Hybrid Residual Learned Modeling for Nonlinear Unmanned Surface Vehicle (USV) Hydrodynamics

Aaron E. Wu
Georgia Institute of Technology
Atlanta, GA, USA
aewu@gatech.edu

Owen R. Anderson
Georgia Institute of Technology
Atlanta, GA, USA
oanderson36@gatech.edu

Teghpreet Singh Mago
Georgia Institute of Technology
Atlanta, GA, USA
tmago3@gatech.edu

Rachit Gupta
Georgia Institute of Technology
Atlanta, GA, USA
rgupta432@gatech.edu

Abstract

Modeling Unmanned Surface Vehicle (USV) dynamics is difficult due to nonlinear hydrodynamics and unmodeled disturbances. We propose a hybrid residual learning framework that augments Fossen's 3-DOF model with a neural network trained on Blue Robotics BlueBoat data in Gazebo. Sequence models (LSTMs, TCNs) capture temporal effects like wake buildup, while physics-informed regularization enforces energy and damping consistency. This approach enhances robustness and generalization while maintaining physical interpretability for reliable autonomous navigation.

1 Introduction

Modeling the hydrodynamic behavior of Unmanned Surface Vehicles (USVs) remains a key challenge in marine robotics due to complex, nonlinear fluid interactions and environmental disturbances such as currents and waves. Traditional physics-based models, such as Fossen's three-degree-of-freedom (3-DOF) maneuvering equations [1], provide interpretable dynamic equations for surge, sway, and yaw motion. However, these models assume linear damping and decoupled dynamics, which fail to capture nonlinear drag, thruster asymmetries, and unsteady flow coupling encountered in real-world operations. This leads to persistent prediction errors, reduced control stability, and inefficient navigation. Improving the accuracy of USV dynamic models is critical for enabling robust path-following, disturbance rejection, and adaptive control in autonomous surface operations, particularly for small platforms such as the Blue Robotics BlueBoat [2].

Problem Definition and Inputs/Outputs: The goal of this work is to develop a *hybrid residual learning model* that augments a known physics-based hydrodynamic model with a data-driven correction term. The system takes as input the instantaneous body-frame velocities and thruster commands:

$$s_t = [u, v, r, T_{\text{port}}, T_{\text{stbd}}],$$

where u, v, and r represent surge, sway, and yaw velocities, and $T_{\text{port}}, T_{\text{stbd}}$ are port and starboard thruster forces. The output is a set of predicted residual accelerations:

$$\dot{s}_{\text{res}} = [\dot{u}_{\text{res}}, \dot{v}_{\text{res}}, \dot{r}_{\text{res}}],$$

which are added to the baseline Fossen model to form the hybrid prediction:

$$a_{\text{hybrid}}(t) = a_{\text{phys}}(t) + f_\theta(s_t).$$

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Training Conditions and Data: Training is performed primarily on simulated data generated from a Fossen-based dynamic simulator augmented with synthetic noise and disturbance terms to emulate real environmental effects. The simulator publishes data to `ROS` topics analogous to real sensor streams:

- `/imu/data` - linear accelerations and angular velocities.
- `/odom` - pose and velocity for computing $\dot{u}, \dot{v}, \dot{r}$.
- `/thrusters/cmd_thrust` - left/right thrust inputs.

2 Related Works

Modeling marine vehicle dynamics through hybrid physics-ML approaches has gained attention in recent years. Skulstad et al. [3] proposed a cooperative model combining a first-principles ship dynamics model with a feedforward neural network that learns residual acceleration errors from unmodeled forces such as waves and currents. Validated on RV Gunnerus drift experiments, their approach outperformed purely analytical methods, demonstrating the value of combining physical interpretability with data-driven flexibility. However, their framework is stateless, limiting its ability to capture temporal dependencies across dynamic sequences.

Miao et al. [4] introduced DyTR, a transformer-based residual learning framework for vehicle dynamics correction that refines state predictions via high-dimensional residual queries, reducing prediction error by up to 92% over baseline physical models. While effective for terrestrial vehicles, it lacks physical constraints and temporal regularization, raising stability concerns. Similarly, Lim and Oh [5] developed a hybrid neural maritime CO_2 emission model combining kinetic equations with deep networks, underscoring the predictive benefits of hybrid architectures.

Wang et al. [6] proposed a hybrid physics-ML model augmenting Fossen's 3-DOF maneuvering equations with a feedforward residual network for USV trajectory prediction. Incorporating trigonometric features like $\cos(\psi)$ and $\sin(\psi)$ enabled the model to capture periodic hydrodynamic effects without extra sensors. Tested on real lake data, it outperformed both physics-only and data-driven baselines in open-loop accuracy but lacked temporal memory and closed-loop validation.

We extend prior work by introducing sequence-based residual models (LSTMs and TCNs) to capture history-dependent effects such as wake buildup and nonlinear drag. Physics-informed regularization enforces energy and damping consistency, and validation is conducted in Gazebo using the Blue Robotics BlueBoat [2]. All training and evaluation data, such as GPS, imagery, and odometry, are collected from the BlueBoat's onboard ROS topics to ensure high fidelity and real-world consistency.

3 Problem Decomposition and Technical Approach

3.1 Hypotheses (H1–H3)

- H1: [A physics-only model captures coarse-scale system dynamics but fails to reproduce unmodeled nonlinearities.]
- H2: [A purely data-driven model (e.g., MLP or LSTM) learns flexible mappings but generalizes poorly and violates known physical constraints.]
- H3: [A hybrid residual model combining physics and learned corrections improves predictive accuracy by compensating for structured modeling error.]
- H4: [A Physics-Informed Neural Network (PINN) enforcing physical consistency via the loss function further improves extrapolation and physical fidelity.]

3.2 Approach Overview

(1) Hybrid MLP Residual Model. The hybrid model augments a known physics-based acceleration model $a_{\text{phys}}(t)$ with a learned residual f_θ:

$$a_{\text{hybrid}}(t) = a_{\text{phys}}(t) + f_\theta(s_t), \tag{1}$$

where $s_t = [u, v, r, \tau_L, \tau_R]$ denotes the instantaneous surge, sway, yaw, and left/right thruster inputs. The residual network f_θ is implemented as a 3-layer MLP with ReLU activations, trained using mean squared error (MSE) between predicted and measured accelerations.

(2) Hybrid LSTM Residual Model. To capture temporal correlations and actuator delays, we extend the residual model with a recurrent architecture:

$$a_{\text{hybrid}}(t) = a_{\text{phys}}(t) + g_\theta(s_{t-k:t}), \tag{2}$$

where g_θ is an LSTM operating over the most recent k timesteps. This model learns temporally correlated residual dynamics that an instantaneous MLP cannot express. The loss is identical to the MLP hybrid:

$$\mathcal{L}_{\text{LSTM}} = \|a_{\text{meas}}(t) - a_{\text{hybrid}}(t)\|_2^2. \tag{3}$$

(3) Physics-Informed Neural Network (PINN). The PINN directly enforces physical consistency within the learning objective. Let

$$f_{\text{dyn}}(s, \dot{s}) = M\dot{v} + D(v) + C(v) - \tau, \tag{4}$$

denote the residual of the Fossen 3-DOF dynamic model, where M, D, and C are inertia, damping, and Coriolis terms. The total PINN loss combines data fidelity and physics residual penalties:

$$\mathcal{L}_{\text{PINN}} = \lambda_{\text{data}}\|a_{\text{pred}} - a_{\text{meas}}\|_2^2 + \lambda_{\text{phys}}\|f_{\text{dyn}}(s, \dot{s})\|_2^2. \tag{5}$$

The physics term is evaluated via automatic differentiation to compute $\dot{v}$ from the network outputs.

3.3 Implementation Details

Our implementation consists of a data preprocessing pipeline, a physics derived baseline model, and a residual neural network for learning unmodeled dynamics. We implemented everything in Python 3.11 using PyTorch for model training and SciPy and Pandas for processing.

For our data pipeline, raw experimental logs were exported from ROS bag files to long-form CSV format containing timestamped IMU and odometry data. We developed a preprocessing script (`make_dataset.py`) that pivots these logs into wide format, synchronizes IMU and odometry streams, and resamples the data to a 10Hz frequency. The script computes body-frame velocities $[u, v, r]$, quaternion-derived yaw angle ψ, filtered signals using a Butterworth low-pass filter (cutoff 1.2Hz), and the derivatives $\dot{u}, \dot{v}, \dot{r}$ from the central differences. We normalized these values and output an additional JSON file containing the normalization statistics for consistency.

Our baseline model was implemented with Fossen's 3-DOF equations as a differentiable module in PyTorch (`Fossen3DOF`) that predicts surge, sway, and yaw accelerations given current velocities and thrust inputs. The model accounts for rigid body inertia, added mass, and both linear and quadratic damping terms. This module provides physically grounded baseline accelerations for future residual learning.

The learned correction is modeled by a fully connected residual multilayer perceptron with two hidden layers of 128 units and ReLU activation. The network takes as input normalized velocities, trigonometric yaw encodings ($\cos\psi$, $\sin\psi$), and thrust commands, and outputs predicted residual accelerations $[\dot{u}_{res}, \dot{v}_{res}, \dot{r}_{res}]$. The final acceleration estimate is computed as

$$\dot{\nu}_{pred} = \dot{\nu}_{Fossen} + \dot{\nu}_{res}.$$

Training minimizes mean squared error between predicted and measured accelerations.

3.4 Baseline and Preliminary Results

Our baseline is the Fossen 3-DOF analytic model for surge, sway, and yaw dynamics, parameterized using approximate mass, added-mass, and damping terms for the Blue Robotics BlueBoat. This model provides interpretable, physically consistent estimates of body accelerations from measured velocities $[u, v, r]$ and thrust commands $[T_{\text{port}}, T_{\text{stbd}}]$. We chose it as the foundation because it captures first-order hydrodynamic effects and serves as a reliable benchmark for assessing the contribution of learned residuals.

3.5 Evaluation Procedure and Metrics

The proposed residual model is trained to minimize the discrepancy between the measured accelerations (derived from synchronized IMU and odometry data) and the combined hybrid predictions:

$$\dot{\nu}_{\text{hybrid}} = \dot{\nu}_{\text{phys}} + f_\theta(\nu, \tau)$$

We evaluate both the baseline physics model and the learned hybrid model using the root-mean-square error (RMSE) against measured accelerations:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \|\dot{\nu}_{\text{pred},i} - \dot{\nu}_{\text{meas},i}\|^2} \tag{6}$$

where $\dot{\nu}_{\text{pred},i}$ is either the physics-based or hybrid prediction and $\dot{\nu}_{\text{meas},i}$ is the reference acceleration computed from filtered IMU and odometry data. The coefficient of determination (R^2) is also reported to quantify linear agreement between predicted and measured accelerations.

Dataset splits follow a chronological 70/15/15 training, validation, and test partition to preserve temporal dependencies and prevent data leakage across sequences.

4 Intermediate / Preliminary Results

The dataset comprises synchronized sensor logs from a simulated BlueBoat in Gazebo, containing thruster commands, body-frame velocities (u, v, r), and ground-truth accelerations. These data were used to train and validate the physics-based and hybrid residual models. Future work will extend this dataset with real-world trials to capture richer hydrodynamic effects and sensor noise.

Figure 1 compares measured accelerations with physics-based, residual, and hybrid predictions. The physics-only model underfits nonlinear and coupled effects, while the residual network learns accurate corrective accelerations, yielding a hybrid output closely matching measurements.

Figure 2 shows the residual MLP's training loss over 100 epochs, exhibiting smooth convergence toward near-zero error and confirming that the hybrid approach effectively models unmodeled dynamics.

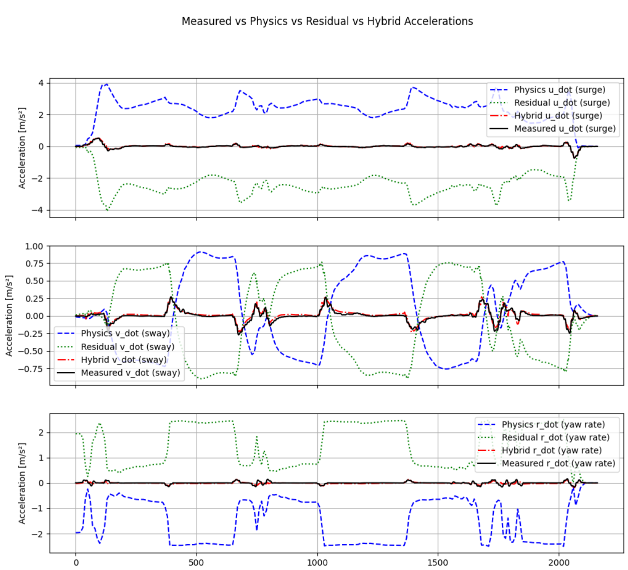

Figure 1: Comparison of measured, physics-based, residual, and hybrid accelerations for surge ($\dot{u}$), sway ($\dot{v}$), and yaw rate ($\dot{r}$).

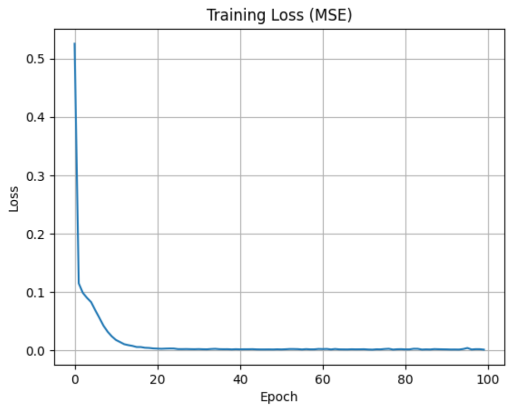

Figure 2: Training loss (mean squared error) over 100 epochs for the residual MLP.

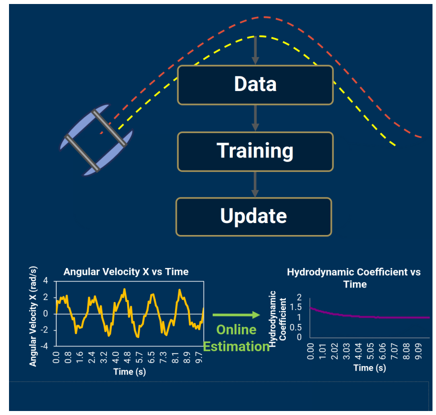

Figure 3: Draft/teaser of ideal implementation of the hybrid-residual model, where the model estimates and updates hydrodynamic coefficients while accounting for nonlinear effects.

5 Next Steps

As of November 3, 2025, the hybrid MLP residual model has been implemented and trained, achieving stable convergence (final MSE ≈ 0.0018) with initial visualization of learning curves. The remaining work focuses on extending the residual formulation to temporal and physics-informed variants.

Period	Key Tasks and Milestones
Week 1	Finalize MLP evaluation and visualizations. Implement hybrid **LSTM residual model** to capture temporal dependencies. Compare LSTM vs. MLP using MSE, MAE, and R^2.
Week 2	Develop **PINN variant** integrating physics-based constraints into the residual loss. Conduct ablation study across Physics-only, MLP, LSTM, and PINN models.
Week 3	Integrate hybrid models into Gazebo for simulation validation. Prepare final report and figures. Submit **Final Report by Nov 30, 2025**.

Table 1: Condensed project timeline for hybrid residual modeling.

5.1 Challenges and Mitigation

- *Data collection:* Real-robot deployment may be limited by time and setup complexity. To mitigate this, we will first validate models in Gazebo and use logged simulation data to supplement physical trials.
- *Physics-informed loss tuning:* Balancing data and physics terms is nontrivial; we will start with adaptive loss weighting based on gradient magnitudes.
- *Integration risks:* To minimize errors during deployment, we will standardize model I/O interfaces and develop lightweight wrappers for ROS/Gazebo.

5.2 Success Criteria

- LSTM and PINN outperform the baseline MLP in validation loss and stability metrics.
- Models demonstrate physical consistency (low residuals, stable trajectory predictions).
- Successful simulation or real-robot deployment showing plausible dynamic behavior.
- Final report includes quantitative, visual, and physical performance comparisons.

References

[1] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[2] B. Robotics, "Blueboat general integration guide," October 2023. Accessed: 2025-11-01.

[3] R. Skulstad, G. Li, T. I. Fossen, T. Wang, and H. Zhang, "A co-operative hybrid model for ship motion prediction," *Modeling, Identification and Control*, vol. 42, no. 1, pp. 17–26, 2021.

[4] J. Miao, R. Yan, B. Zhang, T. Wen, K. Jiang, M. Yang, J. Huang, Z. Zhong, and D. Yang, "Residual learning towards high-fidelity vehicle dynamics modeling with transformer," *arXiv preprint arXiv:2502.11800*, 2025. Accessed: 2025-11-01.

[5] S. Lim and J. Oh, "Hybrid neural network-based maritime carbon dioxide emission prediction: Incorporating dynamics for enhanced accuracy," *Applied Sciences*, vol. 15, no. 9, p. 4654, 2025. Accessed: 2025-11-01.

[6] Z. Wang, J. Cheng, L. Xu, L. Hao, and Y. Peng, "Hybrid physics-ml modeling for marine vehicle maneuvering motions in the presence of environmental disturbances," *arXiv preprint arXiv:2411.13908v1*, 2024. Accessed: 2025-11-01.