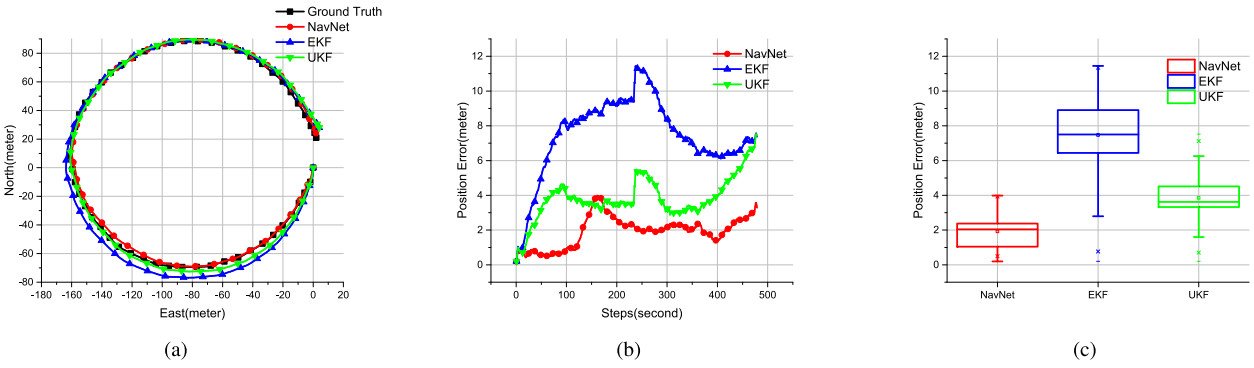


FIGURE 15. The motion trajectory results and position error of Case 3.

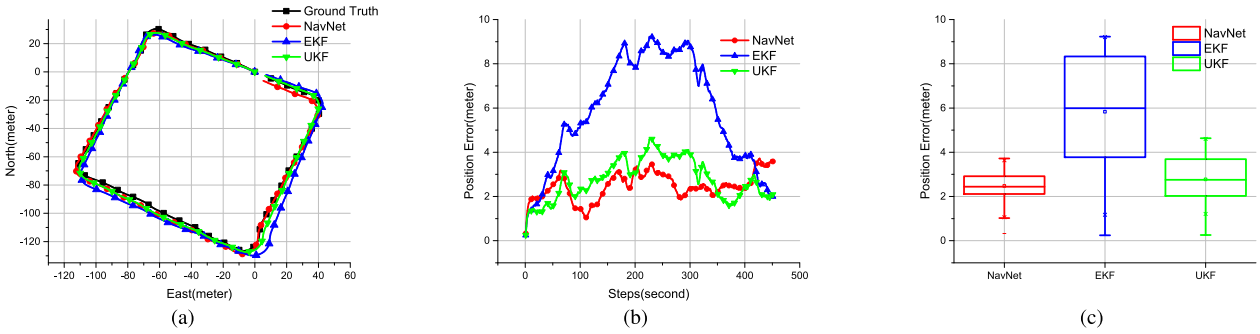


FIGURE 16. The motion trajectory results and position error of Case 4.

Fig. 13 shows the estimated motion trajectory of Case 1, with a length of about 108 meters and 73 seconds in time. As we can see from Fig. 13, the EKF algorithm has the worst performance for AUV navigation, which is furthest away from the GPS trajectory. In addition, the proposed deep framework has a comparative navigation capability with the UKF algorithm, which can also be seen in Table 5. However, the RMSE of NavNet is the smallest while UKF has a minimum value of end point position error.

TABLE 5. Overall position RMSE and end point position error of Case 1 to Case 4.

Case	Criterion	NavNet	EKF	UKF
Case 1	Overall position RMSE (m)	1.7984	3.8664	2.0886
	End point position error (m)	2.7435	5.7769	1.3591
Case 2	Overall position RMSE (m)	1.9538	3.6520	3.5215
	End point position error (m)	0.9324	3.8534	1.4077
Case 3	Overall position RMSE (m)	2.1145	8.1301	4.3910
	End point position error (m)	3.2894	7.3772	7.4267
Case 4	Overall position RMSE (m)	2.5404	6.0237	2.6439
	End point position error (m)	3.5220	1.9393	2.0625

In case 2, AUV performs a pentagon task track with the length of side around 40 meters, and the estimated motion trajectory is shown in Fig. 14. The total length of Case 2 is approximately 184 meters and the running time is 130 seconds. In this case, the performance of EKF and UKF is similar, which is worse than the performance of NavNet. Table 5 also shows the effectiveness of the deep framework, which has the smallest RMSE.

Fig. 15 shows the estimated trajectory of Case 3, which is a circular track with a radius of 80 meters. The total running distance is around 491 m with the running time of 479 seconds. The results in this case is similar to that in in Case 2, with the proposed NavNet performing the best. Moreover, NavNet has a smaller overall position RMSE and position error than the other two algorithms.

The track of Case 4 is square with its side length of 120 meters and the gross cruising distance is 468 meters. The voyage takes 452 seconds and the estimated trajectory is shown in Fig. 16. The performance of AUV navigation with NavNet is similar to UKF while the RMSE of NavNet is a bit smaller. Both of them are better than the performance of EKF. In terms of the position error, the performance of NavNet is also similar to that of UKF. The position error of EKF is much larger than the other two algorithms in the task, but the end point position error is the smallest. At the same time, the computation time of four cases is shown in Table 6. As seen from Table 6, the proposed NavNet has a shorter computation time compared to EKF and UKF.

TABLE 6. Computation time of Case 1 to Case 4.

Case	Criterion	NavNet	EKF	UKF
Case 1	Computation time (ms)	22.323	38.553	102.396
Case 2	Computation time (ms)	26.281	54.712	154.881
Case 3	Computation time (ms)	41.548	145.75	500.398
Case 4	Computation time (ms)	38.103	136.129	481.867

Overall, our experimental results show that the proposed deep framework can realize a reliable AUV navigation and localization, and has an equivalent or better performance than UKF, which is better than EKF.

2) FAULT TOLERANCE

We also take the sensor errors into account for designing the proposed deep framework, especially the DVL low-frequency errors. In this section, we test and verify the fault tolerance ability of the the proposed framework.

We consider two cases to test the fault tolerance of NavNet, in consideration of the running environment of Sailfish and the characteristics of sensors. The results are shown in Fig. 17 and Fig. 18.

Case 5 is a track with a total length of 175 meters and the running time of 125 seconds. In the task, limited by the property of AUV, the forward velocity of DVL is always between 1 m/s to 2.5 m/s. To simulate the situation that DVL acquired incorrect data, we inserted a wrong forward velocity of DVL—10 m/s — at the time of 49 seconds, as shown in Fig. 17(a). Fig. 17(d) shows the estimated trajectory and Fig. 17(d) shows the corresponding position error. In order to show the results more clearly, the region that was influenced by the wrong data is locally enlarged in original dealing frequency of each algorithm, as shown in Fig. 17(c) and Fig. 17(e). From Fig. 17 we can obviously seen that, both EKF and UKF are badly affected by the wrong data at the corresponding time step. However, the AUV navigation with the proposed framework is unaffected and achieves a higher navigation accuracy. Moreover, in terms of position error, we found that the position error of EKF and UKF increases rapidly at the time of inserting the wrong data, while the position error of NavNet has a mild increase.

The application scenarios of Sailfish are mainly the shallow sea conditions which means that the operating distance of DVL is always within the sensor threshold. Generally speaking, under these circumstances, the DVL is always available for AUV. Nevertheless, in the complex underwater environments, some unforeseeable cases may happen to affect the validity of DVL. For instance, when the seafloor is covered with strong sound absorption materials such as sludge, or the AUV encounters larger pitch attitude and roll attitude, the transmitted acoustic pulses of DVL cannot be reflected back, which will make the DVL measurements failure temporarily. Therefore, in Case 6, we focus on the invalid circumstance of DVL. In Case 6, the AUV is running for 96 seconds and a total length of 140 meters. The invalid data of DVL forward velocity is interpolated at the 80th time step, as shown in Fig. 18(a). Fig. 18(b) shows the estimated trajectories of NavNet, EKF, UKF and the ground truth. The region that was influenced by the invalid data is also locally enlarged, as shown in Fig. 18(c). From Fig. 18(c) we can see that, the tracking trajectories of EKF and UKF have a tiny turn at the time of interpolating the invalid data. However, the performance of NavNet is unaffected. Nevertheless, since the speed of AUV is small, the invalid data does not cause fatal effect for conventional algorithms. In this case, the proposed framework also achieves a better performance. The corresponding RMSE and end point position error of Case 5 and 6 are shown in Table 7. Together with results in Table 7, we can see that our proposed NavNet can improve the navigation accuracy and has good fault tolerance capacity at the same time. The computation time of these two cases is shown in Table 8, and the proposed NavNet has a shorter running time than others.

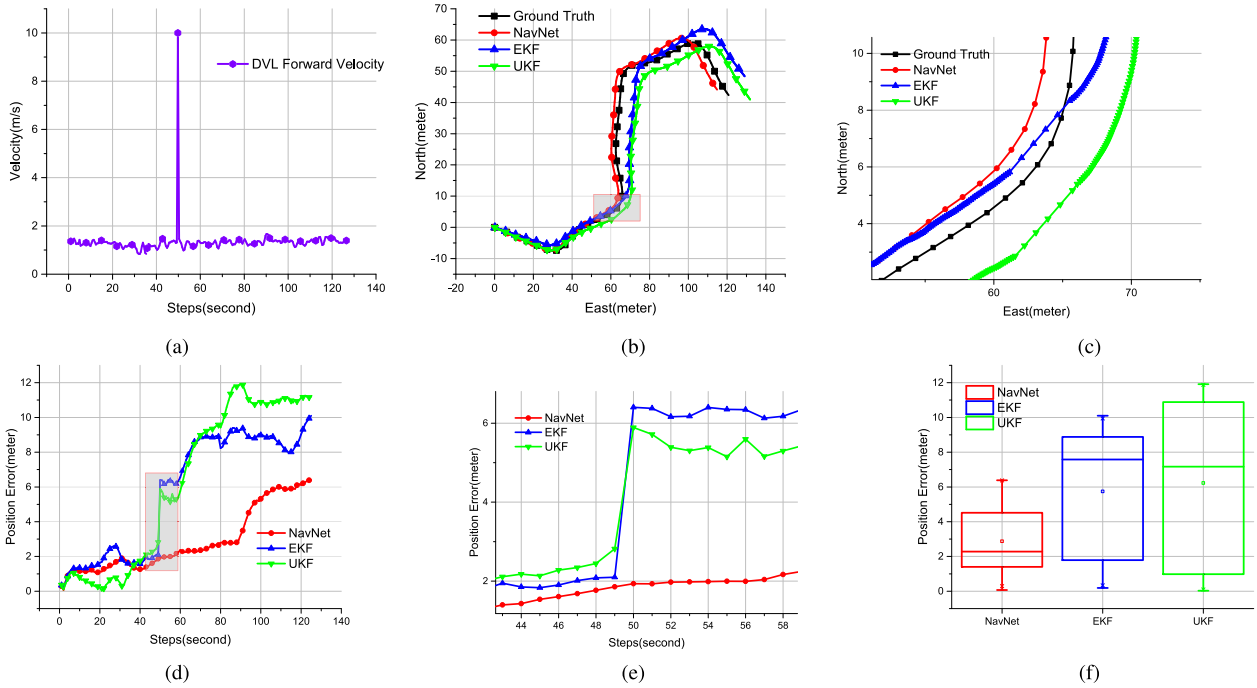


FIGURE 17. Corresponding results of Case 5. (a) represents the forward velocity of DVL, (b) and (d) are respectively the trajectory and position error results. The enlargement is the region of the red boxes. (f) is the boxplot of the position error.

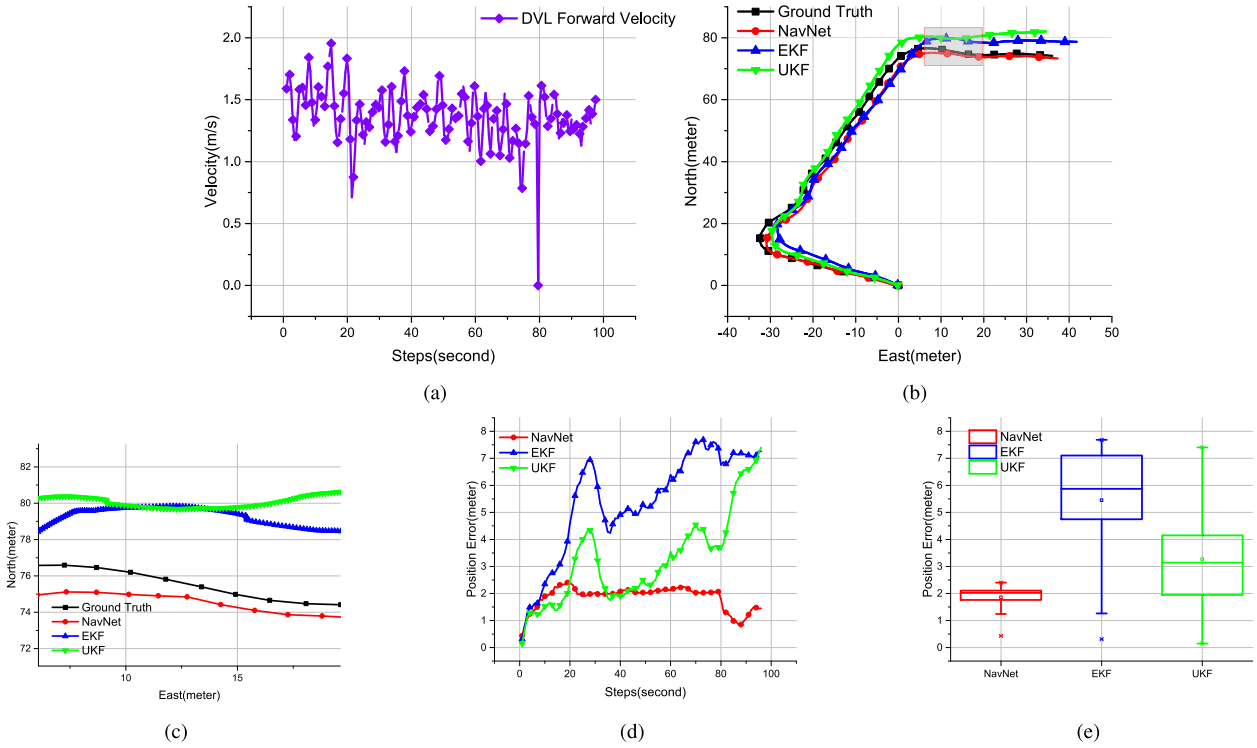


FIGURE 18. Corresponding results of Case 6. (a) represents the forward velocity of DVL, (b) and (d) are respectively the trajectory and position error results. The enlargement is the region of the red boxes. (e) is the boxplot of the position error.

3) COMBINING TRADITIONAL METHOD WITH DEEP FRAMEWORK

The experimental results of combining the proposed framework with EKF are shown in Fig. 19, 20 and 21. The corresponding evaluation criterions are described in Table 9. Fig. 19(a), 20(a) and 21(a) shows the estimated trajectories of the fusion strategy, EKF and the ground-truth. In addition, the yaw, forward velocity, and starboard velocity for the fusion strategy, EKF and the raw sensory data are also shown in Fig. 19, 20 and 21, respectively.

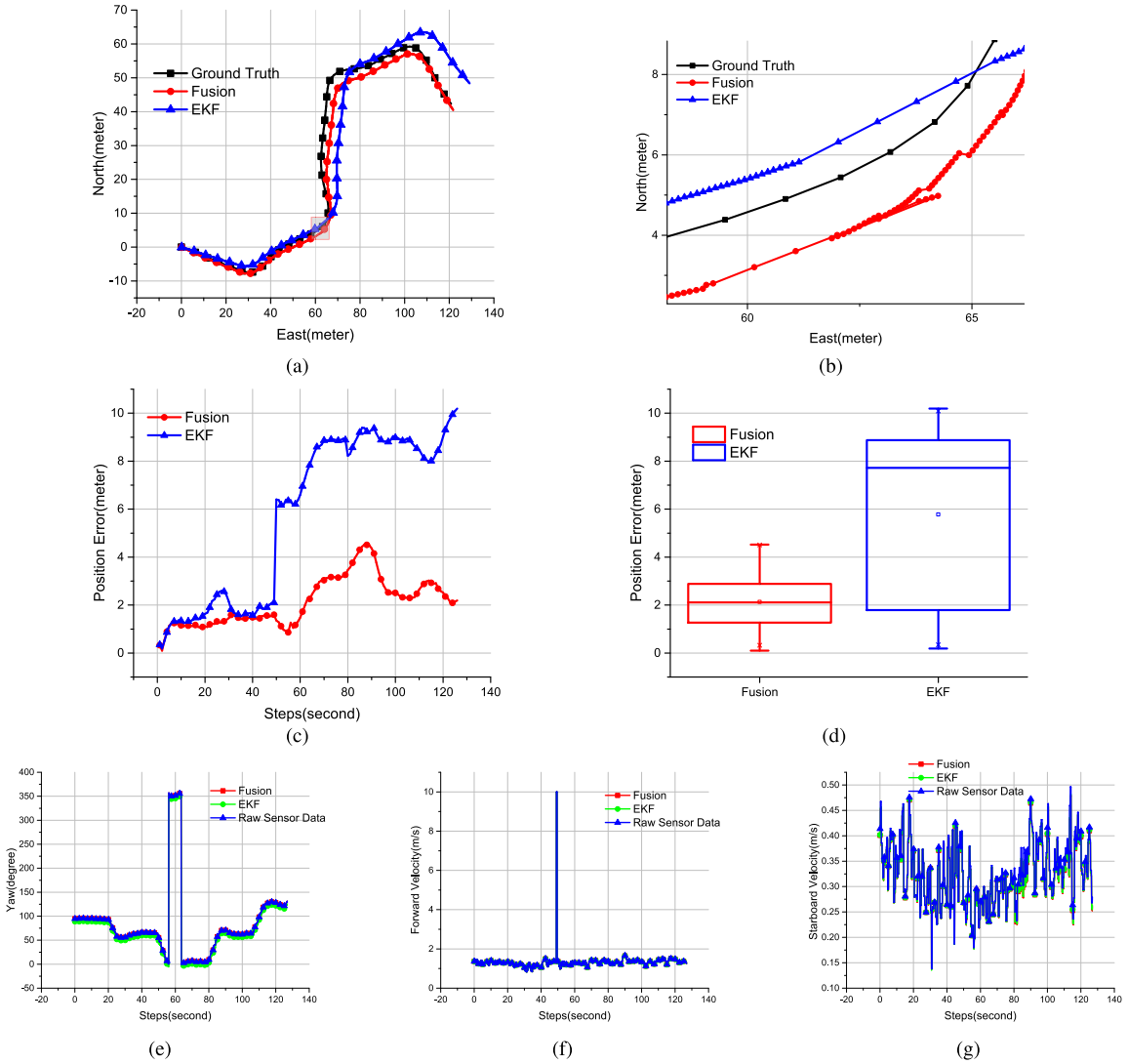


FIGURE 19. The trajectory and position error results of Case 5. The enlargement is the region of the red boxes.

TABLE 7. Overall position RMSE and end point position error of Case 5 to Case 6.

Case	Criterion	NavNet	EKF	UKF
Case 5	Overall position RMSE (m)	3.4035	7.4135	6.4151
	End point position error (m)	6.3783	10.0972	11.1906
Case 6	Overall position RMSE (m)	1.9032	5.3554	3.6349
	End point position error (m)	1.4532	7.2586	7.4049

TABLE 8. Computation time of Case 5 to Case 6.

Case	Criterion	NavNet	EKF	UKF
Case 5	Computation time (ms)	25.367	56.547	159.793
Case 6	Computation time (ms)	23.848	44.622	130.29

Firstly, we study the fusion outcome in Case 5 where a wrong forward velocity data was artificially inserted at the time of 49 seconds. From the the enlarged Fig. 19(b) we can see that the trajectory of fusion strategy were drifted at the time of inserting the wrong data. However, in the next moment when the NavNet output arrives, the position of the fusion strategy has been limited and pulled back. Finally, the fusion strategy achieves a better accuracy than EKF. This can also be shown by the results in Table 9 in terms of RMSE and end point position error.

TABLE 9. Overall position RMSE and end point position error of corresponding Cases.

Case	Criterion	Fusion	EKF
Case 5	Overall position RMSE (m)	2.4671	6.4151
	End point position error (m)	2.1943	10.1910
Case 2	Overall position RMSE (m)	1.7287	3.6520
	End point position error (m)	1.7226	3.7616
Case 7	Overall position RMSE (m)	2.2314	5.3999
	End point position error (m)	5.5220	9.7324

Case 2 is also used to demonstrate the superior performance of the fusion strategy. The trajectory of fusion method is closer to the ground truth and the accuracy of EKF has been greatly increased. Case 7 is a track with a total length of 194 meters and running time of 126 seconds. The experimental results in Case 7 are similar to those in

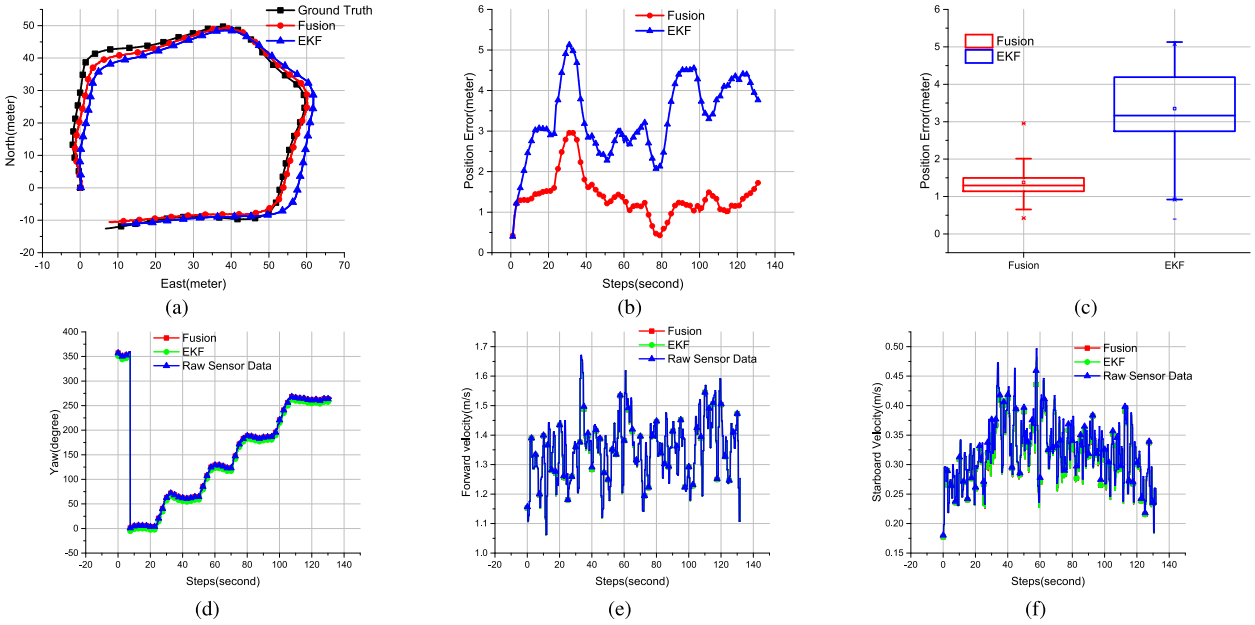


FIGURE 20. The trajectory and position error results of Case 2.

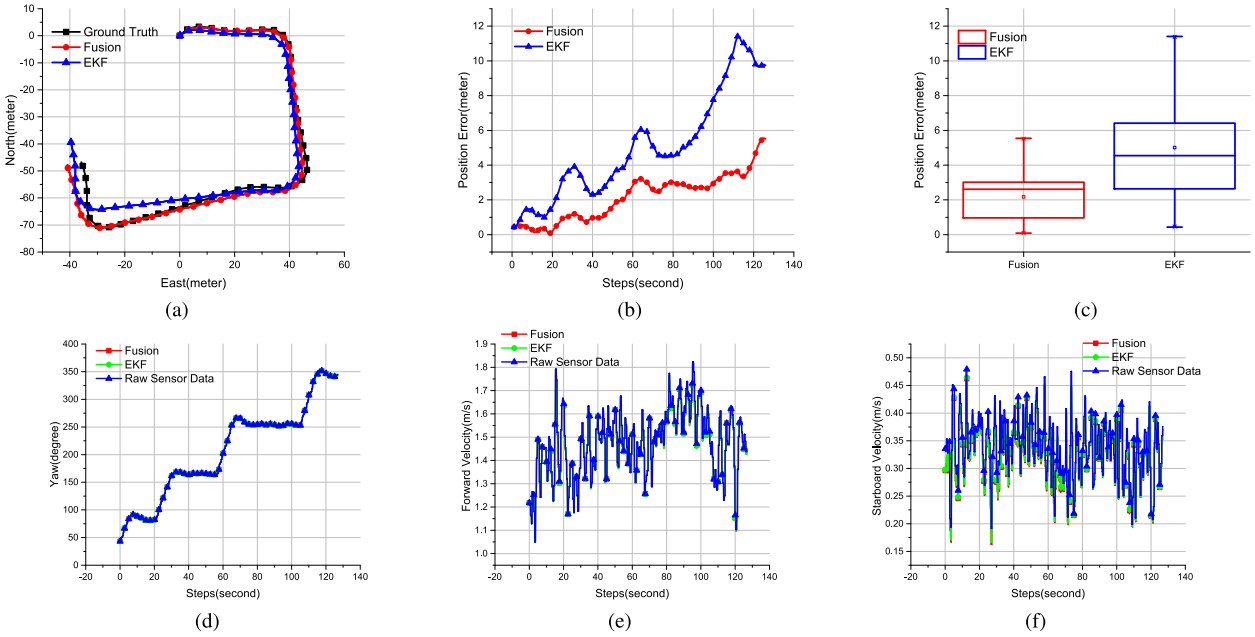


FIGURE 21. The trajectory and position error results of Case 7.

Case 2. In this case, the performance of EKF is unsatisfactory but the fusion method has an excellent performance. Table 9 also shows the advantage of the fusion strategy in terms of the overall position RMSE and end point position error. The computation time is shown in Table 10. Because the fusion method combines the proposed NavNet with EKF, its running time is longer than EKF.

From the above results we can see that, the fusion strategy not only achieves high-frequency navigation but also improves the navigation accuracy to a large extent. Meanwhile, the fusion strategy has a good tolerance of fault data and can limit the error influence. Therefore, the proposed deep framework can be used as a reliable supplement for conventional algorithms to obtain a more robust AUV navigation system.

TABLE 10. Computation time of corresponding Cases.

Case	Criterion	Fusion	EKF
Case 5	Computation time (ms)	71.828	42.19
Case 2	Computation time (ms)	73.445	48.005
Case 7	Computation time (ms)	72.22	45.963

VI. CONCLUSION

In this paper, we proposed a deep framework — NavNet — by taking AUV navigation as a deep sequential learning problem. Different from traditional navigation algorithms, such as Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF), which require the system model and measurement model for state estimation to obtain the AUV position, NavNet does not need the model building. Therefore, it can avoid the import of modeling errors and state estimation errors. We compare the performance of NavNet to EKF and UKF using collected data by running Sailfish in the sea. Experimental results show that NavNet has an excellent performance in terms of both the navigation accuracy and fault tolerance. In addition, a reliable fusion strategy of NavNet and conventional method is applied to achieve high-frequency AUV navigation. The experimental results show that the proposed architecture can be a reliable supplement to limit the error growth of conventional algorithms.

In the future, we would like to investigate the generalization of our method to different and complex underwater environments. For example, data from more sensors can be augmented to sense the surrounding environments in a more precise way so as to improve the generalization ability of the proposed framework. In addition, we will focus on the real-time application of the fusion strategy and take the system uncertainty into account to improve the performance of the framework.

REFERENCES

[1] R. B. Wynn, V. A. I. Huvenne, T. P. Le Bas, B. J. Murton, D. P. Connelly, B. J. Bett, H. A. Ruhl, K. J. Morris, J. Peakall, D. R. Parsons, E. J. Sumner, S. E. Darby, R. M. Dorrell, and J. E. Hunt, "Autonomous underwater vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience," *Mar. Geol.*, vol. 352, pp. 451–468, Jun. 2014, doi: 10.1016/j.margeo.2014.03.012.

[2] M. Purcell, D. Gallo, G. Packard, M. Dennett, M. Rothenbeck, A. Sherrell, and S. Pascaud, "Use of REMUS 6000 AUVs in the search for the air France flight 447," in *Proc. OCEANS MTS/IEEE KONA*, Waikoloa, HI, USA, Sep. 2011, pp. 1–7.

[3] S. Panzieri, F. Pascucci, and G. Ulivi, "An outdoor navigation system using GPS and inertial platform," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 2, pp. 134–142, Jun. 2002, doi: 10.1109/TMECH.2002.1011250.

[4] B. Sinopoli, M. Micheli, and G. Donato, "Vision based navigation for an unmanned aerial vehicle," in *Proc. ICRA*, Seoul, South Korea, Oct. 2001, pp. 1757–1764.

[5] 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. 2014, doi: 10.1109/JOE.2013.2278891.

[6] J. J. Leonard and A. Bahr, "Autonomous underwater vehicle navigation," in *Springer Handbook of Ocean Engineering*. Cham, Switzerland: Springer, 2016, pp. 341–358, doi: 10.1007/978-3-319-16649-0_14.

[7] Y. Watanabe, H. Ochi, and T. Shimura, "A tracking of AUV with integration of SSBL acoustic positioning and transmitted INS data," in *Proc. OCEAN EUROPE*, Bremen, Germany, May 2009, pp. 1–6.

[8] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms," in *Proc. OCEANS MTS/IEEE*, Washington, DC, USA, Sep. 2005, pp. 1086–1092.

[9] F. C. Teixeira and A. M. Pascoal, "Geophysical navigation of autonomous underwater vehicles using geomagnetic information," *IFAC Proc. Volumes*, vol. 41, no. 1, pp. 178–183, 2008, doi: 10.3182/20080408-3-IE-4914.00032.

[10] D. Titterton, J. L. Weston, and J. Weston, "Strapdown inertial navigation technology," *IET*, vol. 17, 2004, doi: 10.1049/PBRA017E.

[11] N. Y. Ko and S. Jeong, "Attitude estimation and DVL based navigation using low-cost MEMS AHRS for UUVs," in *Proc. URAI*, Washington, DC, USA, Nov. 2014, pp. 605–607.

[12] M. T. Sabet, H. M. Daniali, A. Fathi, and E. Alizadeh, "A low-cost dead reckoning navigation system for an AUV using a robust AHRS: Design and experimental analysis," *IEEE J. Ocean. Eng.*, vol. 43, no. 4, pp. 927–939, Oct. 2018, doi: 10.1109/JOE.2017.2769838.

[13] R. Martins, J. B. de Sousa, C. C. Afonso, and M. L. Incze, "REP10 AUV: Shallow water operations with heterogeneous autonomous vehicles," in *Proc. OCEANS IEEE-Spain*, Jun. 2011, pp. 1–6.

[14] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. Comput. Sci., Univ. North Carolina Chapel Hill, Chapel Hill, NC, USA, Tech. Rep. TR95041, 1995, doi: 10.1145/800233.807054.

[15] A. Alcocer, P. Oliveira, and A. Pascoal, "Study and implementation of an EKF GIB-based underwater positioning system," *Control Eng. Pract.*, vol. 15, no. 6, pp. 689–701, Jun. 2007, doi: 10.1016/j.conengprac.2006.04.001.

[16] A. Mallios, P. Ridao, D. Ribas, F. Maurelli, and Y. Petillot, "EKF-SLAM for AUV navigation under probabilistic sonar scan-matching," in *Proc. IROS*, Taipei, Taiwan, Oct. 2010, pp. 4404–4411.

[17] B. Allotta, L. Chisci, R. Costanzi, F. Fanelli, C. Fantacci, E. Meli, A. Ridolfi, A. Caiti, F. Di Corato, and D. Fenucci, "A comparison between EKF-based and UKF-based navigation algorithms for AUVs localization," in *Proc. OCEANS*, Genoa, Italy, May 2015, pp. 1–5.

[18] M. Barisic, A. Vasilijevic, and D. Nad, "Sigma-point unscented Kalman filter used for AUV navigation," in *Proc. MED*, Barcelona, Spain, Jul. 2012, pp. 1365–1372.

[19] J. Melo and A. Matos, "On the use of particle filters for terrain based navigation of sensor-limited AUVs," in *Proc. MTS/IEEE OCEANS*, Bergen, Norway, Jun. 2013, pp. 1–8.

[20] F. Maurelli, S. Krupinski, Y. Petillot, and J. Salvi, "A particle filter approach for AUV localization," in *Proc. OCEANS*, Quebec City, QC, Canada, Sep. 2008, pp. 1–7.

[21] D. Ribas, P. Ridao, J. Neira, and J. Tardos, "SLAM using an imaging sonar for partially structured underwater environments," in *Proc. IROS*, Beijing, China, Oct. 2006, pp. 5040–5045.

[22] B. He, Y. Liang, X. Feng, R. Nian, T. Yan, M. Li, and S. Zhang, "AUV SLAM and experiments using a mechanical scanning forward-looking sonar," *Sensors*, vol. 12, no. 7, pp. 9386–9410, Jul. 2012, doi: 10.3390/s120709386.

[23] D. Forouher, J. Hartmann, M. Litza, and E. Maehle, "Sonar-based FastSLAM in an underwater environment using walls as features," in *Proc. ICRA*, Tallinn, Estonia, Oct. 2011, pp. 588–593.

[24] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "HCP: A flexible CNN framework for multi-label image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1901–1907, Sep. 2016, doi: 10.1109/TPAMI.2015.2491929.

[25] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware CNN model," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1134–1142.

[26] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, Vancouver, BC, Canada, Oct. 2013, pp. 6645–6649.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[28] V. Mohanty, S. Agrawal, S. Datta, A. Ghosh, V. Dutt Sharma, and D. Chakravarty, "DeepVO: A deep learning approach for monocular visual odometry," 2016, *arXiv:1611.06069*. [Online]. Available: http://arxiv.org/abs/1611.06069

[29] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. ICRA*, Singapore, May 2017, pp. 2043–2050.

[30] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. CVPR*, Providence, RI, USA, Jun. 2012, pp. 3354–3361.

[31] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 513–542, Apr. 2018, doi: 10.1177/0278364917734298.

[32] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular visual odometry through unsupervised deep learning," in *Proc. ICRA*, Brisbane, QLD, Australia, May 2018, pp. 7286–7291.

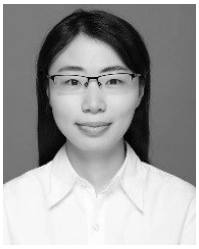
[33] R. Clark, S. Wang, and H. Wen, "VINet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Proc. AAAI*, Feb. 2017, pp. 3995–4001.
[34] M. Turan, Y. Almalioglu, and H. Araujo, "Deep EndoVO: A recurrent convolutional neural network (RCNN) based visual odometry approach for endoscopic capsule robots," *Neurocomputing*, vol. 275, pp. 1861–1870, Jan. 2018, doi: 10.1016/j.neucom.2017.10.014.
[35] R. Skulstad, G. Li, T. I. Fossen, B. Vik, and H. Zhang, "Dead reckoning of dynamically positioned ships: Using an efficient recurrent neural network," *IEEE Robot. Autom. Mag.*, vol. 26, no. 3, pp. 39–51, Sep. 2019, doi: 10.1109/MRA.2019.2918125.
[36] H. Tang, Y. Yin, and H. Shen, "A model for vessel trajectory prediction based on long short-term memory neural network," *J. Mar. Eng. Technol.*, vol. 18, pp. 1–10, Sep. 2019, doi: 10.1080/20464177.2019.1665258.
[37] D. Ribas, P. Ridao, and J. Neira, *Underwater SLAM for Structured Environments Using an Imaging Sonar*, vol. 65. Berlin, Germany: Springer, 2010, doi: 10.1007/978-3-642-14040-2.
[38] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Chelmsford, MA, USA: Courier Corporation, 2007.
[39] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Proc. SPIE*, vol. 3068, pp. 182–193, Jul. 1997, doi: 10.1117/12.280797.
[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 275, 2012, pp. 1861–1870, doi: 10.1145/3065386.
[41] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
[42] C. Raffel and D. P. W. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," 2015, *arXiv:1512.08756*. [Online]. Available: http://arxiv.org/abs/1512.08756
[43] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, Dec. 2015.
[44] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.
[45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[47] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.
[48] T. Tieleman and G. Hinton, "Lecture 6.5-RMSprop, coursera: Neural networks for machine learning," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 4, 26-31, 2012.

XIN ZHANG received the B.S. degree from the Department of Electronic Engineering, Ocean University of China, in 2016, where she is currently pursuing the Ph.D. degree. Her research interests include AUV navigation and vision-aided inertial navigation.

BO HE (Member, IEEE) was born in Qingdao, Shandong, China, in 1971. He received the M.S. and Ph.D. degrees from the Harbin Institute of Technology, China, in 1996 and 1999, respectively. From 2000 to 2003, he was a Postdoctoral Fellow with Nanyang Technological University, Singapore, where he worked on mobile robots, unmanned vehicles, research works included precise navigation, control, and communication. In 2004, he joined the Ocean University of China (OUC), where he is currently a Full Professor of OUC and the Deputy Head of the Department of Electronics Engineering, College of Information Science and Engineering. His current research interests include AUV design and applications, AUV SLAM, AUV control, and machine learning.

GUANGLIANG LI (Member, IEEE) received the bachelor's and M.Sc. degrees from the School of Control Science and Engineering, Shandong University, in 2008 and 2011, respectively, and the Ph.D. degree in computer science from the University of Amsterdam, The Netherlands, in 2016. He was a Visiting Researcher with the Delft University of Technology, The Netherlands, and a Research Intern with Honda Research Institute Japan Company Ltd., Japan. He is currently a Lecturer with the Ocean University of China. His research interests include reinforcement learning, human agent/robot interaction, and robotics.

XIAOKAI MU received the M.S. degree from the School of Physics and Electronic-Electrical Engineering, Ningxia University, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Ocean University of China. His research interests include integrated navigation and SLAM.



YING ZHOU received the B.S. degree from the School of Communication Engineering, Qingdao Agriculture University, in 2018. She is currently pursuing the master's degree with the Department of Electronic Engineering, Ocean University of China. Her research interests include acoustics navigation and AUV navigation.

TANJI MANG received the B.S. degree from the School of Information Engineering, Shanghai Maritime University, in 2018. She is currently pursuing the master's degree with the Department of Electronic Engineering, Ocean University of China. Her research interests include acoustic navigation and AUV navigation.