Learning an AUV docking maneuver with a convolutional neural network

Albert Sans-Muntadas, Kristin Y. Pettersen, Edmund Brekke and Eleni Kelasidi
Norwegian University of Science and Technology (NTNU)
Centre for Autonomous Marine Operations and Systems (NTNU-AMOS)
Otto Nielsens veg 10, NO-7491 Trondheim, Norway
Email: {albert.sans, kristin.y.pettersen, edmund.brekke, eleni.kelasidi}@ntnu.no

***Abstract*—This paper proposes and implements a convolutional neural network (CNN) that maps images from a camera to an error signal to guide and control an autonomous underwater vehicle into the entrance of a docking station.**

The paper proposes to use an external positioning system synchronized with the vehicle to obtain a dataset of images matched with the position and orientation of the vehicle. By using a guidance map the positions are converted into desired directions that guide the vehicle to a docking station. The network is then trained to estimate, for each frame, the error between the desired direction and the orientation. After training, the CNN can estimate the error without using the external positioning system, creating an end-to-end solution from image to a control signal.

I. Introduction

Autonomous underwater vehicle (AUV) technology has proven to be sufficiently mature to perform a variety of underwater missions autonomously. However, the battery duration of AUVs is generally a limiting factor for a mission. This restriction adds the need of a surface support vessel to launch and recover the vehicle, something which increases the cost of the mission and makes the operational outcome more dependent on sea conditions.

A permanent docking station on the seafloor, where the vehicle could charge the batteries and transfer the results of a mission, would reduce the need for frequent launch and recovery operations at the surface, making the technology more cost effective, safer and more robust. Also, it would enable the possibility of permanently residing AUVs ready for subsea operations, which would further extend the capabilities of the AUV technology.

The lack of precise positioning systems such as GPS underwater represents one of the most challenging aspects of a docking operation. This can be compensated for by using alternative positioning methods, a summary of which can be found in [1]. In some cases, tailor-made solutions specific to the docking task are applied to improve the navigation accuracy when the vehicle is close to the docking station. In [2] a single beacon solution is described. An induced local electromagnetic field is used in [3] to obtain more precise positioning data. A computer vision approach to identify and locate the docking station is proposed in [4]. These solutions often require a certain level of human abstraction such as recognizing features of the docking station, producing navigation data and generating control commands. [5] shows how recent advances in convolution neural networks allow, given the right conditions, to produce end-to-end solutions that optimize all these elements simultaneously. Examples where neural networks have been applied to learn and perform a complex control task, by itself and even outperform humans are presented in, [6], [7]. In the context of self driving cars, [5], [8] show that a neural network is able to produce a regression that maps image data into angular steering wheel commands, enabling a car, under certain conditions, to drive autonomously.

Motivated by these advances, this paper proposes a framework for obtaining data and training a convolutional neural network (CNN) for docking an AUV. The proposed CNN uses raw images from a front facing camera as input and as output it produces the error signal that can later be fed to a controller to steer the vehicle into the docking station. In the proposed framework the data required for training the CNN to perform a docking maneuver is obtained in a controlled environment, such as a tank equipped with a motion capture system or an underwater operation where a supply vessel equipped with GNSS-USBL is present and able to provide accurate measurements. The external positioning system in

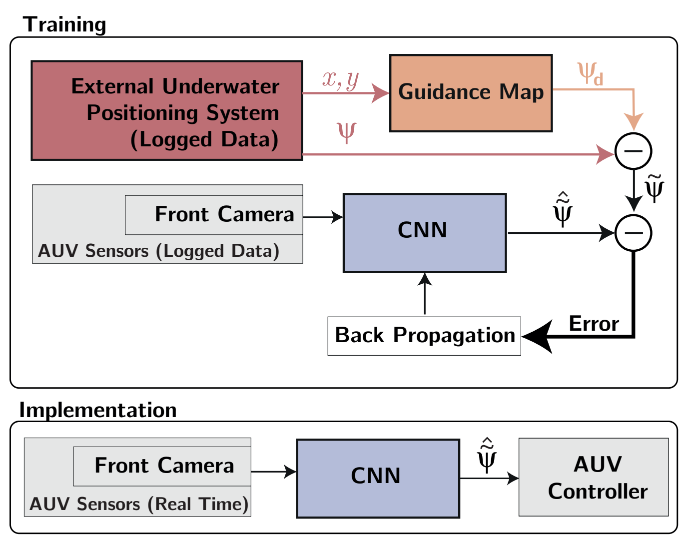

Fig. 1: Block diagram showing how the CNN is trained and the implementation after training

this controlled environment allows for producing very accurate measurements of the position and attitude (x, y, ψ) of the vehicle during the training period. Simultaneously and in a synchronized way, the data from the AUV's internal sensors is recorded from a large set of different states/locations and paired with the precise data from the external sensors.

As represented in Figure 1, all the data measured by external positioning system is mapped trough a guidance map, to generate desired direction ψ_d that guides the AUV towards the docking station. The end goal of this application is to produce a regression from image/sensor information to an estimate of the heading error, $\tilde{\psi}$, that does not use any of the external sensors. This end-to-end solution combines the detection of the docking station, navigation and guidance all in one single network. To achieve this, the CNN takes frames of the training dataset as input, and the output from the CNN is compared with the ground-truth values obtained from the external measurement system.

The paper is organized as it follows: Section II proposes a guidance map and a transformation of the coordinate system to map each position to the desired direction for the AUV. Section III explains how the data for training the network is obtained and pre-processed. Section III-A describes the model of the network. Section IV explains the results and Section V draw some conclusions about the design, training, and performance of the network.

II. Guidance

This section presents two elements that will be used for the guidance of the vehicle: A transformation of the polar coordinate system that takes into account the size of the docking station, and a guidance map that prescribes the desired direction at any given point.

A. Transformation of the polar coordinates

Polar coordinates are sometimes useful in underwater navigation, especially for systems that measure range and bearing. Also descriptions of spiral paths and trajectories become simpler when using polar coordinates. However, in polar coordinates any displacement close to the origin and perpendicular to the radial direction results in large changes in the angular coordinate. This singularity can make polar coordinates unattractive because some controllers may become unstable when the system comes close to the origin. Since a docking station has tolerance to accommodate a certain lateral offset, we propose a transformation of the coordinate system that can take into account the size of the entrance of a funnel shaped docking station such as the one used in [9]. In this paper we propose a transformation of the polar coordinates $(r, \phi) \rightarrow (r^*, \phi^*)$ that, far from the origin, behaves as polar coordinates, but unlike polar coordinates when close to the origin the points in front of the entrance of the docking station will all have similar range and angular position (see figure 2). Since this avoids the singularity of polar coordinates at the entrance, this choice of coordinates can provide a more robust

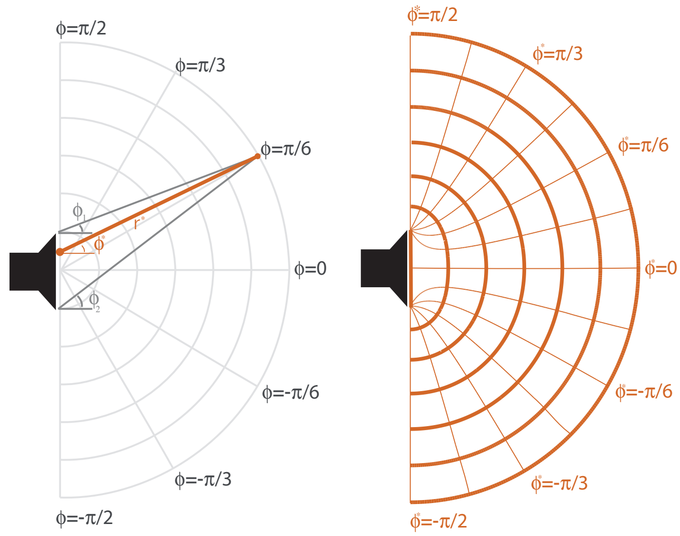

Fig. 2: Transformation of the polar coordinates

control system when the vehicle moves towards the origin, i.e. the docking station.

The transformation, that could also be interpreted as "stretching the center point" of the polar coordinates along the entrance, is described based on two virtual poles that are placed at each side of the docking station's entrance (at a distance $d/2$ from the center). The first parameter of the proposed system, the transformed angular coordinate ϕ^*, is described as the average angle $\phi^* = \frac{1}{2}(\phi_1 + \phi_2)$, where ϕ_1 and ϕ_2 are defined by the position of the virtual poles as shown in Figure 2.

By definition, the new angular coordinate is thus always between ϕ_1 and ϕ_2.

$$\phi^* = \frac{1}{2}\left(\underbrace{\operatorname{atan}\left(\tan(\phi) + \frac{\frac{d}{2}}{r\cos(\phi)}\right)}_{\phi_1} + \underbrace{\operatorname{atan}\left(\tan(\phi) - \frac{\frac{d}{2}}{r\cos(\phi)}\right)}_{\phi_2}\right) \tag{1}$$

The transformation for the range r^* is described as the length of a segment, that has an angle ϕ^* with respect to docking station centerline, and connects the point (r, ϕ) with the docking station's entrance. Note that the point of the segment at the entrance is not a fixed point and can move between the two poles depending on (r, ϕ). The length of the segment r^* and can be found from the following expression:

$$r^* = r\frac{\cos(\phi)}{\cos(\phi^*)} \tag{2}$$

B. Guidance Map

In [10] a spiral path was proposed for reaching a docking station and at the same time preserve the field of view (FOV) of the transmitter or landmark mounted on the docking station for navigation purposes. In this paper, we propose an alternative solution to avoid the path planning. Instead, the new approach uses a mapping of each position to a desired direction ψ_d . The desired direction is designed such that when the vehicle follows this direction it will reach the entrance of the docking station. In particular, we choose the desired direction as:

$$\psi_d(\phi) = \phi + \operatorname{atan}\left(\tanh\left(\frac{2\phi}{\tan(\theta_{\max})}\right)\tan(\theta_{\max})\right) \quad (3)$$

where $\theta_{\max}$ is the maximum FOV. Figure 3 shows the benefit of using the transformation of the coordinates; the left plot shows the result of applying polar coordinates to the guidance map $\psi_d(\phi)$, and the right plot shows the guidance map when using transformation of the coordinates $\psi_d(\phi^*)$. By using the transformation of coordinates, the guidance system allows the vehicle to enter the docking station from any point contained between the sides of the docking station. Note that if the polar coordinates were used instead, this guidance system would only allow the vehicle to enter the docking station trough the centerline.

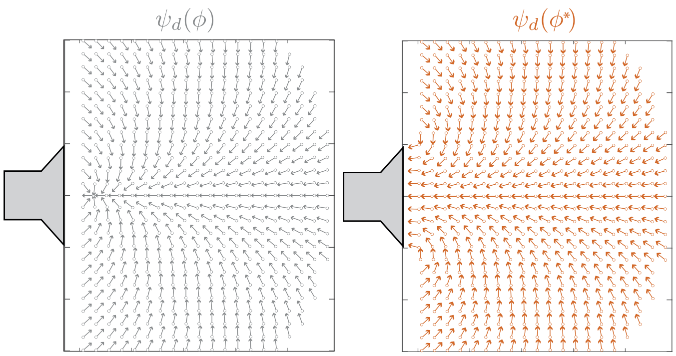

Fig. 3: Guidance map with polar coordinates (left) or the transformed coordinates (right)

III. Data Acquisition and Pre-processing

The images and data used in this paper were obtained during the tuning and calibration phases of the experimental results published in [10]. The required images were obtained in the Marine Cybernetics Laboratory (MC-lab) at NTNU, Trondheim, Norway [11], in a tank of dimensions L: 40 m, H: 1.5 m and W: 6.45 m. The camera used to obtain the images was attached to an underwater vehicle which received real-time measurements of the robot's position and orientation were obtained from an underwater motion capture system, Qualisys, installed in the basin [12]. Note that a template of markers was mounted under the head module of the robot, where the camera was attached, that allowed the positioning system to determine the position and heading of each frame accurately. A flat panel with a reflective region was used as a mockup of the entrance of a docking station. The position of the docking station (x_{ds}, y_{ds}) was also obtained using the underwater positioning system. The data acquired consists of eight different runs of a docking maneuver, resulting in a total of 5358 frames correlated with their position. The frames obtained were resized to 96x112 px to be more computationally

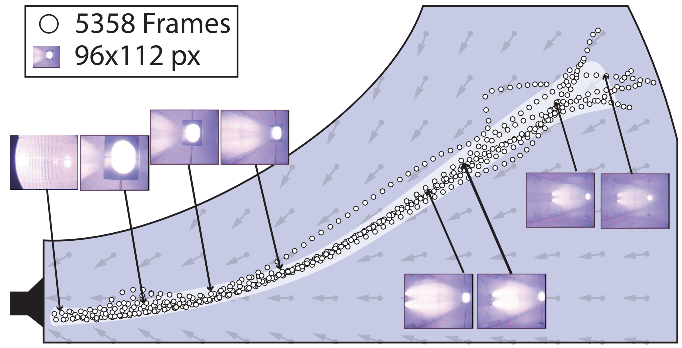

Fig. 4: Representation of the data set used for training the CNN. The background arrows represent the guidance map

efficient but still preserve the important features. The images were then normalized from: $[0, 255] \rightarrow [0, 1]$. The positions and orientation (x, y, ψ) of the camera and the position of the docking station (x_{ds}, y_{ds}), obtained by the Qualisys motion capture system, were first transformed to polar coordinates, with the position of the docking station as origin:

$$\begin{aligned} r &= \sqrt{(x - x_{ds})^2 + (y - y_{ds})^2} \\ \phi &= \operatorname{atan2}(y - y_{ds}, x - x_{ds}) \end{aligned} \quad (4)$$

Then, Equations (1-2) were used to transform the polar coordinates and Equation (3) was used to calculate the desired heading ψ_d, for each position. Finally, the error between the heading ψ and the desired heading given by the guidance map, ψ_d, was calculated:

$$\tilde{\psi} = \psi - \psi_d \quad (5)$$

This produced an array pairing each of the 5358 frames of the set with the error $\tilde{\psi}$. The data are represented in Figure 4 each dot represents the position from where each frame was obtained, and the arrows represent the guidance map used to calculate ψ_d and $\tilde{\psi}$. A 15% of these data (a full run) was saved for future validation.

A. Model of the network and training parameters

The neural network used in this paper follows the structure of the network proposed in [5], but since the scenario for the docking is quite monotone the size of the layers is reduced. This also reduces the complexity of the network, thus reducing the risk of overfitting.

The model of the network is illustrated in Figure 5 and its parameters are displayed in Table 1. The network begins with five convolution layers, the first three have 5x5 kernels and the two last have 3x3 kernels. The outcome of the last convolution layer is then flattened into a long array which is then connected to the fully connected layers (FCL). The FCL layers consist of four layers that reduce in size, for which each neuron is connected with all the neurons from the previous layer. The single neuron of the last layer returns the estimated error $\hat{\tilde{\psi}}$. The network is trained by finding the weights and biases that minimize the difference between the error $\tilde{\psi}$ produced by the guidance map and the $\hat{\tilde{\psi}}$ estimated by the neural network.

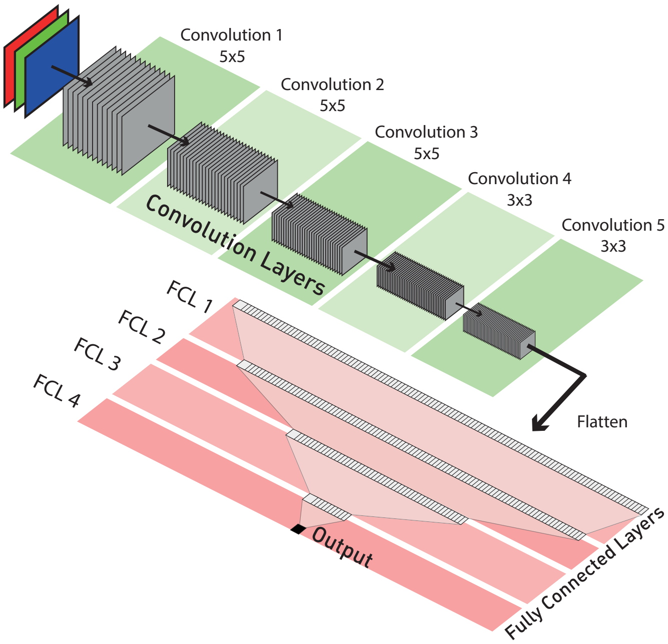

Fig. 5: Architecture of the network

Convolution Layers	Kernel Size	Stride	Padding	Filters	Activation
Conv 1	5x5	2	0	6	RelU
Conv 2	5x5	2	0	12	RelU
Conv 3	5x5	2	0	18	RelU
Conv 4	3x3	0	0	24	RelU
Conv 5	3x3	0	0	24	RelU
Fully Connected Layers	**Size**				**Activ.**
FCL 1	32				RelU
FCL 2	16				RelU
FCL 3	8				RelU
Output	1				-

TABLE I. Parameters of the model

The model and the training of the CNN was programmed using the Tensorflow library [13]. It was trained during 160 epochs in batches of 100 frames and the loss function used was the L2 norm distance with weight regularization (10^{-5}). All the weights were randomly initialized using a truncated normal distribution (STD:0.01). The network was then trained using the Adam optimization method [14] with a learning rate of (10^{-4}). To avoid overfitting, during the training phase, the images were augmented by randomly changing the contrast and brightness and imposing random dropouts in 20% of the connections of the fully connected layers [15], [16].

IV. Results

This section shows the results of training the system for 160 epochs. The resulting network has a size $>$ 300 Kb, and the computation time of an image could be executed in an order of magnitude faster than the camera frame rate. Figure 6 shows the performance of the CNN at estimating $\tilde{\psi}$ for the testing dataset. The performance of the CNN is illustrated in Figure 6. The figure shows two very different performances in the main and the late part of the frames, which are separated into two regions with different backgrounds colors. In the white region, which composes the initial and largest part of the docking maneuver, the network shows to be very accurate at estimating $\tilde{\psi}$ (RMSE: 0.0345 rad $|1.98$ deg). In the second region with gray background, however, the network suddenly becomes very inaccurate (RMSE: 0.1949 rad $|11.17$ deg). An explanation of the drop in performance can be found by observing the frames displayed in Figure 7. Here we see that the frames that give an inaccurate estimate are the last frames of the docking maneuver, i.e. when the vehicle is very close to the docking station. At such a close distance there are no references that the network can use to perceive neither the direction nor the distance to the docking station, such as the walls of the tank, the focus or the frame of the docking station.

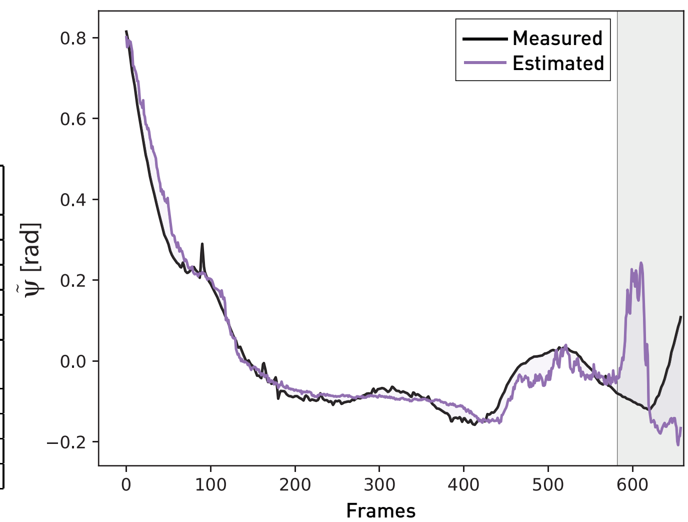

Fig. 6: Performance of the trained network estimate compared with the measured error

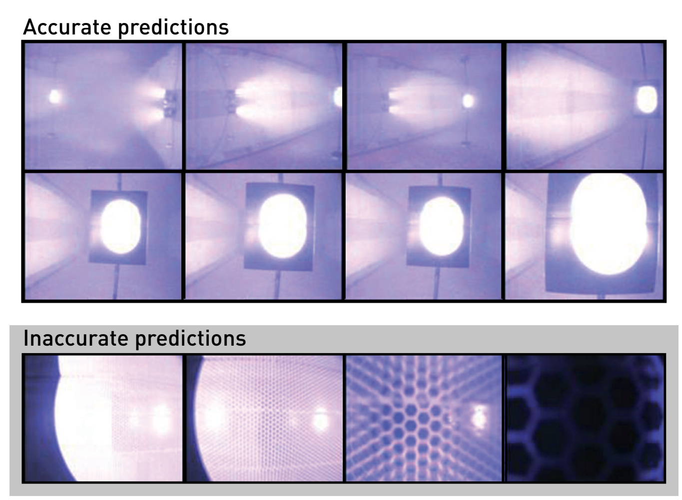

Fig. 7: Frames classified according to the accuracy of the estimate

V. Conclusions

In this paper we have proposed a framework for end-to-end learning of an AUV docking maneuver. In particular, we have shown how to gather the data and train a convolutional neural network to perform a docking task. The paper presents a guidance approach that maps the position and orientation of the vehicle with respect to the docking station, to a desired direction given by the yaw angle ψ_d. When the AUV follows this desired yaw angle, the system guides the vehicle to the docking station, while preserving the FOV. To train the CNN, the paper proposes to use an external positioning system synchronized with the vehicle to obtain a dataset of images, i.e. sensor measurements frames matched with the vehicle position. When combined with the guidance map, this data describes for each frame the error between the desired direction and the orientation of the vehicle. Then the dataset is used to train a convolutional neural network, which afterwards can estimate the error in the direction $\tilde{\psi}$, without using an external positioning system.

The results have shown that in general, the vehicle is able to accurately estimate the error, producing a solution that embeds the recognition of the docking station, the navigation and the guidance in one network. Therefore, a yaw control law using the estimated error in direction, $\tilde{\psi}$, as input, will make the vehicle follow a trajectory that leads into the docking station.

The lack of information when the vehicle is very close to the docking station has shown to produce inaccurate estimates. To overcome this issue, the last phase of the docking maneuver might need a special guidance law, for instance one that maintains a straight course. The neural network could also be trained to recognize when the vehicle is very close to the docking station and switch to an alternative guidance law.

The results presented in this paper only train the network to perform in a small area. Outside of the training set the vehicle would probably not be able to estimate the error precisely, but the experiment shows that if implemented for a larger training set, the network would be able to provide a more general and robust solution. The training set has a unique lighting condition, a more robust training would also need training data with different lighting conditions.

Future work may include experimental validation of the trained network as well as using larger training sets, data augmentation by virtually panning the camera, adding other sensors or using recurrent neural networks.

Acknowledgments

The authors gratefully acknowledge Trym Vegard Haavardsholm, FFI, for his inputs in the design and training of the network.

This work was partly supported by the Research Council of Norway through the Centers of Excellence funding scheme, project No. 223254 NTNU AMOS, and by VISTA – a basic research program in collaboration between The Norwegian Academy of Science and Letters, and Statoil.

References

[1] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.

[2] P. Baccou and B. Jouvencel, "Homing and navigation using one transponder for AUV, postprocessing comparisons results with long base-line navigation," in *Proc. International Conference on Robotics and Automation*, Washington, DC, 2002.

[3] M. D. Feezor, F. Y. Sorrell, P. R. Blankinship, and J. G. Bellingham, "Autonomous underwater vehicle homing/docking via electromagnetic guidance," *Journal of Oceanic Engineering*, vol. 26, no. 4, 2001.

[4] J.-Y. Park, B.-h. Jun, P.-m. Lee, and J. Oh, "Experiments on vision guided docking of an autonomous underwater vehicle using one camera," *Ocean Engineering*, vol. 36, no. 1, 2009.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[8] D. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems 1*, D. Touretzky, Ed. Morgan Kaufmann, 1989.

[9] R. S. McEwen, B. W. Hobson, L. McBride, and J. G. Bellingham, "Docking control system for a 54-cm-diameter (21-in) AUV," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 550–562, 2008.

[10] A. Sans-Muntadas, E. Kelasidi, K. Y. Pettersen, and E. Brekke, "Spiral path planning for docking of underactuated vehicles with limited FOV." Kohala Coast, HI: Proc. IEEE Conference on Control Technology and Applications (CCTA), 2017.

[11] "Marine cybernetics laboratory (MC-lab)." https://www.ntnu.edu/imt/lab/cybernetics, accessed: 2017-02-16.

[12] "Qualisys-motion." http://www.qualisys.com/, accessed: 2017-02-16.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.