Journal of *Marine Science and Engineering*

MDPI

Article

Towards Realizing Intelligent Coordinated Controllers for Multi-USV Systems Using Abstract Training Environments

Sulemana Nantogma [1], Keyu Pan [1], Weilong Song [2], Renwei Luo [1] and Yang Xu [1,*]

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; 201814080012@std.uestc.edu.cn (S.N.); 201822080224@std.uestc.edu.cn (K.P.); 202022080430@std.uestc.edu.cn (R.L.)

[2] China North Vehicle Research Institute, No. 4 Huaishuling, Fengtai District, Beijing 100072, China; songweilong8896@gmail.com

* Correspondence: xuyang@uestc.edu.cn

Abstract: Unmanned autonomous vehicles for various civilian and military applications have become a particularly interesting research area. Despite their many potential applications, a related technological challenge is realizing realistic coordinated autonomous control and decision making in complex and multi-agent environments. Machine learning approaches have been largely employed in simplified simulations to acquire intelligent control systems in multi-agent settings. However, the complexity of the physical environment, unrealistic assumptions, and lack of abstract physical environments derail the process of transition from simulation to real systems. This work presents a modular framework for automated data acquisition, training, and the evaluation of multiple unmanned surface vehicles controllers that facilitate prior knowledge integration and human-guided learning in a closed-loop. To realize this, we first present a digital maritime environment of multiple unmanned surface vehicles that abstracts the real-world dynamics in our application domain. Then, a behavior-driven artificial immune-inspired fuzzy classifier systems approach that is capable of optimizing agents' behaviors and action selection in a multi-agent environment is presented. Evaluation scenarios of different combat missions are presented to demonstrate the performance of the system. Simulation results show that the resulting controllers can achieved an average wining rate between 52% and 98% in all test cases, indicating the effectiveness of the proposed approach and its feasibility in realizing adaptive controllers for efficient multiple unmanned systems' cooperative decision making. We believe that this system can facilitate the simulation, data acquisition, training, and evaluation of practical cooperative unmanned vehicles' controllers in a closed-loop.

Keywords: unmanned surface vehicles; training system; intelligent autonomous systems; fuzzy learning classifier systems; multi-agent systems; artificial immune system; reinforcement learning

Citation: Nantogma, S.; Pan, K.; Song W.; Lu, R.; Xu, Y. Towards Realizing Intelligent Coordinated Controllers for Multi-USV Systems Using Abstract Training Environments. *J. Mar. Sci. Eng.* **2021**, *9*, 560. https://doi.org/10.3390/jmse9060560

Academic Editor: David Moreno-Salinas

Received: 19 April 2021
Accepted: 18 May 2021
Published: 22 May 2021

1. Introduction

Intelligent decision-making and the coordinated control of multiple unmanned systems such as unmanned surface vehicles (USVs), unmanned aerial vehicles (UAVs), unmanned ground vehicles (UGVs), and unmanned underwater vehicles (UUVs) have become an intense research area because of their high performance, efficiency and extensive application potentials. With the increasing development and application of unmanned systems, intelligent decision-making and the coordinated behavior of these systems are receiving more attention [1]. For unmanned systems to be able to reach a maximum level of autonomy, they must be able to make decisions under unpredictable situations taking into considerations its performance constraints and other vehicles acting in the environment. However, a prior specification of the optimal and robust coordination is difficult due to the complexity and dynamics of their operating environment and missions. The situations in the operation environment of these systems can change rapidly. Hence, it is imperative for these systems to adapt their decision-making strategy to accommodate changes in the

environment. One way to achieve robust coordinated behaviors and dynamism for these systems on their various missions is training and learning using virtual environments dovetail with machine learning algorithms. By integrating machine learning in virtual environments with human-guided learning and guided interventions, multi-agent systems can be trained on mission specific scenarios that closely resemble the physical environment of the unmanned systems. Moreover, data which can be valuable in the training and evaluation of these systems using state-of-the-art methods such as deep-reinforcement learning techniques are scarce, and it is difficult to obtain experimental data in a real environment since training and/or testing can be costly and sometimes dangerous in the physical environments or using the real systems. In order to obtain large chunks of operational data and optimized decision making, abstract (virtual) environments are powerful tools that can facilitate this process by allowing machine learning algorithms to manipulate parameters, store data and visualize results. They can provide cost-effective and risk-free training and testing mechanisms while advancing research and development with humans in the loop. By using abstract environments with machine learning approaches, agents can be built as physical controllers and trained in settings and scenarios that would have been either too costly or too difficult to replicate in the field. Specifically, this study envisions a multi-USV training system for realizing autonomous coordinated control in a variety of maritime applications such as escort missions, search and destroy, maritime patrol missions, etc., as the rapid growth of maritime activities extends multi-USV research for applications in civilian [2–5] and military services [6–8]. USVs are marine vessels capable of performing various marine operations with no crew on-board in a variety of complex and dynamic ocean environments. In comparison with other unmanned systems, the USV offers a significant number of advantages such as resources localization [9], the use of traditional communication capabilities [10], as well as payload and energy capacities [11].

In the literature, several attempts have been made towards the development of multi-USV systems for various maritime missions. Advances in areas such as the use of statistical or machine learning techniques to derive knowledge from data as well as through qualitative logic-based approaches [12] can facilitate intelligent or robust behavior realization of unmanned systems, especially in multi-agent missions. Indeed, recent designs approaches for facilitating the execution of autonomous unmanned systems missions employs simulations for modeling and qualitative logic-based approaches as well as machine learning techniques such reinforcement learning and bio-inspired approaches for control systems design. For instance, intelligent decision-making for multiple unmanned vehicles using genetic fuzzy trees is presented by the authors in [13]. The proposed system, as demonstrated in combat scenarios was capable of obtaining strategies that are robust, aggressive and responsive against opponents. A localization framework for underwater robotic swarms to dynamically fuse multiple position estimates of an autonomous underwater vehicle while using a fuzzy decision support system is presented in [14]. The authors in [15] used deep q-neural networks to obtain combat strategies in an attack–defense pursuit–warfare of multiple unmanned systems in a simplified environment. A heuristic planning approach for guarding a valuable asset by a team of USVs operating in a continuous state-action space is presented in [16]. By evolving planning decision trees, they succeeded in automatically generating decision trees expressing a blocking policy for the USVs. In contrast, our approach allows the actual organization and optimization of behaviors peculiar to the mission. These behaviors are provided by the designer and fined tuned using immune network dynamics and clonal selection. On the other hand, refs. [16,17] employs deep reinforcement learning for path planning and the formation of USVs. In relation to one of the scenarios being considered in this work, ref. [18] employs deep reinforcement learning in order to train a team of escorts to maintain payload safety while navigating alongside the payload. We extend this case to the complex maritime environment with increased complexity in input and output space. While minimizing assumptions and prior knowledge may also result in realizing more robust controllers, the complexity of the learning process is greatly reduced when prior knowledge is utilized in complex multi-agent problems

peculiar to multi-USV missions. The authors in [19] present a multi-agent based intelligent training system for USVs where the authors proposed the use of genetic fuzzy trees [13] to realize controllers for the multi-USV system. The fuzzy tree is a tree-based algorithm with branches, where each branch handles a sub-task of the control problem and employs genetic algorithms for optimization. On the contrary, in this work, behaviors and sub-tasks are independently defined while immune dynamics are employed for behavior activation and action selection.

On the other hand, ref. [20] presents an overview and comparative study of free simulation software for mobile robots and concluded that simulators supporting USV operations as compared to other platforms are lacking. This claim was validated in a more recent work in [21] where the authors evaluated several realistic simulators and presented a simulation environment integrated with robotic middleware which models the forces that act on a USV in a disaster scenario. The authors in [22,23] present a simulation strategy and experimental design for developing and testing controllers for UAVs and USVs coordination with the aim of significantly reducing development and delivery times by providing an off-the-shelf simulation environment and a step-by-step implementation guideline. A virtual RobotX simulation capable of approximating the behavior of USVs operating in complex ocean environments is presented in [24]. Moreover, the authors in [25] developed a platform to model and visualize the behavior of marine vehicles in three-dimensional space for surface and subsurface applications.

To contribute to the available literature and provide simulation support for the emerging domain of autonomous combat USVs, this work presents a modular framework for the automated training, simulation and evaluation of multiple unmanned surface vehicles controllers that facilitate prior knowledge integration and human-guided learning through designer-provided behaviors in a closed loop. To realize this, we first present a digital maritime environment of multiple unmanned surface vehicles that abstracts the real-word dynamics in our application domain. Then, a behavior-driven artificial immune-inspired fuzzy classifier systems approach that is capable of optimizing agents' behaviors and action selection in a multi-agent environment is presented. More specifically, we modeled the training systems as artificial immune system with agents as organs and behavior models as immune cell containers of fuzzy learning classifier systems whose classifiers were modeled as B-Cells of the artificial immune system.

Learning classifier system is a machine learning approach that evolves a group of if–then rules by employing evolutionary machine learning to solve practical learning problems that is general enough for a wide range of tasks [26–30]. In fuzzy learning classifier system, which is an extension of the learning classifier system (LCS), classifiers are modeled as fuzzy rules and are applied to realize tactical behavior [31] of robotic systems. The LCS [32,33] concept in general has inspired a multitude of implementations adapted to manage the different problem domains to which it has been applied. On the other hand, various control problems have benefited from fuzzy if–then rules [31,34] with the advantage of easy comprehension as compared to 'blackbox' methods such as deep-reinforcement learning (DRL), dynamic programming and policy functions. Unlike research in DRL methods that focus on the training of networks in various problem settings, a fuzzy LCS framework in general trains classifiers (rules).

On the other hand, the artificial immune system is a typical multi-agent and decentralized information processing system, capable of learning and remembering, which was inspired by the working mechanisms exhibited by the biological immune system [35,36]. The dynamics exhibited by the biological immune system has inspired various theories and models which represent the different aspects proposed under the artificial immune system such as the immune network [37], clonal selection [38], and negative selection [39] and several applications have been demonstrated based on these theories [40].

The immune network theory is a critical theory of the artificial immune system which exhibits characteristics such as learning and memorizing in immune system. The immune network theory proposed by Jerne [37] suggests that the immune system is capable of

achieving immunological memory by the presence of a mutually reinforcing network of B-Cells by producing the interaction mechanism between the network cells. The interaction of cells happens regardless of the presence of harmful foreign agents. Jerne's theory stipulates that the antibody of an immune cell's epitope is recognized by a set of different antibodies (paratopes) with various levels of precision. The idiotope of one antibody can be recognized by the paratope of another antibody with or without the presence of an antigen that possesses an epitope (analogous to an idiotope). This recognition and interaction results in a network that is dynamic and leads to stimulation and suppression. The recognized antibody is suppressed while the recognizer antibody is simulated. In the robotics domain, ref. [36] proposed a computational model of Jerne's idiotypic network theory which has been notable as a means of inducing adaptive behavior mediation and has demonstrated some encouraging results. In these idiotypic networks, competence modules (antibodies) are linked not only to environmental stimuli (antigens) but also to each other, which leads to the formation of a dynamic chain of suppression and stimulation that affects their concentration levels globally.

On the contrary, negative selection abstracts an aspect of the immunological mechanism of organisms that deals with self-non-self-classification. This process of negative selection of B-Cells in the biological immune system involves the destruction of B-Cells that react against the 'self' and the promotion of B-Cells that attack only foreign agents. This is the underlying principle of the negative selection algorithms and their modifications [39].

On the other hand, clonal selection provides the immune system the ability to adapt B-Cells to new types of antigens. This adaptation is proportionate to the degree of matching between B-Cells and antigens. Hence, a stronger match causes a B-Cell to be cloned many times compared to a weaker match. Cloned B-Cells undergo mutation from the originals at a rate inversely proportional to the matched strength. This mechanism is the inspiration behind the artificial clonal algorithms [41] and their applications in different tasks and domains [42].

Comparatively, multi-agent coordinated control is similar in characteristics with those in the biological immune system (BIS), in that, there is the need for coordination and the adaptive control of agent's behaviors in a dynamic environment. In our approach, we combined immune-based methods with fuzzy classifier systems to find an appropriate amount of the suppression and simulation of behaviors in the architecture, in addition to learning the internal mechanism of each behavior so that behaviors are adaptive to the agent's environment. The contributions of this work are as follows:

- A generic framework for an autonomous unmanned systems training system design that supports operational data collection in a closed-loop was developed;
- We present a realistic abstract digital maritime environment for interactive multi-USV systems that can be used for multi-agent reinforcement learning;
- A behavior-driven immunized fuzzy classifier system approach for multi-USV coordinated intelligent control and decision-making is presented;
- We demonstrate the feasibility of our approach in realizing improved decision-making in multi-USV missions.

The rest of this work is organized as follows. Section 2 presents the background and motivation of this work. Section 3 presents the approach and system architecture of the training system. Section 4 presents the details of the digital maritime environment design and modeling. Section 5 details the training and learning approach and the experimental evaluation scenarios and results are presented in Section 6. Section 7 concludes this work.

2. Background and Motivation

The increasing number of maritime assets and infrastructures, ocean exploration and the military's need to operate in littoral and asymmetric warfare situations are all factors influencing USV research and development. This work is part of a larger project aimed at designing artificial general intelligence control systems for coordinating unmanned systems in multiple and complex missions. In particular, this paper focuses on tasks that requires

multiple USVs to coordinate in the guarding, selecting and intercepting hostile threats, performing intelligent combat maneuvers, surveillance and taking counter measures in the presences of several environmental disturbances. This is necessitated by the widening possible applications and happenings in the maritime domain. For instance, in January 2017, for the first time an unmanned surface vehicle was operated from a distance in a real warfare environment with its full operational capability being demonstrated when a vessel of the Saudi navy was damaged in an attack carried out by the Houthis using an unmanned suicide vehicle [7]. Prior to this, in 2000, a similar event occurred when the USS Cole-guided missile was a target of a terrorist attack during a refuel [8]. This attack was executed by a small fiberglass boat carrying *C*4 explosives. These two events could have been avoided by taking advantage of autonomous USVs. The main question is how does a USV employed for this mission become informed of the appropriate behaviors or actions to take under the different situations they will encounter on their missions? This process is called training.

Leveraging the potential and practicality of learning classifier systems and the artificial immune algorithms, a hybrid approach for multi-USV control learning is discussed and its applicability to realizing the coordinated control and decision making of a multi-USV system is presented. Ultimately, a modular multi-USV training framework for the automated training, simulation and evaluation of multi-USV systems in a closed loop was realized. Particularly, the training of USV is realized by designing behavior-driven fuzzy classifier systems whose working mechanism in this work was inspired by the biological immune system. Using prior behavior encoded by domain experts, appropriate primitive actions can be learned to realize robust internal behavior mechanisms. This way, we can accelerate the learning while reducing the computational requirements. In order to make the system's results a reference for real-system decision making, there is the need for a physics-based meta-model of the USVs and the environment in order to realize a digital maritime environment that abstracts the real process and constraints while conforming to the physical rules governing the multi-USV operations.

3. Multi-USVs Training System Design

In this section, we present the design and approach used to realize the training system. We first introduced the schematic framework of the system and then followed it with the component description.

System Architecture

As shown in Figure 1, this approach presents a layered architecture. In the first level from the top is the real or simulated environment of the controlled platforms. The second level, referred to as the USV (platform) abstraction layer, represents the core of the system as it provides the framework for the resources and data management of the vehicles in the environment. The third level is the immunized strategy and decision-making layer which provides the necessary mechanisms for behavior definitions and learning during training. The last level of the architecture is the optimization (behavior learning) layer which provides learning algorithms for the implementation of the appropriate behavior. The third and fourth layers together provide the training and learning mechanism. This is done by the creating, grouping and mapping of behaviors that provides the learning objective of an agent to the individual optimization algorithms. This design approach enables the implementation of different behavior controllers while enabling the implementation of several learning algorithms based on the scenario being modeled. This also enables centralized learning and distributed control implementation. Due to the high computational requirements of both machine learning algorithms and simulations, the modular design adopted allows for a distributed and parallel processing since each part of the system is designed as a separate process or subsystem. Different components interface and exchange data with other processes and subsystems through generic interface of sockets and/or a message passing interface. These design approaches ensure the effective evolution of data services

and control requirements of the different unmanned systems. The proposed framework offers the necessary interfaces for collecting data through a data engine and decision-making mechanism that interfaces with agents of the unmanned system for receiving observations and sending commands.

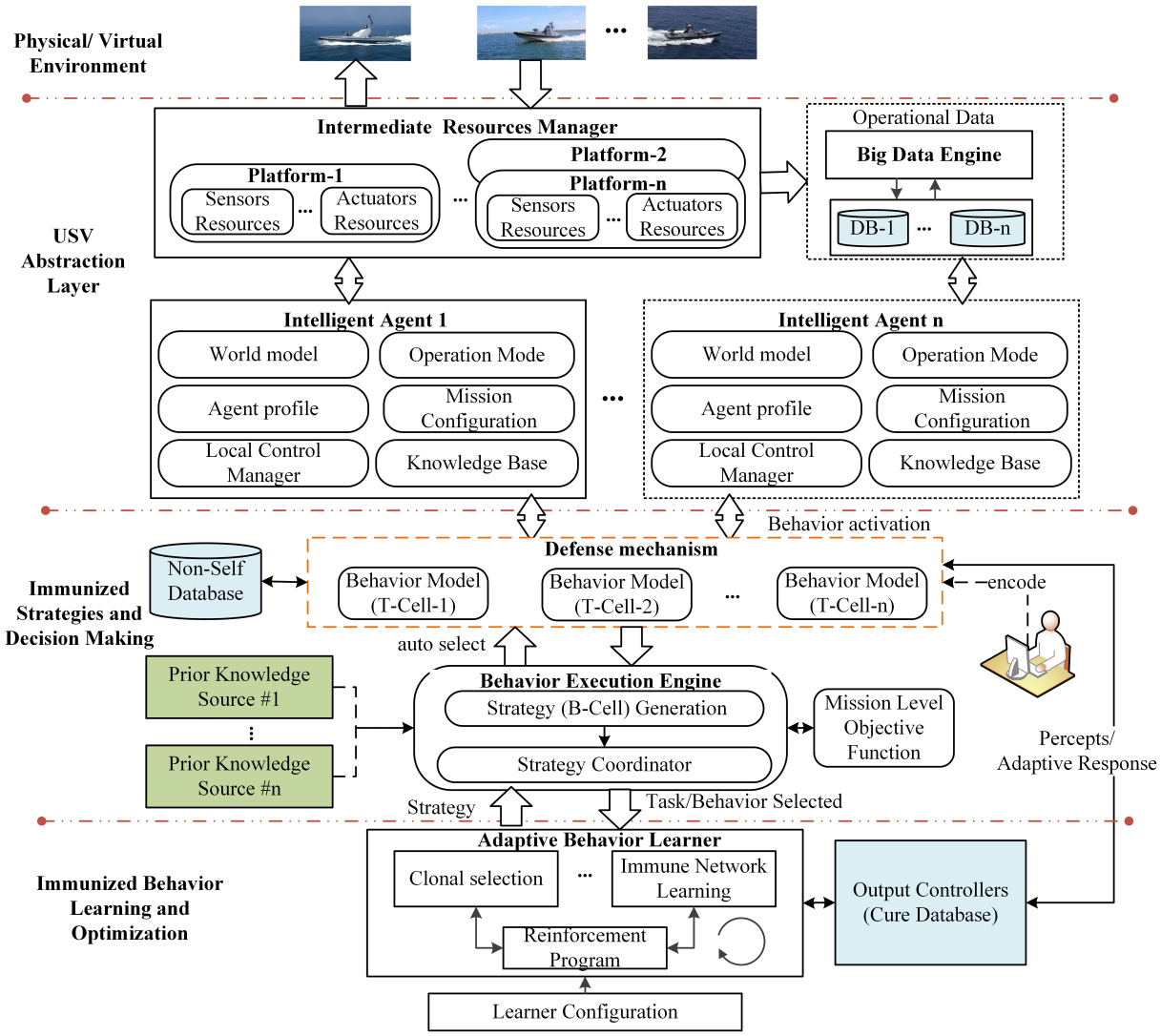


Figure 1. The proposed multi-USV training system architecture.

During training, the data of relevant variables are fetched and fed to the intelligent agent that interfaces the learning algorithm for individual behaviors. In order to obtain a large chunk of operational data and optimized decision making through data-oriented learning methods, the training system can be used to obtain a database of observed cases, actions taken to respond to those cases and the results of the action through the data engine. The data engine stores the data as a resource tree which can be transformed into common formats such as excel data sheets. The agent layer provides the abstraction and interface for simulated or real USVs. Thus, to achieve a seamless transition between the physical systems and their abstract simulation, this layer provides a data pipeline for the various sensors and actuators of the control platforms through the intermediate resource manager. Using this data pipeline, the important sensor data are extracted through platform-dependent virtual sensors.

In the proposed system, the values extracted from the real/simulated sensor readings from the environment are computed by a predefined set of virtual sensor functions. The world model of the intelligent agent provides an environment, detection, relational

and velocity virtual sensor functions to process the sensor data in the context of high-level functionalities. The detection virtual sensors return the relative positions and distances of other objects with a classification function that determines and classifies objects. The relational virtual sensor returns information about how an agent is situated to and from other objects. While the velocity sensor returns, the velocity of the objects remains within the detection range of the model. The environment virtual sensor returns normalized values of the observed ocean wind, currents and waves. Table 1 presents the data models extracted from the environment for agent world modeling.

Table 1. Information extracted from the agent environment as contained in a data model.

Data Item	Description	Examples
Internal state	These data consist of information that is internal to the USV	Rudder angle, radar range, engine power, weapon type, number of ammunition, etc.
External state	This consists of USV external information in relation to the environment coordinates	position, orientation, speed, heading, etc.
Observation	These data include the external state information of objects detected by USV sensors	distance, relative heading, relative position, dimension, etc.
Weather	This data model holds the environment data contents of wind, currents and waves data	wind speed, wind direction, waves height, currents speed, etc.
Simulation info	The information and data pertaining to the abstract environment of the USV	simulation mode, simulation time steps, and configuration data

Figure 2 shows an abstract data model representing the data concerning the state and observation of controlled USVs and environment. The content data model represents the structure that is used to hold the actual value of the data instance. The properties of the content data model are the type and value fields. This data model consists of the state, observation, and weather information data contents. The observation includes information on detected objects computed from the virtual sensors. The state includes the internal state such as the rudder angle, radar range, etc. and the external state such as the vehicle speed, position and heading. The platform (simulation) information consists of the task-specific data, decision times and other relevant information. The weather information model contains information about the wind, water currents and waves.

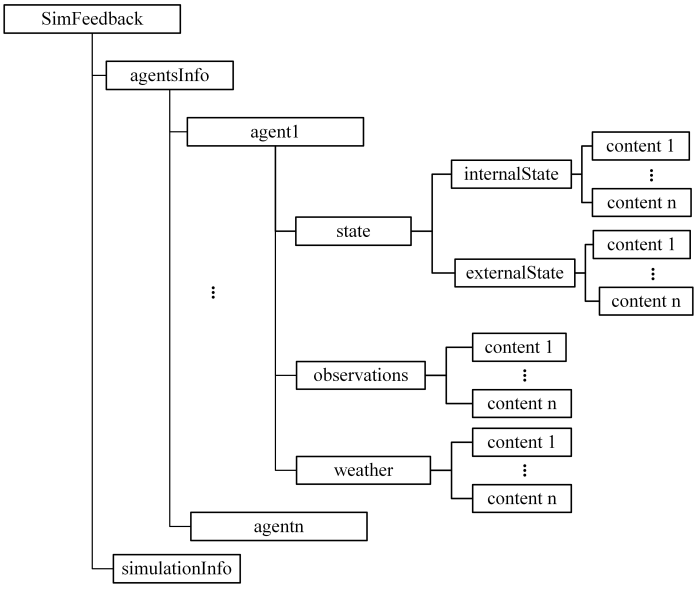


Figure 2. Data model of platform as a resource.

4. Multi-USV Interactive Environment Design

The usefulness of the trained controllers is directly related to the accuracy of the models that are used. As such, significant efforts are required to ensure the models used for vehicles dynamics, sensors, and the environment are realistic. Despite the availability of third party open source resources such as Webots [43], Gazebo [44] and UWSim [45] that can provide the USV physical engine, environment, and sensors modeling, these resources are limited in some instance such as combat vehicles modeling and in some instances specific to one or two types of vehicles. For instances, Gazebo focus on UGVs and UUVs with few works introducing new features designed for USVs. In UWSim, the wind simulation appears not to affect the vehicle movement and its focus is clearly on UUVs. Hence, in this section, we present the details of the digital maritime multi-USV environment design of the training system that provides a straightforward realistic behavior modeling of combat capable USVs.

4.1. USV Physical Engine

The USV abstract model presented accounts for wind and wave-induced currents to make it applicable to a wider range of sea conditions. Figure 3 shows USV motions in 6-*DOF*. Considering the USV movement dimension in 6-*DOF* of freedom, the basic USV model used is based on Fossen's 6-*DOF* model for marine vehicles [46]. This model expresses the resulting movement of USV as the combined effect of five main forces as shown in Equation (1)

$$\tau_{RF} = \tau_{hsf} + \tau_{hydf} + \tau_{wind} + \tau_{waves} + \tau \tag{1}$$

where τ_{hsf} is the hydro-static forces, τ_{hydf} is the hydrodynamics forces, τ is the control and propulsion forces, while τ_{wind}, τ_{waves} are the wind and wave forces, respectively. The kinematic and kinetic model with wind and wave disturbances is defined in Equation (2). The complete kinematic and kinetic model including the perturbations due marine currents can be found in [47]:

$$\begin{cases} \dot{\eta} = J(\eta)\nu \\ M_{RB}\dot{\boldsymbol{\nu}} + C_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + M_A\dot{\nu}_r + C_A(\nu_r)\boldsymbol{\nu}_r + D(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + g(\eta) = \tau_E + \tau \end{cases} \tag{2}$$

where:

- $\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^T$ is a vector of position and euler angles in the m-frame;
- $\boldsymbol{\nu} = [u, v, w, p, q, r]^T$ is a vector of linear and angular velocities in the d-frame;
- v_r is the hydrodynamic terms of relative velocities vector, i.e., the difference between the vessel velocity relative to the fluid velocity and of the velocity of marine currents expressed in the reference frame;
- τ_E is the forces and moments of environmental disturbances of superimposed wind, currents and waves;
- The parameters J, M, D, C are the rotational transformation, inertia, damping and the coriolis and centric fugal matrices, respectively.

The USV state model integrated the state of a physical dynamical model to the parameters of the corresponding constructed geometric model of the USVs, given external forces and torques. Currently, the physical properties are only available to describe the ranges and activation of the available sensors of an USV entity including their physical properties. In order to compute the drag and lift forces of the USVs, the coefficients and the USVs' velocity relative to the water velocity are used. To calculate these forces, the apparent velocity of the difference between the model and the ocean current is used together with lift and drag coefficients, as described in [48].

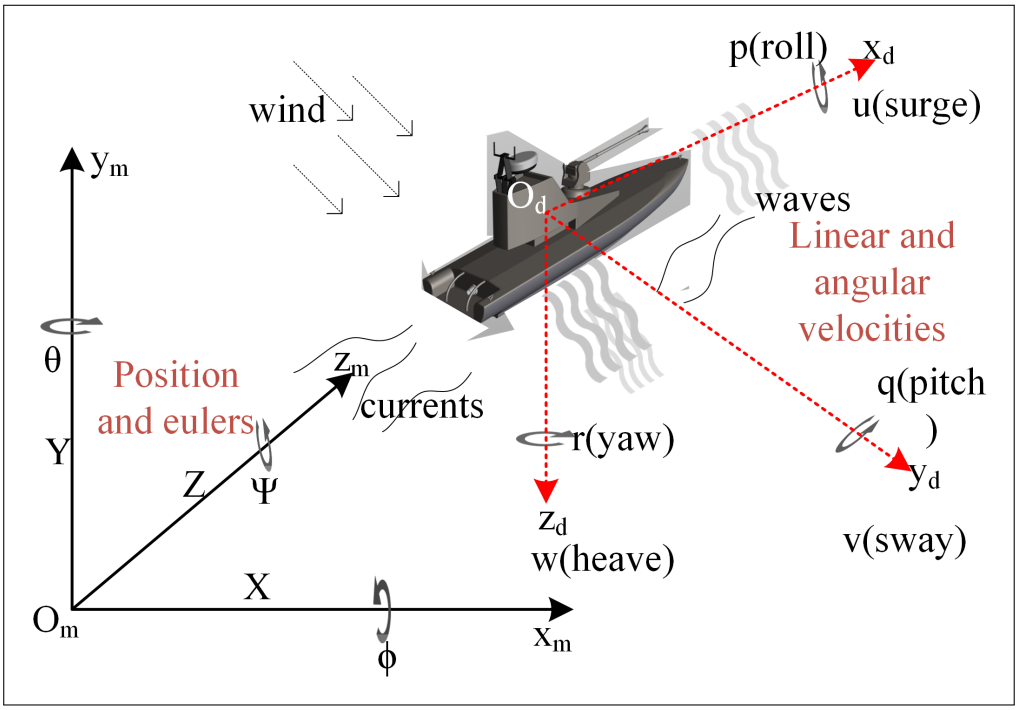


Figure 3. USV motion model in 6 DOF.

4.1.1. Modeling Buoyancy

To represent the buoyancy effects on the vehicles in the environment, the Archimedes principle as shown in Equation (3) is used to compute the buoyant force:

$$\|\mathbf{F}_b\| = \rho \cdot v_f \cdot g \tag{3}$$

where g is the gravitational acceleration constant, ρ is the known density of the water and v is the volume of the water displaced. Because each USV entity can be approximated with basic geometries, the volume of the displaced water is computed every time step of the simulation using the submerged region of the basic geometries of the vehicle. The buoyancy effects on the USVs consider the height of the water due to waves. This is particularly important for combat USVs since the orientation of the USV can affect the required elevation of a targeting weapon. The buoyancy is therefore assessed by the buoyant force and the gravitational force (computed from the mass of the vehicle). However, to improve the buoyancy effect on the USV, each USV is represent by six links joined together and the gravity and buoyancy forces applied to each link's center.

4.1.2. Actuator and Sensor Modelling

Marine radar is an important environmental perception sensor for USVs. Considering the problems of noise, jamming, and target lost in marine radar images, as well as the high-speed of the USV to the requirement of realistic representation, Radar and Lidar beams are simulated using the ray casting and ray-geometry collision detection approach, which we implemented through the ODE physics engine. The rays cast are affected or can be blocked by objects making detection performance realistic. By comparing the time the ray left the radar/scanner to the time each return is received, the range measurement can be computed. The range distance can be calculated as

$$\|\overrightarrow{r}\| = c\frac{t}{2} \tag{4}$$

where c is light speed and t is the difference in time between transmission and receiving a pulse. To simulate the LiDAR, in each simulation step, the range of α degrees is sampled N times to generate different rays using Equation (5):

$$\measuredangle \overrightarrow{r} = i\frac{\alpha}{N}\ i\varepsilon[1, \ldots, N] \tag{5}$$

The physics engine reports the collision and returns the range distance of the collided every time a ray hits an object. The detected points are then computed using this information and the current position of the model.

An inertial measurement unit (IMU) and global position system (GPS) for positioning and attitude information were modeled using a Gaussian model to account for sensor noise.

Moreover, to be able to represent combat USVs, a gun component was modeled in our simulated environment. The control parameters of the weapon are the elevation and angle in addition to the fire actuator. In order to produce a more realistic weapon performance, a fired weapon impact velocity at the target location can be estimated taking into consideration the aerodynamic drag of the weapon. For example, the aerodynamic drag force in one dimension of the fired weapon can be estimated in Equation (6):

$$F_{dr} = 0.5C_{dr}A\rho_a v^2; \tag{6}$$

where v is the speed of the ammunition, A is the defined area of the ammunition, ρ_a is the mean density of the air and C_{dr} is the aerodynamic drag coefficient. Based on Equation (6) the impact velocity of a ammunition can be estimated upon which we can estimate the damage to the target hit.

The damage of the ammunition is simplified by Equation (7), where d is the distance between the point where a bullet was fired and the hit target, while v is the impact velocity, and $\varkappa$ and η are the wind and scalar constants. Hence, the ammunition is also simulated as constrained by the physical properties of the weapon type and is affected by the environmental conditions:

$$D(H|b_j) = \frac{|v|}{d * \varkappa * \eta} \tag{7}$$

4.1.3. Wave, Wind and Current Modeling

To provide realistic waves, we adopted the Gerstner swell wave model which is commonly used in computer graphics [49,50] and simulates trochoides. In order to take into account the combination of many different waves for difference training instance, we generated wave trains for specific a wave spectrum inspired by [51] using randomly generated wave parameters within constraints. Using a different set of parameters, we continuously updated the waves and all trains. This representation was used to affect the physical motion of the USVs at the ocean surface.

To represent the wind disturbance for objects at the sea surface, we used the modeling in [24] to generate the wind direction and wind speed. The forcing influence of wind direction and the wind speed in the reference plane on the model above the water surface was modeled as

$$u_{rw} = u - u_w \tag{8}$$

$$v_{rw} = v - v_w \tag{9}$$

where u_w and v_{rw} are the x and y components of the simulated wind velocity in the vessel body frame, expressed as

$$u_w = V_w \cos(\beta_w - \psi) \tag{10}$$

$$v_w = V_w \sin(\beta_w - \psi) \tag{11}$$

The resulting wind force can then be computed using:

$$X_{wind} = \bar{c}_x u_{rw} |u_{rw}| \tag{12}$$

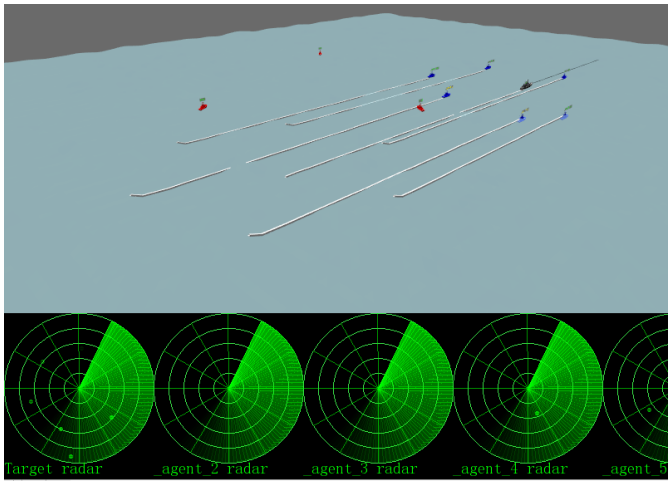
$$Y_{wind} = \bar{c}_y v_{rw} |v_{rw}| \tag{13}$$

$$N_{wind} = -2.0 \bar{c}_n u_{rw} v_{rw} \tag{14}$$

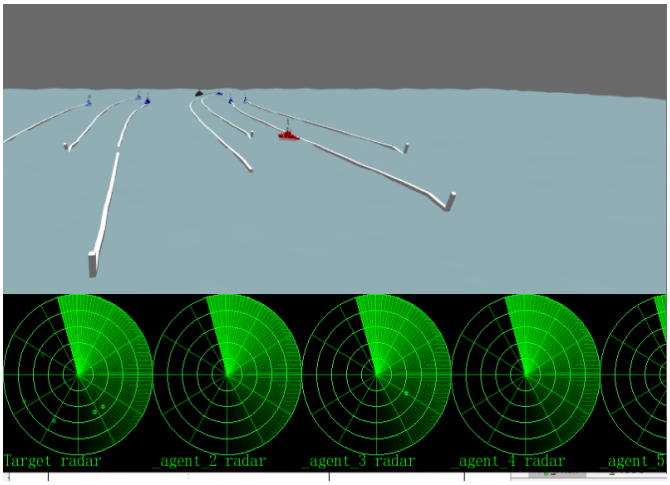
where $\bar{c}_x, \bar{c}_y, \bar{c}_n$ are the dimensional wind coefficients.

To model the effects of water current, a first-order Gauss–Markov processes is used to homogeneously generate the parameters of currents at different nodes of the environment. The forcing terms are then obtained using the current vector v_c^E similar to the wind forcing terms.

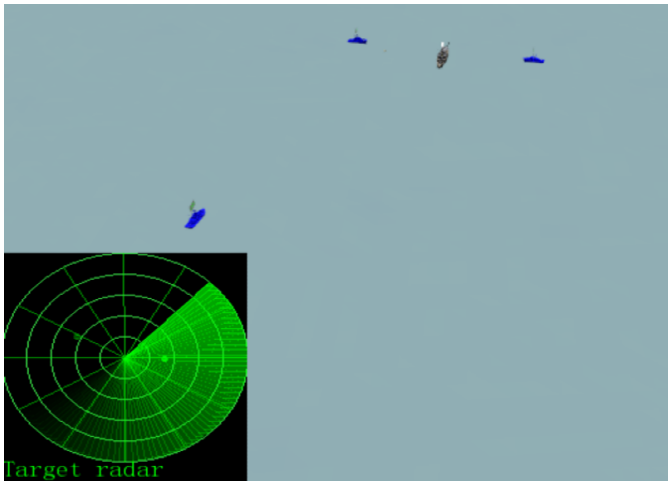
Figure 4 shows the trajectory of the USVs with no disturbance (Figure 4b) and when wind and currents are activated (Figure 4b). In addition, the performance of a USV radar shown at bottom left corner of Figure 4c. In this figure, we only activated the radar of the non-blue boat to demonstrate the detection range and performance. As can be seen in the radar, only two USVs appear on the target radar and the closest USV on the right appears more clearly than the one on the left.



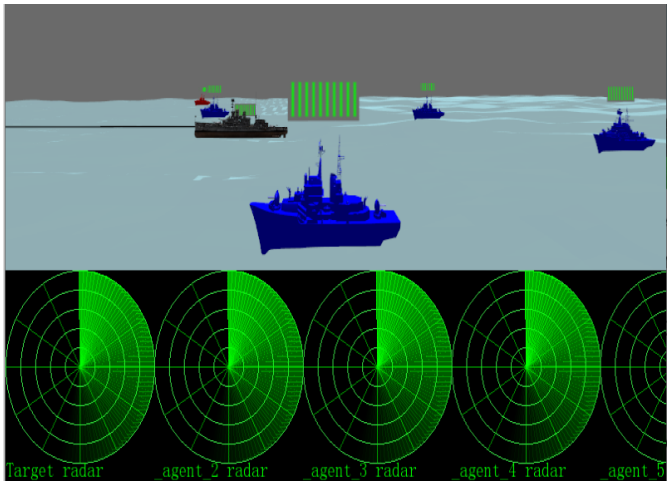
(**a**) Trajectory of USVs on calm sea



(**b**) Trajectory of USVs with disturbances



(**c**) Simulated USV radar detection performance



(**d**) A closer look of USVs in the virtual environment

Figure 4. Snapshot depicting environmental conditions' effect on USVs maneuverability and detection performance.

5. Multi-USV Training Algorithm Design

In the previous sections, we described how the digital environment of the training system is abstracted and simulated by describing how the physical environment and entities are modeled. In this section, we present our approach for learning controllers of USVs to realize multi-USV cooperative decision making.

The approach adopted in this work was inspired by the working mechanism of the biological immune system, which can be regarded as a typical distributed multi-agent system. In this approach, agents are designed as physical controllers of vehicles that owns a set of behaviors models (T-Cells models) encapsulating fuzzy classifier units (FCUs) which output its control decisions (antibodies). Each FCU possesses a detector which matches the attributes (antigens) of the agent environment obtained through its sensors and internal measurement units. The objective is to learn the primitive behaviors of the agents while coordinating the different behaviors of the agent. The key idea in our approach is to generate, activate and assign the behaviors of B-Cells to agents executing a mission so that the entire multi-agent team can learn an optimal control strategy faster. In this case, diverse behaviors that constitute the multi-agent mission serve as motivations for the immune agents. That is, the focus of execution within each behavior model is guided by the underlying objective of that particular behavior with which a local reinforcement program can be defined.

In what follows, we present the detail algorithmic approach towards realizing autonomous multi-USV control decision making in combat tasks.

5.1. Agent Knowledge Modelling and Representation

The agents in the training system are equipped with sensor (radar) and weapon systems for detecting and firing threats, as shown in Figure 5. An agent control system only requires important sensors information abstracted from the raw sensor data. The extracted information is then formatted and sent to the decision-making layer.

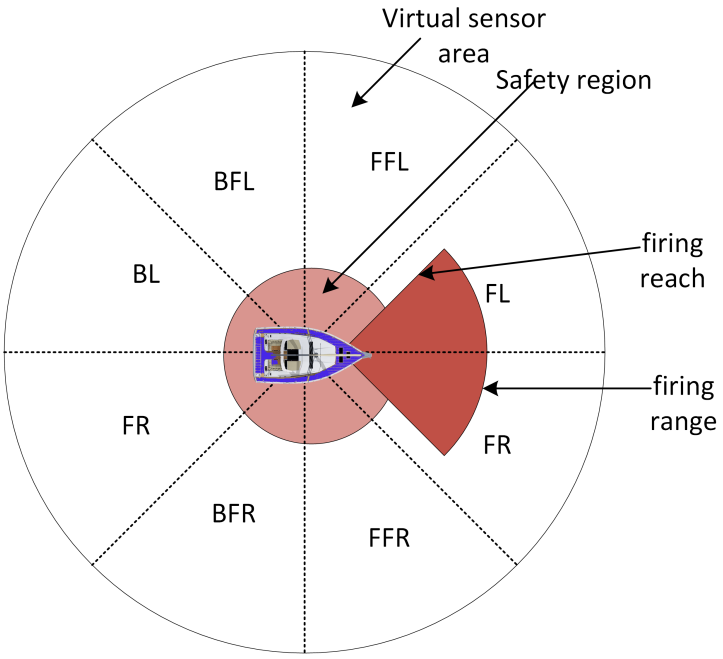


Figure 5. USV agent and virtual sensors modelling.

The equipped detection radar of the agents has a circular detection region with a radius r which determines its range. This radar is partitioned into the virtual sensors of the same range by assigning a reach for each virtual sensor. These virtual sensors return the relative positions and the normalized distance of other objects. The relational virtual

sensor returns information about how an agent is situated to and from other objects, while the velocity sensor returns the velocity of objects within the detection range of the agent.

In our approach, the agent is analogous to an organ in the body and owns a set of T-Cells which encapsulate a behavior or decision model that generates antibodies in response to the environment stimulation. T-Cells perform fuzzy matching between classifiers and environment attributes. A fuzzy classifier is modeled as a B-Cell which consists of several parts: an ID, condition (attributes of T-Cell receptors), consequent (specific antibody or antibodies) and connection (idiotope) parts. A set of B-Cells constitute a strategy. T-Cell models activate B-Cells which together results in an immune network of B-Cells. In the BIS, T-Cell receptors can only recognize antigens that are bound to certain receptor molecules, and undergo a process called rearrangement, causing the recombination of a gene that expresses T-Cell receptors. The process of rearrangement allows for a lot of binding diversity. This process is likened to the generation of the rule base of the FCU inputs. During training, T-Cells activated B-Cells which are optimized to obtain a good performance strategy. Figure 6 shows an abstract immune network of the T-Cell interaction of an agent. Each T-Cell is associated with a decision-making model within the overall mission and is identified by a unique identifier, receptors, specific antibodies and local antibody network which forms the knowledge base of a behavior. The inputs and outputs are defined as fuzzy sets with different degrees of membership and represented as binary strings similar to the representation used in [52]. The detector receives the real values of input variables (antigens) and transforms each of these values into a group of fuzzy sets. In this study, we adopted a triangular membership functions type for defining a range of values for each variable of a behavior model. Figure 7a,b presents the example heading and range inputs membership functions and their encoded strings positions or indexes, which are determined by the designer. For instance, a real-value of 100 for heading can be represented as a *11000* binary string. The possible outputs of a T-Cell represent the constituents of antibodies that can be activated by it, while the inputs define the set of pathogens that a B-Cell reacts to. Here, we loosely model the T-Cell as a container of B-Cells and its responsibility is to activate the B-Cells generated from the recombination of genes (attributes of agent tasks) that express T-Cell receptors.

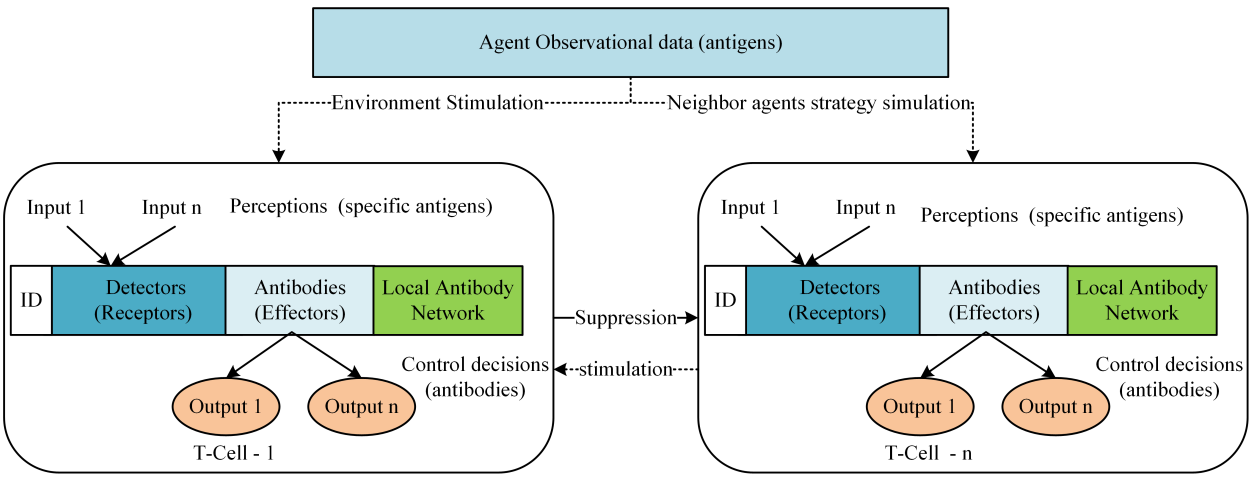
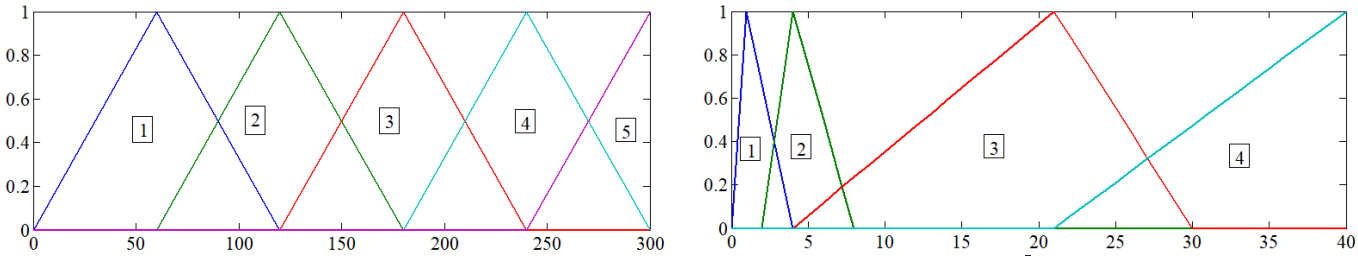


Figure 6. An abstract immune B-Cells generation model.



(**a**) Example heading input fuzzy set and string encoding. (**b**) Example range input fuzzy set and string encoding.

Figure 7. Example inputs (antigens) encoding in an immunized fuzzy classifier system.

5.2. *Decision-Making and Evaluation Mechanism*

Behavior units interact with the environment through the agent, its detectors and effectors, whilst the actions of classifiers (B-Cells) are evaluated through a reinforcement program that captures the motivation or objective of the behavior. Classifiers are strengthened or weakened based on the performance of the behavior unit modeled by the T-Cells. During the execution, behaviors are activated based on the stimulation from the environment and a match set of classifiers which determines the stimulation level of the behavior to the agent's observation is formed for each behavior in parallel. Each matched classifier is collected when the agent is exposed to the environment state and its parent behavior is activated to form a network of B-Cells. To determine the affinity between the behavior unit and the environment's antigens, we used the average affinity of the activated classifiers by the behavior unit. The final actions of the environment is emitted from the classifiers with a higher concentration after interacting with other classifiers. Classifiers that contributed to the overall fitness obtained during the evaluation of a behavior unit receive a relative reward as its contribution to how well the agent performs the behavior based on the reinforcement program of the behavior. This means that individuals are evaluated in parallel and evolved according to its experience in the environment.

The algorithmic approach for the agent's decision making of a multi-USV system consists of the following steps, which are executed at each sample time by each agent:

1. At each decision step, the agent activates behaviors according to the environmental stimulation;
2. For every activated behavior, we generate a set of strategies or determine the match set of strategies (B-Cells). When multiple tasks are detected within the behavior, a match set is formed for each task. For example, when a track behavior is activated after USV detects multiple targets, the behavior model generates a strategy for each target using the attributes of each task (target);
3. Then, the agent establishes connections between B-Cells based on the selected task. Connections between B-Cells are established based on the tasks and behavior under which they are generated or activated.
4. To coordinate with nearby agents, the same is done by the agent with respect to nearby agents and the detected targets. Agents can also communicate with nearby agents within a communication range to obtain the strategy concentration for shared behaviors' activation to select the best strategy with respect to a task and submit to a behavior learner;
5. Apply immune network dynamics to update the concentration of each B-Cell;
6. The final actions of the environment are emitted from the classifiers (B-Cells) with a higher concentration after interacting with other classifiers.

In order to evaluate an agent's performance of a behavior, classifiers within behavior units are evaluated based on the local objective of the behavior and global reward using Equation (15):

$$F(t_{j+1}) = (1 - \alpha)F(t_j) + \alpha_1 R_g(\alpha_1 R_l) \tag{15}$$

where R_g is the team reward return by the environment as computed by the global reinforcement program. R_l is the local reward of a particular behavior, in other words, the estimate of how well an agent performs a particular behavior which is evaluated by the local behavior objective function or human-provided reinforcement during training. The variables α_1 and α are discount factors used to discount the local and global objective functions. The local reinforcement program (evaluation functions) allows for reward shaping, the independent learning of behaviors by agents, and facilitates the sharing of learned policies between team members while the global reinforcement program implements the global objective function that evaluates the team performance on a task or the overall mission performance by the agents. The global reinforcement program in the evaluating cases are designed to capture the overall mission objective using the cumulative rewards received from the environment after an episode ends. In this case, a score was assigned as the strength of each behavior's B-Cells that were triggered during an episode run as obtained from reinforcement programs.

Based on the immune system mechanisms, stimulation and co-stimulation occur among B-Cells. In our case, B-Cell i is said to stimulate B-Cell j if the strength or fitness of j is higher than that of i. This implies that the control action proposed by j leads to a better performance than that of i. At the lower level, when a B-Cell is activated during a behavior execution, this will lead to the stimulation of B-Cells under same conditions but with a different output in the global immune network when the output of the B-Cell leads to a better performance and vice versa. A relative affinity between a B-Cell and environment pathogens can be measured using a modified version of Equation (15) that takes into account the number of times the B-Cell was activated, as shown in Equation (16):

$$Affinity(B_i) = F(t_{j+1}) - \frac{1}{N_t} \tag{16}$$

With the knowledge of B-Cell's affinity and how to obtain the affinity between B-Cells, the concentration (fitness) C_a of the a_{th} B-Cell can be realized using Equation (17) [53]:

$$\frac{C_a(t+1)}{dt} = \left(\alpha \sum_{j=1}^{N} m_{ja} c_j(t) - \beta \sum_{k=1}^{N} m_{ak} c_k(t) + \gamma m_a - \omega \right) c_a(t) \tag{17}$$

where:

- N is the number of B-Cells that have an inhibitory or stimulating effect on the B-Cell;
- m_a is the affinity between B-Cell a and current stimuli (antigens);
- m_{ja} is the mutual stimulus coefficient of antibody j into B-Cell a;
- m_{ki} represents the inhibitory effect of B-Cell k into B-Cell a;
- ω is the rate of the natural death rate of B-Cell a;
- $c_a(t)$ is the bounded concentrations imposed on a B-Cell modeled as a squashing function for normalized concentration values [53];
- The coefficients α, β and γ are weight factors that determine the significance of the individual terms.

The concentration level of a B-Cell in this case affects the chances of a particular action that the classifier can propose as the optimal action that maps the conditions of the classifier. This means that the objective of the optimization algorithms is to learn the optimal action of a classifier map to a compact number of classifiers necessary to realize an optimal and adaptive execution of the independent behaviors.

5.3. Evolution Mechanism: Clonal and Negative Selection

In classifier systems, genetics-based learning approaches are employed by using genetic operators to realize solutions for different problems and exists in two forms, i.e., Michigan-style [54] and Pittsburgh style [55] approaches. In the Michigan-style approach, an individual is a single rule, while in the Pittsburgh approach, one individual is a

set of rules. In this work, the Pittsburgh approach is used to represent the antibody set as the possible solution or optimal actions mapping for individual classifiers of a behavior. The clonal selection and negative selection mechanism enables the evolution and learning of our optimal internal behavior execution while the immune network dynamics allows the coordination between and within behaviors for effective realization of controllers. In what follows, we present the process for evolving classifiers.

1. At the beginning of training, T-Cell receptors must undergo the rearrangement process for the recombination of genes that express T-Cell receptors to form the knowledge base of all encoded behaviors;
2. Initialize a non-self database to empty or using prior knowledge where the designer encodes inconsistent antibody (control action) mapping as the antibody set for individual behaviors (T-Cells);
3. Randomly initialize an N population of the antibody set for each T-Cell by assigning an antibody from the valid antibodies of the respective T-Cell to each B-Cell to form the initial controllers;
4. Compare the current generated antibodies with those in the non-self database to remove/modify inconsistent antibody-sets;
5. Next, the simulator is run with the current generation of antibodies for each behavior N times to test each set in the population. In each run, we apply the decision-making mechanism described above to select the classifiers whose actions are posted to the environment;
6. At the end of each episode, we apply the evaluation mechanism to evaluate the antibody set of each behavior;
7. At the end of each generation of the population of antibodies, the concentration level of classifiers (B-Cells) based on the performance of individual behaviors' entire antibody set is used to determine the n best antibody sets;
8. Clone and store the classifiers of the elite antibody sets that were triggered and recompose the global cure database with these classifiers. Submit the population of clones to a hyper-mutation scheme by randomly selecting and changing the antibodies of classifiers to form temporary antibody sets to be evaluated next;
9. Add the set of antibodies that results in the poor performance of the behavior to the non-self database. After the B-Cell undergoes mutation by changing the action (antibody) parts of the B-Cells, the resulted antibody set is compared with those in the non-self database and modified if the similarities between them is below a predefined threshold;
10. On the other hand, elite B-Cells resulting from other agents are cloned by other agents when other agents succeed in finding more optimal actions for a behavior execution;
11. Repeat Steps 5 to 10 until a termination condition is met

6. Experiments

In this section, we demonstrate the performance of the training system and approach in the context of training multi-USV systems in two combat tasks. The cases presented are real-world scenarios designed to test the performance generalization and learning speed of the system and approach. Experiments are setup and the USVs trained in the virtual environment and the obtained controllers are evaluated. Moreover, the trained controllers are evaluated in different configurations of the underlying scenarios in all cases. Two training scenarios' designs were based on different missions of unmanned surface vehicles. The cooperative target and escort task saw the island conquered in the realistic environment.

6.1. General System Setup

The USVs behavior configuration consists of several high-level behaviors as T-Cells shown in Table 2. Depending on the mission or tasked to be learned, behaviors peculiar to the task are activated and the execution of these behaviors are tuned and optimized

to maximize USV performance on the mission in different situations. Each behavior has receptors that define a minimum of one antigen it responds to, and as specified by the designer based on the knowledge of the tasks and behavior. Table 3 shows the repository of antigens used during experiment. Table 4 lists our primitive control actions (antibodies of classifiers) that can be suggested by a classifier (B-Cell). The primitives of steer control and throttle control directly translate into fuzzy sets. For weapon control, the fuzzy set of the aim angle include aimLeft, aimCenter and aimRight. Radar control involves turning on/off of the radar system of the USV. In each experiment, the number of antibody sets in a population is set to 40 and a simulation episode lasts for 3 min. The detection and firing range of both teams are set to 300 and 80 m, respectively, with a maximum speed of 25 m/s during training, while the maximum turning angle of the USV is 30. The wind and current speed are set between [0–10] and [0–8] meters per second with a variable direction across the sea surface. The different configurations used during testing are shown in Table 5.

Table 2. Behaviors defined in the training system.

Behavior Module (T-Cell)	Control Actions (Antibodies)	Description
Alignment	Speed, direction	Align with other agents or an object
Avoid collision	speed, direction	Avoid colliding with a static or dynamic object
Pursue	Speed, direction	Chase a target
Detour	Speed, direction	Get behind a target as soon as possible
Track	Speed, direction	Follow a target at a specific distance
Search	Speed, direction	Search an area
Attack	Weapon angle, salvo	fire at a target with an appropriate number of salvos
Assist teammate	Speed, direction	Move to a teammate performing a task
Conquer	Speed, direction	Move to an island location

Table 3. Attributes of the tasks and environment for the different behavior models.

Attribute (Antigen or Input Variable)	Description
V_A	The current velocity of usv
V_E	The velocity of a detected threat usv
V_T	The velocity of the protected target
H_A	Current heading of usv
H_E	Current heading of detected threat usv
H_T	Heading of protected target.
D	Distance to task (enemy usv or island to be conqured)
D_T	Distance to protected target
H_{Diff}	Heading difference of task and current usv
D_{scout}	Distance to the assigned neighbor
N_c	Number of threats detected by this usv
N_t	Number of threats detected by neighbor USVs
T_{st}	Number of available neighbors
C_{un}	Number of un-responded calls
C_{st}	Number of objects or agents in collision region
DC_c	distance to potential collision point
W_v	Speed of wind
W_d	Direction of wind
C_v	Speed of water current
C_d	Direction of water current
W_h	Wave height

Table 4. Primitive control actions.

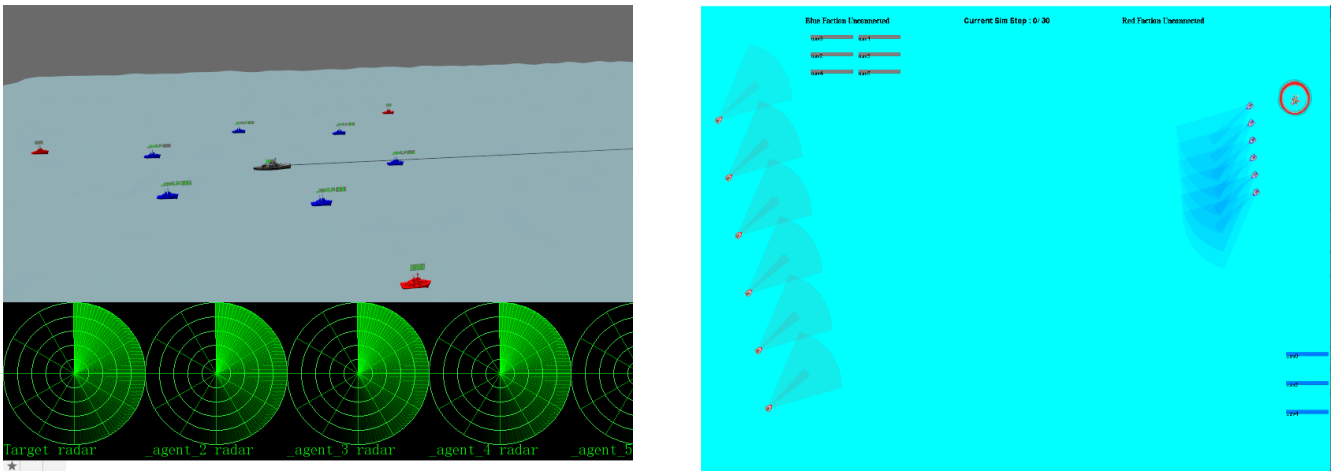
Type	Primitive Fuzzy Terms (Antibodies)
Steer control	straightAhead, turnSlightlyLeft, turnVeryLeft, turnLeft, turnExtremelyLeft, turnVeryRight, turnRight, turnSlightlyLeft, turnExtremelyRight,
Throttle control	reverseSpeed, verySlowSpeed, lowSpeed, normalSpeed, fastSpeed
Gun and radar control	fire(angle), performDetection

Table 5. USVs and team configuration in evaluation scenarios.

Scenario/ Settings	B6vsR3 (Scene 1)		B10vsR7 (Scene 2)		B10vsR10 (Scene 4)		B7vsR10 (Scene 3)	
	Blue	Red	Blue	Red	Blue	Red	Blue	Red
Radar range	300 m	300 m	200m	300 m	300 m/s	250 m/s	250 m/s	300 m/s
Firing range	95 m	95 m	75 m	95 m	75 m	90 m	70 m	80 m
Max gun turn	30°	30°	25°	30°	25°	25°	20°	30°
Max turn angle	30°	30°	25°	30°	25°	25°	20°	30°
Max speed	30 m/s	30 m/s	30 m/s	25 m/s	25 m/s	30 m/s	30 m/s	30 m/s
Wind speed	3 m/s		3 m/s		5 m/s		8 m/s	
current speed	2 m/s		1.5 m/s		3 m/s		6 m/s	

6.2. Case 1: Multi-USV Target Escort

This scenario is motivated by the practical domain of protecting oil tankers or cargo ships, and protecting forces and maritime warfare in general. These assets of tremendous economic value may be the target of terrorist organizations or pirates using small boats parked with remotely or manually controlled explosives or as in the case of pirates boarding small boats to carry out attacks. In this case, small and fast USVs with combat capabilities can provide protection in such situations, especially when multiple hostile threats are involved.



(**a**) Simulation in virtual environment (**b**) Simulation setup in 2D environment

Figure 8. Snapshot of target escort scenario setup of the environment. In (**a**), the controlled agents are blue boats and the red boats are the threats. The protected target is in the middle and the blue frames show the radars of the blue USVs. In (**b**), the setup is shown in 2D with a smaller detection angle and firing range for evaluation.

In the training setup, six USVs (blue team) are required to protect a dynamic target from the hostile boats (red team). Figure 8 shows a typical setup of the scenario during training. The blue team must coordinate their actions so that they result in creating a safety fence around the target while intercepting and neutralizing any incoming threat.

The goal of the team is to ensure the safety of the target and successfully destroy any detected threat (red USVs) by engaging in combat. In this scenario, a successful cooperative control enables the blue team to balance their resources between several main tactical behaviors. A dynamic redeployment or formation may maximize the coverage area and an interception and combat tactics may destroy the detected threats. Based on this criterion, the global reinforcement program implements the following objective function:

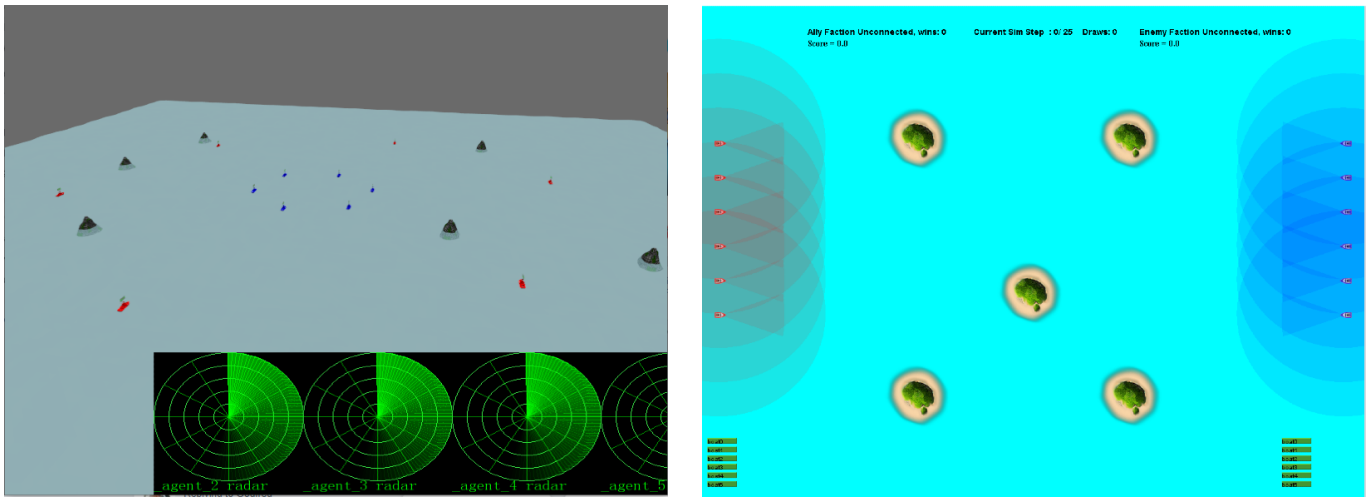
$$\langle \mathcal{R}^{team}, \pi^* \rangle = \arg\max_{\mathcal{R},\pi} E[(\beta + \gamma) - (\alpha + \frac{\rho - 1}{\rho}] \tag{18}$$

where β is the damage caused to the opponent team, γ is the time taken for the target to be destroyed—normalized between 0 and 1—α the damage caused by opponents to the team members and ρ is the total resource utilization of the team. On the other hand, the staying power of a USV is the health of the USV which deteriorates as the amount of damage caused by its opponent's fired weapon increases. In addition to the global objective function, the local reinforcement program implements the local objective functions based on the objective of a defined behavior.

The design of the opposing (red) team control policy is crucial for providing realistic challenge blue team. In this regard, the red team control policy is designed to be competitive in the sense that their action selection within executing behaviors is not fixed to similar states, hence no predefined pattern can be easily deduced by the blue force. Table 6 shows an example encode rules used for the red team behavior selection during simulation.

Table 6. Example encoded task selection rules used by red teams and baseline policy.

Protected Target	Enemy Detected	Behavior Output
Destroyed	Is none	Retreat
Destroyed	Is more	Intercept closest enemy
Destroyed	Is behind me	Perform detour
Is alive	Is many	Intercept closest enemy
Is alive	Not attacking	Attack target
Is alive	Is behind me	Perform detour
Is in firing range	-	Fire at target
Is alive or dead	Is in firing range	fire at enemy
Is alive	Is none	Attack target



(**a**) Simulation setup in 3D environment (**b**) Simulation setup in 2D environment

Figure 9. Island conquering simulation setup in 3D settings (**left**) and the simplified 2D environment (**right**).

6.3. Case 2: Cooperative Islands Conquering

In this scenario, multiple USVs competing for conquering more islands while engaging in combat is simulated as described in [19]. We implement this scenario in our virtual environment while using the environment in [19] for evaluation. Figure 9 shows simulation setup for this scenario in two different environments. The environment consisted of N islands and two teams of unmanned surface vehicles (boats). Each team had information on the location and number of islands and their states, whether conquered or unconquered by the team. An island is said to be conquered by a team if a member of the team moves to the coordinate of the island and stays there for that time period and no opponent boats move to that particular island conquered by the team. If two opponent boats occupy an island at the same time, the island is not awarded to any team for the elapsed time steps. The red force uses fixed rules while the blue force is trained with our approach, which implements or modifies the additional behaviors described in [19]. We compared the performance of these policies and the trained controllers in both environments and settings. In this case, the global objective function as defined in [19] is as follows:

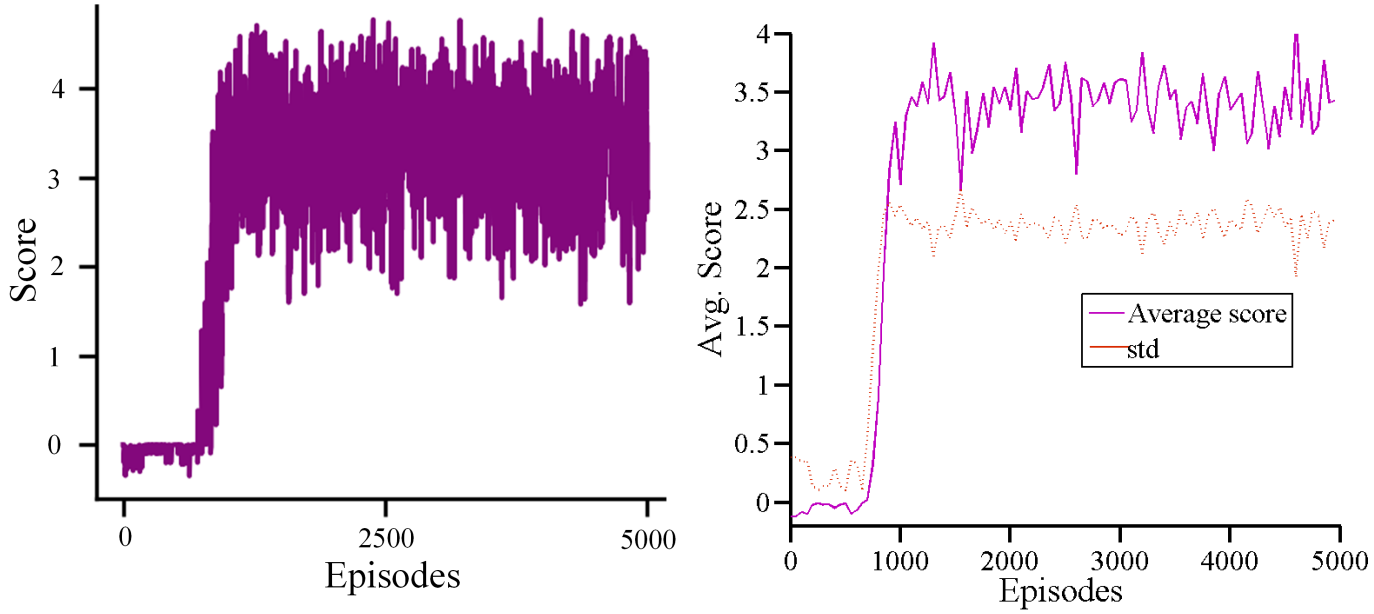
$$\langle \mathcal{R}^{team}, \pi^* \rangle = \arg\max_{\mathcal{R},\pi} E[(\beta + \gamma) - \alpha] \tag{19}$$

where β is the damage caused to opponent team, γ is the time taken for the target to be destroyed, and α the damage caused by opponents to the team.

6.4. Results and Discussion

In the first case, the blue team is declared the winner of the confrontation if the target is not destroyed within the episode run. After a number of training episodes, the obtained results show a better performance corresponding to a higher score and an average score for our agents as the training progresses. This can be seen in the results of Figure 10. Figure 10a shows the score obtained by the team based on the global objective function while Figure 11b shows the average score per generation. To evaluate the trained controllers, the results of different configuration of the case is presented in a simplified 2D environment and the virtual environment in this work. The results in both settings are being compared with a baseline heuristic policy. In the heuristic strategy, the USVs are uniformly distributed around the protected target(s) and moves according to a pre-planned way points. When the intruder boats appear, the blue USVs intercept the closest by simultaneously moving closer to the intruder with a fixed speed and fixed turning angles. The intruder is automatically fired at when it is within the firing range of the blue USVs. Alternatively, selecting the intruder for interception is done randomly and one on one, i.e., only one blue USV intercepts an intruder or two blue USVs randomly intercept a selected intruder two to one, where the USVs employ similar rules defined in Table 6. Figure 11a,b presents the evaluation results as compared to the baseline heuristics in both the virtual environment and simplified environment with a varied team configuration to demonstrate the transferable and scalable nature of the learned controllers. While the 2D environment assumes calm sea conditions in all settings, different sea conditions are specified in the 3D environment to test the performance in both settings. The results show that in two of the environment settings, the blue team achieved the highest success rate in setup $B6vsR3$ when all teams had the same configuration. There is no significant drop in performance in $B10vsR7$ despite the constraints in the capability of the blue team. This may be as a result of the team's size and speed advantage of the blue team. However, there is further drop in performance in both environments with the equal team size and when the red team has an advantage over the team size. A significant drop in performance was witnessed in $B7vsR10$ setup. This can be attributed to the impact of large environmental disturbances caused by the wind and currents in addition to the team size advantage of the red team. The impact of wind and current can be seen in all settings as the performance in the 2D environment (with no environmental disturbances modeled) is better in all

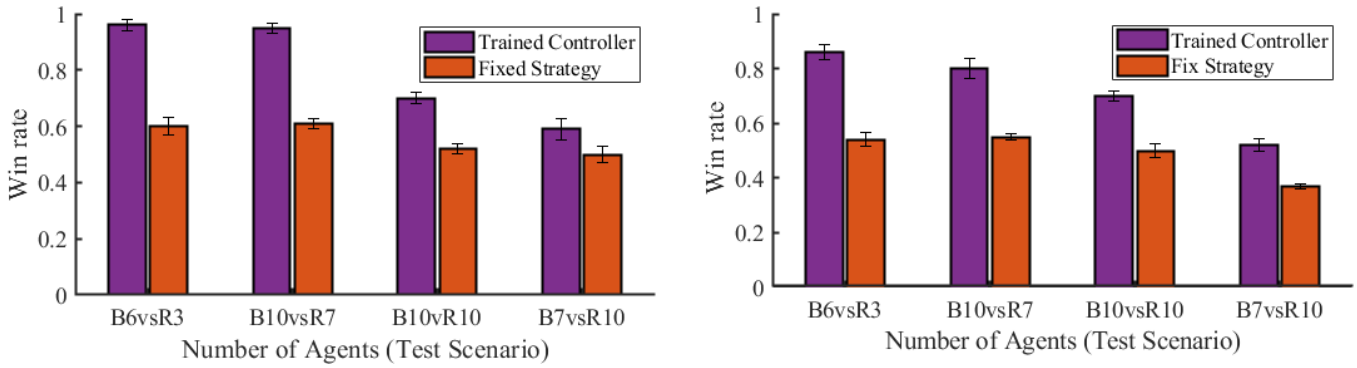
scenarios. Moreover, in all setups of the escort task, the trained controllers outperformed the heuristic-based fixed strategies with a win rate between 10% and 30%.



(**a**) Learning agents score per episode

(**b**) Average score over 50 episodes of simulation run

Figure 10. The learning curve of agents based on the proposed approach during training.



(**a**) Win rate under variable team size in simplified environment

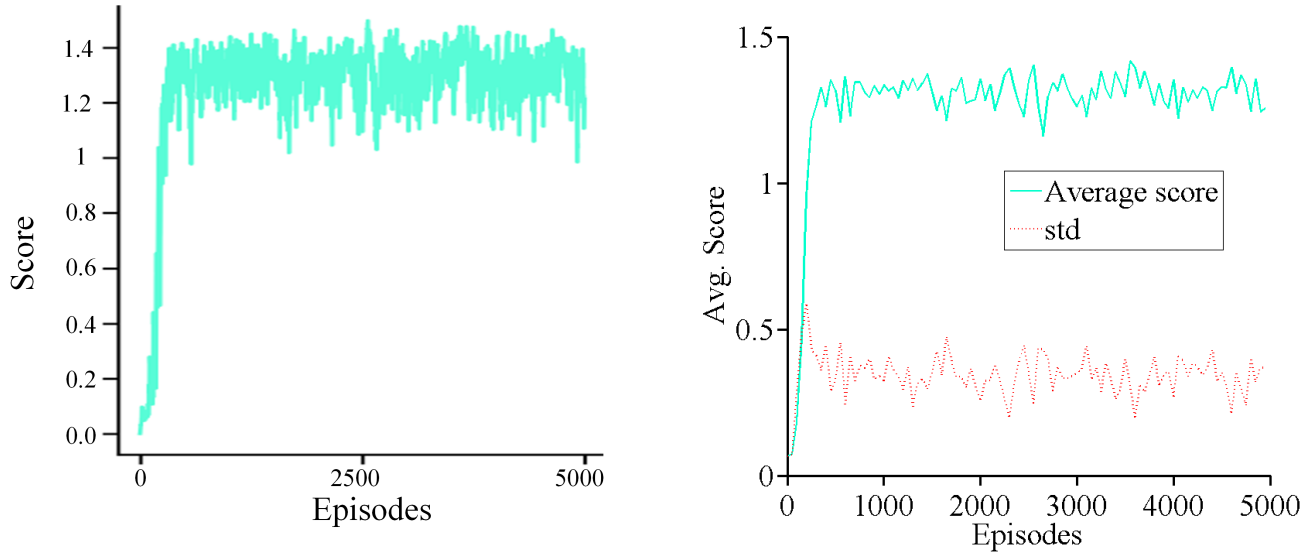
(**b**) Win rate under variable team size in virtual environment

Figure 11. Simulation results under variable team size in different environment configurations. Here, the control team wins the confrontation if the target is not destroyed and at least one USV is still escorting it or all the intruder USVs are destroyed.

On the other hand, Figure 12 shows the episodic score and average score over 50 episodes (generation) of simulation in the case 2 scenario. The average score per generation of antibodies and score per episode during training increases as training progresses as shown in Figure 12a,b, respectively. To evaluate the performance of the output controllers after training, the different configuration of the scenario is also run on both the virtual environment and the environment developed in [19] without any major changes to the output controllers and compared with a heuristic baseline approach which comes in two forms in terms of selecting a task to perform.

CCI: Under this strategy, each member of the blue force selects and conquers the closest island. When an opponent appears, it also selects the closest to attack. The intruder is automatically fired at when it is within the firing range of the blue USVs. Fixed rules for behaviors such as detour, track and retreat can be employed by the USV during combat engagement, similar to one shown in Table 6.

CIU: This is also known as conquer in units, a strategy which works by grouping team members to perform the conquering in units. In this case, the groups can select the islands to conquer randomly. However, when multiple opponents are detected, the units maybe dissolved during combat.

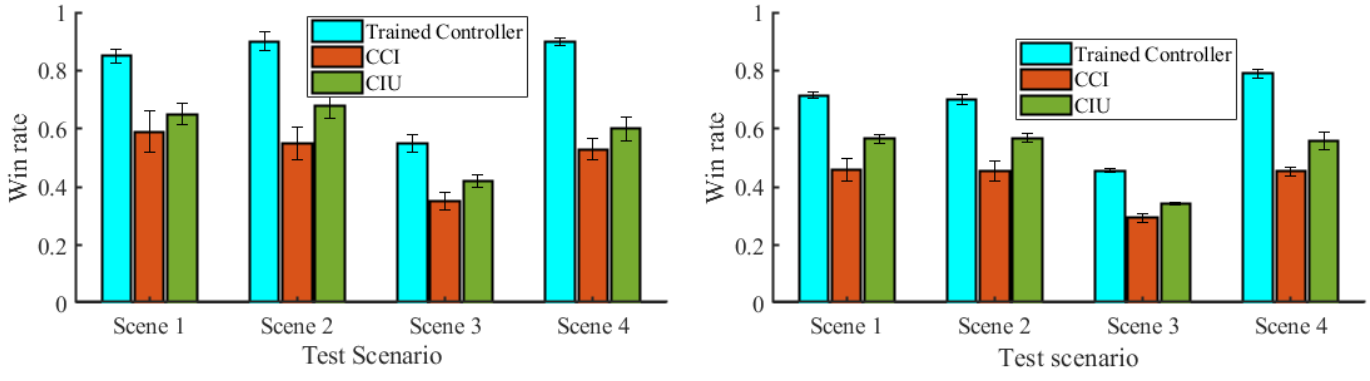


(**a**) Score per episodes obtained by agents

(**b**) Average score over 50 episodes of simulation run

Figure 12. The learning curve of agents based on the proposed approach during training.

The average wining rate, defined by the number of islands, the losses of the team and the damages to the opponent faction, is shown in Figure 13a,b, respectively. A win here means that the blue team scores higher than the red team based on the objective function. In this scenario, interestingly, the trained controllers obtained its maximum performance in the Scene 4 setup despite the more significant environment disturbances compared with Scenes 1 and 2. Moreover, as can be observed, the highest performance obtained in this scenario was about 4% less that of the one obtained in the escort task. However, it appears to have performed very well when evaluated in the 2D settings. This is also evident in that the trained controllers outperformed both fixed strategies in both environments.



(**a**) Comparison of the win rate in simplified environment

(**b**) Comparison of win rate in realistic environment

Figure 13. Comparison of team win rate under variable team configuration in 3D and 2D environment. Scene 1 consists of 3 islands, and 5 USVs in each team. In Scene 2, there are 15 each with 8 islands. In Scene 3, there are 9 islands, 12 blue USVs and 15 red USVs. Scene 4 is the reverse of Scene 3.

To develop realistic coordinated control strategies, multi-USV systems need an adaptive navigation strategy to face unpredictable environmental forces such as waves, wind, and water currents. A starting step toward this goal is to have a digital environment with re-

alistic modeling where designers can assess their control strategies under different degrees of environmental disturbances. While some physics-based simulators exist for other types of unmanned vehicles, very few exist in the literature for simulating multi-USV missions. Moreover, to the best of our knowledge, the few available simulators do not provide the data storage engine that can facilitate scenario recreations during and after simulation. The digital maritime environment developed in this work provides this functionality.

On the other hand, learning in the digital environment can be time consuming as the complexity of the environment increases. However, our proposed learning algorithms were able to learn a stable control policy that achieved the coordination requirement of the multi-USV systems after a few generations and the evolution of classifiers as can be seen in the learning performance graphs. This, we believe, is facilitated by the behavior-driven approach as the objective within each behavior can easily be formulated. By modeling the physical environment and constraints of USVs in a digital environment, the obtained control systems could be implemented in real systems, as the controllers were able to perform credibly in settings they were not trained on. The above results show the robustness and scalability of the controllers realized by our training system and approach.

7. Conclusions and Future Work

This work presents a training simulation platform that includes a realistic simulation environment. We present a generic framework that can be used for training system designs that are flexible enough and easily scalable. An abstraction of the physical platforms and environment dovetail immunized behavior-driven decision-making and behavior learning are the key contributions of this work. Our approach to implementation enables different learning approaches and algorithms to be developed as learning or training algorithms and also allows the various components of the physical simulator to scale at the local level without major changes to other components.

A unified approach to USVs training presents a modular and simple development process for intelligent controllers. The purpose of all the experiments was to obtain realistic controllers that are adaptable and understandable to meet the requirement in real USV and the physical environment. As can be seen in the performance graphs provided, the training system is able to discover policies that improve the performance of the USVs on their tasks. By continually updating the non-self database, agents do not waste time in evaluating antibodies that are already evaluated by others, hence speeding up the training process to realize a stable policy with few simulations. Simulation results show that the resulting controllers can achieve an average wining rate between 52% and 97.6% in all test cases, indicating the effectiveness of the proposed approach and its feasibility in realizing adaptive controllers for efficient cooperative decision-making among multiple unmanned systems. The results further point to the importance of the model's environmental disturbances since the performance of the controllers increases when transferred to calm environments. Even though the controllers' performance decreases with the different teams' configuration, about 20% of this performance degradation can be attributed to the environmental disturbances at sea.

Finally, using the proposed system, real or simulated sensor data can be obtained and processed based on the model definition. This allows for straightforward switching between real and simulated sensors and actuators. We also introduce, as part of our contribution, more realistic multi-USV learning tasks that can be used for evaluating cooperative control strategies and for multi-agent reinforcement learning research advancement. Subsequently we intend to improve parts of the system by mathematical models and real equipment will be used in a human-in-the-loop configuration and to acquire real operational data from field tests to be used by the trainer to improve the tactical behaviors of unmanned systems. In the future, more experiments will be conducted by implementing different learning algorithms on the training system. A field test will be carried out to ascertain the performance of these controllers in the real world. We will also design and

improve the training algorithm and develop more real-world scenarios that can be used as benchmarks for training and designing controls systems for multi-USV systems.

Author Contributions: Conceptualization, supervision and funding acquisition, Y.X.; methodology, software, writing—original draft preparation, S.N.; software and validation, writing—review and editing, K.P.; software and validation, W.S.; validation, experiments and editing, R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61370151 and in part by the National Science and Technology Major Project of China under Grant 2015ZX03003012.

Conflicts of Interest: There is no conflict of interest.

References

1. Dong, Y.; Zou, Q.; Zhang, R.; Kang, L.; Ren, C. An USV controlling autonomy level algorithm based on PROMMETHEE. In Proceedings of the 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 12–15 June 2016; pp. 2460–2465. [CrossRef]
2. Von Ellenrieder, K.D. Development of a USV-based bridge inspection system. In Proceedings of the OCEANS 2015—MTS/IEEE Washington, Washington, DC, USA , 19–22 October 2015; pp. 1–10.
3. Zhang, J.; Xiong, J.; Zhang, G.; Gu, F.; He, Y. Flooding disaster oriented USV UAV system development demonstration. In Proceedings of the OCEANS 2016, Shanghai, China, 10–13 April 2016; pp. 1–4.
4. Shriyam, S.; Shah, B.; Gupta, S. Online Task Decomposition for Collaborative Surveillance of Marine Environment by a Team of Unmanned Surface Vehicles. *J. Mech. Robot.* **2018**, *10*. [CrossRef]
5. Peng, Y.; Yang, Y.; Cui, J.; Li, X.; Pu, H.; Gu, J.; Xie, S.; Luo, J. Development of the USV 'JingHai-I' and sea trials in the Southern Yellow Sea. *Ocean Eng.* **2017**, *131*, 186–196. [CrossRef]
6. Simetti, E.; Turetta, A.; Casalino, G.; Storti, E.; Cresta, M. Protecting Assets within a Civilian Harbour through the Use of a Team of USVs: Interception of Possible Menaces. In Proceedings of the IARP workshop on Robots for Risky Interventions and Environmental Surveillance-Maintenance (RISE'10), Sheffield, UK, 20–21 January 2010.
7. Corfield, S.; Young, J. Unmanned Surface Vehicles—Game Changing Technology for Naval Operations. In *Advances in Unmanned Marine Vehicles*; IET: London, UK, 2006; pp. 311–328. [CrossRef]
8. Pinko, E. Unmanned Vehicles in the Maritime Domain Missions, Capabilities, Technologies and Challenges. 2019. Available online: https://www.researchgate.net/publication/332420996 (accessed on 19 April 2021).
9. Jakuba, M.V.; Kinsey, J.C.; Partan, J.W.; Webster, S.E. Feasibility of low-power one-way travel-time inverted ultra-short baseline navigation. In Proceedings of the OCEANS 2015—MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015; pp. 1–10. [CrossRef]
10. Suzuki, N.; Kitajima, H.; Kaba, H.; Suzuki, T.; Suto, T.; Kobayashi, A.; Ochi, F. An experiment of real-time data transmission of sonar images from cruising UUV to distant support vessel via USV: Development of underwater real-time communication system (URCS) by parallel cruising. In Proceedings of the OCEANS 2015—Genova, Genova, Italy, 18–21 May 2015; pp. 1–6. [CrossRef]
11. Claus, B.; Kinsey, J.; Girdhar, Y. Towards persistent cooperative marine robotics. In Proceedings of the 2016 IEEE/OES Autonomous Underwater Vehicles (AUV), Tokyo, Japan, 6–9 November 2016; pp. 416–422. [CrossRef]
12. Mitra, S.; Hayashi, Y. Neuro-fuzzy rule generation: Survey in soft computing framework. *IEEE Trans. Neural Netw.* **2000**, *11*, 748–768. [CrossRef] [PubMed]
13. Ernest, N. Genetic Fuzzy Trees for Intelligent Control of Unmanned Combat Aerial Vehicles. Ph.D. Thesis, University of Cincinnati, Cincinnati, OH, USA, 2015. [CrossRef]
14. Sabra, A.; Fung, W.K. A Fuzzy Cooperative Localisation Framework for Underwater Robotic Swarms. *Sensors* **2020**, *20*, 5496. [CrossRef] [PubMed]
15. Ma, X.; Xia, L.; Zhao, Q. Air-Combat Strategy Using Deep Q-Learning. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 3952–3957. [CrossRef]
16. Raboin, E.; Svec, P.; Nau, D.; Gupta, S. Model-predictive target defense by team of unmanned surface vehicles operating in uncertain environments. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 3517–3522.
17. Woo, J.; Yu, C.; Kim, N. Deep reinforcement learning-based controller for path following of an unmanned surface vehicle. *Ocean Eng.* **2019**, *183*, 155–166. [CrossRef]
18. Garg, A.; Hasan, Y.A.; Yañez, A.; Tapia, L. Defensive Escort Teams via Multi-Agent Deep Reinforcement Learning. *arXiv* **2019**, arXiv:1910.04537.
19. Han, W.; Zhang, B.; Wang, Q.; Luo, J.; Ran, W.; Xu, Y. A Multi-Agent Based Intelligent Training System for Unmanned Surface Vehicles. *Appl. Sci.* **2019**, *9*, 1089. [CrossRef]
20. Torres-Torriti, M.; Arredondo, T.; Castillo-Pizarro, P. Survey and comparative study of free simulation software for mobile robots. *Robotica* **2014**, *1*, 1–32. [CrossRef]

21. Paravisi, M.; dos Santos, D.H.; Jorge, V.A.M.; Heck, G.; Gonçalves, L.M.G.; Amory, A.M. Unmanned Surface Vehicle Simulator with Realistic Environmental Disturbances. *Sensors* **2019**, *19*, 1068. [CrossRef]
22. Velasco, O.; Valente, J.; Alhama Blanco, P.J.; Abderrahim, M. An Open Simulation Strategy for Rapid Control Design in Aerial and Maritime Drone Teams: A Comprehensive Tutorial. *Drones* **2020**, *4*, 37. [CrossRef]
23. Borreguero, D.; Velasco, O.; Valente, J. Experimental Design of a Mobile Landing Platform to Assist Aerial Surveys in Fluvial Environments. *Appl. Sci.* **2019**, *9*, 38. [CrossRef]
24. Bingham, B.; Agüero, C.; McCarrin, M.; Klamo, J.; Malia, J.; Allen, K.; Lum, T.; Rawson, M.; Waqar, R. Toward Maritime Robotic Simulation in Gazebo. In Proceedings of the OCEANS 2019 MTS/IEEE SEATTLE, Seattle, WA, USA, 27–31 October 2019; pp. 1–10. [CrossRef]
25. Garg, S.; Quintas, J.; Cruz, J.; Pascoal, A.M. NetMarSyS–A Tool for the Simulation and Visualization of Distributed Autonomous Marine Robotic Systems. In Proceedings of the 2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV), St. Johns, NL, Canada, 30 September–2 October 2020; pp. 1–5. [CrossRef]
26. Smith, R.; Dike, B.; Mehra, R.; Ravichandran, B.; El-Fallah, A. Classifier systems in combat: Two-sided learning of maneuvers for advanced fighter aircraft. *Comput. Methods Appl. Mech. Eng.* **2000**, *186*, 421–437. [CrossRef]
27. Studley, M.; Bull, L. X-TCS: Accuracy-based learning classifier system robotics. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2–5 September 2005; Volume 3, pp. 2099–2106.
28. Wang, C.; Wiggers, P.; Hindriks, K.; Jonker, C.M. Learning Classifier System on a humanoid NAO robot in dynamic environments. In Proceedings of the 2012 12th International Conference on Control Automation Robotics Vision (ICARCV), Guangzhou, China, 5–7 December 2012; pp. 94–99.
29. Smith, R.E.; Dike, B.A.; Ravichandran, B.; El-Fallah, A.; Mehra, R.K. Two-Sided, Genetics-Based Learning to Discover Novel Fighter Combat Maneuvers. In *Applications of Evolutionary Computing*; Boers, E.J.W., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; pp. 233–242.
30. Tosik, T.; Maehle, E. MARS: A simulation environment for marine robotics. In Proceedings of the 2014 Oceans—St. John's, St. John's, NL, Canada, 14–19 September 2014; pp. 1–7.
31. Bonarini, A. An Introduction to Learning Fuzzy Classifier Systems. In Proceedings of the IWLCS 1999, Orland, FL, USA, 13 July 1999; pp. 83–106. [CrossRef]
32. Booker, L. Classifier Systems that Learn Internal World Models. *Mach. Learn.* **1988**, *3*, 161–192. [CrossRef]
33. Booker, L.; Goldberg, D.; Holland, J. Classifier systems and genetic algorithms. *Artif. Intell.* **1989**, *40*, 235–282. [CrossRef]
34. Nantogma, S.; Ran, W.; Yang, X.; Xiaoqin, H. Behavior-based Genetic Fuzzy Control System for Multiple USVs Cooperative Target Protection. In Proceedings of the 2019 3rd International Symposium on Autonomous Systems (ISAS), Shanghai, China, 29–31 May 2019; pp. 181–186. [CrossRef]
35. Hunt, J.E.; Cooke, D.E. Learning using an artificial immune system. *J. Netw. Comput. Appl.* **1996**, *19*, 189–212. [CrossRef]
36. Farmer, J.; Packard, N.H.; Perelson, A.S. The immune system, adaptation, and machine learning. *Phys. D Nonlinear Phenom.* **1986**, *22*, 187–204. [CrossRef]
37. Jerne, N. Towards a network theory of the immune system. *Ann. Immunol.* **1974**, *125 C*, 373–389.
38. Burnet, F.M.F.M. *The Clonal Selection Theory of Acquired Immunity*; Vanderbilt University Press: Nashville, TN, USA, 1959; p. 232. Available online: https://www.biodiversitylibrary.org/bibliography/8281 (accessed on 6 February 2021).
39. Matzinger, P. The danger model: A renewed sense of self. *Science* **2002**, *296*, 301–305. [CrossRef]
40. Kong, X.; Liu, D.; Xiao, J.; Wang, C. A multi-agent optimal bidding strategy in microgrids based on artificial immune system. *Energy* **2019**, *189*, 116154. [CrossRef]
41. De Castro, L.; Von Zuben, F. The Clonal Selection Algorithm with Engineering Applications. *Artif. Immune Syst.* **2001**, *8*, 36–39.
42. Youssef, A.; Osman, M.; Aladl, M. A Review of the Clonal Selection Algorithm as an Optimization Method. *Leonardo J. Sci.* **2017**, *16*, 1–4.
43. Michel, O. Webots: Professional Mobile Robot Simulation. *J. Adv. Robot. Syst.* **2004**, *1*, 39–42.
44. Koenig, N.; Howard, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2149–2154. [CrossRef]
45. Prats, M.; Pérez, J.; Fernández, J.J.; Sanz, P.J. An open source tool for simulation and supervision of underwater intervention missions. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012; pp. 2577–2582. [CrossRef]
46. McCue, L. Handbook of Marine Craft Hydrodynamics and Motion Control [Bookshelf]. *IEEE Control Syst. Mag.* **2016**, *36*, 78–79. [CrossRef]
47. Velueta, M.; Rullan, J.; Ruz-Hernandez, J.; Alazki, H. A Strategy of Robust Control for the Dynamics of an Unmanned Surface Vehicle under Marine Waves and Currents. *Math. Probl. Eng.* **2019**, *2019*, 1–12. [CrossRef]
48. Xiao, L.; Jouffroy, J. Modeling and Nonlinear Heading Control of Sailing Yachts. *IEEE J. Ocean. Eng.* **2014**, *39*, 256–268. [CrossRef]
49. Tosik, T.; Schwinghammer, J.; Feldvoß, M.J.; Jonte, J.P.; Brech, A.; Maehle, E. MARS: A simulation environment for marine swarm robotics and environmental monitoring. In Proceedings of the OCEANS 2016, Shanghai, China, 10–13 April 2016. [CrossRef]
50. Hinsinger, D.; Neyret, F.; Cani, M.P. Interactive Animation of Ocean Waves. In Proceedings of the ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA), San Antonio, TX, USA, 21–22 July 2002. [CrossRef]

51. Thon, S.; Dischler, J.; Ghazanfarpour, D. Ocean waves synthesis using a spectrum-based turbulence function. In Proceedings of the Computer Graphics International 2000, Geneva, Switzerland, 19–24 June 2000; pp. 65–72. [CrossRef]
52. Nantogma, S.; Xu, Y.; Ran, W. A Coordinated Air Defense Learning System Based on Immunized Classifier Systems. *Symmetry* **2021**, *13*, 271. [CrossRef]
53. Raza, A.; Fernandez, B.R. Immuno-inspired robotic applications: A review. *Appl. Soft Comput.* **2015**, *37*, 490–505. [CrossRef]
54. Holland, J.H. Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems. In *Machine Learning: An Artificial Intelligence Approach*; Michalski, R.S., Carbonell, J.G., Mitchell, T.M., Eds.; Morgan Kaufmann: Los Altos, CA, USA, 1986; Volume 2.
55. Smith, S.F. A Learning System Based on Genetic Adaptive Algorithms. Ph.D. Thesis, University of Pittsburgh, Pittsburgh, PA, USA, 1980.