Simulation of Embodied Cyber Physical System Based on Modelica/MWORKS: A Case Study of Intelligent Unmanned Surface Vessel

Zhiguo Zhou[1,4,5]; Xuehua Zhou[1]; Lin DU[2]; Peiquan Ma[1]; Xiang Wang[1]; Ying chen[1]; Mingjia Liu[1]; Tengyue Wang[1]; Lixin Hui[3]; Cun Zeng[3]

[1]School of Integrated Circuits and Electronics, Beijing Institute of Technology, China, {zhiguozhou, xuehuazhou, 3220242141, 3220231862, 3120221304,3220242100}@bit.edu.cn

[2]Maritime and Transportation Department, Ningbo University, China, dulin1@nbu.edu.cn

[3]Suzhou Tongyuan Soft Control Information Technology Co., Ltd, China, {huilx,zengc}@tongyuan.cc

[4]Beijing Institute of Technology, Zhuhai 519088 No. 6 Jinfeng Road, Tangjiawan Town, Xiangzhou District, Zhuhai City, Guangdong Province, China

[5]Tangshan Research Institute of BIT, Tangshan 063000,China

Abstract

This paper proposes a new paradigm of the Embodied Cyber-Physical System (Embodied CPS, ECPS) to address the issues of the disconnection between physical laws and intelligent decision-making and the insufficient interaction with dynamic environments in the modeling and simulation of traditional CPS. ECPS achieves unified modeling of physical laws and autonomous decision-making through the "perception-decision-action" closed loop. To verify ECPS, an embodied space framework based on Modelica/MWorks is designed. Through three major technological innovations: constructing an embodied domain modeling specification and embedding the Navier-Stokes equations into the training of the policy network; expanding the syntax and semantics of Modelica, encapsulating physical constraint reinforcement learning components, and establishing a gradient interaction protocol between the Physics-Informed Neural Network (PINN) and Modelica equations; building a digital twin-hardware-in-the-loop co-simulation platform based on the FMI/SSP protocol to establish a collaborative verification link between high-precision physical simulation and real-time decision-making. Taking the Unmanned Surface Vehicle (USV) as the carrier, the full-process method from dynamic modeling, reinforcement learning strategy training to virtual-real environment co-simulation is demonstrated. Experiments verify the effectiveness of this framework in achieving the closed-loop coupling of physical simulation and intelligent decision-making under complex sea conditions, providing a methodological foundation for interpretable modeling and verifiable simulation in the development of embodied intelligence.

Keywords: embodied intelligence, embodied domain, embodied spatial, physical information neural network

1 Introduction

1.1 Background

With the increasing demand for the adaptability of autonomous systems in dynamic physical environments, Cyber-Physical Systems (CPS) [1] are gradually evolving towards Embodied Cyber-Physical Systems (Embodied CPS, ECPS). Its core feature lies in that intelligent agents need to conduct real-time interactions with the physical environment through the "perception-decision-action" closed loop [2]. In the traditional CPS framework, physical models and intelligent algorithms are often designed in a segmented way, resulting in bottlenecks such as the lack of compliance with physical laws and insufficient dynamic interaction modeling when intelligent agents are deployed in real environments. For example, in scenarios where dynamic physical constraints (such as the torque limitations of robotic arm joints and the dynamic boundaries of mobile robots) coexist with high-dimensional environmental uncertainties (such as the random distribution of obstacles and multimodal disturbances), existing methods find it difficult to achieve the deep integration of physical laws, environmental interactions, and autonomous decision-making. How to construct a simulation system with the ternary coupling of "physics-information-environment" [3], and organically unify the strong constraints of physical laws with the data-driven environmental adaptability has become the core challenge in the development of embodied intelligence systems.

1.2 Related Work

In terms of ECPS simulation methods, current research presents two typical paths: One is based on the Modelica multi-domain modeling tool (such as MWORKS), which accurately describes the constitutive relations of physical systems through acausal equations. However, in terms of the deep integration of the intelligent decision-making layer and the deep learning framework, it is still limited by the difficult problem of collaborative solving of discrete-continuous systems. The other is to adopt the Physics-Informed Neural Network (PINN) [4]. By embedding physical laws such as the Navier-Stokes equations into the neural network training process through the residual constraints of differential equations, it demonstrates the advantage of following physical constraints in embodied intelligence scenarios such as fluid mechanics prediction and robotic motion control. It is worth noting that recent research attempts to integrate PINN with Modelica [5], and realizes the co-simulation of the neural network model and the multi-domain physical model through the Functional Mock-up Interface (FMI) standard. However, there is still a lack of standardized protocols in key technical aspects such as the gradient backpropagation mechanism and the alignment of spatio-temporal scales between the two. Overall, the development of current ECPS simulation tools is still in the exploratory stage. Existing research generally faces three major bottlenecks: the lack of a semantic-level integration mechanism between physical equations and data-driven models; the insufficient standardization of the protocol stack for real-time interaction between heterogeneous simulation platforms; and the contradiction between the efficiency and interpretability of the representation of the embodied space in a high-dimensional dynamic environment.

1.3 Challenges

The core challenges of ECPS simulation are as follows:
It is difficult to couple multi domain models: the traditional Modelica standard library lacks a specific agent interface, and cannot uniformly describe the real-time interaction between physical equations (such as ship dynamics) and intelligent algorithms (such as DRL) [6];
Physical constraints are separated from intelligent decision-making: data-driven methods are easy to generate actions that violate physical laws (such as exceeding the safety threshold of roll angle), while traditional control theory is difficult to deal with high-dimensional environmental uncertainty; Lack of virtual and real collaborative verification system: the existing tool chain (such as Gazebo+ROS) [7] has a gap between the physical simulation accuracy (microsecond step size) and the efficiency of intelligent algorithm (millisecond response), and lacks standardized interface protocol.

1.4 Proposed Approach

In response to the above challenges, this paper proposes an ECPS simulation framework based on the embodied space, and achieves methodological breakthroughs through a three-level technical path. With the modeling specification of the embodied domain as the technical foundation, this framework expands the Modelica language to define standardized interaction interfaces (such as sensors/actuators) and physical constraint description grammars, and supports the semantic-level integration of complex physical models such as the six-degree-of-freedom dynamics of ships and reinforcement learning strategies. Relying on the embodied space architecture, a collaborative simulation link of "physical domain (high-precision fluid mechanics equations) - information domain (DRL decision-making) - embodied domain (PINN gradient interaction)" is constructed. The constraints of the Navier-Stokes equations are embedded in the strategy training through the PINN-Modelica protocol to ensure that intelligent decisions strictly follow the physical boundaries. Based on the FMI/SSP protocol, multi-time scale scheduling (1ms-level physical simulation/10ms-level decision-making) is realized. In combination with ROS/Unity, the co-verification of virtual and real environments is achieved, providing a methodological paradigm for the engineering of ECPS in dynamic environments.

2 Methodology

2.1 Modelica Modeling Specification for the Embodied Domain

To meet the needs of intelligent design for modern autonomous systems and address the issues of insufficient intelligence in traditional cyber-physical integration systems and weak theoretical constraints of intelligent algorithms, this paper constructs a three-tier technical system for the Modelica modeling specification of the embodied domain: forming the embodied domain specification based on the extensible syntax and semantics of Modelica; constructing a verification protocol stack for virtual and real systems based on API interfaces, the FMI standard, and the SSP protocol; and building a physical-intelligence integration engine based on the PINN-Modelica gradient interaction protocol. This specification incorporates intelligent considerations on the basis of the physical domain and the information domain to achieve the unification of physical laws and autonomous decision-making. On the one hand, by utilizing the extensible syntax and semantics of Modelica, a general modeling specification and architecture for embodied entities are formed. The model components of different objects are standardized, and generalized models of general autonomous systems such as autonomous vehicles and unmanned surface vehicles are constructed. This provides theoretical constraint boundaries for the embodied data training of various unmanned systems and scenario working conditions, and is verified in real time through the physics-informed neural network, breaking through the collaborative bottleneck between semantic ambiguity and physical compliance. On the other hand, based on the PINN-DRL architecture, physical laws such as the Navier-Stokes equations are embedded in the neural network training.

DOI 10.3384/ecp218591

Through the regularization constraints of physical equations, the model not only fits the data but also follows physical laws, thereby improving generalization ability, data efficiency, and interpretability of time decision-making in dynamic environment.

2.2 Extension Specification for Modelica Embodied Domains

Traditional engineering often faces problems such as poor compatibility of heterogeneous interface protocols, discretization of physical constraint description, and difficulty in cross domain simulation coupling, resulting in low efficiency of agent modeling and limited simulation credibility. Although the Modelica standard library can solve the unified modeling and simulation problems in many fields, such as Electromechanical, hydraulic and thermal control, it lacks standardized support for perception action closed-loop, real-time physical constraints and intelligent strategy fusion, and cannot realize the deep coupling between the agent and the physical environment. It is difficult to cope with the rapid development of embodied intelligence in the fields of robots, unmanned systems and so on.

can switch autonomously according to demand. The information domain includes reinforcement learning strategies and neural network reductions, achieving closed-loop optimization of physical laws and autonomous decision-making. The embodied domain is responsible for the environmental interaction capabilities of the entire embodied space, including communication, 3D visual scenes, external devices, etc. The various parts are connected through the embodiment domain language extension, achieving a technical closed loop of "perception-decision-action".

Therefore, in order to support the formal modeling of embodied agents and improve the modeling ability of complex systems, it has three core functional modules: standardized interaction framework: defining the base class and open interface of embodied agents, unifying the sensor actuator communication standard and reducing the amount of hardware adaptation code. Integration of physical intelligence: innovative introduction of PINN hybrid modeling syntax and physical constraint reinforcement learning library to optimize AI strategy in strict accordance with dynamic laws. Multi domain collaborative support: define

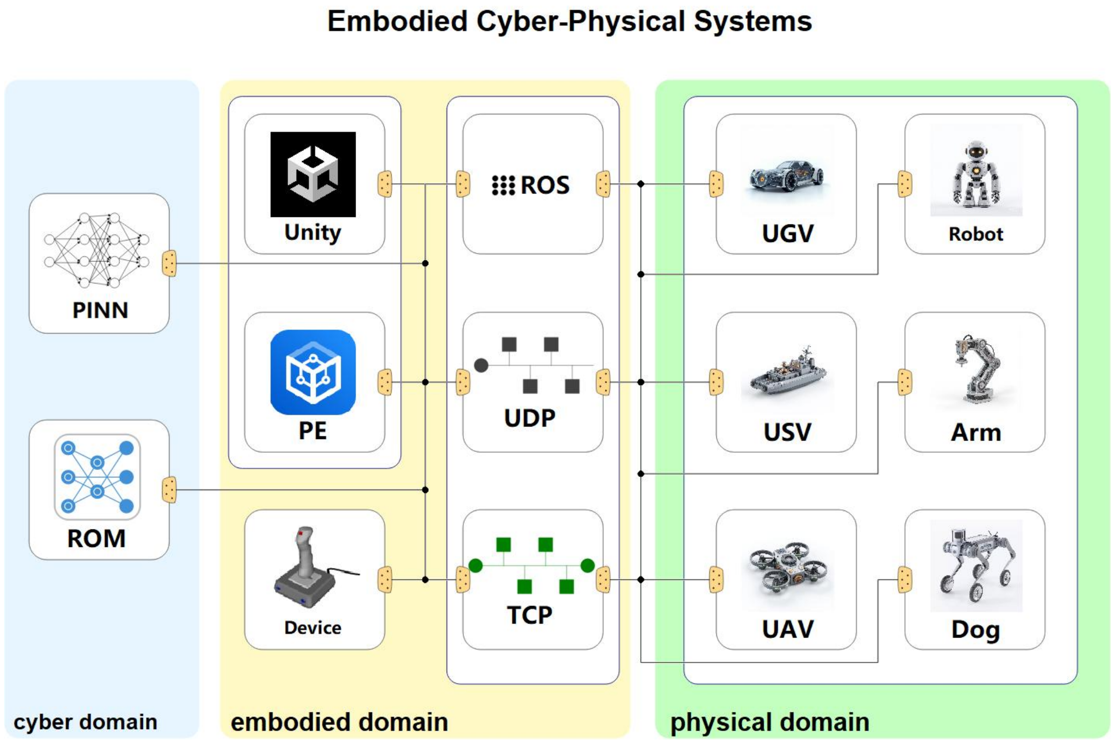

Figure 1. Embodied Cyber-Physical Systems.

Modelica general standard interface based on API interface or communication protocol to realize the coupling simulation of control logic, physical model and environmental factors under a unified time base.

The embodied cyber-physical system consists of three parts: the physical domain, the information domain, and the embodied domain. The physical domain includes various embodied intelligent agents, such as autonomous vehicles, unmanned ships, drones, robots, robotic arms, robotic dogs, and so on. Embodied intelligent agents satisfy the concept of Modelica replaceable classes and

Specifically, this specification extends the following syntax for Modelica language:

- add the embodied keyword to support the declaration of embodied agents.

The traditional Modelica modeling is centered on physical entities and lacks the explicit expression of the cognitive behavioral characteristics of agents. The embodied keyword is realized through the reconstruction of the syntax layer:

1. Ontology mapping: forcibly declare the interfaces of perceptual input and action output, so that the code structure is strictly isomorphic with the perceptual action cycle of embodied intelligence.
2. Quantification of cognitive delay: by using cognitivedelay and other built-in parameters, the biologically inspired nerve conduction delay (about 50-200ms) is incorporated into the dynamic equation to avoid the distortion problem that the control delay is simplified to the ideal zero delay in the traditional modeling.
3. Constraint prefix declaration: physical constraints are required to be declared through constraint blocks at the agent definition stage to ensure that subsequent control strategy development naturally meets the physical feasibility.

The extension syntax is as follows:

Listing 1. Embodied keyword

```
embodied <AgentName>
  extends <BaseClassName>;
  parameter <Type> <paramName> =
<value> "Description";
  input <Type> <sensorName>[N] ";
  output <Type> <actuatorName>[M];
  replaceable
<PolicyPackage>.<Algorithm> policy
    constrainedby EmbodiedPolicyInterface;
  constraint <ConstraintType>
<constraintName>
equation
  <equation>;
end <AgentName>;
```

- physical constraint fusion modeling, introducing the physical constraint description equation, allowing the PINN prediction results to be embedded in the Modelica model.

Traditional physical modeling and data-driven modeling have been separated for a long time, mainly in the following aspects:

1. White box model dilemma: pure physical equations are difficult to accurately describe complex nonlinear phenomena (such as turbulence and material fatigue), and the error of traditional Navier Stokes equations in a ship motion case is significant.
2. Black box model risk: the pure neural network prediction may violate the law of conservation of physics, and the non-conservation of energy in the trajectory prediction of a manipulator leads to simulation collapse.

The differential equation level fusion of physical equations and neural networks is realized through the PINN_interface syntax. The extended syntax includes:

Listing 2. PINN_interface

```
equation
 <OutputVariable> = PINN_Interface(
   inputs = {<InputVariable1>,
<InputVariable2>, ...},
   modelFile = "pinn_model.onnx")
  <PhysicalVariable> = <PhysicsEquation>
+ PINN_CorrectionTerm ;
```

- physical constraint Reinforcement Learning Library.

Develop physical component library and provide DRL algorithm with physical residual regularization (such as PPO Physics). Its loss function is defined as:

$$L = \alpha \cdot L_{DRL} + \beta \cdot \|N(u) - PDE(U)\|^2 \tag{1}$$

Define DRL algorithm class with physical constraints. Take PPO physics algorithm as an example:

Listing 3. PPO physics

```
package PhysicsRL
model PPO_Physics
   extends DRL.BasePPO;
   parameter Real lambda_physics = 0.5 ;
  protected
   function compute_loss
     input Real[:,:] trajectories;
     input PhysicalModel physics_model;
     output Real total_loss;
   algorithm
     Real policy_loss :=
compute_policy_loss(trajectories);
     Real value_loss :=
compute_value_loss(trajectories);
     Real physics_residual :=
physics_model.evaluate_residual(
       states = trajectories[:,1:state_dim],
       actions =
trajectories[:,state_dim+1:end]);
     total_loss := policy_loss + value_loss +
       lambda_physics * physics_residual^2;
   end compute_loss;
  end PPO_Physics;
end PhysicsRL;
```

The specification provides the underlying support for industrial 4.0 core scenarios such as digital twins and agent group collaboration, and promotes the leap of embodied intelligence from theoretical verification to engineering implementation.

The embodied space architecture based on Modelica mainly includes four core modules:

- Physical domain modeling layer: build a 6-DOF ship dynamics model based on Modelica/MultiBody, integrate wave disturbance

DOI
10.3384/ecp218591

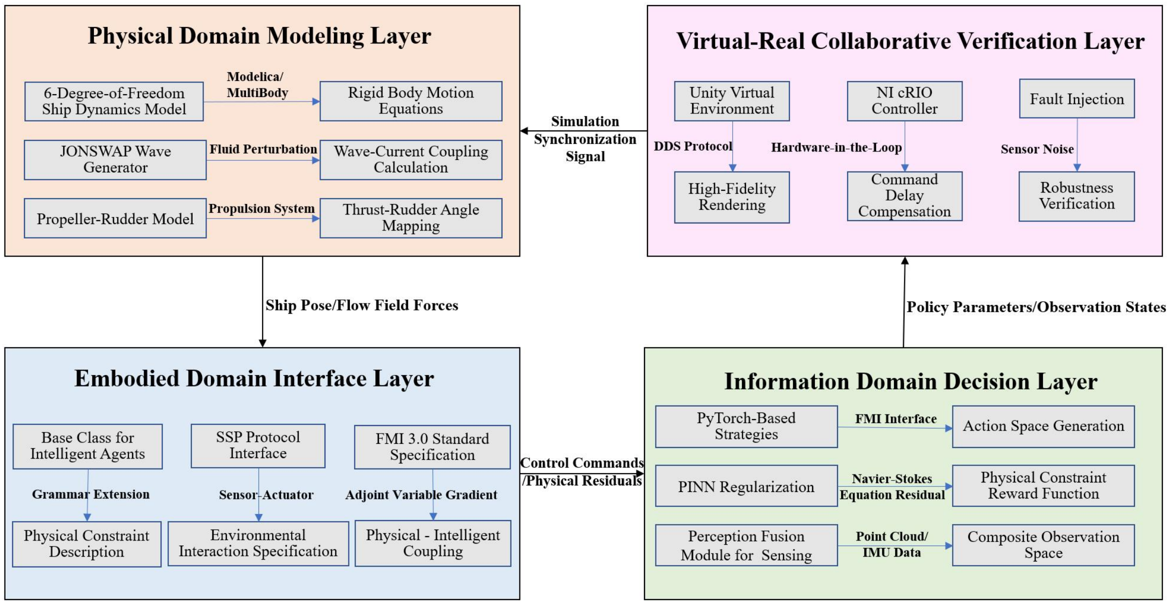

Figure 2. Embodied space architecture diagram based on Modelica.

(JONSWAP spectrum generator) and propulsion system (propeller actuator joint model);

- Embodied domain interface layer: extend Modelica language to define the embodied agent base class, support sensor actuator interface (SSP protocol) and physical constraint description syntax;
- Information domain decision-making layer: call the DRL policy network implemented by PyTorch through Python Modelica bidirectional interface (FMI standard) to dynamically generate control instructions;
- Virtual and real collaborative verification layer: build a high fidelity marine virtual environment based on unity engine, and realize millisecond data synchronization with Modelica simulation through DDS (data distribution service).

2.3 Modelica Embodied Space Architecture

2.3.1 Technical route

Traditional engineering often faces problems such as The main design idea of the embodied space based on Modelica is to achieve the two-way cognitive coupling between intelligent agents and the dynamic environment through the physical-information fusion mechanism, forming a three-layer architecture of "physical entity-digital twin-embodied intelligence". For different physical entities of autonomous systems, a system model and control algorithm are constructed based on the multi-disciplinary unified modeling software MWORKS.Sysplorer to achieve a control closed loop between the system model and the control algorithm. A visualization environment for the autonomous system is built based on the Unity 3D virtual engine to form a digital prototype of the autonomous system. Based on the virtual-real collaborative verification protocol stack of API interfaces, the SSP protocol, and the FMI standard, a unified solution architecture for physical models and autonomous decision-making algorithms is realized, breaking down the barriers between the virtual world and the physical world, expanding the boundary conditions of the digital prototype, and enhancing the ability of the physical entities of the autonomous system to learn independently and adapt to complex environments, thus forming a digital twin in the embodied space. Based on the PINN-Modelica gradient interaction protocol, an embodied intelligence training environment is added on the basis of the digital twin of the autonomous system, integrating key technologies such as perception fusion and decision-making reasoning to improve the intelligent characteristics of the embodied entity. The overall technical roadmap and framework diagram of the embodied space are as shown in the figure 2.

2.3.2 Implementation framework

The embodied space architecture based on Modelica mainly includes four core modules, as shown in the figure 3:

- Physical domain modeling layer: build a 6-DOF ship dynamics model based on Modelica/MultiBody, integrate wave disturbance (JONSWAP spectrum generator) and propulsion system (propeller actuator joint model);
- Embodied domain interface layer: extend Modelica language to define the embodied agent base class, support sensor actuator interface (SSP protocol) and physical constraint description syntax;

- Information domain decision-making layer: call the DRL policy network implemented by PyTorch through Python Modelica bidirectional interface (FMI standard) to dynamically generate control instructions;
- Virtual and real collaborative verification layer: build a high fidelity marine virtual environment based on unity engine, and realize millisecond data synchronization with Modelica simulation through DDS (data distribution service).

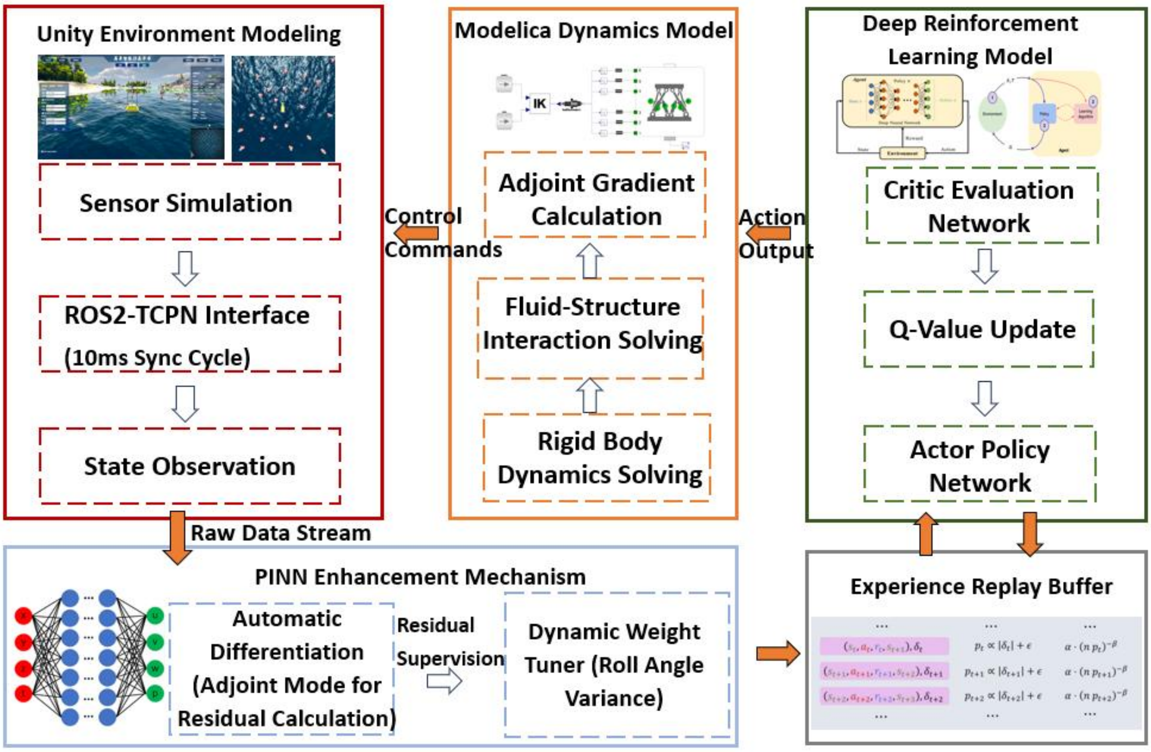

Figure 3. PINN enhanced DRL framework.

2.3.3 Virtual-Physical Co-Simulation Protocol Stack

The virtual real collaborative verification protocol stack of embodied space is the core module to realize the system integration and co-simulation of embodied information physical system. On the one hand, the virtual real collaborative verification protocol stack can organically integrate the digital sample ship and intelligent algorithm to form an intelligent ship model for virtual simulation verification; On the other hand, the virtual and real collaborative verification protocol can complete the data interaction between the virtual model and the physical ship, forming the virtual and real collaborative verification of unmanned ship.

1. Cross-Platform Interface Design.

The interface design of embodied information physical system needs to consider three levels of platform data interaction: physical modeling system, virtual training environment and physical deployment environment. For the Modelica model of the unmanned ship, a general theoretical model of the unmanned ship is formed based on the physical modeling system platform, and a physical simulation interface is formed based on the standard API interface or communication protocol to interact with the three-dimensional visual environment; For the intelligent algorithm construction and data training scenario, based on the virtual training environment and the universal model of unmanned ship, the virtual environment interface is formed for the physical simulation interface, and the theoretical calculation is introduced as the constraint boundary in the three-dimensional visual environment and the large data sample training environment, so as to form a decision-making strategy model suitable for the actual unmanned ship. For the physical deployment environment of unmanned ship, the strategy deployment interface is constructed, and the algorithm model of rapid deployment training is completed, so as to realize the virtual real combination of unmanned ship and build a complete unmanned ship body space.

Physical simulation interface: a collaborative simulation interface for building physical models based on the MWORKS platform API interface or the general communication protocol standard. By analyzing the differential algebraic equation (DAE) generated by Modelica multi domain unified modeling language, the 6-DOF model of the unmanned ship can realize data

DOI
10.3384/ecp218591

interaction on different platforms, and finally realize the decoupling deployment of the physical simulation unit. Through the topology description file based on the physical simulation interface, the port connection relationship and parameter metadata are automatically maintained, which significantly reduces the repeated modeling cost of data collaboration.

Virtual environment interface: Based on the ROS2 distributed communication framework, develop customized TCPN (timed colored Petri net) plug-in to build a deterministic data channel. The high fidelity 3D visualization model is embedded in unity virtual environment, and the status synchronization with Physical model of unmanned ship is realized through ROS2-TCPN middleware. A 10ms fixed cycle hard real-time communication link is established.

Policy deployment interface: build AI policy deployment interface based on ONNX (Open Neural Network Exchange) open format. After compatibility check and optimization, the deep reinforcement learning strategy network trained by PyTorch/TensorFlow is transformed into a lightweight ONNX model. By integrating the ONNX algorithm engine, the end-to-end reasoning strategy deployment is realized on embedded controllers (such as NVIDIA Jetson AgX Xavier), which meets the real-time control requirements of edge devices under complex conditions.

2. Multi-Temporal Scheduling Mechanism.

The interface design of embodied information physical system under different platform levels and scenarios, the interface scheduling frequency is different, so it is necessary to add multi time scale scheduling module to the virtual and real collaborative verification protocol stack, including hard real-time layer, soft real-time layer and asynchronous event layer.

Hard real-time layer (1ms level): the hard real-time layer refers to building a real-time unmanned ship system model based on Modelica in the physical modeling system platform, and using the DAE (Differential Algebraic Equation) implicit solution algorithm to deal with the coupling effect between the six degree of freedom motion of the unmanned ship rigid body and fluid viscosity. By introducing variable step size integration algorithm or fixed step size integration algorithm, the real-time simulation control is realized on FPGA hardware accelerator card, which provides the bottom constraint boundary for the UAV body space frame.

Soft real-time layer (10ms level): soft real-time layer refers to the realization of data interaction between virtual environment and unmanned ship digital model based on virtual environment interface when intelligent strategy algorithm reasoning and 3D Virtual Environment Rendering of unmanned ship are carried out in virtual environment platform. The soft real-time layer has no real-time requirements for the embodied space frame, and its overall delay is stably controlled within 10ms.

Asynchronous event layer: when unmanned ships handle discrete events such as obstacle generation and task reset, it is necessary to design event driven architecture (EDA) to design discrete event processing mechanism to realize decoupling scheduling of aperiodic operations such as obstacle generation and task reset.

3 Case Study: Intelligent USV

3.1 Case Architecture Design

The embodied domain, as an intermediary layer between the physical domain and the information domain, realizes the deep integration of physical laws and information algorithms through the dynamic perception and adaptive decision - making of embodied agents. The experimental case selects the Embodied Space Simulation Platform on MoHub (https://mohub.net/). This platform is built based on MWORKS and is a comprehensive artificial intelligence platform focusing on embodied space modeling, unmanned system modeling, and simulation testing. Through the integration of multiple spaces and the collaboration of modular engines, it achieves a full - link closed - loop of the physical domain - embodied domain - information domain.

In this case, the USV - 130 unmanned surface vehicle is taken as the research object. Aiming at its navigation and obstacle - avoidance problems during rescue and search operations under uncertain conditions, an embodied intelligence modeling and training framework based on the MWORKS platform and PINN is proposed. Through modeling, simulation, and experimental verification, PINN is used to provide kinematic and dynamic constraints that conform to physical laws for the obstacle - avoidance algorithm. The influence of three environmental factors, namely wind, waves, and currents, is considered, and the calculation of the corresponding disturbing forces and torques is added to the motion mathematical model to make the algorithm simulation more consistent with reality.

3.2 Implementation of Embodied Domain Modeling

In the modeling and simulation of unmanned surface vehicles, high - precision hydrodynamic modeling is a key challenge for achieving the "simulation - to - reality" transfer. Traditional fluid resistance prediction methods based on empirical formulas, due to simplified assumptions, struggle to accurately describe the nonlinear hydrodynamic behavior in complex environments. In this case, PINN (Physics - Informed Neural Network) is introduced. By integrating the Navier - Stokes equations with measured flow field data, a data - physics dual - driven fluid resistance prediction model is constructed. In the MoHub Embodied Space

Simulation Platform, this method has established a multi - scale hydrodynamic calculation framework. At the macroscopic level, the fluid motion control equations are embedded as soft constraints, and at the microscopic level, the data from the pressure sensors on the hull surface are used to optimize the network parameters. This hybrid modeling approach significantly improves the calculation accuracy of complex flow states, reduces the error in the hull resistance characteristics between the virtual and real environments. Moreover, by encoding the spatio - temporal coupling characteristics of the wind, wave, and current disturbance fields into the input dimensions of PINN, the real - time calculation of dynamic environmental disturbance forces is realized, providing a basis for the physically reliable training of motion control algorithms.

3.2.1 Physical Domain Modeling

USV130 is an intelligent unmanned surface vehicle specially designed for scientific research and education. It integrates multi-source sensing system and has intelligent control modules such as speed and heading locking cruise, autonomous return path planning and obstacle avoidance. The main structure of the equipment is made of glass fiber composite materials. The whole ship is 1.3m (L) × 0.64M (W). It is equipped with a dual water jet propulsion system to achieve mobile control. The technical parameters show that its navigation performance includes: 18 cm working draft, 4 m/s peak speed, and 5 m minimum radius of gyration. The propulsion system can support high-precision motion control in complex hydrological environment. As shown in the following figure 3.

Figure 3. Physical Image of USV130 Unmanned Submarine

USV130 is equipped with Ubuntu 20.04 operating system and ROS noetic robot framework, and the underlying motion control adopts PID algorithm. The driving system supports multimodal operation: it can be controlled by the physical remote-control device, the upper computer software and the handle linkage or executed by the autonomous command based on the algorithm. The communication system is equipped with a dedicated base station to achieve stable data transmission with an effective radius of 200 meters.

The physical parameters required to build the unmanned ship are as follows:

Table 1. USV130 partial physical parameter table.

physical parameter	
Hull mass	30kg
Total length	1.3m
Geometric center	0.45m
Total width	0.64m
Hull width	0.27m
Draft	0.18m

3.2.2 Physics-Information Modeling

In this case, a dynamic model based on MWORKS platform is constructed to describe the motion behavior of the unmanned ship under complex hydrodynamic conditions. The dynamic model framework is divided into three parts: dynamic modeling framework, dynamic formula framework and additional mass matrix Mv framework.

- Dynamic equation modeling: the three degree of freedom dynamic equation of the system includes longitudinal velocity, lateral velocity and yaw angular velocity. The specific expression is as follows.

$$Mv + C(v)\mathrm{v} + D(v)\mathrm{v} = \tau \quad (2)$$

Where, M is the mass matrix, $C(v)$ is the Coriolis force matrix, $D(v)$ is the resistance matrix, and τ is the control input generated by the water jet propulsion system.

- Adaptive backstepping controller:

1. Speed control: realize the stable control of longitudinal speed through feedback linearization, and introduce adaptive compensation term to eliminate the uncertainty of resistance.
2. Heading control: Lyapunov function is designed to ensure heading error convergence. The controller expression is as follows:

$$\tau_z = k_1 e_\psi + k_2 r \quad (3)$$

Where, e_ψ is heading error and r is yaw rate.

3. Adaptive parameter estimation.

The adaptive law is introduced to estimate the resistance coefficient and added mass online, and compensate the uncertain disturbance of the system in real time to ensure the robustness and stability of the control system.

4. Improvement of integrated PINN.

In order to further improve the ability of the model to follow the physical laws, the physical information neural network (PINN) can be combined with the unmanned ship dynamics model. The Navier Stokes equation is embedded into the neural network as a physical constraint, and the PINN interface is introduced into the Modelica model to modify the hydrodynamic term in the dynamic equation of the unmanned ship, so that the model can strictly follow the hydrodynamic law while learning the data.

The core of the modeling framework of the dynamic model is to integrate the physical parameters and dynamic characteristics of the unmanned ship. The dynamic equation takes the force and motion state of the hull as the core, and calculates the motion response of the unmanned ship through moment balance. The framework contains the following key inputs:

- physical parameters.

Mass: including the influence of load change on the inertial characteristics of the hull.

Geometric center (LCG): describes the longitudinal position of the center of gravity of the hull.

Drag coefficient (CD) and water density (ρ): affect the hydrodynamic characteristics of unmanned ships.

Hull size parameters: including length (L), draft (T), and hull width (Bhull).

- function and modeling of hydrodynamic coefficient.

Hydrodynamic coefficients (HC) are the key parameters connecting hydrodynamic characteristics and motion equations, and their accuracy directly affects the prediction ability of the simulation model. For the catamaran USV, the hydrodynamic coefficients are divided into linear and nonlinear categories: linear damping coefficient (such as) represents the viscous resistance at low speed; The nonlinear damping coefficient (such as) describes the resistance characteristics dominated by the square term of speed under high-speed conditions. In addition, the additional mass coefficient (such as) reflects the inertial effect of the surrounding fluid when the hull accelerates.

3.3 Embodied Domain Simulation Analysis

After constructing the physical model of the unmanned ship based on the MWORKS.Sysplorer platform and implementing the virtual-real collaborative verification protocol stack based on the ROS framework, by randomly setting routes and obstacles, multi-level and multi-scenario data simulation can be carried out within the embodied space. The simulation data can further improve the training samples, enabling the self-improvement and autonomous learning of the PINN algorithm.

Environmental factors are one of the main factors that separate the unmanned ship model from the actual unmanned ship. By setting dynamic environmental parameters in the virtual environment engine, the ROS framework can transform the perception of disturbance factors such as wind, waves, and currents, achieving the deep coupling between the additional disturbances of the dynamic environment and the physical model of the unmanned ship. This adds physical boundary constraints on the basis of the real embodied training environment. The figure 4 below shows the disturbances transmitted to the physical model of the unmanned ship through the ROS framework.

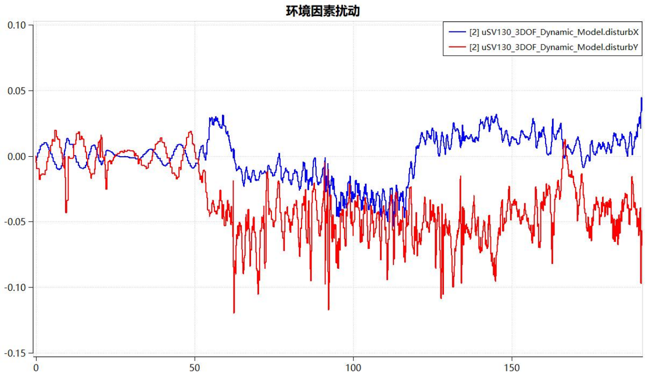

Figure 4. Dynamic Environmental Disturbances of the Unmanned Ship

After setting the route, the entire unmanned ship automatically locates its current position and the target route landmarks. It conducts navigation control by adopting the approach of global path planning and local obstacle avoidance, thus achieving the automatic navigation of the unmanned ship. After the simulation runs, the path changes of the unmanned ship in the dynamic environment can be observed, as shown in the figure 5 below.

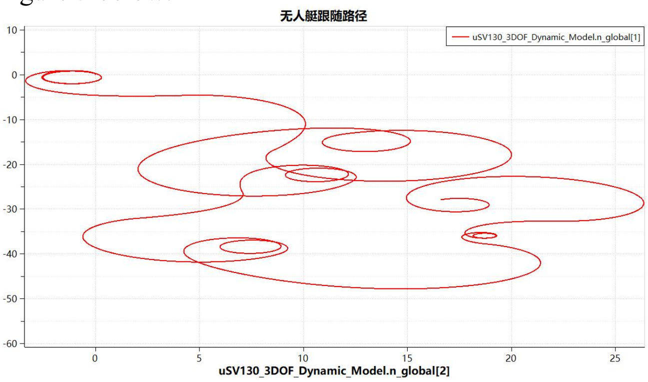

Figure 5. Response Path of the Unmanned Ship

3.4 Simulation Training in the Embodied Space

Traditional physical - based modeling methods have always been unable to clearly describe the hydrodynamic laws of the unmanned ship system. To achieve more accurate virtual - real collaborative verification, the Physics - Informed Neural Network (PINN) is introduced. It corrects the additional hydrodynamic effects in the unmanned ship system, further enhancing the model's ability to follow physical laws and adapt to dynamic environments, and ensuring the accuracy and reliability of the simulation results.

3.4.1 Hydrodynamic Laws

- Navier - Stokes Equation

The N-S equation, as the governing equation for describing the motion of viscous fluids, has its conservation form embedded in the loss function of the Physics - Informed Neural Network (PINN):

$$Residual = \frac{\partial u}{\partial t} + (u \cdot \nabla)u - \nu\nabla^2 u + \frac{1}{\rho}\nabla p \qquad (4)$$

By forcing the prediction results of the neural network to minimize the equation residuals, it ensures that the flow field predictions (such as the flow velocity around the hull and the pressure distribution) conform to the principles of viscous fluid mechanics.

- Added Mass Effect

Based on potential flow theory, the inertial effect of the surrounding fluid when the hull accelerates is calculated as follows:

$$M_A = diag(X_{\dot{u}}, Y_{\dot{v}}, N_{\dot{r}}) \qquad (5)$$

The added mass coefficients are determined by empirical formulas based on the hull's geometric parameters, reflecting the coupling characteristics of the hull shape on the fluid inertial response.

- Nonlinear Viscous Resistance Model

A resistance decomposition dominated by the velocity - squared term is adopted:

$$D_{(v)} = \begin{bmatrix} X_{u|u|}|u|u & 0 & 0 \\ 0 & Y_{v|v|}|v|v + Y_r r & N_v|v|v + N_r r \end{bmatrix} \qquad (6)$$

The longitudinal resistance coefficient ($X\{u|u|\}$) is calibrated according to the wetted surface area of the hull and the Reynolds number, and the lateral/rotational damping coefficients ($Y\{v|v\}, N_r$) are fitted from the ship model towing test data.

Coriolis - Centrifugal Force Coupling Effect

It is explicitly expressed in the hull motion equation as:

$$D_{(v)} = \begin{bmatrix} 0 & 0 & -m(y_g r + v) \\ 0 & 0 & m(x_g r - u) \\ m(y_g r + v) & -m(x_g r - u) & 0 \end{bmatrix} \qquad (7)$$

where (x_g, y_g) is the position of the center of gravity, which directly affects the dynamic coupling of the roll - yaw motion.

3.4.2 Application of the PINN Algorithm

- PINN - corrected Resistance Prediction

Since the traditional resistance model has significant errors in turbulent flow conditions, for the calculation of the fluid resistance of the unmanned ship, the N - S equation is embedded as a soft constraint in the PINN algorithm for training, and a hybrid resistance model is constructed as follows:

$$\dot{D}(v) = D_{phys}(v) + PINN(v;\theta) \qquad (8)$$

- Dynamic Compensation of Added Mass

Considering that the added mass coefficient of the unmanned ship changes with the draft (load fluctuations cause the draft to change by ±10 cm), a PINN surrogate model $M_A = f(T, nabla)$ is established for real - time online estimation:

- Fluid - Motion Coupling Decision - making

Traditional DRL strategies tend to generate actions beyond the feasible domain of fluid dynamics (for example, a sharp turn causing the lateral velocity to exceed the limit). Therefore, it is necessary to introduce a fluid - constraint reward term in the PPO algorithm:

$$r_{physics} = -\,||v_y|| \cdot \prod(|v_y| > v_y^{max}) - ||r|| \cdot \prod(|r| > r^{max}) \qquad (9)$$

and calculate the gradient penalty of the Navier - Stokes equation on the policy parameters through automatic differentiation.

- Wave Disturbance Modeling

Considering the phase lag of the heave/pitch excitation force generated by the JONSWAP wave spectrum, a wave force transfer function is established based on potential flow theory:

$$F_{wave} = \int_{-\infty}^{\infty} H(\omega) \cdot S(\omega) e^{i\omega t} d\omega \qquad (10)$$

Where ω is calculated by the hull strip theory, and $S(\omega)$ is the wave spectrum density.

3.4.3 Deployment and Operation of the Unmanned Ship System

In the deployment and operation phase, the preset water scene and the digital prototype of the ship are loaded based on the unity virtual simulation environment, and the sensor data stream (lidar, visual camera) and physical engine parameters are bound through the ROS-TCP communication protocol; Start the obstacle avoidance algorithm node within the ROS framework, load the pre trained PPO/TD3 reinforcement learning model and supervised learning strategy, and run MWORKS Sysplorer carries out ship dynamics calculation; The motion control module drives the digital sample ship model according to the real-time decision-making instructions (path planning, obstacle avoidance angular speed), and feeds back the ship position and attitude, environment interaction status to the simulation interface through ROS Topic; Finally, a real-time closed loop of "perceptual data acquisition - algorithm online reasoning - control instruction execution - physical state return" is formed. During operation, the system supports dynamic adjustment of environmental parameters (wind wave current intensity, obstacle distribution), and continuously optimizes the strategy network through PyTorch's online learning mechanism to ensure the autonomous navigation robustness and task adaptability of unmanned craft in complex scenes.

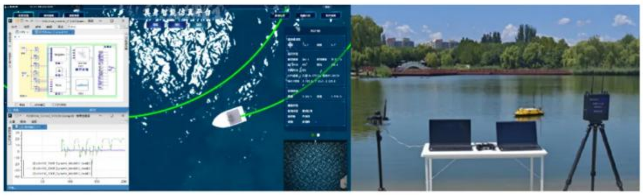

Figure 6. Digital prototype and physical prototype

DOI 10.3384/ecp218591

4 Conclusion

This study proposes an innovative framework of embodied information physical system (ECPS) based on Modelica/MWORKS, and constructs a "physical information embodied" ternary collaborative architecture through the extension of modeling language, which breaks through the separation limitations of physical laws and intelligent decision-making in traditional tools. The framework innovatively integrates the physical information neural network and multi domain modeling theory, and realizes the deep coupling of physical simulation and autonomous decision-making in the field of intelligent unmanned system. By constructing the reinforcement learning paradigm enhanced by PINN, the fluid dynamics equation constraint is embedded in the strategy network training, which significantly improves the ability of the agent to follow the physical laws; Relying on the FMI/SSP standardized protocol, a virtual and real collaborative verification system is built to effectively solve the timing conflict between high-precision simulation and real-time decision-making. The research results provide an integrated methodology of interpretable modeling, verifiable decision-making and scalable deployment for the upgrading of industrial intelligence, promote the leap of embodied intelligence from theoretical verification to engineering implementation, and establish a new benchmark for the development of independent systems in the era of digital transformation.

References

[1]Lin J, Yu W, Zhang N. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications[J]. IEEE Internet of Things Journal, 2017, 4(5): 1125-1142.

[2]Gratch J. The promise and peril of interactive embodied agents for studying non-verbal communication: a machine learning perspective[J]. Philosophical Transactions of the Royal Society of London, Series B. Biological Sciences, 2023, 378(1875).

[3]Liu Y, Chen W, Bai Y, et al. Aligning Cyber Space with Physical World: A Comprehensive Survey on Embodied AI[M]. arXiv, 2024.

[4]Rong J, Luan N. Safe Reinforcement Learning with Policy-Guided Planning for Autonomous Driving[J]. IEEE, 2020.

[5] China Automotive Engineering Research Institute Co., Ltd. (Guangzhou), China Automotive Data Co., Ltd. (Tianjin), & China Automotive Data Co., Ltd. (2024). An intelligent prediction method for Modelica fluid models (CN Patent No. CN202311575134.1). China National Intellectual Property Administration. Issued April 9, 2024.

[6]Penwarden M, Zhe S, Narayan A, et al. A Metalearning Approach for Physics-Informed Neural Networks (PINNs): Application to Parameterized PDEs[M]. (2023-01-19).

[7]Sivashangaran S, Zheng M. Intelligent Autonomous Navigation of Car-Like Unmanned Ground Vehicle via Deep Reinforcement Learning[J]. IFAC-PapersOnLine, 2021, 54(20): 218-225.

[8]Marian M, Stinga F, Georgescu M T, et al. A ROS-based Control Application for a Robotic Platform Using the Gazebo 3D Simulator[C]//2020 21th International Carpathian Control Conference (ICCC). High Tatras, Slovakia: IEEE, 2020: 1-5.

[9]Chen Y, Koohy S. GPT-PINN: Generative Pre-Trained Physics-Informed Neural Networks toward non-intrusive Meta-learning of parametric PDEs[M]. arXiv, 2023.

[10]Shi H, Shi L, Xu M, et al. End-to-End Navigation Strategy With Deep Reinforcement Learning for Mobile Robots[J]. IEEE Transactions on Industrial Informatics, 2020, 16(4): 2393-2402.