*Article*

# Towards Integrated Digital-Twins: An Application Framework for Autonomous Maritime Surface Vessel Development

Minahil Raza *, Hanna Prokopova, Samir Huseynzade, Sepinoud Azimi and Sebastien Lafond

Faculty of Science and Engineering, Åbo Akademi University, 20540 Turku, Finland
* Correspondence: minahil.raza@abo.fi

**Abstract:** The use of digital twins for the development of Autonomous Maritime Surface Vessels (AMSVs) has enormous potential to resolve the increasing need for water-based navigation and safety at the sea. Aiming at the problem of lack of broad and integrated digital twin implementations with live data along with the absence of a digital twin-driven framework for AMSV design and development, an application framework for the development of a fully autonomous vessel using an integrated digital twin in a 3D simulation environment has been presented. Our framework has 4 layers which ensure that simulation and real-world vessel and the environment are as close as possible. Åboat, an in-house, experimental research platform for maritime automation and autonomous surface vessel applications, equipped with two trolling electric motors, cameras, LiDARs, IMU and GPS has been used as the case study to provide a proof of concept. Åboat, its sensors, and the environment have been replicated in a commercial, 3D simulation environment, AILiveSim. Using the proposed application framework, we develop obstacle detection and path planning systems based on machine learning which leverage live data from a 3D simulation environment to mirror the complex dynamics of the real world. Exploiting the proposed application framework, the rewards across training episodes of a Deep Reinforcement Learning model are evaluated for live simulated data in AILiveSim.

**Keywords:** maritime autonomy; autonomous ship; safety; digital twin; deep reinforcement learning; collision avoidance; situational awareness

## 1. Introduction

The increasing need for water-based navigation has created a strong demand for autonomous maritime surface vessels (AMSVs). In particular, research in the fields of autonomous navigation and collision avoidance has started to become increasingly relevant for maritime transport. Several projects develop and explore the viability of autonomous vessels, such as Roboat [1], the MUNIN project [2] and the YARA Birkeland project [3]. The definition of AMSV varies from project to project. In this work, we follow the definition given by the International Maritime Organisation (IMO). IMO defines an autonomous vessel as a vessel which, to a varying degree, can operate independently without human interaction. The degrees of autonomy are classified into four categories. Degree one includes vessels with automated processes and decision support along with human intervention. Degree two includes remotely operated vessels with seafarers on board. Degree three includes remote-controlled vessels without seafarers on board. Lastly, degree four refers to a fully autonomous vessel. The operating system of a fully autonomous vessel is capable of making decisions without any human intervention [4].

The key motivator for unmanned operations is to reduce human error, avoid accidents and tackle the predicted shortage of human resources. In 2020 alone, a total of 2632 accidents causing 36 deaths and 618 injuries were reported by the European Maritime Safety Agency and 43% of these were caused by navigational events such as collisions, grounding, and contact [5]. Formal investigation revealed that human factors and operational issues

were major causal factors behind these accidents. Thus, the development of AMSVs can aid in providing a countermeasure against such navigational events [6].

Artificial intelligence (AI), machine learning (ML), Internet-of-Things and big data have played a huge role in directing research efforts toward the design and development of AMSVs [6]. While many recent studies [1,3,6] have focused on developing collision avoidance systems for autonomous vessels, the scarcity of data and resources, along with the difficulty of conducting multiple and risky experiments, limit their potential. Within this context, digital twin vessels have started gaining popularity over the last decade [7]. The idea of digital twins originated in the aerospace industry to replicate complex physical systems. These replicas (i.e., digital twins) were used for system integration and personnel training before the deployment of systems [8].

More recently, the use of digital twins has been increasing owing to the advances in simulation methods for engineering. Formally, a digital twin is a representation of a real-life object or system (such as a car, vessel or part of the equipment) in a digital format. Digital twins provide a possibility for interaction between the physical and the digital world. The typical application areas vary between manufacturing, robotics and the autonomous vehicle industry. In order to achieve intelligent machines, application frameworks such as Pipe Machining Production Line are being designed to visualize the system workflows [7]. However, most existing digital twins lack universality and cannot be easily integrated across different settings. They are either in the preliminary stages of development or target a specific problem area [9,10]. Although prior literature indicates that the concept is already being applied narrowly to specialized areas in the maritime industry, no clear method exists for creating broader interdisciplinary digital twins [11]. In this work, we present an end-to-end system for AMSVs using digital twin along with a pipeline for deployment based on obstacle detection and Deep Reinforcement Learning (DRL). Additionally, we present an application framework for the development of an AMSV using an integrated digital twin in a 3D simulation environment.

The rest of the paper is structured as follows. Firstly, Section 2 describes the related work in the field of AMSVs and digital twins. Next, Section 3 describes the application framework of the digital twin-driven AMSV system. Section 4 presents the experiments using digital twins for data collection and model training. Lastly, Section 5 concludes this paper.

## 2. Related Works

Autonomous vessel navigation aims to increase safety and efficiency at the seas. AMSVs have two main capabilities: situational awareness and autonomous path planning. Situational awareness refers to the ability to perceive the environment. That is achieved through sensors such as cameras, LiDARs, IMU, GPU and RADAR. Autonomous path planning includes global and local route planning and optimization. Multi-sensor situational awareness and control have been thoroughly researched in the field of autonomous cars [12]. On the contrary, autonomous vessels are still an emerging area of research and development with a lot of untapped and unexplored potential.

The autonomous vessel research has received some attention in Northern Europe, specifically in Finland and Norway [13]. More recently, Japan, China, Korea and Singapore have also taken initiatives in this area [14]. Several ongoing industrial projects focus on the design and development of AMSVs. The MUNIN project [2] focuses on an autonomous cargo ship and autonomous vessel control. It introduces an autonomous ship controller (ASC), a Shore Control Centre (SCC) and an Advanced Sensor System (ASS). The AAWA project [15] provides a system of four modules- Situational Awareness (SA), Collision Avoidance (CA), Route Planning (RP) and Ship State Definition (SSD).

AI/ML approaches for autonomous vessel navigation have been the primary focus of academic research. While some papers focus on classical algorithms [16,17], more recently the interest has shifted towards DRL. In [14,18], the authors present a decision-making system for ASMVs based on DRL using the Rizhao port as a case study. The work in [19]

presents a DRL approach based on the Proximal Policy Optimization algorithm (PPO) for collision avoidance. In [20], the authors design a new reward function which takes the waiting time at the corner of the path as the optimization goal to minimize the total travel time of autonomous vessels. However, these works use traditional 2D simulation-based conditions that neglect the complex nature of the real world. While more recent works have incorporated 3D simulations in their experiments [21], they do not present an end-to-end AMSV system. In addition, most of these works use prerecorded data, which limits their accuracy and efficacy. Existing research works face the challenge of incomplete or oversimplified perception of the environment, which limits their applicability to real-life scenarios.

It is necessary to create an exact replica of the vessel for more accurate experimental evaluations in addition to 3D simulations. This replica, called a digital twin, simulates the behaviours of its physical counterpart. Digital twin vessel literature indicates that this concept is being applied to highly focused areas such as structural fatigue estimation [22], manufacturing systems [7], condition monitoring [23] and personnel training [24,25]. In [11], digital twins are categorized into three constituents: asset representation, behavioural models, and measured data [26,27]. Asset representation is a 3D model usually consisting of CAD files complemented with metadata. Behavior models bridge the gap between digital and physical representations using physics or statistics. The measured data category refers to the injection of live (online) or prerecorded (offline) data into the simulation environment. However, these implementations lack the incorporation of live data, AI/ML training and evaluation and 3D modelling in a single system. Moreover, they lack a comprehensive implementation framework to bridge the gap between the real and the simulated vessel. Therefore, we present an end-to-end application framework for a digital twin AMSV which is cohesively tied to the real vessel in a 3D, replicated simulation environment.

### 3. Application Framework of Digital Twin-Driven AMSV System

An innovative application framework for a digital twin-driven AMSV system is shown in Figure 1. The application framework consists of four layers: physical layer, digital twin layer, data layer and application layer. The physical layer and digital twin layer are cognates concerning sensor configuration, control structure and dynamics of the environment. This replication provides a unique opportunity to collect data from the digital twin layer and augment it with offline data collected from the physical layer. The training and evaluation of machine learning models are carried out in the digital twin layer before deploying them on the physical layer. After the initial verification, the models are moved to real-world testing. Next, we discuss the function of these layers in detail.

### 3.1. Physical Layer

A high-level overview of the physical layer can be seen in Figure 2. The physical layer consists of the environment and the vessel. The environment refers to the vessel's surroundings including other vessels, sea marks etc. For this paper, we consider the area of Turku, Finland. Our Åboat vessel is built on the platform of a 375fc FoldCat Inflatable Fishing Boat with a bow and stern trolling electric motors of type MotorGuide Xi5 55SW54-GPS. Each motor can be controlled by setting its angle (from $-180$ to $180$ degrees) and thrust ($0$–$100$ percent of maximum) values. The motor rotation speed is around 4 seconds for a complete 360-degree turn which is approximately 90 deg/s. Detailed characteristics and technical specifications of the Åboat are listed in Table 1.

The Åboat sensors are interfaced with an Nvidia Jetson™ AGX Xavier™ Carrier Board [28]. It serves as the main processor on the vessel which communicates with a PC on the shore over 4G. The shore PC has a GeForce GTX 1080 for handling high compute loads from high-dimensional sensors (camera and LiDAR). The application of AI/ML techniques on the data from these sensors incorporates situational awareness into the Åboat. The Åboat sensors include four 2MP cameras which are a part of SurveilsQUAD—A synchronized multi-camera system developed by Sony. The system is designed specifically

for NVIDIA ® Jetson AGX Xavier containing a baseboard that interfaces with the Xavier and camera boards. The cameras provide high-resolution RGB images of the environment. A mid-range LiDAR has been used to provide distance and depth information. A GNSS has been used to determine the position of the vessel while an IMU calculates the speed and heading. Additionally, this layer has the control unit which manoeuvres the vessel. It translates the heading angle to two motor angles to follow the desired path. This part of the physical layer can be mapped to the path planning module of the application layer.
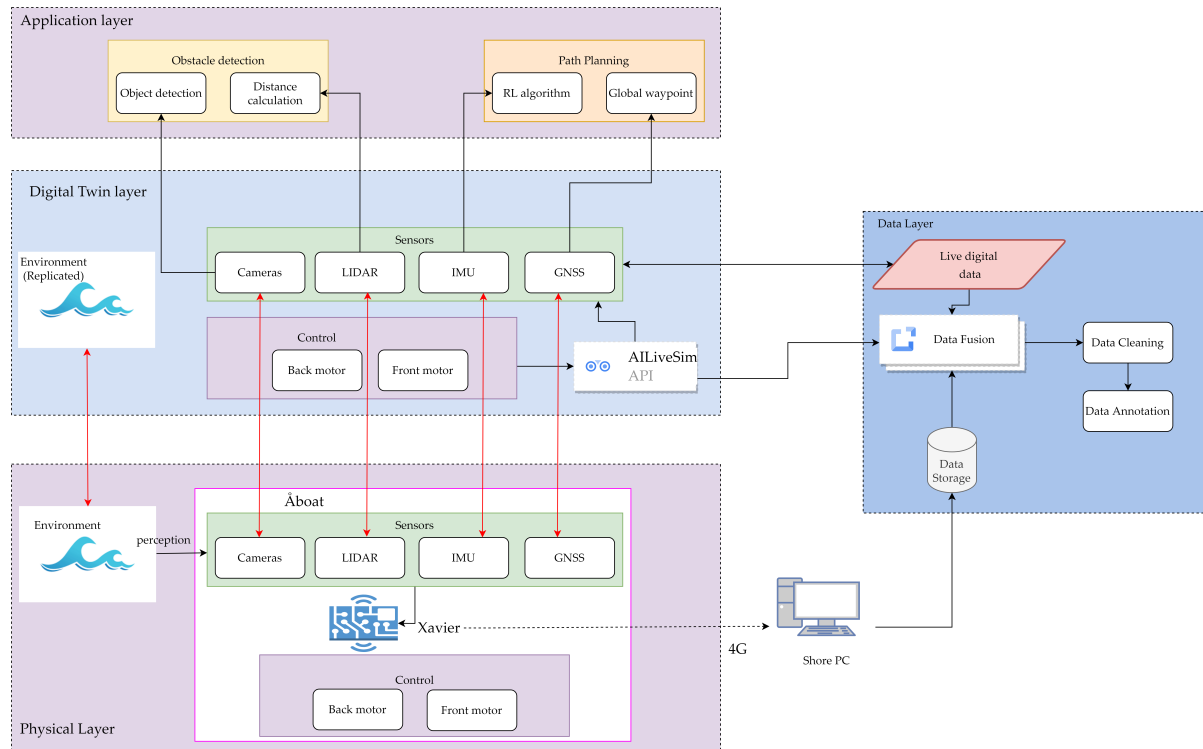


**Figure 1.** Application framework for the digital twin-driven AMSV system. The framework consists of four layers to create a cohesive and integrated digital twin system that accelerates model training and deployment for AMSVs.
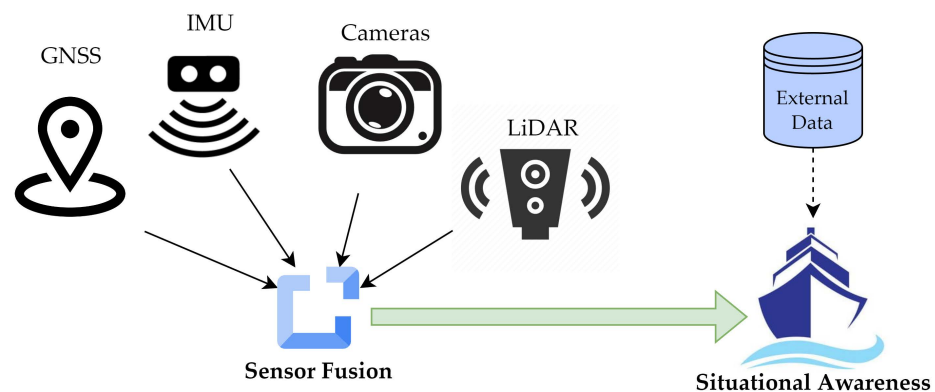


**Figure 2.** High-level Overview of the physical layer. The sensors provide information about the environment to incorporate situational awareness in the vessel.

**Table 1.** Technical specification of the Åboat.

| Items | Characteristics |
|---|---|
| Dimension (L × W × H) | 3.75 m × 1.37 m × 0.44 m |
| Hull Weight | ∼34 kg |
| Load Capacity | 2 Adults or 295 kg |
| Tube Diameter | 0.42 m |
| Onboard Sensors | LiDAR, IMU, 4 Cameras, GPS |
| Motors | 2 MotorGuide Xi5 55SW54-GPS |
| Thrust | 25 kg |

### 3.2. Digital Twin Layer

In this work, we focus on the incorporation of a digital twin layer in traditional AMSV systems. In order to create an integrated and effective digital twin layer, it is necessary to involve both elements of the physical layer: the vessel and the environment. To the best of our knowledge, while most works replicate the vessel to provide a physical or functional representation for training AI/ML algorithms, they discard useful information by neglecting the physical environment. Therefore, similar to the physical layer, our proposed digital twin layer consists of two elements- the digital twin vessel and the replicated digital environment.

#### 3.2.1. Digital Twin of Åboat

The digital twin of the Åboat is a 3D model in a simulation tool, AILiveSim [29]. The vessel architecture and sensor configuration have been replicated as illustrated in Figure 3. The sensors of the digital twin have been configured to reflect those in the physical layer. For example, the resolution, frames per second (FPS) and field-of-view (FOV) angle have been adjusted to match the Full HDR cameras. Similarly, the width and height of the scan area, rotations per second, number of samples taken in horizontal and vertical directions, laser wavelength and length of the scan rays of the LiDAR also match the configurations in the physical layer.



(**a**)  (**b**)  (**c**)
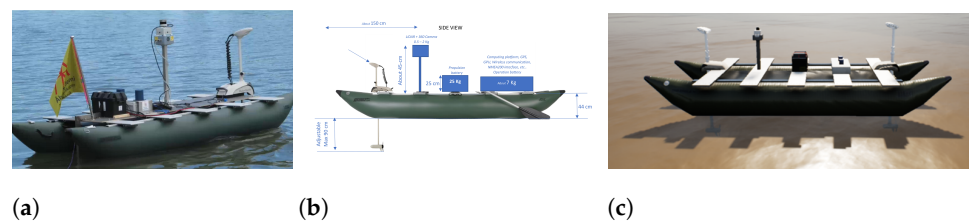
**Figure 3.** Åboat in real-world and AILiveSim simulation. (**a**) Åboat; (**b**) Side view of Åboat; (**c**) Åboat in AILiveSim.

#### 3.2.2. Replicated Digital Environment

The complexities and dynamics of the real-world environment must be taken into consideration at every stage of the design process. These complexities, including the stochastic nature of the environment, are often overlooked in research works focusing on developing AI/ML systems for AMSVs. Our work highlights the importance of incorporating the real-life environment in simulation. For this paper, our primary focus has been around duplicating the waterway route in Turku, Finland. The 3D geometric, physical and kinematic models of the environment as well as the Åboat have been encapsulated into the digital twin in AILiveSim by the software vendor. The GUI provides an integrated and intelligent environment for the efficient training and testing of algorithms for AMSVs with live data from the tool. It is important to note that the distributions of simulated and real-life data are the same. Moreover, with the addition of noise, the simulated data mirror their real-life counterpart.

### 3.3. Data Layer

The data layer is responsible for data management. Data management includes, but is not limited to, data storage, data processing, data annotation, data encoding, data analysis and data fusion. This layer consists of data from sensors in the physical layer as well as the digital twin layer. We collect sensor data from the digital twin layer, annotate them and fuse them with real-life data to train our models in the application layer. The data layer can provide the integration and interoperability of the physical and digital twin layers by merging these two sets of data.

### 3.4. Application Layer

The application layer consists of applications to implement the autonomous functions of an AMSV. In order to achieve autonomous operations, it is essential to move towards the destination while avoiding possible obstacles along the way. Therefore, the application layer can be divided into two main modules: obstacle detection and path planning.

#### 3.4.1. Obstacle Detection

The obstacle detection module detects and localizes obstacles. It uses cameras to detect obstacles (object detection) and LiDAR for distance calculation. We use YOLO (You Only Look Once) [30] to detect the obstacles in RGB images streamed by the cameras. The output of this algorithm is in the form of bounding boxes around the obstacles. Moreover, each bounding box is categorized into two categories: dynamic obstacle (vessels) and static obstacle (buoys and rocks). This algorithm provides high inference speed along with good detection accuracy [31,32]. After detecting the bounding boxes around the obstacles, LiDAR is utilized for distance estimation. LiDAR and cameras complement each other and provide more accurate results than just using a single sensor [15]. A comparison of the two sensors is shown in Table 2.

**Table 2.** Comparison of Camera and LiDAR for Maritime Situational.

|  | **Visual HD Cameras** | **LiDAR** |
|---|---|---|
| Spatial Accuracy | Accurate | Accurate |
| Field of view | Limited | Limited |
| Distance measurement | No | Accurate |
| Object identification | Accurate | Limited |
| Availability, any weather | No | Limited |
| Computational load of analysis | No | No |
| Marine robustness | Yes | No Data |

#### 3.4.2. Path Planning

The path planning module finds a collision-free path to the destination. For navigation purposes, the path to the goal is represented as a series of waypoints. These waypoints serve as intermediate goals. Path planning is done in two stages: global planning and local planning. The global planner calculates static waypoints to the destination using the available information about the environment. However, the AMSV can encounter unforeseen obstacles along its way. This is where the local planner is used. When an obstacle has been detected within the safety radius of the AMSV, the local planner takes over the control and guides the vessel until the collision threat disappears. The local planner uses DRL to achieve this. DRL aims to learn intelligent behaviors through trial and errors by interacting with the environment. DRL has greatly enhanced navigation in related fields. Therefore, we focus on this approach instead of classical ones. Each interaction with the environment provides information to the algorithm which can be used to identify the suitable course of action. Once the vessel is aware of its speed, location, heading and obstacles in its surroundings, this information is used to train a DRL agent.

### 3.4.3. Problem Formulation

We use the digital twin to train the DRL algorithm. It provides a safer and more robust way of training and evaluating the agent. The following must be defined in a DRL problem: agent, action, observation and reward. In our case, the agent is the vessel which would take actions in our simulation environment. The agent observes the environment. This observation is an input for the DRL algorithm which provides an action as an output. Based on the action, the agent gets a reward.

### 3.4.4. Observation Space and Action Space

The vessel receives an observation vector from the environment (based on live data in simulation) which includes its distance from the destination, the next waypoint and the obstacle (if detected) as well as the angle to the obstacle, waypoint and destination. These values have been normalized to facilitate training and avoid neural network saturation. These distances have been marked in Figure 4.
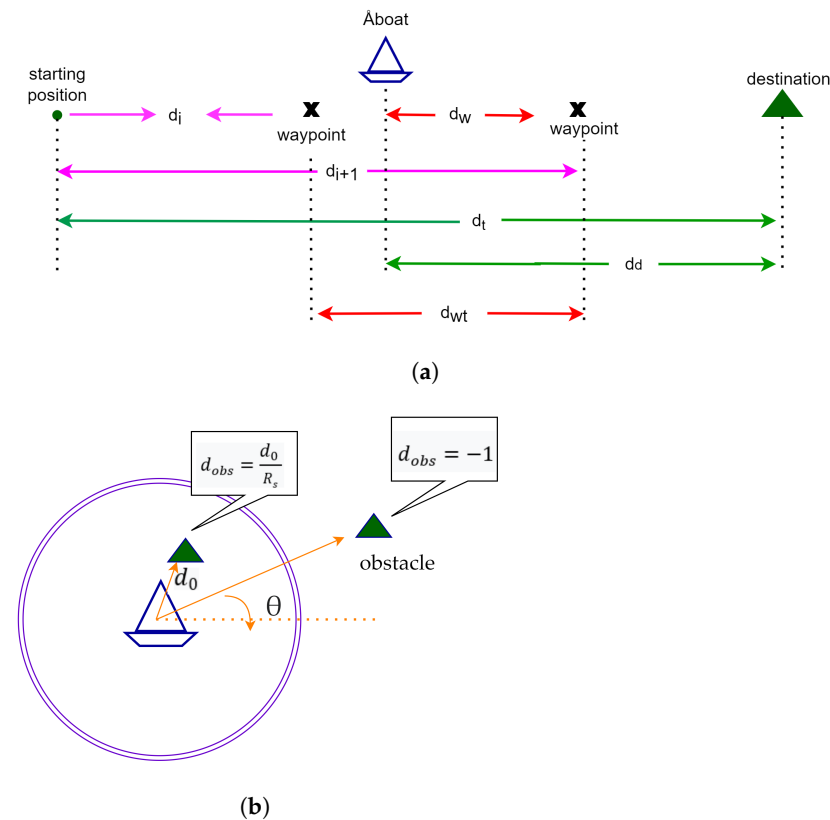
(a)

(b)

**Figure 4.** Distance of Åboat from the obstacles, waypoints and destination. (**a**) Distance from destination and waypoints; (**b**) Distance from obstacles.

The distance to the destination $d_d$ has been normalized between 0 and 1 by dividing it by the total distance to the destination $d_t$ as follows:

$$d_g = \frac{d_d}{d_t},\tag{1}$$

Similarly, the distance to the next waypoint $d_w$ is normalized by dividing it by the total distance between the previous waypoint $d_i$ and the next one $d_{i+1}$ as follows

$$d_{wp} = \frac{d_w}{d_{i+1} - d_i},\tag{2}$$

For the obstacle, it is not possible to normalize the distance as shown above. Therefore, we consider the safety radius of the vessel. The safety radius $R_s$ is the minimum distance at which the local planner is engaged. We normalize the distance to obstacle $d_o$ as follows:

$$d_{obs} = \begin{cases} \frac{d_o}{R_s} & d_o \leq R_s \\ -1 & d_o > R_s \end{cases} \tag{3}$$

Note that $d_{obs}$ is set to -1 when the object is outside the safety radius of the vessel to indicate that a penalty would not be assigned with respect to this obstacle unless it enters the safety radius. The angles to the obstacle, waypoint and destination are calculated using 3D vector calculus. For normalisation, we consider their cosines as follows:

$$\cos \angle(\vec{P}_1, \vec{P}_2) = \frac{\vec{P}_1 \cdot \vec{P}_2}{|P_1||P_2|} \tag{4}$$

The action space consists of an angle which corresponds to the angle at which the vessel is turned. In order to simplify the problem, the vessel's speed is fixed during our experiments. While the vessel can take a 360° turn, in practice a full range is not needed. Therefore, we restrict the action space between $-45°$ and $45°$. Note that we consider a continuous action space.

3.4.5. Reward Function

The reward function guides the vessel to avoid collisions and reach the goal. It considers several factors. Based upon the distance to the destination $d_g$, the distance to the next waypoint $d_w$ and the distance to the obstacle $d_{obs}$, the reward function $R$ is defined by the following equation:

$$R = 100.d_g + 50.d_w + R_o \tag{5}$$

$$R_o = \begin{cases} 20d_{obs} & d_o \leq R_s \\ 0 & d_o > R_s \end{cases} \tag{6}$$

3.4.6. DRL Algorithm

Continuous control tasks such as the navigation considered in this paper are approached through actor-critic based DRL methods. An actor (policy) identifies the course of action, while a critic evaluates how good the action was by calculating Q-values. In this work, we use a state-of-the-art DRL algorithm, Soft Actor-Critic (SAC) [33]. SAC is an algorithm that aims to maximize the reward (the standard objective in RL problems) along with the entropy. Entropy is a measure of the randomness of the policy. By encouraging a higher entropy, we encourage the agent to explore different states, which makes the algorithm more robust. In SAC, deep neural networks are used to parameterize the actor and critic functions. The actor is a Gaussian policy whose mean and diagonal covariance are produced by a neural network. The critic consists of double Q-networks to reduce the variance in Q values. We refer the reader to the original paper for more details on the algorithm.

**4. Experiments and Results**

In this section, we present experiments using the digital twin for data collection and model training. First, we discuss the experiments and results for training obstacle detection algorithms with the addition of simulated data. Then, we present the model training and evaluation results for the DRL algorithm for live simulated data in AILiveSim.

*4.1. Obstacle Detection*

For obstacle detection, YOLO was trained on the combination of a publicly available dataset, ABOShips [34], along with simulated data collected from AILiveSim, SimuShips, which is available from [35]. The dataset contains simulated images under different weather,

visibility, occlusion and lighting conditions, as shown in Figure 5. Table 3 provides the details of data distribution within the training and test set.



**Figure 5.** Images from SimuShips dataset

**Table 3.** Dataset for training the object detection model. We combine real images from Turku River from ABOShips with simulated ones.

|  | Our Dataset SimuShip | ABOShips | Combined Dataset |
|---|---|---|---|
| Total Images | 9471 | 9880 | 19,351 |
| Images in Training Set | 7576 | 8073 | 15,479 |
| Images in Test Set | 1895 | 1807 | 3871 |

We compared model performance on real-life data with and without the augmentation of simulated data. Model A is the YOLO model trained on ABOShips only. Model B is the same model trained on the combined dataset. We use recall as the evaluation metric for the models. In simple words, 'recall' tells us about how many obstacles our model was able to capture as compared to all the obstacles present in the data. Since overlooking obstacles in the maritime domain is of the most critical nature, having a higher recall leads to safer navigation. The augmentation of real-life data with our dataset during training led to a 2.9% increase in recall, as indicated by the results in Table 4.

**Table 4.** Effect of simulated data addition on object detection. The model's recall improved to 0.577 from 0.561, leading to a reduction in false positive predictions).

| Class | Number of Occurrences | Recall | |
|---|---|---|---|
|  |  | Model A | Model B |
| All | 8362 | 0.561 | 0.577 |
| Dynamic Obstacle | 6776 | 0.653 | 0.667 |
| Static Obstacle | 1586 | 0.468 | 0.486 |

### 4.2. Path Planning

In order to train and evaluate the path planning module, we used the Turku scenario in AILiveSim. First, we placed multiple waypoints to simulate the trajectory of the global path planner. Then, we introduced an obstacle in the middle of the path. The experimental setup is shown in Figure 6.

**Figure 6.** DRL model training setup in AILiveSim. The pink cylinders represent the global static waypoints while the rock is an obstacle amidst the planned trajectory. Note that the static waypoints are not recomputed when an obstacle is encountered.

In the training phase, we trained the DRL agent for 6000 episodes with a maximum of 400 steps in an episode. Live data from the simulation were used for training. Noise was added to the data to keep the data distribution close to real-life. During the training phase, we stored transitions (state, reward, next state, and action) in a replay buffer and sampled 128 samples at each step to train the agent. We updated the target networks every 25 steps. The rewards across training episodes are shown in Figure 7. The figure shows the SAC agent's progress during training. The agent learnt to follow waypoints and avoid obstacles which led to an increase in the average reward accumulated across episodes as indicated by the running averages. Our vessel learnt to avoid collision. The trajectories taken by the vessel after 500, 5000 and 50,000 steps of training have been visualized in Figure 8. The agent learns to avoid collisions and follow waypoints to get to the destination.
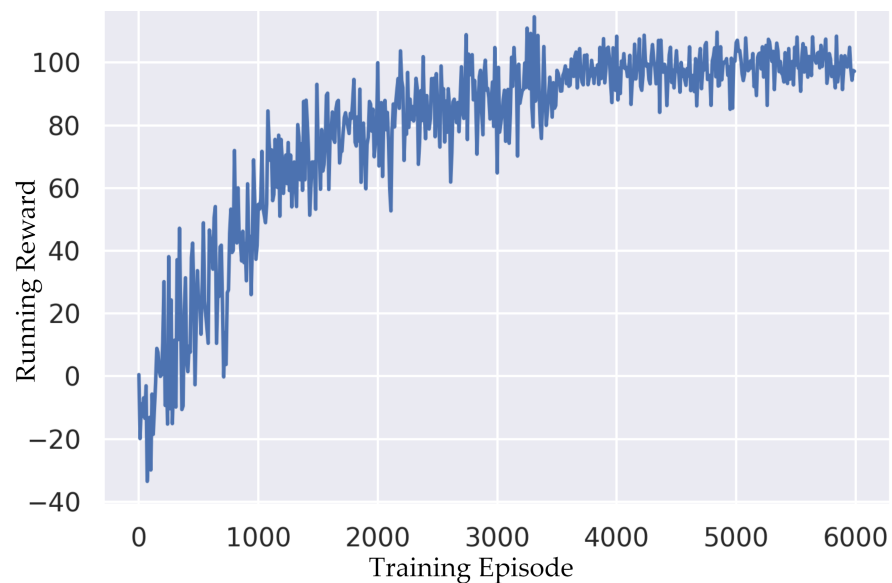


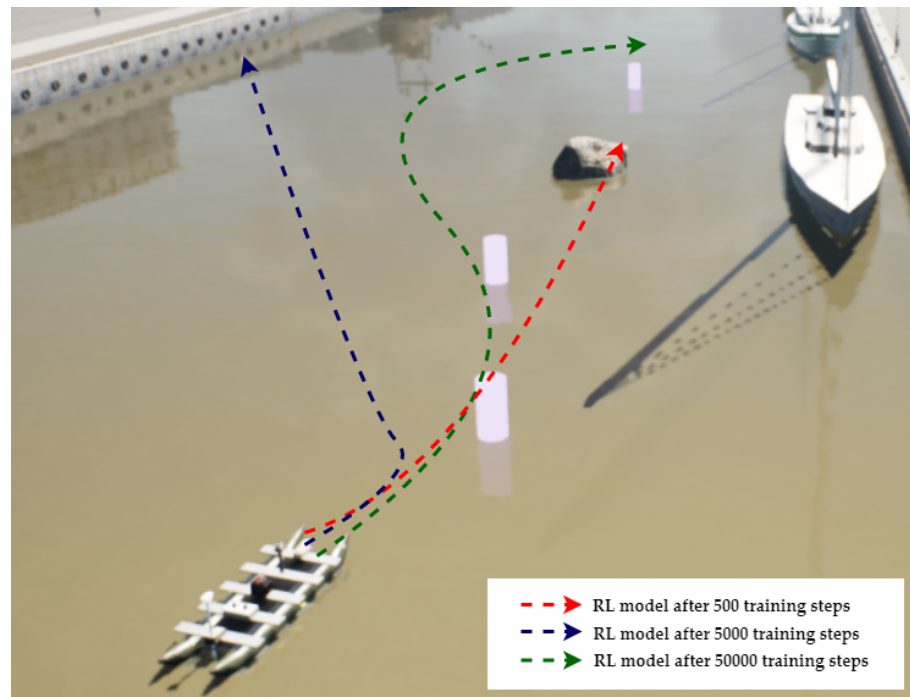**Figure 7.** Running reward across training episodes for the SAC Agent.

**Figure 8.** Trajectory Analysis for the SAC Agent.

## 5. Conclusions

In this paper, we presented an end-to-end application framework for AMSVs, which leverages integrated digital twins. This system has been applied and tested for our in-house development platform, Åboat, to perform autonomous obstacle detection and path planning. Through the use of a digital twin in a 3D replicated simulation environment, we showcased the possibility of collecting data from the simulation. Moreover, our integrated 3D digital twin enabled the creation of multiple scenarios for developing systems for AMSVs. Overall, we provided a proof-of-concept that digital twins are an agile and safe method for the design and development of AMSVs.

Our work opens possibilities in multiple directions. First, the digital twin can be enriched with additional information, such as sea charts and estimates around the wave disturbances, to optimize the path planning models. Second, data visualization applications can be added to the application layer. Third, field experiments and surveys can be conducted with digital twins to include more relevant information and increase the adoption of digital twins in the industry.

Exploiting the proposed application framework, the rewards across training episodes of a Deep Reinforcement Learning model were evaluated for live simulated data in AILiveSim.

At the current stage, our work only includes sensor data. In the future, we plan to extend the data sources as well as add more modules to the application layer. We also plan to explore more efficient methods for AMSVs using the developed framework.

**Author Contributions:** All the authors developed the initial concepts together and contributed to the planning of the methodology. M.R. designed the application framework for the AMSV system and the path planning module. H.P. analyzed sensor data from the digital twin. S.H. wrote scripts for the object detection models and deployment. S.L. and S.A. jointly supervised the project.All authors have read and agreed to the published version of the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wang, W.; Shan, T.; Leoni, P.; Fernández-Gutiérrez, D.; Meyers, D.; Ratti, C.; Rus, D. Roboat II: A novel autonomous surface vessel for urban environments. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2020; pp. 1740–1747.
2. Maritime Unmanned Navigation through Intelligence in Networks. Available online: http://www.unmanned-ship.org/munin/ (accessed on 28 May 2022).
3. Autonomous Ship Project, Key Facts about Yara Birkeland. Available online: https://www.kongsberg.com/maritime/support/themes/autonomous-ship-project-key-facts-about-yara-birkeland/ (accessed on 28 May 2022).
4. Autonomous Shipping. Available online: https://www.imo.org/en/MediaCentre/HotTopics/Pages/Autonomous-shipping.aspx (accessed on 28 May 2022).
5. Annual Overview of Marine Casualties and Incidents. 2021. Available online: http://www.emsa.europa.eu/newsroom/latest-news/download/6955/4266/23.html (accessed on 28 May 2022).
6. Wróbel, K.; Montewka, J.; Kujala, P. Towards the assessment of potential impact of unmanned vessels on maritime transportation safety. *Reliab. Eng. Syst. Saf.* **2017**, *165*, 155–169. [CrossRef]
7. Wu, Q.; Mao, Y.; Chen, J.; Wang, C. Application Research of Digital Twin-Driven Ship Intelligent Manufacturing System: Pipe Machining Production Line. *J. Mar. Sci. Eng.* **2021**, *9*, 338. [CrossRef]
8. Boschert, S.; Rosen, R.; Hehenberger, P.; Bradley, D. Mechatronic Futures. *Digital Twin—The Simulation Aspect*; Peter, H., David, B., Eds.; Springer: Cham, Switzerland, 2016; pp. 59–74.
9. Bolbot, V.; Theotokatos, G.; Hamann, R.; Psarros, G.; Boulougouris, E. Dynamic Blackout Probability Monitoring System for Cruise Ship Power Plants. *Energies* **2021**, *14*, 6598. [CrossRef]
10. Stoumpos, S.; Theotokatos, G.; Mavrelos, C.; Boulougouris, E. Towards Marine Dual Fuel Engines Digital Twins—Integrated Modelling of Thermodynamic Processes and Control System Functions. *J. Mar. Sci. Eng.* **2020**, *8*, 5140. [CrossRef]
11. Fonseca, Í.A.; Gaspar, H.M. Challenges when creating a cohesive digital twin ship: A data modelling perspective. *Ship Technol. Res.* **2021**, *68*, 70–83. [CrossRef]
12. Wang, Z.; Wu, Y.; Niu, Q. Multi-Sensor Fusion in Automated Driving: A Survey. *IEEE Access* **2019**, *8*, 2847–2868. [CrossRef]
13. Brekke, E.F.; Wilthil, E.F.; Eriksen, B.O.H.; Kufoalor, D.K.M.; Helgesen, Ø.K.; Hagen, I.B.; Breivik, M.; Johansen, T.A. The Autosea project: Developing closed-loop target tracking and collision avoidance systems. *J. Phys. Conf. Ser.* **2019**, *1357*, 012020. [CrossRef]
14. Zhang, X.; Wang, C.; Liu, Y.; Chen, X. Decision-Making for the Autonomous Navigation of Maritime Autonomous Surface Ships Based on Scene Division and Deep Reinforcement Learning. *Sensors* **2019**, *19*, 4055. [CrossRef] [PubMed]
15. Thombre, S.; Zhao, Z.; Ramm-Schmidt, H.; Vallet García, J.M.; Malkamäki, T.; Nikolskiy, S.; Hammarberg, T.; Nuortie, H.; Bhuiyan, M.Z.; Särkkä, S.; et al. Sensors and AI Techniques for Situational Awareness in Autonomous Ships: A Review. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 64–83. [CrossRef]
16. Mousazadeh, H.; Jafarbiglu, H.; Abdolmaleki, H.; Omrani, E.; Monhaseri, F.; Abdollahzadeh, M.r.; Mohammadi-Aghdam, A.; Kiapei, A.; Salmani-Zakaria, Y.; Makhsoos, A. Developing a navigation, guidance and obstacle avoidance algorithm for an Unmanned Surface Vehicle (USV) by algorithms fusion. *Ocean Eng.* **2018**, *159*, 56–65. [CrossRef]
17. Liu, T.; Dong, Z.; Du, H.; Song, L.; Mao, Y. Path following control of the underactuated USV based on the improved line-of-sight guidance algorithm. *Pol. Marit. Res.* **2017**, *24*, 3–11. [CrossRef]
18. Wang, C.; Zhang, X.; Li, R.; Dong, P. Path Planning of Maritime Autonomous Surface Ships in Unknown Environment with Reinforcement Learning. In *Proceedings of the Cognitive Systems and Signal Processing*; Sun, F., Liu, H., Hu, D., Eds.; Springer: Singapore, 2019; pp. 127–137.
19. Artusi, E.; Chaillan, F.; Napoli, A. Path planning for a maritime suface ship based on Deep Reinforcement Learning and weather data. In Proceedings of the OCEANS 2021, San Diego, CA, USA, 20–23 September 2021; pp. 1–8. [CrossRef]
20. Wu, D.; Lei, Y.; He, M.; Zhang, C.; Ji, L. Deep Reinforcement Learning-Based Path Control and Optimization for Unmanned Ships. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 7135043 . [CrossRef]
21. Yan, N.; Huang, S.; Kong, C. Reinforcement Learning-Based Autonomous Navigation and Obstacle Avoidance for USVs under Partially Observable Conditions. *Math. Probl. Eng.* **2021**, *2021*, 5519033 . [CrossRef]
22. Caspeele, R.; Taerwe, L.; Frangopol, D.M. *Life Cycle Analysis and Assessment in Civil Engineering: Towards an Integrated Vision: Proceedings of the Sixth International Symposium on Life-Cycle Civil Engineering (IALCCE 2018), Ghent, Belgium, 28–31 October 2018*; CRC Press: Boca Raton, FL, USA, 2018; Volume 5.
23. Lindstad, E.; Bø, T.; Eskeland, G. MARINE DESIGN XIII. In *Reducing GHG Emissions in Shipping—Measures and Options*; Kujala, P., Lu, L., Eds.; CRC Press: Boca Raton, FL, USA, 2018; pp. 923–930.
24. Tofte, B.L.; Vennemann, O.; Mitchell, F.; Millington, N.; McGuire, L. How digital technology and standardisation can improve offshore operations. In Proceedings of the Offshore Technology Conference, OnePetro, Houston, TX, USA, 16–19 August 2019.
25. Dufour, C.; Soghomonian, Z.; Li, W. Hardware-in-the-loop testing of modern on-board power systems using digital twins. In Proceedings of the 2018 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), Amalfi, Italy, 20–22 June 2018; pp. 118–123.

26. Cameron, D.B.; Waaler, A.; Komulainen, T.M. Oil and Gas digital twins after twenty years. How can they be made sustainable, maintainable and useful? In Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59), Oslo, Norway, 26–28 September 2018.
27. Cabos, C.; Rostock, C. Digital Model or Digital Twin? In Proceedings of the 17th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'18), Pavone, Italy, 14–16 May 2018.
28. Deploy AI-Powered Autonomous Machines at Scale. Available online: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/ (accessed on 28 May 2022).
29. Leudet, J.; Christophe, F.; Mikkonen, T.; Männistö, T. AILiveSim: An Extensible Virtual Environment for Training Autonomous Vehicles. In Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, 15–19 July 2019; Volume 1, pp. 479–488. [CrossRef]
30. YOLOv5 Documentation. Available online: https://docs.ultralytics.com/ (accessed on 28 May 2022).
31. Choudhari, V.; Phadtare, M.; Pedram, R.; Vartak, S. Comparison between YOLO and SSD MobileNet for Object Detection in a Surveillance Drone. *Int. J. Sci. Res. Eng. Man* **2021**, *5*, 51688. [CrossRef]
32. Iyer, R.; Bhensdadiya, K.; Ringe, P. Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for Real-Time Mask Detection. *Int. J. Res. Eng. Technol.* **2021**, *8*, 1156–1160.
33. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International conference on machine learning. PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
34. Iancu, B.; Soloviev, V.; Zelioli, L.; Lilius, J. ABOships—An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations. *Remote Sens.* **2021**, *13*, 988. [CrossRef]
35. Raza, M.; Hanna, P.; Samir, H.; Azimi, S.; Lafond, S. SimuShips—A High Resolution Simulation Dataset for Ship Detection with Precise Annotations. *Zenodo* 2022, *preview*. [CrossRef]