

Set-Transformer BeamsNet for AUV Velocity Forecasting in Complete DVL Outage Scenarios

Nadav Cohen*, Zeev Yampolsky and Itzik Klein

*The Hatter Department of Marine Technologies,
Charney School of Marine Sciences, University of Haifa
Haifa, Israel*

Abstract—Autonomous underwater vehicles (AUVs) are regularly used for deep ocean applications. Commonly, the autonomous navigation task is carried out by a fusion between two sensors: the inertial navigation system and the Doppler velocity log (DVL). The DVL operates by transmitting four acoustic beams to the sea floor, and once reflected back, the AUV velocity vector can be estimated. However, in real-life scenarios, such as an uneven seabed, sea creatures blocking the DVL's view and, roll/pitch maneuvers, the acoustic beams' reflection is resulting in a scenario known as DVL outage. Consequently, a velocity update is not available to bind the inertial solution drift. To cope with such situations, in this paper, we leverage our BeamsNet framework and propose a Set-Transformer-based BeamsNet (ST-BeamsNet) that utilizes inertial data readings and previous DVL velocity measurements to regress the current AUV velocity in case of a complete DVL outage. The proposed approach was evaluated using data from experiments held in the Mediterranean Sea with the Snapir AUV and was compared to a moving average (MA) estimator. Our ST-BeamsNet estimated the AUV velocity vector with an 8.547% speed error, which is 26% better than the MA approach.

Index Terms—Autonomous underwater vehicle (AUV), Inertial navigation system (INS), Doppler velocity log (DVL), Deep Learning, Transformer

I. INTRODUCTION

Autonomous underwater vehicles (AUVs) are used in different applications. Autonomous navigation with sufficient accuracy is one crucial aspect allowing the AUV to successfully complete its task. Therefore, the AUV is equipped with several navigation-enabling sensors [1]. Often, the inertial navigation system (INS) and Doppler velocity log (DVL) are used to provide the navigation solution [2]–[4]. The INS consists of inertial sensors that measure the specific force and angular velocity vectors. These are then integrated to provide the navigation solution, namely the AUV's position, velocity, and orientation. [5], [6]. However, due to the system's nature, the navigation solution accumulates error that needs to be corrected using an external aiding sensor [7], [8] such as a DVL.

The DVL transmits four acoustics beams to the seafloor that are reflected back to the sensor. Through this process, known as bottomlock, beam velocity measurements are obtained. It has been shown that DVL provides high-accuracy velocity estimates and is, therefore, considered a reliable sensor [9]. To further improve the DVL velocity estimate, we introduced

BeamsNet, an end-to-end deep learning framework to estimate the AUV velocity vector using DVL data or inertial/DVL information [10]. However, in real-world scenarios, not all the DVL beams are reflected back due to several reasons, such as uneven seabed, sea creatures blocking the sensor's view, and during pitch/roll maneuvers [11], [12]. If fewer than three beam measurements are available the DVL cannot provide the AUV velocity vector and consequently the AUV would usually abort its mission and be forced back to the surface.

To obtain continuous DVL data flow, several solutions have been introduced addressing both partial (less than three beams) and complete (no beams) DVL outages. In partial DVL scenarios, a virtual beam was generated using past measurements and additional data [13]. Further development expanded this approach by considering the cross-covariance matrix of the correlated INS and DVL noise [11]. Later, a non-linear filter was used for INS/DVL fusion using partial DVL measurements [14]. Recently, data-driven approaches were employed for navigation purposes [15] allowing for partial DVL measurements learning approach to emerge. Furthermore, a long short-term memory (LSTM) has been suggested as a model to compensate when one beam is missing [16], and later using least squares support vector machine-aided virtual beam construction algorithm [17]. Previously, we addressed a scenario of two missing DVL beams using a modified BeamsNet architecture [18].

To address complete DVL outages an analytical closed-form solution was developed to estimate the AUV velocity, based on past DVL measurements [12], [19]. Later, a nonlinear autoregressive model based on INS and DVL measurements was suggested and evaluated using a ship [20].

This paper proposes, Set-Transformer based BeamsNet (ST-BeamsNet), a novel solution to handle complete DVL outages. ST-BeamsNet utilizes past AUV velocities and inertial data to forecast the current AUV velocity vector. In the first phase, the output of the two sensors goes through a patch-embedding layer that extracts features, and then utilizes the attention mechanism to learn data dependencies, disregarding their distance in the input. Later, the data is aggregated using a learnable aggregation block and finally, the features and dependencies from both sensors are linked to create the AUV velocity vector. ST-BeamsNet was evaluated using data from experiments in the Mediterranean Sea using the University of Haifa's Snapir AUV. Approximately four hours of recorded IMU and DVL data were collected and used to train, validate,

*Corresponding author: N. Cohen (email: ncohe140@campus.haifa.ac.il).

and test our suggested approach.

The rest of the paper is organized as follows: Section II introduces the DVL equations and error models. Section III discusses the model's learning blocks. Section IV describes the proposed approach, ST-BeamsNet, while our sea experiments and dataset are presented in Section V. The results are provided in Section VI and the conclusions are discussed in Section VII.

II. DVL VELOCITY UPDATES

The DVL produces measurements based on the Doppler effect by transmitting four acoustic beams to the seafloor, commonly using a Janus Doppler configuration, an 'x' shape configuration, as can be seen in Fig. 1. By calculating the frequency shift between the transmitted beam and the one reflected to the sensor, beam velocity measurements are obtained [21].

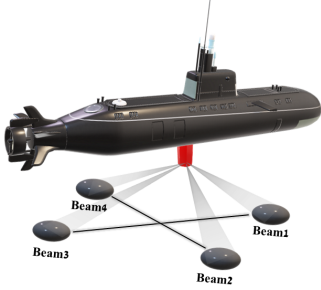


Fig. 1. A DVL transmits acoustic beams in 'x' shape configuration".

In the 'x' shape configuration the beams direction vectors are defined by

$$\text{beam}_i = [\cos \psi_i \sin \theta \quad \sin \psi_i \sin \theta \quad \cos \theta]_{1 \times 3} \quad (1)$$

where the index i represents the beam number, θ is the pitch angle, and the heading angle ψ is

$$\psi_i = (i - 1) \cdot \frac{\pi}{2} + \frac{\pi}{4} [\text{rad}], \quad i = 1, 2, 3, 4 \quad (2)$$

The relationship between the AUV velocity in the body frame, \mathbf{v}_b^b , to the beam velocity measurements, \mathbf{v}_{beam} , can be expressed by a transformation matrix \mathbf{H} [14]:

$$\mathbf{v}_{\text{beam}} = \mathbf{H} \mathbf{v}_b^b, \quad \mathbf{H} = \begin{bmatrix} \text{beam}_1 \\ \text{beam}_2 \\ \text{beam}_3 \\ \text{beam}_4 \end{bmatrix}_{4 \times 3} \quad (3)$$

To solve (3), a linear least squares (LS) filter is used:

$$\hat{\mathbf{v}}_b^b = \underset{\mathbf{v}_b^b}{\text{argmin}} \|\mathbf{y} - \mathbf{H} \mathbf{v}_b^b\|^2 \quad (4)$$

where \mathbf{y} is the measured beam velocity. The solution for the LS problem (4) is [22]:

$$\hat{\mathbf{v}}_b^b = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y} \quad (5)$$

The filter takes on two roles, filtering the errors and transforming the vector from the beam's coordinate frame to the AUV body frame.

III. LEARNING BLOCKS

We review here the basic learning blocks later used in our proposed Set-Transformer BeamsNet approach.

A. Patch Embedding

As suggested in [23], the data is split into patches, flattened, and then linearly projected using a single one-dimensional convolution operation with a kernel size α , stride size β , and patch size γ . This patch embedding is represented by $\mathbf{x}_p \in \mathbb{R}^{N \times D}$, where N is the number of samples generated by the one-dimensional convolution layer and D is the number of filters the layer uses and the latent space dimension fed to the Set-Transformers blocks.

B. Attention

The attention mechanism is defined by

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_q}}\right) \mathbf{V} \in \mathbb{R}^{n \times d_v} \quad (6)$$

where n query vectors of size d_q are represented by $\mathbf{Q} \in \mathbb{R}^{n \times d_q}$ and n_v key - values pairs are $\mathbf{K} \in \mathbb{R}^{n_v \times d_q}$ and $\mathbf{V} \in \mathbb{R}^{n_v \times d_v}$, respectively, d_v is the corresponding values vector dimensions. The Softmax function is [24]

$$\text{Softmax}(\mathbf{Z})_i = \frac{e^{Z_i}}{\sum_{n=1}^M e^{Z_j}} \quad (7)$$

and M is the number of vectors.

C. Multihead Attention

Multiheads are defined to extend the attention mechanism. The multihead attention model projects query Q , keys K , and values V vectors onto h different heads and \hat{d}_q , \hat{d}_k , \hat{d}_v dimensional vectors, respectively. The previously introduced mechanism (6) is applied to each h projections.

Using multihead attention, the model can utilize information from different representation subspaces and positions. The model consists of an attention mechanism with trainable parameters $\mathbf{W}_j^Q, \mathbf{W}_j^K \in \mathbb{R}^{d_q \times \hat{d}_q}$ and $\mathbf{W}_j^V \in \mathbb{R}^{d_v \times \hat{d}_v}$

$$\text{head}_j = \text{Attention}(\mathbf{Q}\mathbf{W}_j^Q, \mathbf{K}\mathbf{W}_j^K, \mathbf{V}\mathbf{W}_j^V), \quad j = 1, \dots, h \quad (8)$$

The multiple heads are linked together in a series and multiplied with a trainable parameter $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d}$ resulting in the following:

$$\text{Multihead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{concatenate}(\text{head}_1, \dots, \text{head}_h) \mathbf{W}^O \quad (9)$$

A common dimension hyper-parameters choice is $\hat{d}_q = \frac{d_q}{h}$, $\hat{d}_v = \frac{d_v}{h}$ and $d = d_q$.

D. Set Attention Block

The multihead attention block (MAB) is defined by modifying the encoder block of the Transformer introduced in [25] by removing the positional encoding and dropout operation. Given two matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times d}$ the parameter \mathbf{L} is defined as:

$$\mathbf{L} = \text{LayerNorm}(\mathbf{X} + \text{Multihead}(\mathbf{X}, \mathbf{Y}, \mathbf{Y})) \quad (10)$$

and the MAB is defined as

$$\text{MAB}(\mathbf{X}, \mathbf{Y}) = \text{LayerNorm}(\mathbf{L} + \text{FFN}(\mathbf{L})) \quad (11)$$

where FFN is a fully connected feed-forward network built from a layer that expands the dimension to the size of FFE , the feed-forward expansion hyper-parameter. After applying a ReLU activation function, the second layer transforms it back to the original dimension

$$\text{FFN}(\mathbf{X}) = \max(0, xW_1 + b_1)W_2 + b_s \quad (12)$$

The set attention block (SAB) performs self-attention between the elements in the set and can be defined using the MAB as follows:

$$\text{SAB}(\mathbf{X}) := \text{MAB}(\mathbf{X}, \mathbf{X}) \quad (13)$$

E. Pooling by Multihead Attention

This block introduces a learnable aggregation scheme by defining a set of k vectors of size n and stacking them in a matrix $\mathbf{S} \in \mathbb{R}^{k \times n}$. To that end, by assuming that an encoder produces a set of features $\mathbf{Z} \in \mathbb{R}^{n \times d}$, then the pooling by multihead attention block (PMA) is defined as follows:

$$\text{PMA}_k(\mathbf{Z}) = \text{MAB}(\mathbf{S}, \text{FFN}(\mathbf{Z})) \quad (14)$$

where k is a hyper-parameter.

F. Overall Architecture

The architecture is built from an encoder followed by a decoder. Given a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ the encoder outputs the extracted features matrix $\mathbf{Z} \in \mathbb{R}^{n \times d}$. The encoder is assembled by stacking SAB blocks (13):

$$\mathbf{Z} = \text{Encoder}(\mathbf{X}) = \text{SAB}_1 \circ \dots \circ \text{SAB}_b(\mathbf{X}) \quad (15)$$

The decoder used the learnable aggregation to transform \mathbf{Z} into a set of vectors that passes through a feed-forward network to receive the outputs [26].

$$\text{Decoder}(\mathbf{Z}) = \dots \circ \text{FFN}_b(\text{SAB}(\text{PMA}_k(\mathbf{Z}))) \in \mathbb{R}^{k \times d} \quad (16)$$

where b is the number of stacked blocks.

To conclude, the hyper-parameters that need to be determined in the blocks above are:

- α - Patch-Embedding kernel size
- β - Patch-Embedding stride size
- γ - Patch-Embedding patch size
- D - Number of filters in the 1D convolution layer
- d - Latent space dimension equaling to D
- h - Number of attention heads
- FFE - Feed forward expansion
- b - Number of stacked SAB blocks
- k - Number of trainable vectors for aggregation

IV. SET TRANSFORMER BASED BEAMSNET

Our proposed approach utilizes data from the inertial sensors, namely the accelerometers and gyroscopes, and n past DVL velocity vector measurements. The motivation behind the specific architecture is to learn features from each sensors' data using the patch embedding layer, which is later on inspected for dependencies using the multihead attention mechanism. Then, a learnable pooling aggregation is applied to obtain the most valuable features and dependencies, which are used to find links between the sensors using fully connected layers to regress the AUV velocity vector.

The flow of information is as follows: first, the data goes through a patch-embedding layer, then it passes through b blocks of encoders or decoders. Finally, the decoders' output from both sensors' data is combined and processed using two fully connected layers separated by the ReLU activation function. Since the network's purpose is to forecast the $n + 1$ velocity vector, the loss function is the mean squared error (MSE). A block diagram showing our proposed approach is presented in Fig. 2. The hyper-parameters that were used in the set-transformer-based BeamsNet are presented in Table. I.

TABLE I
ST-BEAMSNET'S HYPER-PARAMETERS AND VALUES

Hyper-parameter	α	β	γ	D	h	ffe	b	k
Value	2	1	1	128	8	256	2	3

V. DATASET

The dataset was collected using the "Snapir" AUV, an A18-D ECA GROUP mid-size AUV shown in Figure 3. Snapir's length and diameter are 5.5 [m] and 0.5 [m], respectively. Snapir can perform missions of up to 24 hours and reach 3000 [m] depth. The experiments took place in the Mediterranean Sea, and data were collected from the AUV's inertial sensors (iXblue Phins Subsea IMU [27]) and DVL (Teledyne RD Instruments, Navigator DVL [28]) over nine missions. The DVL has a sampling rate of 1 [Hz], and the inertial sensors sample at 100 [Hz]. The total duration of the recorded data is approximately four hours and include 13,886 DVL and 1,388,600 IMU measurements. The suggested network was trained on eight randomly chosen missions and tested on the remaining mission.

As the Snapir AUV carries only one DVL sensor, to mimic a scenario of a sensor that provides ground truth and a unit under test we added a noise term to the DVL measurements. To that end, a scale factor of 0.7%, a bias of 0.0001[m/s], and zero-mean white Gaussian noise of 0.042[m/s] were added. Once the data was created, a scenario of a complete DVL outage was created by removing the AUV velocity vector every four seconds.

VI. RESULTS

To evaluate the suggested approach, four metrics were used:

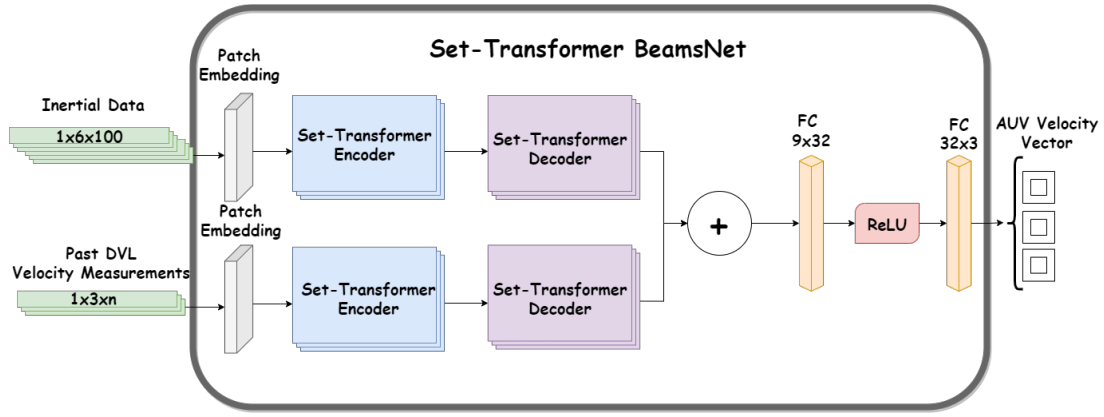


Fig. 2. A block diagram of the set-transformer based BeamsNet

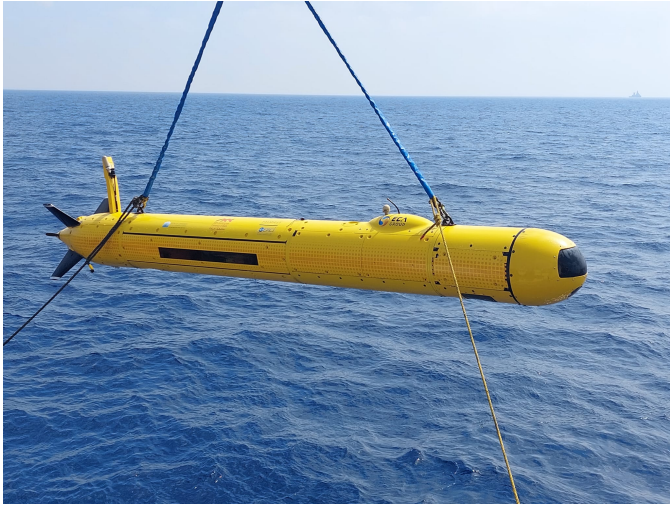


Fig. 3. The Snapir AUV before starting the experiment in the Mediterranean Sea.

1) **Root Mean Squared Error (RMSE)**

$$RMSE(x_i, \hat{x}_i) = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}} \quad (17)$$

2) **Mean Absolute Error (MAE)**

$$MAE(x_i, \hat{x}_i) = \frac{\sum_{i=1}^N |x_i - \hat{x}_i|}{N} \quad (18)$$

3) **The coefficient of determination (R^2)**

$$R^2(x_i, \hat{x}_i) = 1 - \frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{\sum_{i=1}^N (x_i - \bar{x}_i)^2} \quad (19)$$

4) **The variance account for (VAF)**

$$VAF(x_i, \hat{x}_i) = [1 - \frac{var(x_i - \hat{x}_i)}{var(x_i)}] \times 100 \quad (20)$$

The RMSE and MAE represent the velocity error in units of [m/s], while R^2 and VAF are unitless. An optimal solution would be if the VAF is 100, the R^2 is 1, and the RMSE and MAE are 0 [29]. In the above equations:

- N is the number of samples
- x_i is the ground truth velocity vector norm of the DVL
- \hat{x}_i is the predicted velocity vector norm of the AUV given by the network
- \bar{x}_i is the mean of the ground truth velocity vector norm of the DVL
- var stands for variance.

In addition, for a fair comparison, the Moving Average (MA) was calculated to predict the AUV velocity measurements based on $n = 3$ past AUV velocity measurements [30]. Furthermore, the magnitude of the speed error was calculated by comparing the results to the mean of the ground truth velocity vector norm. The results are presented in Table II.

TABLE II
COMPARISON BETWEEN ST-BEAMSNET TO THE MOVING AVERAGE METHOD.

Evaluation Metrics	ST-BeamsNet	Moving-Average
RMSE [m/s]	0.098	0.134
RMSE [%]	8.574	11.714
MAE [m/s]	0.064	0.096
MAE [%]	5.649	8.395
R^2	0.978	0.968
VAF	97.997	96.971

The mean of the ground truth velocity vector norm for the test set is 1.14 [m/s]. These results indicate that the suggested approach can better forecast the AUV velocity vector, using $n = 3$ previous DVL velocity vector measurements and the inertial data readings, compared to the MA algorithm in each of the selected metrics. Besides obtaining a speed error of 8.571% and 5.649% with respect to the RMSE and MAE metrics, respectively, compared to the mean of the ground truth velocity vector norm, the set-transformer based BeamsNet outperforms the MA estimator and provides a 26% improvement.

Fig. 4 presents the probability density error graph for both methods. The ST-BeamsNet's mean error value is 0.022[m/s], and the MA estimator mean error value is 0.028[m/s]. Also, a difference in the error variance value was obtained where for

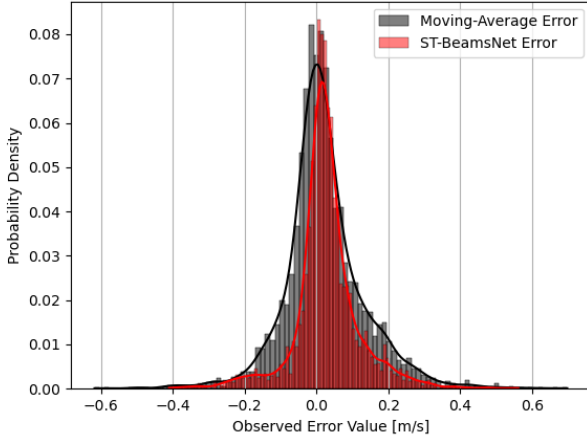


Fig. 4. A probability density error graph for the suggested approach (in red) and the MA method (in black).

the ST-BeamsNet approach the value is 0.008[m/s] while for the MA method 0.013[m/s].

VII. CONCLUSIONS

This paper introduced ST-BeamsNet, an end-to-end set-transformer based architecture to forecast to AUV velocity vector in a case of a complete DVL outage. The suggested approach utilizes data from the inertial sensors and past AUV velocity measurements to obtain the current velocity and, therefore, could prevent the navigation solution from drifting. By gathering data from an experiment using Snapir AUV in the Mediterranean Sea, the model was trained, tested, and compared to an MA estimator.

The results show that ST-BeamsNet can forecast the current AUV velocity in the case of a complete DVL outage, yielding a speed error of 5.649% and 8.54% with respect to the MAE and RMSE metrics compared to the average speed, respectively. Furthermore, compared to the MA method, the suggested approach provides an improvement of 26% and is characterized by a smaller variance error.

ACKNOWLEDGMENTS

N.C. is supported by the Maurice Hatter Foundation and University of Haifa presidential scholarship for students on a direct Ph.D. track.

REFERENCES

- [1] S. K. Jain, S. Mohammad, S. Bora, and M. Singh, "A review paper on: autonomous underwater vehicle," *International Journal of Scientific & Engineering Research*, vol. 6, no. 2, p. 38, 2015.
- [2] Z. Yonggang, D. Yan, and L. Ning, "A tightly integrated SINS/DVL navigation method for autonomous underwater vehicle," in *2013 International Conference on Computational and Information Sciences*. IEEE, 2013, pp. 1107–1110.
- [3] Y. Yao, X. Xu, L. Hou, K. Deng, and X. Xu, "A simple and precise correction method for DVL measurements under the dynamic environment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 10 750–10 758, 2020.

- [4] I. Klein and R. Diamant, "Observability analysis of DVL/PS aided INS for a maneuvering AUV," *Sensors*, vol. 15, no. 10, pp. 26 818–26 837, 2015.
- [5] D. Titterton, J. L. Weston, and J. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.
- [6] P. D. Groves, "Principles of GNSS, inertial, and multisensor integrated navigation systems," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 2, pp. 26–27, 2015.
- [7] J. L. Farrell, *GNSS aided navigation & tracking: inertially augmented or autonomous*. American Literary Press Baltimore, Maryland, 2007.
- [8] E.-H. Shin and N. El-Sheimy, "Accuracy improvement of low cost INS/GPS for land applications," in *Proceedings of the 2002 national technical meeting of the institute of navigation*, 2002, pp. 146–157.
- [9] D. Wang, X. Xu, Y. Yao, T. Zhang, and Y. Zhu, "A novel SINS/DVL tightly integrated navigation method for complex environment," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 5183–5196, 2019.
- [10] N. Cohen and I. Klein, "BeamsNet: A data-driven approach enhancing Doppler velocity log measurements for autonomous underwater vehicle navigation," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105216, 2022.
- [11] R. Eliav and I. Klein, "INS/partial DVL measurements fusion with correlated process and measurement noise," *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 4, no. 1, p. 34, 2018.
- [12] I. Klein and Y. Lipman, "Continuous INS/DVL fusion in situations of dvl outages," in *2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*. IEEE, 2020, pp. 1–6.
- [13] A. Tal, I. Klein, and R. Katz, "Inertial navigation system/Doppler velocity log (INS/DVL) fusion with partial DVL measurements," *Sensors*, vol. 17, no. 2, p. 415, 2017.
- [14] P. Liu, B. Wang, Z. Deng, and M. Fu, "INS/DVL/PS tightly coupled underwater navigation method with limited DVL measurements," *IEEE Sensors Journal*, vol. 18, no. 7, pp. 2994–3002, 2018.
- [15] I. Klein, "Data-driven meets navigation: Concepts, models, and experimental validation," in *2022 DGON Inertial Sensors and Systems (ISS)*, 2022, pp. 1–21.
- [16] M. Yona and I. Klein, "Compensating for Partial Doppler Velocity Log Outages by Using Deep-Learning Approaches," in *2021 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*. IEEE, 2021, pp. 1–5.
- [17] Y. Yao, X. Xu, X. Xu, and I. Klein, "Virtual beam aided SINS/DVL tightly coupled integration method with partial DVL measurements," *IEEE Transactions on Vehicular Technology*, 2022.
- [18] N. Cohen and I. Klein, "Libeamsnet: AUV velocity vector estimation in situations of limited DVL beam measurements," *IEEE OCEANS conference*, 2022.
- [19] I. Klein, Y. Gutnik, and Y. Lipman, "Estimating DVL velocity in complete beam measurement outage scenarios," *IEEE Sensors Journal*, vol. 22, no. 21, pp. 20 730–20 737, 2022.
- [20] D. Li, J. Xu, H. He, and M. Wu, "An underwater integrated navigation algorithm to deal with DVL malfunctions based on deep learning," *IEEE Access*, vol. 9, pp. 82 010–82 020, 2021.
- [21] N. A. Brokloff, "Matrix algorithm for Doppler sonar navigation," in *Proceedings of OCEANS'94*, vol. 3. IEEE, 1994, pp. III–378.
- [22] B. Braginsky, A. Baruch, and H. Guterman, "Correction of DVL Error Caused by Seafloor Gradient," *IEEE Sensors Journal*, vol. 20, no. 19, pp. 11 652–11 659, 2020.
- [23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [26] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 3744–3753.
- [27] iXblue, "iXblue Phins Subsea INS," Available: <https://www.ixblue.com/store/phins-subsea/>, accessed: 2022-05-01.
- [28] Teledyne, "Teledyne Marine RD Instruments DVL," Available: <http://www.teledynemarine.com/dvls>, accessed: 2022-05-01.
- [29] D. J. Armaghani and P. G. Asteris, "A comparative study of ANN and ANFIS models for the prediction of cement-based mortar materials

- compressive strength,” *Neural Computing and Applications*, vol. 33, no. 9, pp. 4501–4532, 2021.
- [30] S. Smith, *Digital signal processing: a practical guide for engineers and scientists*. Elsevier, 2013.