Journal of *Marine Science and Engineering*

MDPI

Article

Event-Domain Trajectory Prediction, Adaptation, and Replanning for USV Formations Under Intermittent Communication

Yuchen Yang [1], Zhenyu Yuan [1], Yiding Huang [1], Jun Wang [1], Zhenxia Ye [1] and Guoyuan Tang [1,2,*]

[1] School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; m202472686@hust.edu.cn (Y.Y.)
[2] Hubei Province Key Laboratory for Unmanned Underwater Vehicle and Manipulating Technology, Huazhong University of Science and Technology, Wuhan 430074, China
* Correspondence: tgyuan@hust.edu.cn

Abstract

Event-triggered communication reduces unnecessary transmissions in unmanned surface vehicle (USV) formations, but leaves two connected trajectory-level problems. Between leader state packets, the follower side must maintain a leader trajectory prediction; after a new packet arrives, the current and updated reference trajectories may not connect. We propose an event-domain prediction–adaptation–replanning framework to address both problems. Under ideal-link conditions, the leader- and follower-side synchronized event-domain state–residual physics-informed neural network (PINN) trajectory predictors generate identical finite-horizon leader trajectory predictions from identical event histories and online correction states. Event-synchronized residual adaptation updates a low-dimensional online correction state at each event to reduce local prediction mismatch. At each valid reference update, reference trajectory replanning uses a quintic connection to join the current reference trajectory to the updated reference trajectory. In multiseed Circle and S-bend missions, the frozen state–residual PINN predictor reduced short-horizon prediction error relative to nominal dynamics rollout, and the residual-adapted predictor reduced it further. Relative to Direct replacement, quintic connection reduced the mean normalized total control variation by 69.70% across 100 missions.

Keywords: unmanned surface vehicle; leader–follower formation; event-triggered communication; physics-informed neural network; trajectory prediction; reference trajectory replanning; quadratic programming

1. Introduction

Unmanned surface vehicles (USVs) and autonomous surface vehicles (ASVs) are used in marine inspection, environmental monitoring, hydrographic surveying, target search, and maritime security [1]. These vehicles can reduce personnel risk and improve operational efficiency in hazardous surface missions [2]. Compared with a single vessel, a USV formation can cover a larger area and perform tasks more efficiently [3]. Cooperative operation and collision avoidance, however, require USVs to maintain formation geometry and inter-vessel communication during navigation [4]. Under harsh sea conditions, sensor noise and communication constraints can disrupt this information exchange. Communication and control of USV formations under such constraints therefore remain open challenges [5].

The leader–follower structure is widely used to coordinate USV formations. The leader provides the global motion trend, while each follower constructs a local reference trajectory

Check for updates

Academic Editor: Decheng Wan

Received: 20 September 2026
Revised: 22 October 2026
Accepted: 23 October 2026
Published: 26 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

from leader information and maintains the prescribed formation geometry [6]. Under ideal continuous communication, the followers can continually obtain the latest leader state and promptly update their desired states to maintain the formation geometry [7]. Marine communication links, however, are constrained by bandwidth, energy consumption, delay, reliability, and hardware resources, so leader information often cannot be transmitted continuously and completely [8]. Event-triggered communication offers a way to address these constraints by transmitting a new leader state packet only when a prescribed update condition is satisfied, rather than at every sampling instant. This mechanism has been used in multi-agent consensus control to reduce unnecessary information exchange [9]. For formation control, Cheng et al. [10] adopted an edge-based event-triggering mechanism, while Dai et al. [11] applied model-assisted event-triggered communication to robotic-fish formations. Zhou et al. [12] combined a distributed observer with event-triggered affine formation control and established a positive minimum inter-event interval. Liu et al. [13] derived computable lower and upper bounds on inter-event intervals for position-based USV formation control, while Huang et al. [14] used interval-extended edge-based event-triggered communication to schedule information updates during multi-AUV formation maneuvers. Cai et al. [15] deployed structurally identical least-squares predictors on the leader and follower sides. These predictors estimated the leader state between communication events and resynchronized their prediction information when communication was triggered. These studies address triggering conditions, inter-event intervals, and state estimation between events. To generate follower reference trajectories for the tracking controllers, however, state-level communication and estimation alone are insufficient. Two connected trajectory-level processes are required: maintaining a leader trajectory prediction between state packets and connecting the current and updated reference trajectories after an update.

Maintaining a leader trajectory prediction between successive state packets requires future leader motion to be propagated using event information from the transmitted state packets. Observers and model-based predictors are commonly used for this purpose. A distributed observer can reconstruct a local target trajectory when the leader velocity or global information is unavailable, while dynamics propagation can combine an event-time measurement with a nominal vessel model [12,13]. Reconstructions based on an observer or nominal model remain affected by unmodeled maneuvers, environmental loads, measurement noise, and model simplification. Data-driven predictors can learn effects omitted from the nominal dynamics. Purely data-driven finite-horizon models, however, do not explicitly impose vessel kinematics and dynamics. Their predictions may therefore lack physical consistency, and prediction performance may deteriorate under operating conditions insufficiently represented in the training data [16]. Physics-informed neural networks (PINNs) incorporate known physical relations into data-driven models [17]. Xu et al. [18] used limited maneuvering data to predict planar USV dynamics, and Cheng et al. [19] applied physics-informed learning to finite-horizon motion prediction. Physics-constrained learned USV models have also been used for predictive control [20]. These studies address inter-event state reconstruction, data-driven motion prediction, and physics-informed modeling, but do not directly address the event-domain trajectory prediction task considered here. When the leader state cannot be obtained continuously, a predictor must use noisy state packets received at irregular event times to generate a finite-horizon leader trajectory prediction. This prediction must start from the latest received state, incorporate the 3-DOF dynamics, and provide the leader trajectory needed to construct follower reference trajectories. This study therefore develops an event-domain trajectory predictor that generates the required leader trajectory prediction from the event information available at each update.

An offline-trained predictor may develop local prediction mismatch when it encounters local disturbances that are insufficiently represented in the training data, causing the predicted trajectory to gradually deviate from the actual motion. To correct local prediction mismatch during operation, existing studies typically use online operating information to update either the learned model or its adaptation parameters. Ding et al. [16] identified time-varying vessel hydrodynamic characteristics from online operating data. Jiahao et al. [21] combined a nominal physical model with a neural ODE residual and used continuous measurements to update the learned dynamics model for predictive control, whereas O'Connell et al. [22] learned a transferable disturbance representation offline and adjusted only low-dimensional coefficients online under changing wind conditions. These studies follow two main strategies: updating the learned model or adjusting a small number of online parameters. Updating a complete model generally requires continuous and sufficiently informative measurement data, whereas event-triggered communication provides a new noisy leader state packet only at discrete transmission events. Such sparse event data provide limited support for continuous online updates of a complex model. An online update suitable for sparse event data is therefore needed to correct local prediction mismatch without retraining the complete predictor and keep the leader trajectory prediction usable over the next inter-event interval.

After a new leader state packet arrives, the updated leader trajectory prediction must be converted into an updated leader-side reference trajectory from which follower reference trajectories can be generated and supplied to the tracking controllers. At this point, each follower is still tracking the current reference trajectory generated from the preceding event prediction through the leader–follower reference mapping. Directly replacing the current reference trajectory with the updated reference trajectory can abruptly change the reference position and heading and, in turn, produce transients in the follower control inputs. A short transition is therefore needed to connect the current and updated reference trajectories without a hard switch in the follower reference trajectory. C. Liu et al. [23] developed event-triggered nonlinear MPC for an underactuated ship subject to multiple obstacles and input–output constraints, with the triggering condition determining when the trajectory-tracking problem is solved again. Dileep et al. [24] combined a physics-informed manipulator model with event-triggered MPC, where an event condition schedules the solution of a finite-horizon trajectory-planning problem. These two studies optimize trajectory tracking under a prescribed reference and trajectory planning toward a specified target, respectively, rather than the short connection between the current and updated reference trajectories considered here. The remaining problem is therefore to replan the reference trajectory under explicit connection conditions so that the current reference trajectory joins the updated reference trajectory.

In each event-update cycle, the leader trajectory prediction must be maintained and corrected between successive state packets, and after each state update, the current reference trajectory must be connected smoothly to the updated leader-side reference trajectory. For an underactuated 3-DOF USV formation comprising one leader and two followers, we propose an event-domain prediction–adaptation–replanning framework. Its event-domain state–residual PINN trajectory predictor, event-synchronized residual adaptation, and reference trajectory replanning via quintic connection perform inter-event prediction, local prediction correction, and reference trajectory connection, respectively. The leader- and follower-side synchronized predictors have the same structure and update rules and generate identical predictions under the ideal-link conditions in Assumption 1. The leader-side predictor evaluates the triggering condition, while each follower-side predictor uses its locally received event history and online correction state for leader trajectory prediction. The leader–follower reference mapping then converts the processed leader-side reference trajectory into the two follower reference trajectories. Appendix E derives a conditional

local bound on formation position error near straight-line and constant-turn equilibria, together with a bound on the online correction state.

The main methodological contributions are summarized as follows:

1. **An event-domain state–residual PINN trajectory predictor generates a finite-horizon six-state leader trajectory prediction from noisy leader state packets sent at triggering events.** The auxiliary state correction network (PINN-X) constructs an auxiliary state trajectory. The generalized residual estimation network (PINN-D) uses this trajectory to estimate the generalized force and moment omitted by the nominal model. The corrected dynamics rollout generates the final prediction.
2. **Event-synchronized residual adaptation updates a low-dimensional online correction state at the current event using a force-balance correction estimate and endpoint error feedback while keeping the PINN parameters fixed.** The updated state is used to correct local prediction mismatch over the subsequent inter-event interval.
3. **Reference trajectory replanning integrates quintic connection into the framework for event-triggered reference updates.** It connects the current and updated reference trajectories on a common time axis for both seam directions. The connection smooths reference updates without imposing vessel or actuator dynamic constraints.

Simulation studies evaluate the framework under measurement noise, plant parameter variation, and spatially varying disturbances. Two representative disturbance-region traversals illustrate how its stages operate in closed loop. A multiseed evaluation set comprising 50 Circle missions and 50 S-bend missions evaluates short-horizon prediction performance, event-time control input changes over all reference updates, and continuous closed-loop execution.

Notation. *Bold lowercase and uppercase symbols denote vectors and matrices, respectively; calligraphic symbols denote sets, histories, mappings, or loss functions according to context. The subscript k indexes communication events, with event time t_k, inter-event interval $h_k = t_{k+1} - t_k$, and local prediction time $s = t - t_k$. The symbols $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote the spaces of real n-dimensional vectors and $n \times m$ matrices, respectively; $\boldsymbol{I}_n$, $\mathrm{diag}(\cdot)$, and $\odot$ denote the n-dimensional identity matrix, diagonal-matrix construction, and elementwise multiplication, respectively. The notation $\|\cdot\| = \|\cdot\|_2$ denotes the Euclidean norm, whereas $\|\cdot\|_{\mathrm{N}}$ denotes the Euclidean norm after componentwise nondimensionalization. For $\boldsymbol{Q} \succeq 0$, $\|\boldsymbol{z}\|_{\boldsymbol{Q}}^2 = \boldsymbol{z}^{\mathsf{T}}\boldsymbol{Q}\boldsymbol{z}$ is a weighted quadratic form and is a squared norm when $\boldsymbol{Q} \succ 0$. The operator $\mathrm{wrap}_\pi(\cdot)$ maps a scalar angular difference to $[-\pi, \pi)$; $\mathrm{wrap}_\eta(\cdot)$ wraps only the heading component of a pose difference. The six-state difference operator $\ominus$ likewise wraps the heading component. A hat denotes a predicted or estimated quantity, a tilde denotes an auxiliary state, and superscript d denotes a desired or reference quantity. In reference trajectory replanning, superscripts $(\cdot)^-$ and $(\cdot)^+$ denote the current and updated reference trajectories at the same event, respectively. The subscript F in $\widehat{\boldsymbol{x}}_{F,k}$ denotes the final prediction, not a follower F_i; other context-specific superscripts and subscripts are defined where introduced.*

2. Preliminaries and Problem Formulation

2.1. Underactuated 3-DOF USV Model

Consider the set of USVs $\mathcal{V} = \{L, F_1, F_2\}$, where L denotes the leader and F_1 and F_2 denote the two followers. For each $i \in \mathcal{V}$, its earth-fixed pose, body-fixed velocity, planar position, and full state are defined as

$$\boldsymbol{\eta}_i = \begin{bmatrix} x_i & y_i & \psi_i \end{bmatrix}^{\mathsf{T}}, \qquad \boldsymbol{\nu}_i = \begin{bmatrix} u_i & v_i & r_i \end{bmatrix}^{\mathsf{T}}, \tag{1}$$

$$\boldsymbol{p}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^{\mathsf{T}}, \qquad \boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{\eta}_i^{\mathsf{T}} & \boldsymbol{\nu}_i^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \tag{2}$$

The horizontal-plane motion of each USV is described by the standard 3-DOF marine craft model [25]:

$$\dot{\boldsymbol{\eta}}_i = \boldsymbol{R}(\psi_i)\boldsymbol{\nu}_i, \tag{3}$$

$$\boldsymbol{M}_i\dot{\boldsymbol{\nu}}_i + \boldsymbol{C}_i(\boldsymbol{\nu}_i)\boldsymbol{\nu}_i + \boldsymbol{D}_i(\boldsymbol{\nu}_i)\boldsymbol{\nu}_i = \boldsymbol{\tau}_i + \boldsymbol{d}_i, \tag{4}$$

where

$$\boldsymbol{R}(\psi_i) = \begin{bmatrix} \cos\psi_i & -\sin\psi_i & 0 \\ \sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

The inertia, Coriolis/centripetal, and damping matrices are

$$\boldsymbol{M}_i = \operatorname{diag}(m_{11,i}, m_{22,i}, m_{66,i}), \tag{6}$$

$$\boldsymbol{C}_i(\boldsymbol{\nu}_i) = \begin{bmatrix} 0 & 0 & -m_{22,i}v_i \\ 0 & 0 & m_{11,i}u_i \\ m_{22,i}v_i & -m_{11,i}u_i & 0 \end{bmatrix}, \tag{7}$$

$$\boldsymbol{D}_i(\boldsymbol{\nu}_i) = \operatorname{diag}\Big(d_{u,i} + d_{u|u|,i}|u_i|, d_{v,i} + d_{v|v|,i}|v_i|, d_{r,i} + d_{r|r|,i}|r_i|\Big). \tag{8}$$

The matrix $\boldsymbol{M}_i$ contains the vessel mass, yaw moment of inertia, and corresponding added-mass terms. The term $\boldsymbol{C}_i(\boldsymbol{\nu}_i)\boldsymbol{\nu}_i$ describes the dynamic coupling among surge, sway, and yaw. The term $\boldsymbol{D}_i(\boldsymbol{\nu}_i)\boldsymbol{\nu}_i$ represents the linear and quadratic hydrodynamic resistance. The vector $\boldsymbol{d}_i = [d_{1,i}, d_{2,i}, d_{3,i}]^{\mathsf{T}}$ represents the generalized force and moment omitted by the nominal 3-DOF model. Because each USV has no independent sway actuator, only the surge force and yaw moment can be controlled directly:

$$\boldsymbol{\tau}_i = \begin{bmatrix} \tau_{u,i} & 0 & \tau_{r,i} \end{bmatrix}^{\mathsf{T}}. \tag{9}$$

Remark 1. *The three components of $\boldsymbol{d}_i$ correspond to the surge, sway, and yaw directions, respectively. Each component may include contributions from plant parameter variation, unmodeled hydrodynamics, and wind, wave, and current loads. The prediction and adaptation modules estimate only the total force or moment omitted by the nominal model in each of the three motion directions.*

2.2. Event-Triggered Communication and Synchronized Predictors

In leader–follower communication, periodic communication transmits leader state packets at fixed time intervals, with transmission times independent of the prediction error. In contrast, event-triggered communication checks a prescribed update condition on the leader side and transmits a new leader state packet only when the condition is satisfied. This mechanism can substantially reduce communication use. The specific triggering condition and inter-event interval constraints adopted in this study are given in Section 3.4.

To provide the follower side with an estimate of the leader state between consecutive communications, Cai et al. [15] deployed predictors with the same structure but different roles on the leader and follower sides. The leader-side predictor determines whether to transmit a new leader state packet, whereas the follower-side predictor continues estimating the leader state when no new packet is received. Following that deployment principle, this study uses leader- and follower-side synchronized predictors with identical structures and update rules. Sections 3.2 and 3.3 present the predictor structure and online update method, respectively. The sequence of event times and the adjacent inter-event intervals are defined as

$$0 = t_0 < t_1 < \cdots < t_k < \cdots \qquad h_k = t_{k+1} - t_k, \tag{10}$$

Here, t_k is the event time associated with the kth packet transmission, and h_k is the interval between two adjacent events. At t_k, the true six-dimensional leader state is

$$\boldsymbol{x}_L(t_k) = [x_L(t_k), y_L(t_k), \psi_L(t_k), u_L(t_k), v_L(t_k), r_L(t_k)]^{\mathsf{T}} \in \mathbb{R}^6. \tag{11}$$

The leader state packet contains the noisy state measurement $\boldsymbol{y}_k$. The measurement and its noise are defined as

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{x}_L(t_k) + \boldsymbol{n}_k, \\ \boldsymbol{n}_k &= [n_{x,k}, n_{y,k}, n_{\psi,k}, n_{u,k}, n_{v,k}, n_{r,k}]^{\mathsf{T}} \in \mathbb{R}^6. \end{aligned} \tag{12}$$

The six components of $\boldsymbol{n}_k$ are the measurement errors in positions x_L and y_L, heading ψ_L, surge velocity u_L, sway velocity v_L, and yaw rate r_L. Their statistics satisfy

$$\mathbb{E}[\boldsymbol{n}_k] = \boldsymbol{0}, \qquad \operatorname{cov}(\boldsymbol{n}_k) = \mathbb{E}[\boldsymbol{n}_k \boldsymbol{n}_k^{\mathsf{T}}] = \boldsymbol{R}_n. \tag{13}$$

The covariance matrix $\boldsymbol{R}_n \in \mathbb{R}^{6\times 6}$ specifies the noise variance in each measurement channel and the correlations between channels. Prediction and correction use the noisy measurement $\boldsymbol{y}_k$ directly.

The leader state packet sent at event time t_k is denoted by $(\boldsymbol{y}_k, t_k)$ and contains only the noisy state measurement and its timestamp. Under the ideal-link conditions in Assumption 1, both sides update their respective event histories and refresh their predictions at t_k. During $[t_k, t_{k+1})$, they propagate the leader trajectory prediction locally using the same event history and deterministic update rules. This prediction mode, refreshed at communication events and executed locally between consecutive events, is termed event-domain prediction. During operation, the inter-vessel link transmits only leader state packets, not predicted trajectories.

Assumption 1 (ideal-link reference case)**.** *Both sides use common initialization and synchronized clocks, and every leader state packet reaches each follower without delay or loss. After each transmission, the predictors use the same timestamp and event history and execute identical deterministic prediction and update rules. Section 5.4 relaxes the reception conditions to evaluate delay, packet loss, and receiver synchronization.*

2.3. Leader–Follower Formation Structure

This study adopts the leader–follower formation shown in Figure 1. Here, L denotes the leader, F_1 and F_2 denote the two followers, and O–$X_E Y_E$ is the earth-fixed frame. Let $\boldsymbol{\eta}_L^d(s) = [x_L^d(s), y_L^d(s), \psi_L^d(s)]^{\mathsf{T}}$ denote the leader-side reference pose at local prediction time s. For follower F_i, ℓ_i denotes the prescribed distance from its reference position to the leader-side reference position, and δ_i denotes the prescribed bearing of this relative position with respect to the reference heading $\psi_L^d(s)$. The pair (ℓ_i, δ_i) remains fixed throughout a mission. The desired follower pose $\boldsymbol{\eta}_{F_i}^d(s) = [x_{F_i}^d(s), y_{F_i}^d(s), \psi_{F_i}^d(s)]^{\mathsf{T}}$ is given by

$$\begin{cases} x_{F_i}^d(s) = x_L^d(s) - \ell_i \cos\Big(\psi_L^d(s) + \delta_i\Big), \\ y_{F_i}^d(s) = y_L^d(s) - \ell_i \sin\Big(\psi_L^d(s) + \delta_i\Big), \\ \psi_{F_i}^d(s) = \psi_L^d(s), \end{cases} \qquad i \in \{1, 2\}. \tag{14}$$

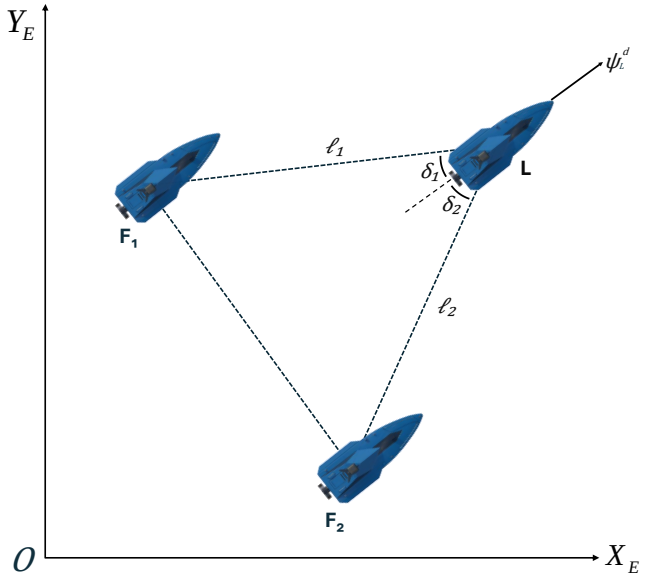


Figure 1. Leader–follower formation geometry.

2.4. Problem Formulation

Section 2.2 defines event-triggered leader state packet transmission and the leader- and follower-side synchronized predictors. Section 2.3 provides the leader–follower reference mapping from the leader-side reference to the reference trajectories of the two followers. Because leader state packets are transmitted only at event times, the follower side must continuously generate usable reference trajectories between successive packets and update them when a new packet arrives. This requires maintaining a continuous leader trajectory prediction and handling the reference switch caused by prediction updates.

The upper part of Figure 2 shows the actual leader motion and the inter-event leader trajectory prediction. The current prediction starts at event time t_k, and t_{k+1} is the next event time. The black solid curve represents the actual leader trajectory, while the blue dashed curve represents the leader trajectory prediction generated from the leader state packet at t_k. Both trajectories start from the same measured state at t_k. As the leader continues to move, effects omitted by the nominal model may cause the actual trajectory to deviate gradually from the prediction. Their separation before t_{k+1} reflects the prediction error accumulated over $[t_k, t_{k+1})$.

The lower part of Figure 2 shows how an event-triggered update of the leader trajectory prediction is propagated to the follower reference trajectories. The blue solid curve represents the follower reference trajectory generated from the leader trajectory prediction at t_k and used throughout the current inter-event interval. The two blue open circles mark the actual follower positions at t_k and t_{k+1}. The new leader state packet received at t_{k+1} is used to update the leader trajectory prediction and generate the leader-side reference trajectory after the event-triggered update. Equation (14) then maps this leader-side reference trajectory to the two follower reference trajectories shown by the red dashed curve.

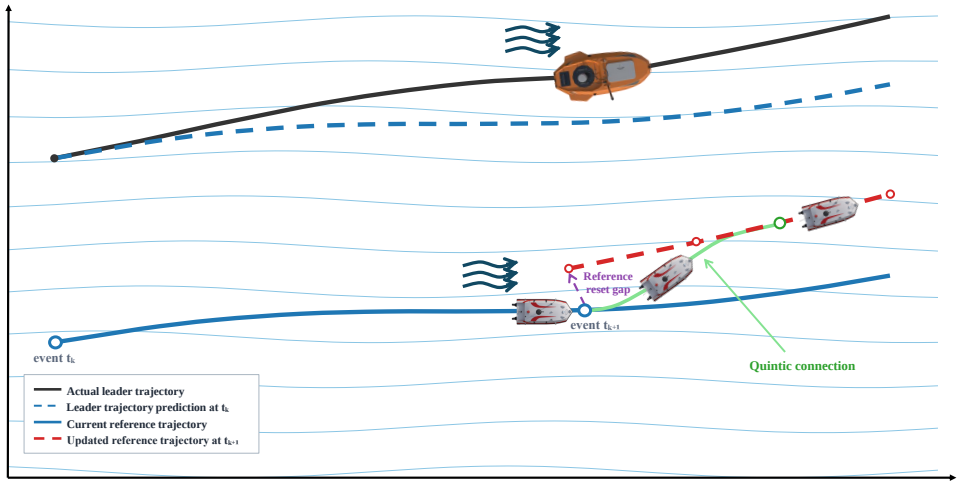


Figure 2. Event-triggered update of the leader trajectory prediction and the resulting reference reset.

At any event time t_k, let $\boldsymbol{p}_{L,k}^{d,-}(s)$ denote the current leader-side reference trajectory and $\boldsymbol{p}_{L,k}^{d,+}(s)$ the updated leader-side reference trajectory generated from the new leader state packet. Under Direct replacement, the first sample of the updated reference trajectory at $s = 0$ immediately replaces the current reference sample, producing the planar leader-side reference reset

$$\Delta \boldsymbol{p}_{L,k}^{d} = \boldsymbol{p}_{L,k}^{d,+}(0) - \boldsymbol{p}_{L,k}^{d,-}(0). \tag{15}$$

In Figure 2, the follower reference trajectory is updated at event time t_{k+1}. In this schematic, the blue open circle at t_{k+1} marks the actual follower position and coincides with the last sample of the current follower reference trajectory. The first open circle on the red dashed curve is the first sample of the updated reference trajectory. The purple dashed segment between them is the follower-side reference reset obtained by mapping the leader-side reference change in Equation (15) through Equation (14). Under Direct replacement, the controller reference jumps directly from the blue sample to the red sample. The resulting simultaneous changes in reference position and heading may cause abrupt changes in the commanded surge force and yaw moment. The green curve represents the follower reference trajectory obtained by applying Equation (14) to the quintic-connected leader-side reference trajectory. The connection combines the current and updated leader-side reference trajectories on a common time axis over a prescribed duration. Section 3.5 presents this construction, and Section 5.3 evaluates the resulting control input response.

Under the USV model, event-triggered communication, and formation mapping defined above, this study addresses the following three problems:

1. Generating a finite-horizon leader trajectory prediction from the historical leader state packets available at event time t_k, allowing the follower side to generate continuous reference trajectories until the next packet arrives.
2. Correcting the leader trajectory prediction over the next inter-event interval using newly obtained state information when the preceding prediction exhibits local prediction mismatch.
3. Connecting the current and updated reference trajectories over a finite duration at each valid reference update to reduce abrupt changes in the follower control inputs.

3. Proposed Methodology

Section 3 presents the methods for the problems formulated in Section 2.4, with Figure 3 first summarizing the overall information flow.

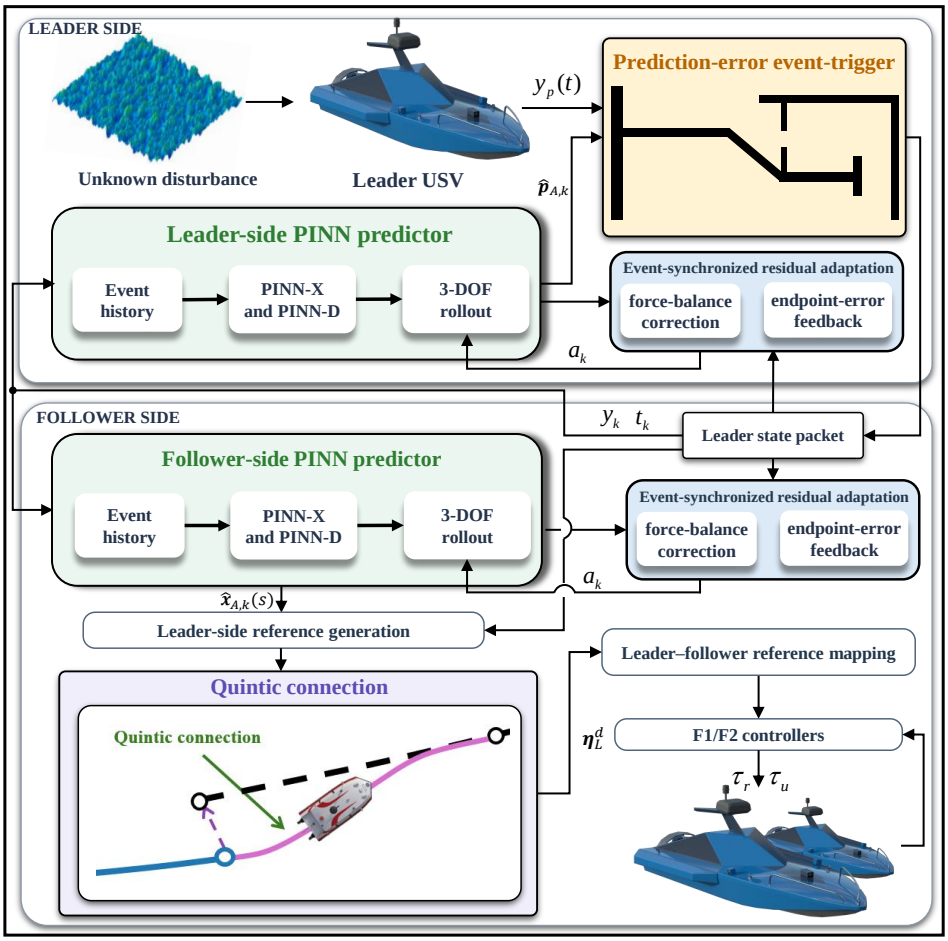


Figure 3. Overall architecture of the proposed method.

3.1. Overall Framework

Figure 3 shows the overall architecture of the proposed method. Its upper half contains the leader-side synchronized predictor, event-synchronized residual adaptation, and prediction-error trigger. The lower half contains the follower-side synchronized predictor, leader–follower reference mapping, and follower controllers. On the leader side, PINN-X constructs an auxiliary state trajectory from the event history. PINN-D uses this trajectory to estimate the force and moment correction, which enters the dynamics rollout to generate the leader trajectory prediction for the current inter-event interval. When a new triggering event generates a state packet, event-synchronized residual adaptation combines the force-balance correction estimate with endpoint error feedback to form a candidate online correction state. Candidates are committed only after the acceptance checks in Section 3.3. The updated state is then used to correct the leader trajectory prediction over the next inter-event interval. The prediction-error trigger compares the current leader position measurement with the predicted position at the same instant. After the minimum inter-event interval has elapsed, the leader side sends a new state packet if the position prediction error reaches the prescribed threshold. If the error remains below the threshold, a state packet is sent at the first monitoring instant reaching the maximum inter-event interval. Until either condition is satisfied, both sides continue using the current leader trajectory prediction.

Under the ideal-link conditions in Assumption 1, after a new leader state packet is sent, each side appends the same packet to its local event history and updates its online correction state using the same rule. Using their identical event histories and online correction states, the synchronized predictors independently generate identical leader trajectory predictions. The leader side uses its prediction to compute the position prediction error and determine when to send the next leader state packet. The follower side uses the same event history and online correction state to generate the new leader trajectory prediction and construct the follower reference trajectories for both followers. With delay or packet loss, each receiver instead uses its locally received packets to update its own event history and online correction state.

On the follower side, the new leader trajectory prediction is first used to generate an updated leader-side reference trajectory. At each valid reference update, a quintic connection joins the current reference trajectory to the updated reference trajectory. If another update arrives before the connection is complete, a new quintic connection starts from the reference trajectory being executed at that time. After this processing, the leader–follower reference mapping in Equation (14) converts the connected leader-side reference trajectory into the follower reference trajectories for F_1 and F_2. Their respective controllers then track these trajectories in closed loop.

3.2. Event-Domain State–Residual PINN Trajectory Prediction

Under the leader- and follower-side synchronized predictor architecture described in Section 2.2, both sides deploy the event-domain state–residual PINN trajectory predictor developed in this section. The predictor contains two sequentially connected networks with distinct roles. The auxiliary state correction network (PINN-X) constructs an auxiliary state trajectory anchored at the latest event state. The generalized residual estimation network (PINN-D) then estimates the generalized force and moment omitted by the nominal model along that trajectory. Starting from the noisy state measurement $\boldsymbol{y}_k$ in the latest leader state packet, the predictor numerically integrates the 3-DOF dynamics with the PINN-D force and moment correction over successive time steps to obtain future states over the prediction horizon. This numerical propagation is termed the dynamics rollout. Figure 4 shows PINN-X, PINN-D, and their inputs and outputs on the left. The discrete physical operators applied to the network outputs are shown in the center, and the corresponding physics-informed losses are shown on the right.

Before the formation mission begins, the leader completes path planning and sends the global formation trajectory information for the mission to the follower side. Both sides use this trajectory information as the basis for subsequent predictions. The leader- and follower-side synchronized predictors are also preconfigured with the same fixed nominal 3-DOF model, nominal leader control law, frozen network parameters, and numerical integration settings. At each event time t_k, both sides use the locally stored global formation trajectory information to construct the task descriptor $\boldsymbol{m}_k$, desired-state sequence $\boldsymbol{x}_k^d(s)$, and nominal feedforward input $\boldsymbol{\tau}_k^{\mathrm{ff}}(s)$. During the mission, the leader side transmits the leader state packet $(\boldsymbol{y}_k, t_k)$ through the inter-vessel link only at event times t_k. Under the ideal-link conditions in Assumption 1, both sides append the same leader state packet to their respective event histories. Using identical locally stored global formation trajectory information and predictor configurations, they then generate identical leader trajectory predictions.

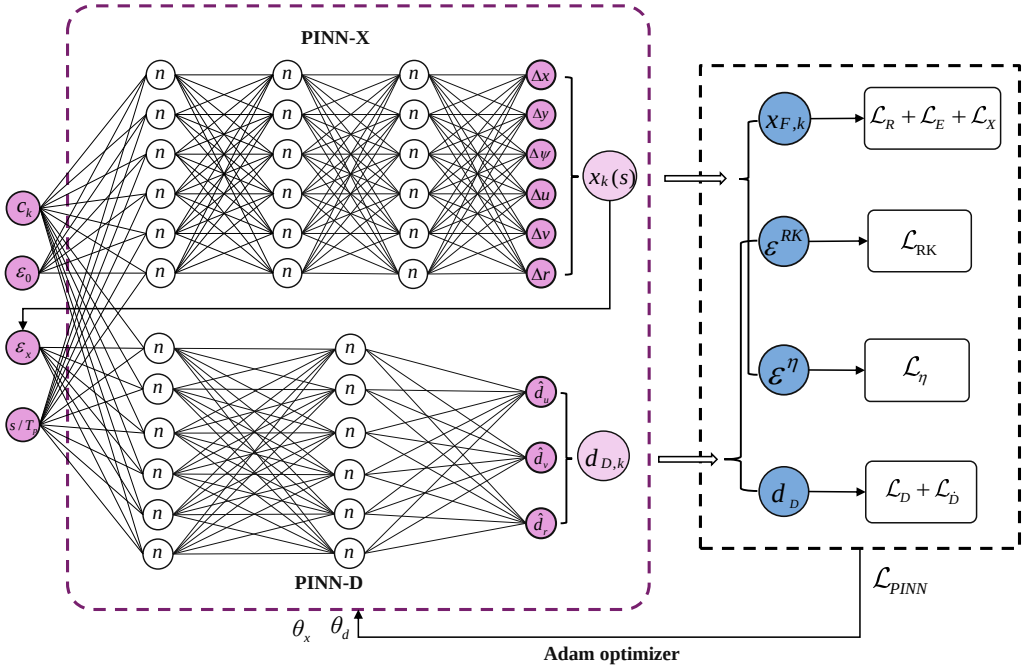


Figure 4. Architecture and offline training of the event-domain state–residual PINN trajectory predictor.

At event time t_k, let $\mathcal{H}_k$ denote the available event history and $\boldsymbol{m}_k$ the six-dimensional task descriptor constructed locally from the stored global formation trajectory information. Define the local prediction time as $s = t - t_k$. The complete prediction task in Figure 4 is

$$\widehat{\boldsymbol{x}}_{F,k}(s) = \mathcal{P}_{\boldsymbol{\theta}_x,\boldsymbol{\theta}_d}(\mathcal{H}_k, \boldsymbol{m}_k, s), \qquad 0 \le s \le T_p, \tag{16}$$

where T_p is the prediction horizon. The parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_d$ belong to PINN-X and PINN-D, respectively, and remain frozen during deployment. Each transmitted leader state packet contains only $(\boldsymbol{y}_j, t_j)$. After a packet becomes available locally, each predictor computes the nominal endpoint error in Equation (18) and stores it with that packet. This error is not transmitted through the inter-vessel link. The event history therefore consists of the latest K locally augmented packet records:

$$\mathcal{H}_k = \left\{ \left(\boldsymbol{y}_j, t_j, \boldsymbol{e}_j^{0,\mathrm{end}} \right) \right\}_{j=k-K+1}^{k}. \tag{17}$$

The six-dimensional task descriptor $\boldsymbol{m}_k$ is constructed locally from the stored global formation trajectory information as

$$\boldsymbol{m}_k = \left[\mathbb{1}_{\mathrm{C}} \quad \mathbb{1}_{\mathrm{S}} \quad \frac{R_{\mathrm{p}}}{50\,\mathrm{m}} \quad \frac{L_{\mathrm{p}}}{150\,\mathrm{m}} \quad \frac{U_{\mathrm{p}}}{1.2\,\mathrm{m\,s^{-1}}} \quad \sigma_{\mathrm{p}} \right]^{\mathsf{T}}.$$

The indicator pair $(\mathbb{1}_{\mathrm{C}}, \mathbb{1}_{\mathrm{S}})$ equals $(1,0)$ for Circle missions and $(0,1)$ for S-bend missions. The planned geometric parameter R_{p} is the Circle radius or the minimum curvature radius of the sinusoidal S-bend. The characteristic length L_{p} is the S-bend spatial wavelength and is set to zero in the descriptor for Circle missions. The quantity U_{p} is the planned leader speed. For Circle missions, $\sigma_{\mathrm{p}} = +1$ and -1 indicate positive and negative planned yaw rates, respectively; $\sigma_{\mathrm{p}} = 0$ for S-bend missions, whose turn direction varies along the path. All six entries are dimensionless and describe the shared plan rather than future measurements or unshared maneuvers.

The operator $\mathcal{P}_{\boldsymbol{\theta}_x,\boldsymbol{\theta}_d}$ in Equation (16) denotes the complete prediction process. It first constructs the event context from $\mathcal{H}_k$ and $\boldsymbol{m}_k$. PINN-X then corrects the nominal state, after which PINN-D estimates the generalized force and moment omitted by the nominal model. The resulting PINN-D force and moment correction is added to the 3-DOF dynamics to produce the predicted trajectory $\widehat{\boldsymbol{x}}_{F,k}(s)$. The remainder of this section describes these components in sequence.

For each receiver, adjacent history indices refer to consecutive locally accepted packets, and their interval is determined by measurement timestamps. The normalization scale T_p used in the event context, network inputs, and correction envelopes remains fixed at its training value. Evaluating the first 2 s does not rescale these quantities. The internal prediction cache also covers propagation between updates and construction of the reference connection.

Let $\boldsymbol{\Phi}_0$ denote the dynamics-rollout operator based on the fixed nominal matrices $\boldsymbol{M}_0$, $\boldsymbol{C}_0$, and $\boldsymbol{D}_0$. For two adjacent leader state packets, let $\boldsymbol{y}_{j-1}$ and $\boldsymbol{y}_j$ denote their respective noisy state measurements. Starting from $\boldsymbol{y}_{j-1}$, the nominal model propagates the state over the actual inter-event interval $h_{j-1} = t_j - t_{j-1}$ to obtain the predicted state at event time t_j. Equation (18) compares this predicted state with the noisy state measurement $\boldsymbol{y}_j$ in the newly received leader state packet:

$$\boldsymbol{e}_j^{0,\mathrm{end}} = \boldsymbol{y}_j \ominus \boldsymbol{\Phi}_0\Big(h_{j-1}; \boldsymbol{y}_{j-1}, t_{j-1}\Big), \qquad h_{j-1} = t_j - t_{j-1}. \tag{18}$$

Here, $\ominus$ denotes the six-dimensional state difference, with only the heading component wrapped over a 2π period. The nominal endpoint error $\boldsymbol{e}_j^{0,\mathrm{end}}$ measures the six-state prediction mismatch accumulated by the nominal model over one actual inter-event interval and provides the networks with recent model-mismatch information. Each retained packet record is encoded relative to the current event t_k as a 14-dimensional feature $\boldsymbol{q}_{j|k}$. This feature consists of the seven-dimensional local-motion information defined in Equation (21), the relative event time $(t_j - t_k)/T_p$, and the normalized and clipped six-dimensional endpoint error stored locally with the corresponding packet. Equation (19) arranges the features of the latest K packet records in chronological order and appends the task descriptor $\boldsymbol{m}_k$ to form the event context:

$$\boldsymbol{c}_k = \begin{bmatrix} \boldsymbol{q}_{k-K+1|k}^{\mathsf{T}} & \cdots & \boldsymbol{q}_{k|k}^{\mathsf{T}} & \boldsymbol{m}_k^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{14K+6}. \tag{19}$$

This study uses $K = 6$, so $\boldsymbol{c}_k \in \mathbb{R}^{90}$. On the left side of Figure 4, $\boldsymbol{c}_k$ is the common context input to PINN-X and PINN-D. It combines recent leader motion, the nominal model's prediction mismatch over adjacent event intervals, and the current task conditions. The nominal dynamics rollout also uses the desired-state sequence $\boldsymbol{x}_k^d(s)$ and the nominal feedforward input $\boldsymbol{\tau}_k^{\mathrm{ff}}(s)$. Both synchronized predictors construct these two sequences from the locally stored global formation trajectory information. The former specifies the desired motion, whereas the latter provides the corresponding baseline surge force and yaw moment.

The predictor starts from the noisy state measurement $\boldsymbol{y}_k$ in the latest leader state packet and integrates the fixed nominal 3-DOF model over the prediction horizon. This produces the nominal trajectory $\boldsymbol{x}_{0,k}(s)$:

$$\boldsymbol{x}_{0,k}(s) = \boldsymbol{\Phi}_0(s; \boldsymbol{y}_k, t_k), \qquad 0 \le s \le T_p. \tag{20}$$

Here, $\boldsymbol{\Phi}_0$ is the nominal dynamics-rollout operator, and Equation (20) gives the nominal trajectory before network correction. Before PINN-X uses this trajectory, each state along the trajectory is represented relative to the latest event state rather than by its absolute

position and heading. In this way, the same local motion has a consistent representation even when events occur at different positions and headings. Equation (21) defines this transformation. It uses the position $\boldsymbol{p}_k$ and heading ψ_k in the noisy state measurement $\boldsymbol{y}_k$ from the latest leader state packet as the local origin and local direction, respectively. Any state $\overline{\boldsymbol{x}}$ within the prediction interval is then converted into seven-dimensional local-motion information:

$$\mathcal{E}_{\mathrm{loc}}(\overline{\boldsymbol{x}};\boldsymbol{y}_k)=\begin{bmatrix}\left(\boldsymbol{S}_p^{\mathrm{loc}}\right)^{-1}\boldsymbol{R}_2^{\mathsf{T}}(\psi_k)(\overline{\boldsymbol{p}}-\boldsymbol{p}_k)\\ \sin(\overline{\psi}-\psi_k)\\ \cos(\overline{\psi}-\psi_k)\\ \left(\boldsymbol{S}_v^{\mathrm{loc}}\right)^{-1}\overline{\boldsymbol{v}}\end{bmatrix}\in\mathbb{R}^7,\qquad \overline{\boldsymbol{v}}=[\overline{u},\overline{v},\overline{r}]^{\mathsf{T}}. \tag{21}$$

The position term first computes the displacement $\overline{\boldsymbol{p}}-\boldsymbol{p}_k$ from the latest event position. The matrix $\boldsymbol{R}_2^{\mathsf{T}}(\psi_k)$ then rotates this displacement into the local frame aligned with the event heading ψ_k. The relative heading is represented by its sine and cosine to avoid the angular discontinuity at $-\pi/\pi$. The matrices $\boldsymbol{S}_p^{\mathrm{loc}}$ and $\boldsymbol{S}_v^{\mathrm{loc}}$ nondimensionalize the local relative position and body-fixed velocity, respectively, with their values reported in Section 4.2. Setting $\overline{\boldsymbol{x}}=\boldsymbol{x}_{0,k}(s)$ gives the PINN-X local-state input shown in Figure 4. Later, setting $\overline{\boldsymbol{x}}=\widetilde{\boldsymbol{x}}_k(s)$ gives the PINN-D local-state input.

PINN-X constructs an auxiliary state trajectory from the nominal trajectory. Following the time-conditioned state-learning approach of Xu et al. [18], PINN-X takes the event context $\boldsymbol{c}_k$, the local-motion representation of the nominal state $\boldsymbol{x}_{0,k}(s)$, and the normalized prediction time s/T_p as inputs. The network outputs a six-dimensional state correction:

$$\widetilde{\boldsymbol{x}}_k(s)=\boldsymbol{x}_{0,k}(s)+\boldsymbol{w}_X(s)\odot\left[\boldsymbol{\ell}_x\odot\tanh\mathcal{N}_x\left(\boldsymbol{c}_k,\mathcal{E}_{\mathrm{loc}}(\boldsymbol{x}_{0,k}(s);\boldsymbol{y}_k),s/T_p;\boldsymbol{\theta}_x\right)\right]. \tag{22}$$

In Equation (22), $\mathcal{N}_x(\cdot;\boldsymbol{\theta}_x)$ denotes PINN-X, and $\odot$ denotes elementwise multiplication. The tanh function and scale vector $\boldsymbol{\ell}_x$ bound each state-correction component. The time envelope $\boldsymbol{w}_X(s)$ controls how the correction changes after the event. The envelopes for position, heading, and body-fixed velocity are

$$w_x(s)=w_y(s)=\left(\frac{s}{T_p}\right)^2. \tag{23}$$

$$w_\psi(s)=1-\exp\left[-\left(\frac{s}{T_b}\right)^2\right]. \tag{24}$$

$$w_u(s)=w_v(s)=w_r(s)=1-\exp\left(-\frac{s}{T_b}\right). \tag{25}$$

Here, $T_b=1\,\mathrm{s}$ is the transition time of the heading and body-fixed-velocity correction envelopes. This study uses auxiliary functions to embed the event initial state directly into the PINN-X output [26]. Specifically, $\boldsymbol{w}_X(0)=\boldsymbol{0}$; together with $\boldsymbol{x}_{0,k}(0)=\boldsymbol{y}_k$, Equations (22)–(25) enforce $\widetilde{\boldsymbol{x}}_k(0)=\boldsymbol{y}_k$ by construction. The resulting event-anchored auxiliary trajectory starts from the latest event state measurement and is then supplied to PINN-D as its state input.

The auxiliary trajectory supplies the PINN-D state input at each prediction time. Physics-informed architectures can combine known dynamics with neural unknown terms [27]. Here, PINN-D does not directly correct the state derivative. It takes the event context $\boldsymbol{c}_k$, the local-motion representation of the auxiliary state $\widetilde{\boldsymbol{x}}_k(s)$, and the normalized

prediction time s/T_p as inputs. From these inputs, it estimates the bounded generalized force and moment omitted by the nominal 3-DOF model:

$$\widehat{\boldsymbol{d}}_{D,k}(s) = \boldsymbol{\ell}_d \odot \tanh \mathcal{N}_d\big(\boldsymbol{c}_k, \mathcal{E}_{\mathrm{loc}}(\widetilde{\boldsymbol{x}}_k(s); \boldsymbol{y}_k), s/T_p; \boldsymbol{\theta}_d\big). \tag{26}$$

Here, $\mathcal{N}_d(\cdot;\boldsymbol{\theta}_d)$ denotes PINN-D. The tanh function and scale vector $\boldsymbol{\ell}_d$ bound the three output components. These components are the surge force, sway force, and yaw moment and enter the 3-DOF dynamics. Position and heading continue to evolve according to $\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu}$.

The PINN-D force and moment correction is added to the nominal 3-DOF dynamics. At each RK4 evaluation, the nominal leader control law computes a feedback input based on the difference between the current rollout state and the desired state $\boldsymbol{x}_k^d(s)$. It then combines this feedback input with the nominal feedforward input $\boldsymbol{\tau}_k^{\mathrm{ff}}(s)$ to form the underactuated control input $\boldsymbol{\tau}_{\mathrm{pub}}(s) = [\tau_u(s), 0, \tau_r(s)]^{\mathsf{T}}$. The resulting state derivative is

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu}, \qquad \dot{\boldsymbol{\nu}} = \boldsymbol{M}_0^{-1}\Big[\boldsymbol{\tau}_{\mathrm{pub}}(s) - \boldsymbol{C}_0(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{D}_0(\boldsymbol{\nu})\boldsymbol{\nu} + \widehat{\boldsymbol{d}}_{D,k}(s)\Big]. \tag{27}$$

Starting from $\boldsymbol{y}_k$, Equation (28) integrates Equation (27) with a fixed-step fourth-order Runge–Kutta method to obtain the final six-dimensional leader trajectory prediction:

$$\widehat{\boldsymbol{x}}_{F,k}(s) = \boldsymbol{\Psi}_{\mathrm{RK4}}\Big(\boldsymbol{y}_k, \widehat{\boldsymbol{d}}_{D,k}(0:s), \boldsymbol{x}_k^d(0:s), \boldsymbol{\tau}_k^{\mathrm{ff}}(0:s); \Delta s_{\mathrm{dep}}\Big). \tag{28}$$

Here, $\boldsymbol{\Psi}_{\mathrm{RK4}}$ is the fixed-step RK4 propagation operator, and Δs_{dep} is the numerical-integration step used during deployment.

Remark 2. *The PINN-X output $\widetilde{\boldsymbol{x}}_k$ is an auxiliary trajectory that provides the state input to PINN-D; it is not the final prediction. PINN-D uses this auxiliary trajectory to generate its force and moment correction. The final six-dimensional leader trajectory prediction $\widehat{\boldsymbol{x}}_{F,k}$ is obtained by rolling out the 3-DOF dynamics with the PINN-D force and moment correction added.*

During training, the four classes of discrete physical operators in the center of Figure 4 compute the complete dynamics rollout, one-step RK4 state transition, kinematic consistency, and the magnitude and rate of temporal variation of the generalized residual. These quantities are used to construct the trajectory supervision term, discrete physics-consistency term, residual-magnitude regularization, and residual temporal-variation regularization shown on the right of Figure 4. This training objective follows the general PINN principle of combining data with physical constraints [17]. Because different loss terms can produce imbalanced gradient contributions [28], each physical quantity is first normalized by its corresponding scale before fixed weights are applied. The state-transition and kinematic defects are defined next.

The PINN-X auxiliary trajectory and the PINN-D force and moment correction should describe the same dynamic process. A fixed step Δs_{tr} discretizes the prediction horizon into the training grid $s_n = n\Delta s_{\mathrm{tr}}$. Over the interval $[s_n, s_{n+1}]$, the one-step propagation in Equation (29) starts from the auxiliary state $\widetilde{\boldsymbol{x}}_i(s_n)$, adds this correction to the 3-DOF dynamics, and advances the state by one RK4 step. The propagated state is then compared with the next auxiliary state $\widetilde{\boldsymbol{x}}_i(s_{n+1})$. For training event window i and grid interval n, the state-transition defect is

$$\boldsymbol{\epsilon}_{i,n}^{RK} = \widetilde{\boldsymbol{x}}_i(s_{n+1}) \ominus \boldsymbol{\Psi}_{\mathrm{RK4}}^{(1)}\Big(\widetilde{\boldsymbol{x}}_i(s_n), \widehat{\boldsymbol{d}}_{D,i}(s_n:s_{n+1}), \boldsymbol{x}_i^d(s_n:s_{n+1}), \boldsymbol{\tau}_i^{\mathrm{ff}}(s_n:s_{n+1}); \Delta s_{\mathrm{tr}}\Big). \tag{29}$$

A smaller $\boldsymbol{\epsilon}_{i,n}^{RK}$ indicates that the PINN-X state transition more closely matches the result produced by the PINN-D-corrected 3-DOF dynamics.

Equation (29) tests whether the auxiliary trajectory can be produced by the corrected dynamics. Equation (30) further tests whether the position, heading, and velocity along this trajectory are mutually consistent. Over $[s_n, s_{n+1}]$, the first term calculates the average pose rate directly from the endpoint differences in position and heading. The second term takes the body-fixed velocity at the interval midpoint and converts it to the earth-fixed pose rate using the midpoint heading. These two results should be close when the auxiliary trajectory satisfies the 3-DOF kinematic relation. Equation (30) defines their difference as the kinematic defect. The angular wrapping operator $\mathrm{wrap}_\eta(\cdot)$ handles heading changes across $\pm\pi$:

$$\boldsymbol{\epsilon}_{i,n}^{\eta} = \frac{\mathrm{wrap}_\eta[\widetilde{\boldsymbol{\eta}}_i(s_{n+1}) - \widetilde{\boldsymbol{\eta}}_i(s_n)]}{\Delta s_{\mathrm{tr}}} - \boldsymbol{R}(\widetilde{\psi}_{i,n+1/2})\widetilde{\boldsymbol{\nu}}_{i,n+1/2}. \tag{30}$$

A smaller $\boldsymbol{\epsilon}_{i,n}^{\eta}$ means that the change calculated from adjacent poses more closely matches the motion represented by the body-fixed velocity. The auxiliary trajectory therefore more closely satisfies the 3-DOF kinematic relation $\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu}$.

Equation (31) combines the three groups of loss terms into the total loss:

$$\mathcal{L}_{\mathrm{PINN}} = w_R\mathcal{L}_R + w_E\mathcal{L}_E + w_{\mathrm{aux}}\mathcal{L}_X + w_{RK}\mathcal{L}_{RK} + w_\eta\mathcal{L}_\eta + w_D\mathcal{L}_D + w_{\dot{D}}\mathcal{L}_{\dot{D}}. \tag{31}$$

Equation (32) defines the seven loss terms. The trajectory loss $\mathcal{L}_R$ and event-endpoint loss $\mathcal{L}_E$ constrain the complete dynamics rollout and the next event endpoint, respectively. The auxiliary trajectory supervision term $\mathcal{L}_X$ directly supervises the PINN-X auxiliary states using the same training trajectory. The discrete physics-consistency terms $\mathcal{L}_{RK}$ and $\mathcal{L}_\eta$ penalize the state-transition defect in Equation (29) and the kinematic defect in Equation (30), respectively. The terms $\mathcal{L}_D$ and $\mathcal{L}_{\dot{D}}$ constrain the magnitude and rate of variation of the PINN-D correction, respectively. The notation $\|\cdot\|_{\mathrm{N}}$ denotes the Euclidean norm after componentwise nondimensionalization. The index q identifies a prediction sampling point, and h_{k_i} is the interval to the next event. The ground-truth training trajectory starting at event k_i is denoted by $\boldsymbol{x}_{k_i}^{\star}$:

$$\begin{aligned}
\mathcal{L}_R &= \mathrm{mean}_{i,q}\left\|\widehat{\boldsymbol{x}}_{F,k_i}(s_q) \ominus \boldsymbol{x}_{k_i}^{\star}(s_q)\right\|_{\mathrm{N}}^2,\\
\mathcal{L}_X &= \mathrm{mean}_{i,q}\left\|\widetilde{\boldsymbol{x}}_{k_i}(s_q) \ominus \boldsymbol{x}_{k_i}^{\star}(s_q)\right\|_{\mathrm{N}}^2,\\
\mathcal{L}_E &= \mathrm{mean}_{i}\left\|\widehat{\boldsymbol{x}}_{F,k_i}(h_{k_i}) \ominus \boldsymbol{y}_{k_i+1}\right\|_{\mathrm{N}}^2,\\
\mathcal{L}_{RK} &= \mathrm{mean}_{i,n}\left\|\boldsymbol{\epsilon}_{i,n}^{RK}\right\|_{\mathrm{N}}^2, \qquad \mathcal{L}_\eta = \mathrm{mean}_{i,n}\left\|\boldsymbol{\epsilon}_{i,n}^{\eta}\right\|_{\mathrm{N}}^2,\\
\mathcal{L}_D &= \mathrm{mean}_{i,n}\left\|\widehat{\boldsymbol{d}}_{D,i}(s_n)\right\|_{\mathrm{N}}^2,\\
\mathcal{L}_{\dot{D}} &= \mathrm{mean}_{i,n}\left\|\frac{\widehat{\boldsymbol{d}}_{D,i}(s_{n+1}) - \widehat{\boldsymbol{d}}_{D,i}(s_n)}{\Delta s_{\mathrm{tr}}}\right\|_{\mathrm{N}}^2.
\end{aligned} \tag{32}$$

For the paired physics-informed loss ablation, the comparison model is trained with $w_{RK} = w_\eta = 0$. It retains the RK4-based dynamics rollout, auxiliary trajectory supervision, residual-magnitude regularization, and residual temporal-variation regularization. The comparison therefore isolates the two discrete physics-consistency terms, while the prediction structure and all other training terms remain unchanged.

The feedback loop at the bottom of Figure 4 indicates that Adam simultaneously updates PINN-X and PINN-D using the total loss in Equation (31). Training samples are event windows extracted from complete leader trajectories. The training and validation sets are partitioned by complete trajectory, and the network parameters are optimized using only the training set. Checkpoint selection uses the mean horizon-normalized trajectory-level ADE across the prespecified validation horizons. Section 4.2 gives the data split and training settings. During deployment, both networks remain frozen, and online operation updates only the low-dimensional correction state defined in Section 3.3.

3.3. Event-Synchronized Residual Adaptation

After the training described in Section 3.2, the parameters of PINN-X and PINN-D remain fixed. Low-dimensional parameter updates have been used to adapt learned dynamics under changing operating conditions [29]. Here, event-synchronized residual adaptation uses new event information to correct local prediction mismatch without adjusting the complete networks. It concentrates the online adjustment in a six-dimensional correction state. Whenever a new leader state packet becomes available locally, the adaptation updates this correction state using the event information derived from the locally available leader state packets, the nominal 3-DOF model, and the two-network prediction. The updated correction state supplements the PINN-D force and moment correction during the dynamics rollout over the next inter-event interval. The leader and follower sides independently apply the same update rule, and their correction states remain identical under the ideal-link synchronization conditions in Assumption 1.

At event time t_k, the online correction state used for the next prediction is defined as

$$\boldsymbol{a}_k = \begin{bmatrix} \boldsymbol{b}_k^{\mathsf{T}} & \dot{\boldsymbol{b}}_k^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^6. \tag{33}$$

Here, $\boldsymbol{b}_k = [b_{N,k}, b_{E,k}, b_{\psi,k}]^{\mathsf{T}}$ contains corrections to northward acceleration, eastward acceleration, and yaw angular acceleration, while $\dot{\boldsymbol{b}}_k$ describes their local rates of change. Model uncertainty and wind, wave, and current effects are treated jointly when constructing the equivalent acceleration correction [30]. Thus, $\boldsymbol{a}_k$ represents the equivalent acceleration correction not explained by the frozen predictor and its local variation, without separately estimating the individual effects. The translational corrections $b_{N,k}$ and $b_{E,k}$ are expressed in the north–east earth-fixed frame. At prediction time s, $\boldsymbol{b}_k + s\dot{\boldsymbol{b}}_k$ is transformed into body-frame accelerations using the auxiliary heading $\widetilde{\psi}_k(s)$. The nominal inertia matrix $\boldsymbol{M}_0$ then maps these accelerations to generalized forces and moment:

$$\widehat{\boldsymbol{d}}_{A,k}(s) = \boldsymbol{M}_0 \begin{bmatrix} \boldsymbol{R}_{BI}(\widetilde{\psi}_k(s)) & \boldsymbol{0}_{2\times 1} \\ \boldsymbol{0}_{1\times 2} & 1 \end{bmatrix} (\boldsymbol{b}_k + s\dot{\boldsymbol{b}}_k), \qquad \widehat{\boldsymbol{d}}_k(s) = \mathrm{sat}_{\boldsymbol{\ell}_d}\left[\widehat{\boldsymbol{d}}_{D,k}(s) + \widehat{\boldsymbol{d}}_{A,k}(s)\right]. \tag{34}$$

Here, $\boldsymbol{R}_{IB}(\psi) \in \mathbb{R}^{2\times 2}$ is the upper-left planar block of $\boldsymbol{R}(\psi)$ in Equation (5). It maps body-frame translational quantities to the north–east earth-fixed frame, while $\boldsymbol{R}_{BI} = \boldsymbol{R}_{IB}^{\mathsf{T}}$ performs the inverse transformation. Equation (34) adds the online correction $\widehat{\boldsymbol{d}}_{A,k}$ to the frozen PINN-D output $\widehat{\boldsymbol{d}}_{D,k}$ and applies the componentwise bounds $\boldsymbol{\ell}_d$. The resulting total force and moment correction $\widehat{\boldsymbol{d}}_k$ enters the dynamics rollout as generalized forces and moment, producing $\widehat{\boldsymbol{x}}_{A,k}(s)$. The units and componentwise bounds of both corrections are provided in Section 4.2.

When a leader state packet becomes available locally, residual adaptation compares $\boldsymbol{y}_k$ with the preceding prediction evaluated at measurement time t_k. Let $h_{k-1} = t_k - t_{k-1}$. The preceding prediction endpoint is $\widehat{\boldsymbol{x}}_{A,k-1}(h_{k-1})$:

$$\boldsymbol{e}_{x,k}^{\mathrm{EP}} = \boldsymbol{y}_k \ominus \widehat{\boldsymbol{x}}_{A,k-1}(h_{k-1}), \tag{35}$$

Equation (35) gives the six-dimensional state difference in position, heading, body-fixed velocity, and yaw rate between the state in the new leader state packet and the preceding prediction endpoint. This endpoint error is used to construct the state correction for the next event domain. Let $\boldsymbol{e}_{p,k}^{\mathrm{EP}} = [e_{x,k}^{\mathrm{EP}}, e_{y,k}^{\mathrm{EP}}]^{\mathsf{T}}$. The measured and predicted body-fixed velocities are each rotated into the north–east earth-fixed frame using their respective headings, and their difference defines the earth-fixed velocity error $\boldsymbol{e}_{v,k}^{I}$. Equation (36) uses the constant-acceleration relations $e_p \simeq ah^2/2$ and $e_v \simeq ah$ to convert these endpoint errors into an acceleration correction:

$$\boldsymbol{z}_k^{\mathrm{EP}} = \begin{bmatrix} \beta_{\mathrm{EP}} \dfrac{2\boldsymbol{e}_{p,k}^{\mathrm{EP}}}{h_{k-1}^2} + (1-\beta_{\mathrm{EP}}) \dfrac{\boldsymbol{e}_{v,k}^{I}}{h_{k-1}} \\ \beta_{\mathrm{EP}} \dfrac{2e_{\psi,k}^{\mathrm{EP}}}{h_{k-1}^2} + (1-\beta_{\mathrm{EP}}) \dfrac{e_{r,k}^{\mathrm{EP}}}{h_{k-1}} \end{bmatrix}, \tag{36}$$

Equation (36) combines four types of endpoint error into a three-dimensional acceleration correction. The two-dimensional position error $\boldsymbol{e}_{p,k}^{\mathrm{EP}}$ and the two-dimensional earth-fixed velocity error $\boldsymbol{e}_{v,k}^{I}$ jointly determine the northward and eastward acceleration corrections. The heading error $e_{\psi,k}^{\mathrm{EP}}$ and yaw-rate error $e_{r,k}^{\mathrm{EP}}$ determine the yaw-acceleration correction. The weight $\beta_{\mathrm{EP}} \in [0,1]$ controls the relative contributions of the reconstructions based on position and heading and on velocity and yaw rate. The factors h_{k-1}^{-2} and h_{k-1}^{-1} convert these two groups of errors into accelerations using the actual inter-event interval. The resulting $\boldsymbol{z}_k^{\mathrm{EP}}$ is the three-dimensional acceleration-correction target used to update $\boldsymbol{b}_k$. Equation (39) subsequently estimates the rate of this target from the temporal trend of recent event targets.

To construct the interval force-balance acceleration-correction target $\boldsymbol{z}_k^{\mathrm{FB}}$ from adjacent event states, write the event-domain generalized disturbance omitted by the nominal 3-DOF model as $\boldsymbol{d} = \boldsymbol{M}_0\dot{\boldsymbol{\nu}} + \boldsymbol{C}_0(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}_0(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{\tau}_{\mathrm{pub}}$. The three components of $\boldsymbol{d}$ are the surge force, sway force, and yaw moment. Integrating this relation over the actual inter-event interval $[t_{k-1}, t_k]$ and dividing by the interval length h_{k-1} gives its interval-average effect. Because $\boldsymbol{M}_0$ is constant, the interval average of the inertial term can be written as $\boldsymbol{M}_0(\boldsymbol{\nu}_k - \boldsymbol{\nu}_{k-1})/h_{k-1}$. The interval averages of the Coriolis, damping, and public-control terms are approximated by the trapezoidal rule using the two event endpoints. The sum of the first two terms in Equation (37) therefore reconstructs the interval-average event-domain generalized disturbance from the adjacent event states.

The preceding dynamics rollout over $[t_{k-1}, t_k]$ already incorporates the frozen PINN-D force and moment correction $\widehat{\boldsymbol{d}}_{D,k-1}(s)$ into the 3-DOF dynamics. Therefore, the event-domain generalized disturbance reconstructed from adjacent event states contains a component already described by PINN-D. Equation (37) subtracts the interval average of this output, leaving the body-frame generalized force and moment not described by the frozen PINN-D. This remainder consists of surge force, sway force, and yaw moment, whereas $\boldsymbol{z}_k^{\mathrm{EP}}$ in Equation (36) consists of northward acceleration, eastward acceleration, and yaw-acceleration corrections. To match their units and components, define the following transformation for a body-frame generalized force and moment $\boldsymbol{f}$: $\mathcal{P}_{I\leftarrow B}(\boldsymbol{f};\psi) = \mathrm{blkdiag}(\boldsymbol{R}_{IB}(\psi), 1)\boldsymbol{M}_0^{-1}\boldsymbol{f}$. Here, $\boldsymbol{M}_0^{-1}\boldsymbol{f}$ converts the generalized force and moment into body-frame acceleration corrections. The matrix $\boldsymbol{R}_{IB}(\psi)$ then transforms the surge and sway components into northward and eastward components, while the yaw component remains a yaw-acceleration correction. Because the event-domain generalized disturbance is reconstructed from data over the entire inter-event interval, Equation (37) uses $\overline{\psi}_k = \psi_{k-1} + \frac{1}{2}\,\mathrm{wrap}_{\pi}(\psi_k - \psi_{k-1})$ as the representative heading for that interval. After

this transformation, z_k^{FB} and z_k^{EP} both consist of northward, eastward, and yaw-acceleration corrections and can be combined in Equation (38).

$$z_k^{\mathrm{FB}} = \mathcal{P}_{I\leftarrow B}\left(M_0\frac{\nu_k-\nu_{k-1}}{h_{k-1}} + \frac{1}{2}\sum_{j\in\{k-1,k\}}\left[C_0(\nu_j)\nu_j + D_0(\nu_j)\nu_j - \tau_{\mathrm{pub},j}\right] - \frac{1}{h_{k-1}}\int_0^{h_{k-1}}\widehat{d}_{D,k-1}(s)\,\mathrm{d}s;\overline{\psi}_k\right). \quad (37)$$

Equation (37) therefore gives the interval force-balance acceleration-correction target z_k^{FB}.

The target z_k^{FB} comes from the dynamic balance over the preceding event interval, while z_k^{EP} comes from the prediction endpoint error over the same interval. Equation (38) combines them into an event correction target:

$$z_k = \mathrm{Proj}_{\mathcal{Z}}\left(\alpha_{\mathrm{FB}}z_k^{\mathrm{FB}} + \alpha_{\mathrm{EP}}z_k^{\mathrm{EP}}\right), \qquad \alpha_{\mathrm{FB}},\alpha_{\mathrm{EP}}\in[0,1], \qquad \alpha_{\mathrm{FB}}+\alpha_{\mathrm{EP}}=1. \quad (38)$$

Here, $\mathcal{Z}$ is the admissible set of three-dimensional acceleration-correction targets. The projection limits the magnitude of each component of z_k.

Equation (38) provides a correction target whenever the measurement interval is long enough for derivative reconstruction. A new target is formed only for intervals of at least 0.75 s, and the implementation retains at most three recent valid pairs (t_ℓ, z_ℓ). The fitting window can be restarted when a change is suspected or detected. A candidate can be generated only when the window is nonempty and the position innovation exceeds the preset proposal threshold. For shorter intervals, retained targets may be reused without estimating a new derivative. Equation (39) uses the retained indices $\mathcal{B}_k$ to fit the correction target and its local rate of change:

$$(\beta_{0,k}^\star,\beta_{1,k}^\star) = \arg\min_{\beta_0,\beta_1}\sum_{\ell\in\mathcal{B}_k}\|z_\ell-\beta_0-(t_\ell-t_k)\beta_1\|_2^2, \qquad a_k^\star = \left[(\beta_{0,k}^\star)^{\mathsf{T}} \quad (\beta_{1,k}^\star)^{\mathsf{T}}\right]^{\mathsf{T}}. \quad (39)$$

Equation (39) centers the linear fit at the current event time t_k. With a single retained target, the implementation sets the fitted slope to zero. The vectors $\beta_{0,k}^\star$ and $\beta_{1,k}^\star$ represent the correction target at t_k and its local rate of change, respectively. Together, they form the six-dimensional fitted target $a_k^\star$, corresponding to the correction and its rate of change in Equation (33). Equation (40) then updates the selected preceding correction state using this fitted target. In Equation (40), a_{k-1} denotes the preceding state selected after validation of any pending candidate. It is either the previously deployed state or an accepted candidate, both expressed at t_{k-1}:

$$a_k^- = \begin{bmatrix} I_3 & h_{k-1}I_3 \\ 0_{3\times3} & \lambda_{r,k}I_3\end{bmatrix}a_{k-1}, \qquad a_k^{\mathrm{ret}} = \rho_a a_k^-, \qquad a_k^{\mathrm{cand}} = \mathrm{Proj}_{\mathcal{A}}\left[a_k^{\mathrm{ret}} + g_k(a_k^\star - a_k^{\mathrm{ret}})\right], \qquad 0<\lambda_{r,k}\le 1, \quad 0<\rho_a\le 1. \quad (40)$$

Before the state propagation in Equation (40), any candidate pending from the preceding update is evaluated using the newly available position measurement. Its prediction is compared with that of the deployed state at the same measurement time. If the candidate is accepted, it replaces the deployed state; otherwise, the deployed state is retained.

Equation (40) first propagates the preceding correction state to the current event. Specifically, $b_k^- = b_{k-1} + h_{k-1}\dot{b}_{k-1}$ and $\dot{b}_k^- = \lambda_{r,k}\dot{b}_{k-1}$ form the propagated state a_k^-. The factor ρ_a then attenuates the propagated state as a whole to obtain the retained historical state a_k^{ret}. The update gain g_k controls how far this state moves toward the fitted target $a_k^\star$. The projection $\mathrm{Proj}_{\mathcal{A}}$ imposes componentwise bounds and event-to-event change limits on the update, producing the candidate state a_k^{cand}. The adapter then evaluates this candidate using a finite-horizon dynamics rollout and a residual-saturation check. Outside the active burst-validation phase, a candidate that passes the finite-horizon dynamics rollout and

residual-saturation checks is committed immediately, giving $\boldsymbol{a}_k = \boldsymbol{a}_k^{\text{cand}}$. During an active burst, the candidate remains pending until the next position measurement and is accepted only if it reduces the prediction discrepancy relative to the deployed state by the prescribed margin. Until it is accepted, or when no new candidate passes the checks, $\boldsymbol{a}_k = \boldsymbol{a}_k^-$. The final correction state is substituted into Equation (34) to generate the trajectory prediction for the next inter-event interval.

The fixed empirical settings combine both correction sources, assigning slightly greater weight to endpoint error feedback and equal weights to its two component reconstructions. During stable operation, smaller update gains and stronger slope forgetting limit changes to the online correction state. When a change is suspected or detected, larger gains are used and more of the recent slope is retained. Phase selection is based on packet-derived innovations and event information. As defined in Equation (40), the slope-retention factor $\lambda_{r,k}$ and the overall retention factor ρ_a have distinct roles. Their fixed values are given in Section 4.2. Section 5.5 evaluates local sensitivity to the fusion weights, β_{EP}, and the update gains while holding both retention factors fixed.

3.4. Prediction-Error-Triggered Communication with Confined Inter-Event Times

The confined inter-event timing rule provides intermittent leader state updates for leader trajectory prediction, event-synchronized residual adaptation, and reference trajectory replanning. Between packet transmissions, the leader-side residual-adapted predictor propagates the leader trajectory prediction. Qu et al. [31] used predicted phase information in event-triggered broadcasting among multiple USVs. Here, the leader monitors the difference between its measured and predicted positions and transmits a leader state packet when the position prediction error reaches the prescribed threshold, subject to the inter-event bounds. The position prediction error and its dimensionless trigger indicator are

$$\boldsymbol{e}_{p,k}(t) = \boldsymbol{y}_p(t) - \widehat{\boldsymbol{p}}_{A,k}(t - t_k), \qquad \gamma_k(t) = \left(\frac{\|\boldsymbol{e}_{p,k}(t)\|_2}{\delta_p} \right)^2. \tag{41}$$

Here, $\boldsymbol{y}_p(t)$ is the current leader-side position measurement, and $\widehat{\boldsymbol{p}}_{A,k}(t - t_k)$ is the predicted position at the same instant. Current leader-side measurements are used only to evaluate the triggering condition; prediction updates use leader state packets. Subtracting the predicted position from the measurement gives the position prediction error $\boldsymbol{e}_{p,k}(t)$. The preset trigger threshold δ_p defines the position prediction error level used in Equation (42), and $\gamma_k(t)$ is the dimensionless squared ratio of the magnitude of the measured position prediction error to this threshold. When $\gamma_k(t) < 1$, the prediction error remains within the prescribed tolerance. When $\gamma_k(t) \geq 1$, the error condition in Equation (42) is satisfied, but immediate packet transmission must also satisfy the inter-event interval constraints.

Equation (42) constrains the timing of leader state packet transmissions through minimum and maximum inter-event intervals. The minimum interval $h_{\min}$ prevents overly frequent transmissions, while $h_{\max}$ requires a refresh at a monitoring instant if no packet has been sent earlier. Liu et al. [13] derived computable lower and upper inter-event bounds for USV formation communication, and Huang et al. [14] used an event-driven estimator to extend communication intervals. In this study, these timing constraints are implemented at discrete monitoring instants. Let $s_m = m\Delta t_{\text{mon}}$, $m = 1, 2, \ldots$, denote the elapsed time from event t_k to the mth monitoring instant, where $\Delta t_{\text{mon}} > 0$ is the monitoring step. For $0 < h_{\min} < h_{\max}$, the next leader state packet transmission time is defined as

$$t_{k+1} = t_k + \min\{s_m : s_m \geq h_{\min} \ \wedge \ [\gamma_k(t_k + s_m) \geq 1 \ \vee \ s_m \geq h_{\max}]\}. \tag{42}$$

Equation (42) has three cases, determined by the elapsed time after event t_k. If $s_m < h_{\min}$, no state packet is sent even when the position prediction error has reached its threshold. If $h_{\min} \leq s_m < h_{\max}$, the packet is sent at the first monitoring instant satisfying $\gamma_k(t_k + s_m) \geq 1$. If the position prediction error does not reach the threshold during this interval, transmission is forced at the first monitoring instant satisfying $s_m \geq h_{\max}$. Therefore, for the actual inter-event interval $h_k = t_{k+1} - t_k$,

$$h_{\min} \leq h_k < h_{\max} + \Delta t_{\mathrm{mon}}. \tag{43}$$

The transmission pattern depends on how often the position prediction error reaches its threshold before the maximum inter-event interval. Frequent threshold crossings lead to earlier prediction-error updates. When the threshold is not reached earlier, maximum-interval updates dominate, and the schedule approaches periodic communication with an interval close to $h_{\max}$. Equation (42) defines the event-triggered communication schedule used by the prediction–adaptation–replanning framework. The evaluation compares this schedule with periodic communication in terms of communication load, prediction accuracy, and closed-loop execution.

Wang et al. [32] excluded Zeno behavior by enforcing a positive minimum inter-event interval. Because Equation (42) imposes $h_k \geq h_{\min} > 0$, infinitely many packet transmissions cannot accumulate within a finite interval. The maximum-interval condition requires a finite-time refresh on the leader side; receiver refresh also depends on packet delivery. At the start of a mission, the K locally augmented packet records required by Equation (17) are not yet available. The leader therefore sends a leader state packet every $h_{\min}$ until the event history $\mathcal{H}_k$ contains K packet records. The nonideal-channel experiments use reliable delivery of these initialization packets and synchronized clocks. Equation (42) is then activated, and subsequent transmission times are determined jointly by the position prediction error condition and the maximum-interval condition. The triggering rule determines only when a packet is sent; the transmitted content remains the complete leader state packet $(\boldsymbol{y}_{k+1}, t_{k+1})$. Under Assumption 1, identical event histories and updates generate identical leader trajectory predictions. The simulation values of δ_p, $h_{\min}$, $h_{\max}$, and Δt_{mon} are given in Section 4.1.

Under nonideal reception, each receiver distinguishes the measurement timestamp from the packet arrival time. Residual reconstruction uses intervals between measurement timestamps, whereas reference trajectories are updated when packets arrive. The new trajectory prediction is evaluated at the packet's current age, advancing the prediction from the measurement time to the execution time. Packets with timestamps no later than that of the last accepted packet are discarded. A pending adaptation candidate is rejected if its cached prediction does not cover the interval to the next measurement time. Between arrivals, each receiver uses its local prediction within the available cache coverage. The next successfully received leader state packet refreshes the available information but does not by itself restore full internal-state synchronization. Section 3.5 describes how the current and updated reference trajectories are connected after a reference update.

3.5. Reference Trajectory Replanning via Quintic Connection Under Event-Triggered Updates

After a leader state packet arrives, event-synchronized residual adaptation updates the online correction state. The predictor then refreshes the leader trajectory prediction, from which the updated reference trajectory is generated. At this point, the follower side is still executing the current reference trajectory. Direct replacement would immediately transfer any position or heading difference between the two reference trajectories to the follower USV's tracking target. The quintic connection used here instead joins the current

reference trajectory to the updated reference trajectory over a finite duration. This separates the information update from the connection between the two reference trajectories.

Seams introduced by reference updates may appear either as offsets along the original direction of travel or as backward displacements. Two representative cases are a reference trajectory offset following a prediction-error update and a backward seam created by a maximum-interval update. The prediction-error case reflects how the updated prediction corrects the current reference trajectory. The maximum-interval case shows that the updated reference trajectory may start behind the last executed reference point even when the position prediction error has not reached its threshold. Prediction-error update and maximum-interval update describe why a leader state packet is sent, whereas forward and backward describe the seam's geometric direction. A prediction-error update may therefore also produce a backward seam.

To describe this geometry, let $\boldsymbol{p}^{-}_{\mathrm{cmd},k}$ denote the last executed leader-side reference position before the update. Let $\boldsymbol{e}_{\parallel,k}$ denote the forward unit tangent of the current reference trajectory at that position. Let $\boldsymbol{p}^{-}_{1,k}$ and $\boldsymbol{p}^{+}_{1,k}$ denote the first corresponding position samples of the current and updated reference trajectories, respectively, on the common time axis. The signed along-track displacements are defined as

$$d^{\pm}_{\parallel,k} = \boldsymbol{e}^{\mathsf{T}}_{\parallel,k}\left(\boldsymbol{p}^{\pm}_{1,k} - \boldsymbol{p}^{-}_{\mathrm{cmd},k}\right). \tag{44}$$

When $d^{+}_{\parallel,k} > 0$, the updated reference point lies ahead of the last executed reference point; when $d^{+}_{\parallel,k} < 0$, it lies behind. Numerical classification uses a tolerance around zero to distinguish forward, backward, and near-zero displacements. The quantity $d^{-}_{\parallel,k}$ describes the continuation direction of the current reference trajectory. Direct replacement creates a backward seam when the current reference trajectory continues forward but the updated reference point lies behind the last executed reference point. These quantities are used only to describe seam geometry and group the experimental cases.

After reference initialization, every valid reference update uses the same quintic connection duration and weighting function, regardless of trigger type or seam direction. Direct replacement is the comparison baseline in Section 5.3.

For either follower side, let t_k denote the measurement timestamp carried by the leader state packet and t^a_k its arrival time; under ideal reception, $t^a_k = t_k$. The current and updated reference trajectories are denoted by $\boldsymbol{p}^{-}_k(t), \psi^{-}_k(t)$ and $\boldsymbol{p}^{+}_k(t), \psi^{+}_k(t)$, respectively, and are evaluated at the same absolute time t. For a delayed packet, the updated reference at τ seconds after arrival corresponds to the prediction time $t^a_k - t_k + \tau$ measured from t_k. Thus, the connection uses the updated reference trajectory after advancing it by the packet age, rather than treating a stale measurement as the current state at arrival.

The connection starts from the current reference trajectory being executed and ends at the corresponding position and heading of the updated reference trajectory at the connection endpoint. If the preceding connection is still in progress, $\boldsymbol{p}^{-}_k(t), \psi^{-}_k(t)$ contain its remaining connection segment and the subsequent reference trajectory. The new connection is constructed from the reference trajectory actually being executed. It neither returns to the original prediction generated from the preceding leader state packet nor substitutes the follower USV's actual position for the connection origin.

Given a connection duration $T_c > 0$, the normalized connection time over $t \in [t^a_k, t^a_k + T_c]$ is defined as

$$\sigma_k(t) = \frac{t - t^a_k}{T_c}, \qquad 0 \le \sigma_k(t) \le 1. \tag{45}$$

The connected trajectory equals the current reference trajectory at the start and the updated reference trajectory at the end. To prevent the varying weight from introducing additional velocity and acceleration terms at either endpoint, the weighting function is required to satisfy

$$\begin{aligned} &\beta(0) = 0, \qquad \beta(1) = 1, \\ &\beta'(0) = \beta'(1) = 0, \\ &\beta''(0) = \beta''(1) = 0. \end{aligned} \tag{46}$$

Let $\beta(\sigma) = \sum_{j=0}^{5} a_j \sigma^j$. The conditions at $\sigma = 0$ give $a_0 = a_1 = a_2 = 0$, whereas those at $\sigma = 1$ give $a_3 = 10$, $a_4 = -15$, and $a_5 = 6$, yielding

$$\beta(\sigma) = 10\sigma^3 - 15\sigma^4 + 6\sigma^5. \tag{47}$$

This weight increases monotonically from zero to one over $[0, 1]$. The connected reference position is defined as

$$\boldsymbol{p}_k^C(t) = \big[1 - \beta\big(\sigma_k(t)\big)\big]\,\boldsymbol{p}_k^-(t) + \beta\big(\sigma_k(t)\big)\,\boldsymbol{p}_k^+(t), \tag{48}$$

Here, the superscript C denotes the reference trajectory produced by the connection. The quintic polynomial specifies how the weight varies with time, while the two input reference trajectories continue to advance during the connection. Equation (48) therefore connects the two moving reference trajectories pointwise on their common time axis rather than inserting a curve between two fixed positions. Because $0 \le \beta \le 1$, the connected position at each instant lies between the two corresponding reference positions. Both forward offsets and backward seams enter the connection through the same position difference $\boldsymbol{p}_k^+(t) - \boldsymbol{p}_k^-(t)$ and therefore use the same curve construction.

The heading uses the same weight after accounting for angular periodicity, with wrap_π mapping each angle to $[-\pi, \pi)$. The difference $\psi_k^+(t) - \psi_k^-(t)$ is first wrapped and then unwrapped along the sample sequence to obtain the continuous angular difference $\Delta\psi_k(t)$. The connected heading is

$$\psi_k^C(t) = \mathrm{wrap}_\pi\big[\psi_k^-(t) + \beta\big(\sigma_k(t)\big)\Delta\psi_k(t)\big], \qquad \Delta\psi_k = \mathrm{unwrap}\big[\mathrm{wrap}_\pi\big(\psi_k^+ - \psi_k^-\big)\big]. \tag{49}$$

This procedure prevents an equivalent heading difference across $\pm\pi$ from being interpreted as a full rotation between adjacent samples. The connection begins with the current reference heading and ends at a heading equivalent to the updated reference heading modulo 2π. In implementation, the first and last samples are assigned directly the corresponding poses from the two input reference trajectories.

The endpoint conditions of the quintic weight determine how the connection inherits the local variations of the input reference trajectories. Define $\Delta\boldsymbol{p}_k(t) = \boldsymbol{p}_k^+(t) - \boldsymbol{p}_k^-(t)$ and assume that both position reference trajectories have continuous first and second time derivatives over the interval considered. Differentiating Equation (48) gives

$$\dot{\boldsymbol{p}}_k^C(t) = \dot{\boldsymbol{p}}_k^-(t) + \beta(\sigma_k)\Delta\dot{\boldsymbol{p}}_k(t) + \frac{\beta'(\sigma_k)}{T_c}\Delta\boldsymbol{p}_k(t), \tag{50}$$

$$\ddot{\boldsymbol{p}}_k^C(t) = \ddot{\boldsymbol{p}}_k^-(t) + \beta(\sigma_k)\Delta\ddot{\boldsymbol{p}}_k(t) + \frac{2\beta'(\sigma_k)}{T_c}\Delta\dot{\boldsymbol{p}}_k(t) + \frac{\beta''(\sigma_k)}{T_c^2}\Delta\boldsymbol{p}_k(t). \tag{51}$$

Because $\beta'(0) = \beta'(1) = \beta''(0) = \beta''(1) = 0$, the connection endpoints satisfy

$$\begin{aligned} \left.\frac{\mathrm{d}^\ell \boldsymbol{p}_k^C}{\mathrm{d}t^\ell}\right|_{t=t_k^a} &= \left.\frac{\mathrm{d}^\ell \boldsymbol{p}_k^-}{\mathrm{d}t^\ell}\right|_{t=t_k^a}, \\ \left.\frac{\mathrm{d}^\ell \boldsymbol{p}_k^C}{\mathrm{d}t^\ell}\right|_{t=t_k^a+T_c} &= \left.\frac{\mathrm{d}^\ell \boldsymbol{p}_k^+}{\mathrm{d}t^\ell}\right|_{t=t_k^a+T_c}, \qquad \ell = 0, 1, 2. \end{aligned} \tag{52}$$

Here, $\ell = 0$, 1, and 2 correspond to position, velocity, and acceleration, respectively. When heading is represented by a locally continuous angle and the input heading trajectories have corresponding smoothness, the same endpoint relations apply to the connected heading. In practice, the reference trajectories are constructed and queried using discrete samples. The relations above characterize endpoint matching in the continuous representation, while Section 5.3 evaluates the control response of the sampled implementation.

Equations (50) and (51) also show how the connection duration affects the rate of reference change. A shorter T_c joins the updated reference trajectory sooner, but the additional velocity term caused by the reference offset scales with $1/T_c$. The additional acceleration term also contains a $1/T_c^2$ factor. A longer T_c distributes the same reference change over a longer interval. This study uses the same fixed value, $T_c = 1.8$ s, for both seam types, with sensitivity and computational cost evaluated in Section 5.5. The quintic connection controls how quickly the current reference trajectory joins the updated reference trajectory. The resulting USV motion and control input demand are evaluated with the associated controllers under continuous closed-loop operation.

During continuous operation, a new leader state packet may arrive before the current connection is complete. The connection trajectory currently being executed and its continuation then become the new current reference trajectory. At the new arrival time, the connection time is reset to zero, and a new connection joins this trajectory to the updated reference trajectory generated from the new packet. The follower USV's actual state and controller memory remain continuous. If no new valid reference update arrives, the current connection continues. After $t_k^a + T_c$, the reference proceeds along the unexecuted portion of the updated reference trajectory. The event-triggering mechanism continues monitoring, while leader trajectory prediction and event-synchronized residual adaptation remain active throughout this process.

Before a connection is constructed, both reference trajectories must cover the entire connection interval and provide the preview required by the controller. In the 2 s prediction implementation evaluated in Section 5.5.2, the reference layer retains the body-fixed velocity and yaw rate at the end of the 2 s prediction. It extends the reference trajectory under constant terminal motion to provide the tail segment required for the connection and controller preview. This tail segment extends only the reference layer; it is not written back to the leader trajectory prediction or used to evaluate prediction accuracy. If the prepared trajectories still lack sufficient common coverage, the method retains the current reference trajectory and records the connection as unavailable. The prescribed connection duration remains unchanged, and the method does not switch to Direct replacement.

The leader–follower reference mapping defined in Section 2.3 converts the connected leader-side reference into follower reference trajectories for the two followers, which their respective controllers then track continuously. The quintic connection changes only the connection between the current and updated reference trajectories. Leader trajectory prediction, event-synchronized residual adaptation, and the formation mapping continue under their respective rules. Section 5.1 illustrates the two representative seam types. Section 5.3 evaluates the resulting event-time control input changes and tracking performance through same-state local comparisons and continuous execution over the full trajectory segment.

This construction smooths the connection between reference trajectories; it does not impose vessel dynamics or actuator constraints during reference generation. The controller, input governor, and saturation in Appendix D constrain the applied inputs but do not guarantee dynamic feasibility of the generated reference trajectory. Appendix E gives a conditional local bound on formation position error and a bound on the online correction state under its stated assumptions.

4. Simulation Setup

This section evaluates the framework in Section 3 under measurement noise, plant parameter variations, and spatially varying disturbances.

4.1. Simulation Scenarios and Common Settings

Simulations use the underactuated 3-DOF USV model (Section 2.1) and leader–follower formation (Section 2.3), with parameters in Table 1. Both followers start at their respective prescribed formation reference positions.

Global formation trajectory information is shared before each mission, without advance information about the unshared maneuver.

The simulated plant parameters and environmental conditions vary across scenarios, but the predictor uses the same fixed nominal model throughout. Each trajectory passes through three disturbance regions where unmodeled acceleration and damping vary with position.

Independent zero-mean Gaussian noise is added to the state measurements in transmitted leader state packets, as specified in Section 2.2.

Table 1. Model, mission, and simulation parameters.

Parameter	Value	Parameter	Value
L	2.0 m	B	0.75 m
m	250 kg	I_z	95.05 kg m^2
$X_{\dot{u}}$	-35 kg	$Y_{\dot{v}}$	-180 kg
$N_{\dot{r}}$	-45 kg m^2	Linear damping $[u, v, r]$	$[38, 95, 32]^\mathsf{T}$
Quadratic damping $[u, v, r]$	$[60, 190, 75]^\mathsf{T}$	Leader-side policy input limits $(\tau_{u,L}^{\max}, \tau_{r,L}^{\max})$	(180 N, 85 N m)
Formation edge	8 m	Initial formation error	0
Circle radius	40–50 m	S-bend radius	30–35 m
Circle speed	1.0–1.2 m s^{-1}	S-bend speed	1.0–1.15 m s^{-1}
Circle duration	320 s	S-bend duration	180 s
Initial phase	$\mathcal{U}[0, 2\pi)$	S-bend length	130–150 m
Mass variation	$\pm$8%	Yaw-inertia variation	$\pm$10%
Added-mass variation	$\pm$15%	Damping-parameter variation	$\pm$15%
Disturbance regions	3	Regional damping change	0.20–0.35
Translational acceleration bound	0.120 m s^{-2}	Yaw acceleration bound	0.070 rad s^{-2}
σ_x, σ_y	0.020 m	σ_ψ	0.60°
σ_u	0.020 m s^{-1}	σ_v	0.015 m s^{-1}
σ_r	0.010 rad s^{-1}	δ_p	0.15 m
$h_{\min}$	0.50 s	$h_{\max}$	2.00 s
Monitoring/simulation step	0.05 s	Predictor deployment step	0.025 s
Follower surge force limit	500 N	Follower yaw moment limit	250 N m

The multiseed evaluation set contains 50 Circle and 50 S-bend missions. Prediction comparisons replay the same event histories at common prediction origins. Running both reference-connection strategies on each mission gives 200 continuous executions. Communication and stress tests use two other four-mission cohorts; sensitivity and timing use one development mission per type.

Stress tests either triple the measurement-noise standard deviations or introduce a temporary unshared avoidance maneuver followed by a return to the original plan. Conditions for the same mission share initial states, true disturbances, and standardized noise realizations. Channel tests impose delayed packet arrivals and independent receiver-side packet losses, with a reliable six-packet initialization and synchronized clocks. The protocol uses no acknowledgments, retransmissions, or complete-state snapshots. Section 5.4 specifies the tested condition levels.

4.2. Predictor Training and Deployment Settings

The training, validation, and system-evaluation sets are partitioned by complete trajectory.

Each mission mode contains 100 training trajectories and 20 validation trajectories. The 200 training and 40 validation trajectories yield 10,282 and 1998 windows, respectively. Each window starts at an event and uses the six-record event history and corresponding global formation trajectory information as inputs.

The subsequent leader trajectory and the endpoint state at the next transmission event provide supervision.

Following the curriculum strategy of Krishnapriyan et al. [33], we train PINN-X and PINN-D jointly with Adam in successive stages. Both networks begin training with zero-initialized output weights and biases. The final-stage checkpoint is selected by horizon-normalized ADE on the 40-trajectory validation set. Network parameters remain frozen during deployment; event-synchronized residual adaptation updates only the online correction state defined in Section 3.3.

Table 2 lists the network architecture, optimization settings, scaling parameters, and residual-adaptation parameters.

Table 2. Predictor, training, and residual-adaptation parameters.

Parameter	Value	Parameter	Value
K	6	$\dim(\boldsymbol{c}_k)$	90
Auxiliary state supervision weight	0.25	Activation	tanh
PINN-X hidden layers	3×96	PINN-X outputs	6
PINN-D hidden layers	2×64	PINN-D outputs	3
Random generator	Mersenne Twister	Random seed	610,905
Hidden-layer weights	Glorot	Bias initialization	0
PINN-X output weights	0	PINN-D output weights	0
Optimizer	Adam	Gradient component clip	5.0
PINN-X learning rate	5×10^{-4}	PINN-D learning rate	5×10^{-4}
Training step	0.10 s	Deployment step	0.025 s
Horizons	2.1/5.0/8.0 s	Maximum epochs	2/4/10
Mini-batch size	256/256/256	Training windows per epoch	All (10,282)/All/All
Validation windows	All (1998)/All/All	T_b	1 s
$\mathcal{L}_R$ weight	1	$\mathcal{L}_E$ weight	0.5
$\mathcal{L}_{RK}$ weight	0.3	$\mathcal{L}_\eta$ weight	0.1
$\mathcal{L}_D$ weight	0.005	$\mathcal{L}_D$ weight	0.002
$\boldsymbol{S}_p^{\mathrm{loc}}$	$10\,\mathrm{m}\,\boldsymbol{I}_2$	$\boldsymbol{S}_v^{\mathrm{loc}}$	$\mathrm{diag}(1.2, 0.30, 0.10)$
$\boldsymbol{S}_x$	$[0.50, 0.50, 5^\circ, 0.15, 0.08, 0.04]^\mathsf{T}$	$\boldsymbol{S}_\eta$	$[1, 1, 0.10]^\mathsf{T}$
$\boldsymbol{\ell}_x$	$[8, 8, 0.8, 0.6, 0.35, 0.25]^\mathsf{T}$	$\boldsymbol{\ell}_d$	$[120, 160, 90]^\mathsf{T}$
β_{EP}	0.50	$(\alpha_{\mathrm{FB}}, \alpha_{\mathrm{EP}})$	$(0.40, 0.60)$
$\lvert\mathcal{B}_k\rvert$	3	Minimum derivative interval	0.75 s
$\lambda_{r,k}$, stable/burst	0.25/0.85	ρ_a	0.995
g_k, stable/burst	0.30/0.80	Burst duration	5
$\mathrm{bound}(\mathcal{Z})$	$[0.18, 0.18, 0.0875]^\mathsf{T}$	$\boldsymbol{a}_0$	$\boldsymbol{0}_6$
$\mathrm{bound}(\mathcal{A})$	$[0.18, 0.18, 0.0875, 0.060, 0.060, 0.035]^\mathsf{T}$	Single-event change limit	$0.75\,\mathrm{bound}(\mathcal{A})$
Maximum residual-saturation fraction	0.05		

Hidden-layer sizes are given as depth × width; slash-separated entries follow the curriculum stages, with "All" denoting all available windows.

PINN-X-only and PINN-D-only are trained independently; the baselines are GRU (Appendix A.1) and a six-dimensional state EKF (Qscale = 1000; Appendix A.2). Consistency-term pairs change only state-transition and kinematic penalties, with matched architecture, initialization, data, batch order, training budget, and checkpoint selection. Both retain RK4 dynamics, auxiliary state supervision, and shared residual regularization.

4.3. Evaluation Protocol and Compared Configurations

Both followers track follower reference trajectories using fixed-parameter controller configurations for Circle and S-bend. Within each mission, Direct replacement and quintic connection share controller, filtering, and input-governor settings. Their surge-force and yaw-moment limits are 500 N and 250 N m, respectively (Appendix D).

Each update starts a new 1.8 s quintic connection from the current reference trajectory (Table 3). Later updates interrupt unfinished connections; insufficient cache coverage leaves the current reference trajectory in place.

Table 3. Reference-connection experiment settings.

Setting	Value or Rule
Comparison	Direct replacement/quintic connection at every update
Connection duration	1.8 s
Follower-control and motion-integration step	0.05 s
Internal prediction cache	8 s
Later update	Restart from the current reference trajectory; interrupt the unfinished connection
Insufficient coverage	Retain the current reference trajectory and record the condition

Prediction accuracy is evaluated at each state packet transmission for which the complete subsequent 2 s ground-truth trajectory is available. Let $\widehat{\boldsymbol{p}}_k(s)$ denote the leader position predicted from event k, and let $\boldsymbol{p}_L(t_k+s)$ denote the corresponding true position. For samples $s_n \in [0, 2\,\mathrm{s}]$, the average displacement error (ADE) and final displacement error (FDE) are

$$\mathrm{ADE}_k = \frac{1}{N_k}\sum_{n=1}^{N_k}\left\|\widehat{\boldsymbol{p}}_k(s_n) - \boldsymbol{p}_L(t_k+s_n)\right\|_2, \qquad \mathrm{FDE}_k = \left\|\widehat{\boldsymbol{p}}_k(2) - \boldsymbol{p}_L(t_k+2)\right\|_2. \tag{53}$$

All prediction methods are evaluated using the same event histories, prediction origins, and scoring intervals. State-wise root-mean-square errors (RMSEs) are reported, with heading error calculated as the shortest equivalent angular difference. State-transition and kinematic defects are reported separately from position prediction errors. Prediction scoring uses a 2 s horizon, and Table 3 specifies the internal prediction-cache coverage.

Recovery analysis includes all disturbance-region traversals identified by physical region entry and aligns each traversal to the first leader state packet after entry. The primary window comprises the first 11 packet origins, each with a 2 s endpoint inside the region. After an FDE above 0.07 m, recovery is confirmed by three consecutive valid FDE values at or below this threshold; a missing value interrupts the sequence. Initially low error, recovery, non-recovery, insufficient observation, and renewed threshold exceedance are recorded separately. Recovery confirmation time is measured from physical region entry, with full-region observations reported as a supplementary analysis. Maneuver recovery requires the FDE to remain at or below 0.07 m for 2 s. Onset delay is measured from maneuver completion to the start of the confirmed recovery interval.

For every update, including completed, interrupted, and incompletely observed connections, a paired comparison is constructed using identical pre-update follower states, controller memory, and previous inputs copied from the quintic-connection run. For each input channel, the reported change is the larger value across the two followers.

For follower i, input channel $c \in \{u, r\}$, and connection method m, the normalized event-time control input change is

$$\Delta_{i,c,k}^{(m)} = \frac{\left|\tau_{i,c}^{(m)}(t_k^+) - \tau_{i,c}(t_k^-)\right|}{\tau_c^{\max}}, \qquad m \in \{\text{Direct}, \text{Quintic}\}, \tag{54}$$

Here, t_k^- and t_k^+ denote the adjacent samples immediately before and after the update, and $\tau_c^{\max}$ is the corresponding input limit. Absolute input changes are given by the numerator of Equation (54). Relative reductions compare equally weighted trajectory means with Direct replacement.

Each strategy retains its own reference and controller histories throughout each complete mission. Position tracking RMSE is calculated relative to each strategy's own reference trajectory, whereas actual formation position RMSE is calculated relative to the formation positions mapped from the true leader pose. The normalized total control variation (TV) sums the normalized absolute input changes between adjacent samples and then averages them over both followers and both control channels. We also record raw requests, pre-governor demands, applied inputs, actuator saturation, time spent above 95% of the input limits, and peaks in speed, acceleration, and curvature.

Periodic communication at 0.5 and 2 s and event-triggered communication (ET) are compared on a common 0.05 s true-time grid. Packet counts include initialization. Each packet has a 56-byte application payload, excluding transport overhead. Prediction availability is the fraction of evaluation times for which a complete 2 s prediction is available. Packet arrival and internal synchronization are evaluated separately.

Adaptation parameters are varied one group at a time. Connection-duration sensitivity and post-warm-up runtime measurements use the implementation that applies quintic connection at every update. Runtime measurements include adaptation, leader trajectory prediction, and reference trajectory processing, but exclude packet transmission and the follower control loop. Section 5.5 lists the tested configurations and timing conditions.

Trajectory-level means receive equal weight. We calculate 95% confidence intervals using a paired percentile bootstrap that resamples complete trajectories within each mission type. Appendix B.5 describes the procedure for comparing Direct replacement with quintic connection. Events, receivers, and channel realizations are treated as nested within trajectories. Failed and incomplete runs remain in the record and are not assigned zero errors.

5. Results and Analysis

Leader–follower formation execution under intermittent communication involves two coupled stages: updating the leader trajectory prediction from leader state packets, and generating an updated reference trajectory through the leader–follower reference mapping before connecting it to the current reference trajectory. This section first illustrates this process in the Circle and S-bend missions. It then evaluates leader trajectory prediction and the effects of event-synchronized residual adaptation and quintic connection, and examines changes in communication conditions, parameter sensitivity, and computational cost.

5.1. Prediction Updates and Formation Execution in Representative Missions

Figures 5 and 6 show representative disturbance-region traversals in the Circle and S-bend missions, respectively. Each mission uses one leader USV and two follower USVs.

The global formation trajectory information is shared before the mission, and the follower reference trajectories are generated from the leader trajectory prediction through the leader–follower reference mapping. When a leader state packet arrives, event-synchronized residual adaptation updates the online correction state, and the predictor refreshes the leader trajectory prediction accordingly. The quintic connection then gradually joins the updated reference trajectory generated from the updated prediction to the current reference trajectory.

Figure 5 shows the leader trajectory prediction and follower reference trajectory updates in the Circle mission. The left panel shows the complete formation trajectories and locates the disturbance-region traversal examined here. The right panel enlarges the leader USV's actual trajectory, the leader trajectory prediction produced by the residual-adapted predictor, the follower reference trajectories, and the actual trajectories of the two follower USVs within this traversal. A local prediction mismatch develops between the leader trajectory prediction and the leader USV's actual motion near the region entrance. Subsequent leader state packets update the online correction state and the leader trajectory prediction. The inset further enlarges the local formation geometry at one reference trajectory update, showing the updated reference trajectory generated by the leader–follower reference mapping and its quintic connection to the current reference trajectory.

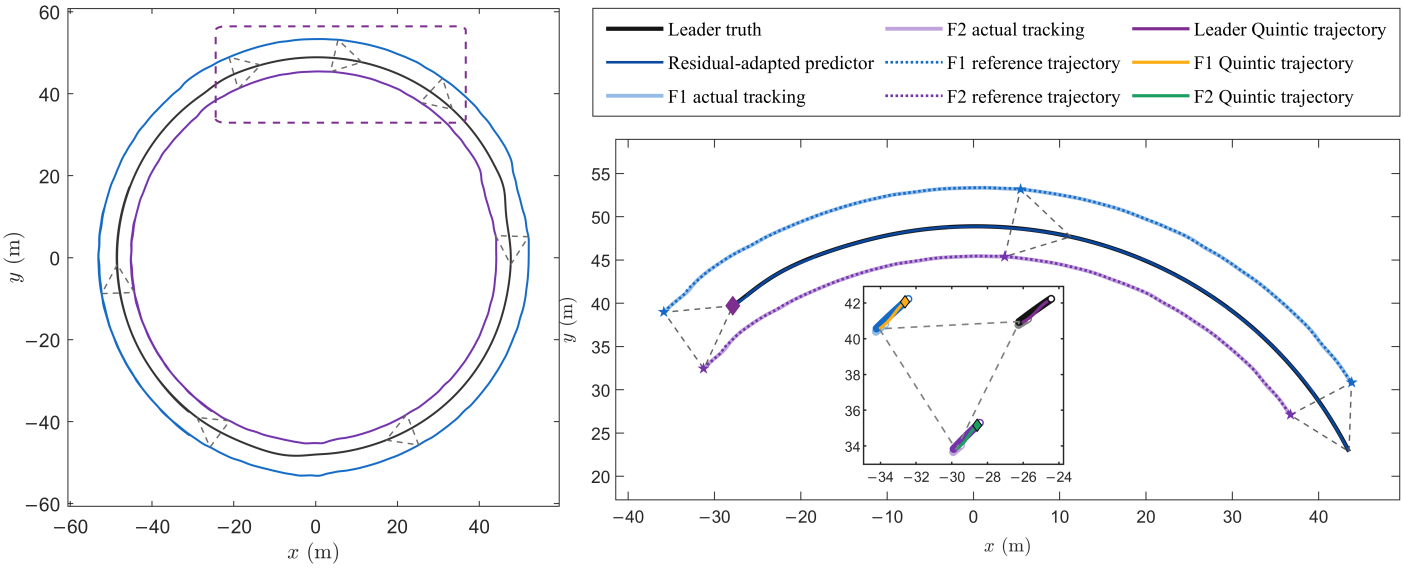


Figure 5. Leader trajectory prediction and formation execution during a Circle disturbance-region traversal. (**Left**): complete Circle formation trajectories; (**Right**): enlarged disturbance-region traversal showing the leader trajectory prediction and the follower reference trajectories and the actual trajectories of the follower USVs; inset: local quintic connection at a reference trajectory update.

Figure 6 shows the S-bend mission, in which path curvature varies continuously. The lower panel locates the disturbance-region traversal on the complete S-bend trajectory, while the upper panel shows the motion of the three USVs, the leader trajectory prediction, and the follower reference trajectories during this traversal. Unlike the circular motion in the Circle mission, the S-bend path changes curvature as the formation advances, and the follower reference trajectories adjust to the changing path geometry. The inset relates the quintic connection to the positions of the three USVs and shows how the same prediction-update process operates under varying path geometry.

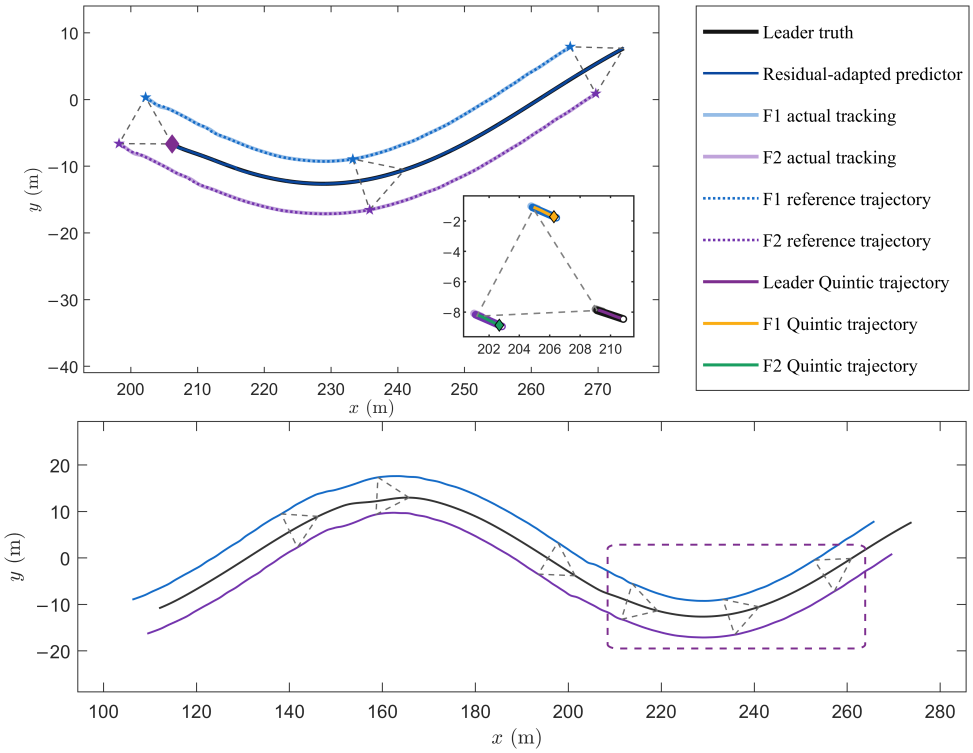


Figure 6. Leader trajectory prediction and formation execution during an S-bend disturbance-region traversal. (**Upper**): enlarged S-bend disturbance-region traversal showing the leader trajectory prediction and the follower reference trajectories and the actual trajectories of the follower USVs; (**Lower**): complete S-bend formation trajectories; inset: local quintic connection at a reference trajectory update.

Figure 7 further examines two types of seam between the current and updated reference trajectories at an event time. Both panels use a local along-track–cross-track coordinate frame whose origin is the position of the current reference trajectory at that event time. The light- and dark-blue curves represent the current and updated reference trajectories, respectively; the gray segment represents Direct replacement, and the orange curve represents the quintic connection. Figure 7a shows a reference trajectory offset following a prediction-error update. Direct replacement would impose this offset immediately on the tracking target. The quintic connection instead starts from the current reference trajectory and gradually increases the weight assigned to the updated reference trajectory over the fixed connection interval.

Figure 7b shows a backward seam produced by a maximum-interval update: the initial position of the updated reference trajectory lies behind the corresponding position on the current reference trajectory. In this case, communication is triggered by the maximum inter-event interval condition and does not require the position prediction error to reach the triggering threshold. In both cases, the same quintic connection is applied at every reference trajectory update to join the current reference trajectory gradually to the updated reference trajectory. This reduces the abrupt change in the tracking target caused by the positional offset in Figure 7a and the initial backward shift caused by the backward seam in Figure 7b. Section 5.3 further evaluates the event-time control input changes in the two cases through same-state local comparisons.

These representative disturbance-region traversals place leader trajectory prediction updates and quintic connection within the same closed-loop process. The following analysis first evaluates leader trajectory prediction and event-synchronized residual adaptation using shared event histories and prediction origins. It then evaluates quintic connection through local and continuous comparisons under matched control conditions.

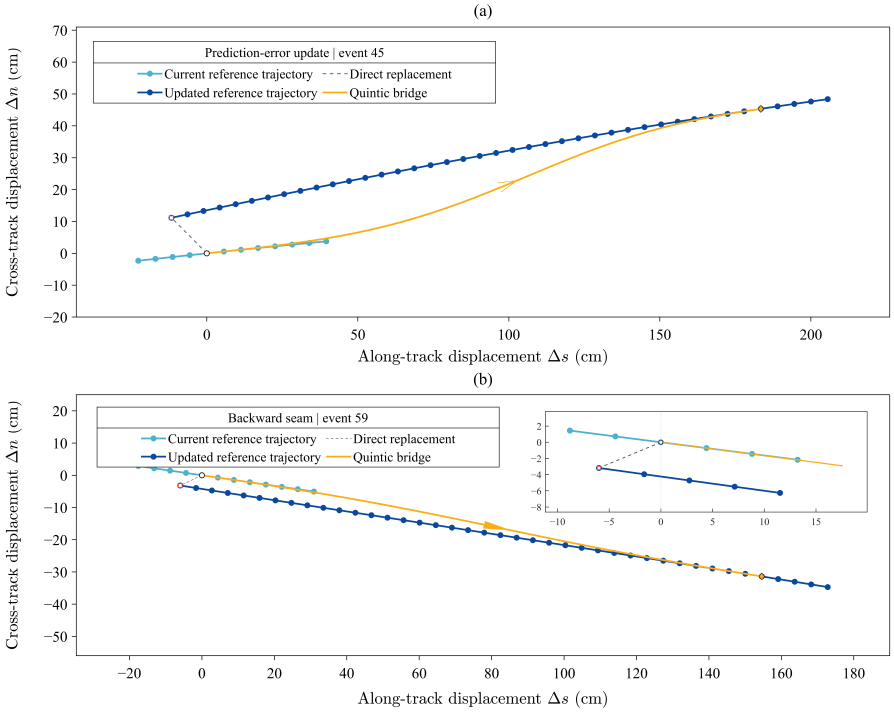


Figure 7. Local quintic connection for two representative seams. (**a**) Reference trajectory offset following a prediction-error update; (**b**) backward seam following a maximum-interval update, with an inset showing the local seam.

5.2. Prediction Performance, Component Roles, and Recovery from Local Prediction Mismatch in Disturbance Regions

5.2.1. Component Ablation and Prediction Comparison with Shared Event Histories

The event-domain state–residual PINN trajectory predictor comprises the auxiliary state correction network (PINN-X) and the generalized residual estimation network (PINN-D). In Table 4, "PINN-X only" and "PINN-D only" denote independently trained single-network configurations, whereas the frozen state–residual PINN predictor uses both networks jointly. Event-synchronized residual adaptation updates the low-dimensional online correction state at the current event without changing the network weights. The full configuration with this correction enabled is termed the residual-adapted predictor.

The prediction comparison reused the multiseed evaluation set comprising 50 Circle and 50 S-bend missions. All leader state packets transmitted during each complete trajectory were collected, and each event time served as a prediction origin. A total of 12,388 2 s trajectory segments were evaluated at origins with a six-packet event history and complete future ground truth. All configurations used the same event histories, global formation trajectory information, prediction origins, and scoring intervals. Segment-level errors were first averaged within each trajectory and then aggregated with equal weight across the 100 complete trajectories; overlapping segments were not treated as independent replicates.

Table 4 reports the means across trajectories and 95% confidence intervals (95% CIs) for average displacement error (ADE) and final displacement error (FDE). The intervals were obtained by bootstrapping the 100 complete trajectories as described in Section 4.3, while retaining 50 Circle and 50 S-bend trajectories in each resample. Each learning-based configuration used one fixed trained model instance, so these intervals reflect variation across trajectories. Complete ADE/FDE comparisons over extended prediction horizons of 5 and 8 s are provided in Appendix A.3.

Table 4. Component ablation and 2 s prediction performance with shared event histories.

Comparison	Prediction Configuration	Mean 2 s ADE (m) [95% CI]	Mean 2 s FDE (m) [95% CI]
Offline components	Nominal dynamics rollout	0.05809 [0.05645, 0.05958]	0.12183 [0.11726, 0.12609]
	PINN-X only	0.05035 [0.04901, 0.05164]	0.09645 [0.09266, 0.10018]
	PINN-D only	0.04246 [0.04142, 0.04352]	0.07644 [0.07319, 0.07954]
	Frozen state–residual PINN predictor	0.04227 [0.04121, 0.04332]	0.07587 [0.07262, 0.07896]
	Frozen state–residual PINN predictor (additional discrete physics-consistency terms disabled)	0.04225 [0.04119, 0.04330]	0.07580 [0.07254, 0.07889]
Residual adaptation	PINN-D only + residual adaptation	0.03486 [0.03437, 0.03541]	0.05201 [0.05056, 0.05362]
	Residual-adapted predictor	0.03475 [0.03425, 0.03531]	0.05170 [0.05022, 0.05334]
Baselines	GRU	0.07387 [0.06610, 0.08530]	0.11300 [0.09866, 0.13429]
	Six-dimensional EKF	0.05872 [0.05704, 0.06025]	0.12295 [0.11832, 0.12731]

The offline component comparison showed that the PINN-D force and moment correction accounted for most of the improvement in position prediction. The "PINN-X only" and "PINN-D only" configurations achieved ADEs of 0.05035 and 0.04246 m, respectively, both below the 0.05809 m obtained by the nominal dynamics rollout. The frozen state–residual PINN predictor further reduced the ADE to 0.04227 m, 0.46% below that of the "PINN-D only" configuration. ADE and FDE alone do not fully characterize the roles of PINN-X and the additional discrete physics-consistency terms. Section 5.2.2 therefore examines their effects on state prediction and auxiliary trajectory consistency.

Event-synchronized residual adaptation further improved the leader trajectory prediction generated by the frozen state–residual PINN predictor. ADE/FDE decreased from 0.04227/0.07587 m to 0.03475/0.05170 m, corresponding to reductions of 17.78% and 31.86%, respectively. The paired absolute change in ADE was −0.00751 m, with a 95% trajectory-level bootstrap interval of −0.00819 to −0.00682 m. Applying the same residual adaptation to the "PINN-D only" configuration produced similar displacement errors, so most of the benefit from online correction did not depend on the small improvement in position prediction contributed by PINN-X.

The baseline comparison included the data-driven GRU sequence predictor and the model-driven six-dimensional EKF; their designs and the corresponding training and filtering settings are given in Appendices A.1 and A.2. With the same event histories and scoring intervals, the residual-adapted predictor achieved an ADE of 0.03475 m, compared with 0.07387 m for the GRU and 0.05872 m for the six-dimensional EKF. The corresponding FDEs were 0.05170, 0.11300, and 0.12295 m, respectively. These comparisons support the role of the PINN-D force and moment correction and event-synchronized residual adaptation in short-horizon prediction under the shared evaluation conditions.

5.2.2. Roles of Auxiliary State Correction and Additional Discrete Physics-Consistency Terms

The contribution of PINN-X was most apparent in heading, sway velocity, and yaw rate, which displacement errors alone do not fully capture. Figure 8 compares the frozen state–residual PINN predictor with the independently trained "PINN-D only" configuration. Figure 8a–c show the six-dimensional state results for all 100 missions, the 50 Circle missions, and the 50 S-bend missions, respectively. For each cell, the RMSE reduction relative to the "PINN-D only" configuration was first calculated over each complete trajectory, after which the mean ± standard deviation was calculated across trajectories. Positive values indicate lower errors for the joint configuration, and percentages were not aggregated across state variables with different units.

At 2 s, the mean per-trajectory reductions in sway-velocity and yaw-rate RMSE were 4.69% and 4.48%, respectively. With longer prediction horizons, heading RMSE decreased by 6.65% and 9.02% at 5 and 8 s, while yaw-rate RMSE decreased by 8.17% and 8.66%.

These state-level improvements show the role of PINN-X more clearly than the changes in position error reported in Table 4. By contrast, surge-velocity RMSE increased by an average of 1.35% at 2 s, while the RMSEs of the x and y position components increased by 0.91% and 1.06% at 8 s. The full six-state results indicate that the effect of auxiliary state correction depends on both the state component and the prediction horizon.

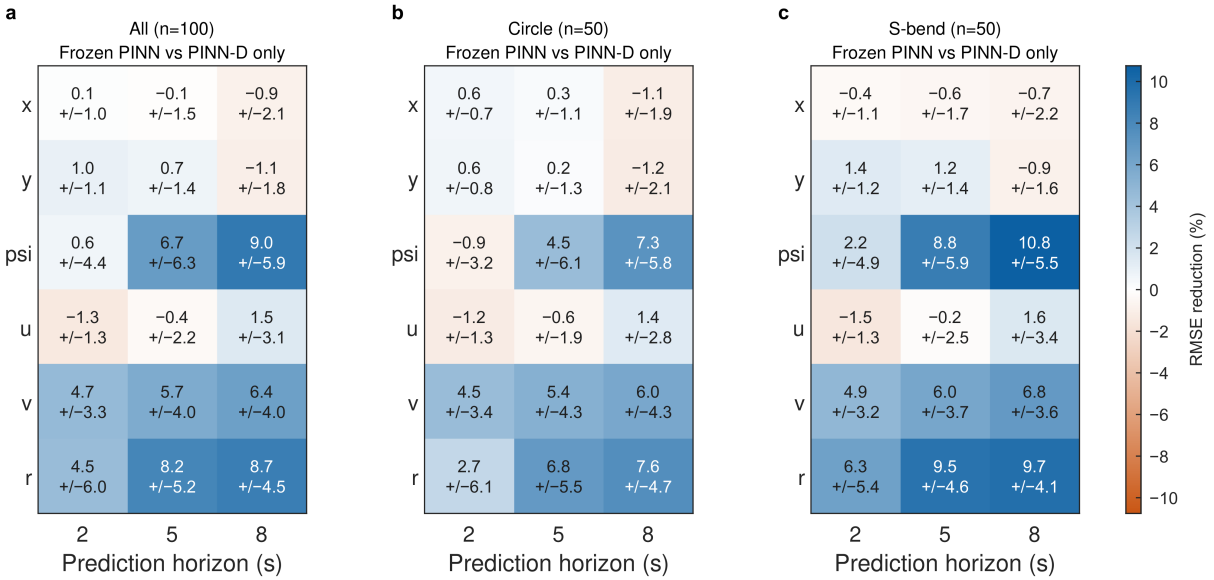


Figure 8. Effect of PINN-X on six-dimensional state prediction in the frozen state–residual PINN predictor. Six-state RMSE reductions for (**a**) all 100 missions; (**b**) the 50 Circle missions; (**c**) the 50 S-bend missions.

To separate the effect of the embedded dynamics structure from that of the physics-informed loss, the two models used identical PINN-X, PINN-D, and RK4 dynamics-based prediction structures, auxiliary trajectory supervision, residual regularization, and training settings. The only difference was whether the discrete dynamics-consistency and kinematic-consistency losses were included. These two losses are jointly referred to below as the physics-informed loss and constitute the additional discrete physics-consistency terms in this ablation. The former constrains adjacent auxiliary states generated by PINN-X to satisfy the RK4 dynamics propagation relation, whereas the latter enforces consistency between changes in position and heading along the auxiliary trajectory and the corresponding body-fixed velocity and yaw rate.

Figure 9 evaluates these constraints using two defects. The state-transition defect is the difference between the auxiliary state at the later time and the state obtained by propagating the preceding auxiliary state through one RK4 step. The kinematic defect is the difference between the position and heading rates along the auxiliary trajectory and the rates implied by the velocity-based kinematic relation. Smaller values indicate closer agreement with the corresponding physical relations. Figure 9a,b show paired comparisons of the two metrics across the 100 missions, respectively. Each point is the paired percentage reduction for one complete trajectory. Blue circles and orange triangles denote Circle and S-bend missions, respectively, and the black lines show the mean $\pm$ standard deviation across the 100 trajectories. The same missions were reused at all three prediction horizons and therefore do not constitute three independent replicate sets.

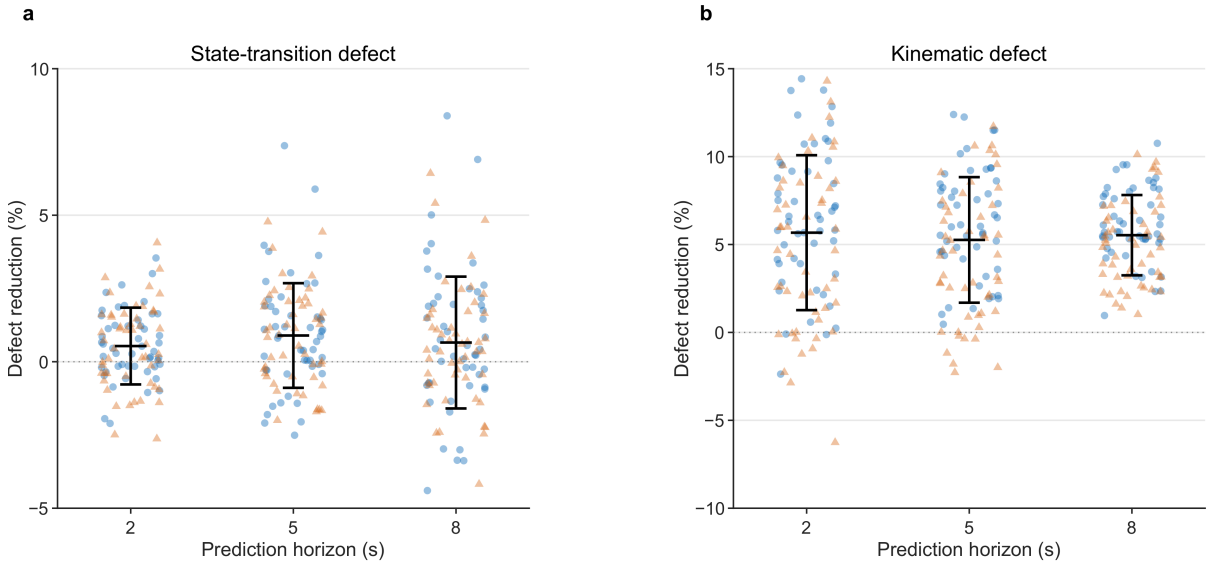


Figure 9. Effects of the additional discrete physics-consistency terms on state-transition and kinematic defects. (**a**) State-transition defect; (**b**) kinematic defect.

With the physics-informed loss included, the mean state-transition defect decreased by 0.53%, 0.89%, and 0.65% at 2, 5, and 8 s, respectively, while the mean kinematic defect decreased by 5.67%, 5.26%, and 5.53%. The main benefit of the physics-informed loss was therefore improved kinematic consistency of the auxiliary trajectory, whereas the improvement in state-transition consistency was smaller. At the same time, the mean ADE and FDE values in Table 4 and Appendix A.3 increased slightly, indicating no additional gain in position prediction accuracy.

These results clarify the roles of the individual components. The PINN-D force and moment correction provides most of the improvement in position prediction, PINN-X supplements the prediction of heading and velocity states, and the physics-informed loss further constrains auxiliary-trajectory consistency. The RK4 dynamics structure contributes directly to prediction generation, whereas the physics-informed loss imposes additional constraints during training; their effects should therefore be evaluated separately.

5.2.3. Prediction Recovery Within Disturbance Regions

The evolution of prediction error after disturbance-region entry was used to examine how event-synchronized residual adaptation corrects local prediction mismatch. The 100 missions in the multiseed evaluation set contained 223 prespecified disturbance-region traversals: 123 Circle traversals and 100 S-bend traversals. Traversals were included solely on the basis of physical entry into a disturbance region, without using the FDE of any predictor as a selection criterion. In Figure 10, N, F, and A denote the nominal dynamics rollout, frozen state–residual PINN predictor, and residual-adapted predictor, respectively. All three configurations use the same event histories and prediction origins.

The upper row of Figure 10 assigns event offset zero to the first leader state packet after region entry. It shows the 2 s FDE at the first 11 packet-based prediction origins, with offsets counted in updates rather than seconds. Traversal errors were first aggregated within each trajectory and then averaged with equal weight across complete trajectories. The shaded bands show the minimum-to-maximum range of the trajectory means, not confidence intervals. The number of evaluable trajectories may differ across event offsets. The lower row gives the complete classification for two observation windows. Sustained recovery requires an observed FDE above 0.07 m followed by three consecutive valid FDE values no greater than 0.07 m. Traversals with low error immediately upon region entry

form a separate category and are not counted as recoveries from high error. A missing value interrupts the consecutive-event criterion, and both unrecovered traversals and traversals with insufficient observations are retained. Because the 223 traversals are nested within 100 complete trajectories, they are not treated as independent statistical samples.

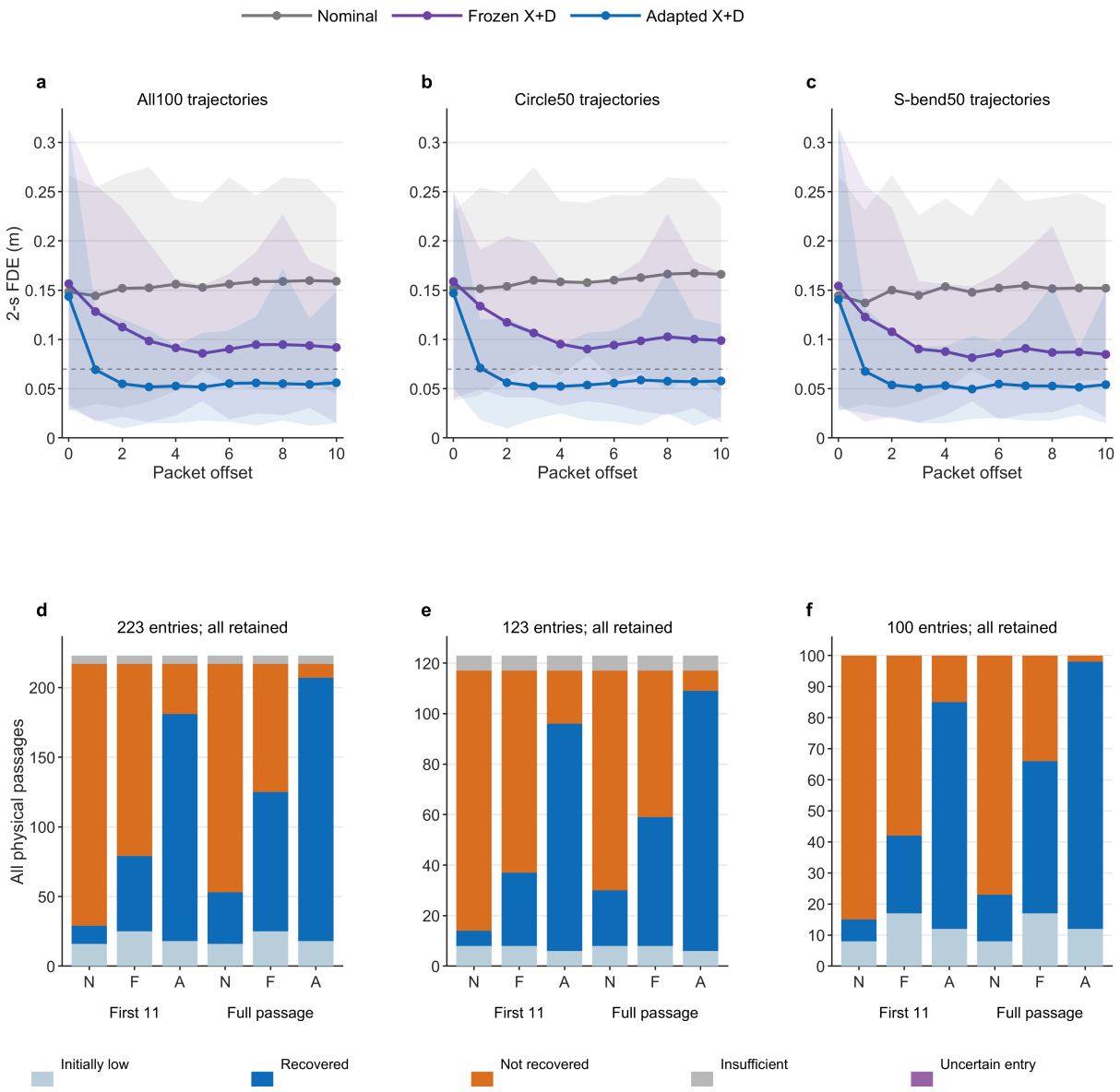


Figure 10. 2 s prediction error and recovery classification after disturbance-region entry. (**a**) Mean 2 s FDE for all 100 missions; (**b**) mean 2 s FDE for the 50 Circle missions; (**c**) mean 2 s FDE for the 50 S-bend missions; (**d**) recovery classification for all 223 disturbance-region traversals; (**e**) recovery classification for the 123 Circle traversals; (**f**) recovery classification for the 100 S-bend traversals.

Figure 10a–c in the upper row show the mean FDE for all 100 missions, the 50 Circle missions, and the 50 S-bend missions, respectively. For both mission types, the nominal dynamics rollout maintains a relatively high mean error. The error of the frozen state–residual PINN predictor decreases gradually but remains above the dashed 0.07 m line. The residual-adapted predictor falls below this line after the first few updates and remains below the other two predictors over the subsequent updates shown.

Figure 10d–f in the lower row summarize all 223 traversals, the 123 Circle traversals, and the 100 S-bend traversals, respectively. In each panel, the left group covers the first 11 packets after region entry, whereas the right group covers the entire disturbance region.

Each bar includes all traversals and distinguishes those with low error at entry, recovery, no recovery, and insufficient observations. Within the first 11 packets, the residual-adapted predictor achieved recovery in 163 traversals, compared with 54 for the frozen state–residual PINN predictor and 13 for the nominal dynamics rollout. The difference between the residual-adapted and frozen predictors appeared in both mission types: 90 versus 29 recoveries for Circle and 73 versus 25 for S-bend. Over the entire disturbance region, the corresponding recovery counts for the residual-adapted and frozen predictors increased to 189 and 100, respectively.

Together, the two rows show that event-synchronized residual adaptation reduces the mean 2 s FDE after disturbance-region entry and enables more traversals to meet the recovery criterion within the first 11 packets.

5.3. Control Response and Continuous Execution with Quintic Connection

Direct replacement switches immediately to the updated reference trajectory, whereas quintic connection introduces it gradually into the current reference trajectory. The two strategies were compared on the 100 archived missions in the multiseed evaluation set, comprising 50 Circle and 50 S-bend missions. Both used the same residual-adapted predictor, event histories, and 500 N/250 N·m control configuration, giving 200 continuous executions in total. Each strategy retained its own follower reference trajectories and controller histories without resetting during execution. The analysis included all missions and all 12,388 updates, including interrupted connections and updates with insufficient observation near the end of a mission. Continuous metrics and same-state local comparison results were first aggregated within each trajectory and then compared with equal weight across the 100 complete trajectories.

Smoothing over the complete mission was measured by the normalized total control variation (TV). This metric sums the absolute changes in surge force and yaw moment between adjacent samples, normalizes them by their respective input limits, and then averages them over the two followers and two control channels. Quintic connection reduced TV in all 100 missions, with a 69.70% reduction in the mean across trajectories and a 95% paired-trajectory bootstrap interval of 68.34–70.88%. To examine the effect of a single reference trajectory update, the follower states, controller memories, and preceding control inputs immediately before each update in the quintic-connection group were copied into two identical simulation branches. Direct replacement was applied in one branch and quintic connection in the other. Both branches recorded the input change between the adjacent samples before and after the update, using the larger value across the two followers for each control channel. The complete-mission TV and same-state local comparisons in Table 5 both show that quintic connection reduces the control input changes caused by reference trajectory updates.

Table 5. Comparison of Direct replacement and quintic connection across 100 missions.

Metric	Direct Replacement	Quintic Connection at Every Update
Full-trajectory continuous execution		
Mean position tracking RMSE (m)	0.25330	0.14889
Median position tracking RMSE (m)	0.04546	0.03254
Mean actual formation position RMSE (m)	0.25988	0.17561
Median actual formation position RMSE (m)	0.05344	0.06320
Mean normalized total control variation (TV)	240.733	72.941
Same-state local updates		
Mean surge force change (N)	112.17	9.01
Mean yaw moment change (N·m)	211.17	12.30

The same-state local comparisons were further grouped according to the two cases shown in Figure 7. The data contained 369 prediction-error updates from 90 missions and 1018 backward seams produced by maximum-interval updates across 100 missions. Results were first aggregated within each trajectory and then compared with equal trajectory weights. Relative to Direct replacement, quintic connection reduced the mean changes in surge force and yaw moment by 91.91% and 89.08%, respectively, for prediction-error updates and by 94.98% and 94.18% for backward seams caused by maximum-interval updates. The same quintic connection therefore reduces control input changes in both reference-update cases without requiring separate activation conditions for the two seam types. Detailed subgroup results are provided in Appendix B.6.

Quintic connection produced a lower position tracking RMSE in most missions, although the median actual formation position RMSE increased. The former is measured relative to the follower reference trajectories generated by each strategy, whereas the latter is measured relative to the ideal formation positions; the difference reflects a trade-off between reference smoothing and formation-position accuracy. Under the common 500 N/250 N·m control configuration, the actual peak surge force and yaw moment were 497.794 N and 248.898 N·m, respectively. Compared with Direct replacement, the proportion of time spent near the input limits decreased from 8.824% to 2.057%. Both groups used the same input governor to limit the actual outputs, so these metrics evaluate continuous execution when reference smoothing is combined with the common control measures. Complete update records, pre-governor input demands, motion peaks, and paired statistics are reported in Appendices B.1–B.5.

5.4. Communication Behavior and Prediction–Execution Responses Under Varying Conditions

This study examines leader trajectory prediction, event-synchronized residual adaptation, and reference trajectory replanning for USV formations under the prescribed confined inter-event timing rule. This section evaluates the communication load, prediction performance, and closed-loop execution of the proposed framework using periodic communication baselines and variations in measurement noise, channel conditions, and task prior information. All experiments use the frozen state–residual PINN predictor and the same event-synchronized residual adaptation rule.

5.4.1. Communication Under Varying Measurement Noise

Table 6 compares periodic communication at 0.5 and 2 s with ET on four missions (two Circle and two S-bend). All six measurement-noise standard deviations were nominal or tripled, with paired runs sharing initial states, true disturbances, and standardized noise realizations. Network weights, adaptation parameters, and the triggering threshold remained unchanged across noise conditions. Within each condition, the measurement sequence and all prediction, adaptation, reference-connection, and controller settings were fixed. Comparing ET with periodic communication at 2 s therefore assesses the effect of adding prediction-error-triggered transmissions to maximum-interval updates.

All entries are equal-weight trajectory means; prediction errors use common-grid rolling 2 s windows. Packet savings use the 0.5 s baseline. Packet counts include six initialization packets; interval and update statistics exclude them. Simultaneous triggers count as prediction-error updates.

Table 6. Communication performance and ET transmission causes under nominal and tripled measurement-noise standard deviations.

Noise Condition	Mission	Strategy	Packet Count	Mean/Median Inter-Event Interval (s)	Packet Reduction Relative to 0.5 s (%)	ADE/FDE (m)	Position Tracking RMSE (m)	Prediction-Error Updates, n (%)	Maximum-Interval Updates, n (%)
Nominal noise	Circle	Periodic communication (0.5 s)	641.0	0.50000/0.50000	0.00	0.04274/0.06717	0.04809	—	—
Nominal noise	Circle	Periodic communication (2 s)	164.0	2.00000/2.00000	74.41	0.05285/0.07190	0.04911	—	—
Nominal noise	Circle	ET	165.0	1.99623/2.00000	74.26	0.05204/0.07194	0.04940	5.0 (3.14)	154.0 (96.86)
3× noise standard deviation	Circle	Periodic communication (0.5 s)	641.0	0.50000/0.50000	0.00	0.09960/0.12505	0.09878	—	—
3× noise standard deviation	Circle	Periodic communication (2 s)	164.0	2.00000/2.00000	74.41	0.14343/0.18511	0.11172	—	—
3× noise standard deviation	Circle	ET	468.0	0.68656/0.57500	26.99	0.15322/0.26603	0.08960	459.0 (99.35)	3.0 (0.65)
Nominal noise	S-bend	Periodic communication (0.5 s)	361.0	0.50000/0.50000	0.00	0.05305/0.09926	0.04554	—	—
Nominal noise	S-bend	Periodic communication (2 s)	94.0	2.00000/2.00000	73.96	0.05876/0.08699	0.04221	—	—
Nominal noise	S-bend	ET	95.5	1.97468/2.00000	73.55	0.05367/0.07879	0.04269	6.0 (6.70)	83.5 (93.30)
3× noise standard deviation	S-bend	Periodic communication (0.5 s)	361.0	0.50000/0.50000	0.00	0.10498/0.15032	0.09386	—	—
3× noise standard deviation	S-bend	Periodic communication (2 s)	94.0	2.00000/2.00000	73.96	0.14165/0.19236	0.08438	—	—
3× noise standard deviation	S-bend	ET	268.5	0.67549/0.56250	25.62	0.15695/0.26894	0.07434	261.5 (99.61)	1.0 (0.39)

Under nominal noise, maximum-interval updates dominated and ET resembled periodic communication at 2 s, with approximately 74% fewer packets than the 0.5 s baseline. Tripled noise shifted transmissions toward prediction-error updates and shortened the mean inter-event interval to approximately 0.68 s. ET then used approximately 26% fewer packets than the 0.5 s baseline and had lower position tracking RMSE but higher ADE/FDE than both periodic baselines. These tests show that the prediction-error condition allows the communication rate to respond to changes in measurement quality. All 12 high-noise cases completed without network retraining or parameter retuning.

5.4.2. Nonideal Channels and Receiver-Side Synchronization

The channel tests evaluate prediction and position tracking when followers update their local states independently under delayed and lossy reception. Each follower separately maintains its own event history, online correction state, and pending candidate. Prediction propagation accounts for packet age using the measurement time and actual arrival time. Table 7 uses a separate set of four missions, comprising two Circle and two S-bend missions, to compare delays of 50, 100, and 200 ms, receiver-side packet-loss rates of 1%, 5%, and 10%, and combined delay-and-loss conditions. The two receivers use independent packet-loss realizations. For each mission and condition, results are first aggregated across the two receivers and then averaged with equal weight across complete trajectories. Hereafter, asynchronous reception refers to differences in packet arrivals and update histories.

Table 7 shows that ADE generally increased with communication delay and packet-loss rate, whereas the position tracking RMSE values for the four missions remained close to their ideal-channel levels. Under the combined condition of 100 ms delay and 5% packet loss, the minimum availability of 2 s predictions was 99.9842%. This measures prediction coverage, not agreement between predictor states.

Table 7. Prediction and tracking under individual and combined channel conditions.

Channel Condition	Circle: 2 s ADE/Position Tracking RMSE (m)	S-Bend: 2 s ADE/Position Tracking RMSE (m)
Ideal channel	0.04994/0.04525	0.05529/0.04365
50 ms delay	0.05081/0.04566	0.05638/0.04391
100 ms delay	0.05169/0.04605	0.05749/0.04407
200 ms delay	0.05349/0.04691	0.05975/0.04478
1% packet loss	0.05042/0.04563	0.05532/0.04351
5% packet loss	0.05252/0.04574	0.05673/0.04345
10% packet loss	0.05565/0.04620	0.06235/0.04501
100 ms delay + 5% packet loss	0.05436/0.04651	0.05895/0.04390

Packet loss causes the two receivers to obtain different leader state packets and therefore to maintain different event histories and online correction states. Across three repeated trials under the combined condition described above, the RMSE of the position prediction difference between the two receivers was 0.03101 m for Circle and 0.02489 m for S-bend. In the test with one missed packet, the next successfully received state packet refreshed the prediction, but the internal states at the two receivers did not return to exact agreement during the observation period. Because a leader state packet does not carry the complete online correction state, the current protocol does not guarantee exact resynchronization after packet loss.

Channel replication tests on another set of four missions completed 35 of 36 executions; one test with 10% packet loss stopped because the reference buffer did not provide sufficient coverage. An ideal-channel reference run was also performed for the previously tested Circle mission. Its position tracking RMSE and peak position tracking error were 0.25199 m and 0.92858 m, respectively.

5.4.3. Communication and Recovery Under Task-Prior Mismatch

To evaluate performance under task-prior mismatch, the four missions from Table 6 were reused with an unshared avoidance maneuver introduced midway through each mission. The maneuver had an amplitude of 1 m and a duration of 16 s, after which the vessel returned to the original plan. The predictor continued to use the original global formation trajectory information and received no advance information about the maneuver. Table 8 compares 2 s prediction, position tracking, and packet counts under the three communication schedules. Mean metrics were aggregated by trajectory, and peak values were taken as the maximum within each mission type.

Under the unshared maneuver, ET reduced state packet transmissions by approximately 74% relative to periodic communication at 0.5 s. For Circle, both the prediction errors and position tracking RMSE under ET lay between those of the two periodic schedules, while its peak position tracking error was lower than both. For S-bend, the three schedules produced similar position tracking RMSE values; ET yielded a slightly lower FDE but a higher peak position tracking error. Of the 12 maneuver–schedule combinations, 11 returned to the prescribed error band during the observation period, defined as a 2 s FDE no greater than 0.07 m sustained for 2 s, while one did not recover. Under ET, three missions entered this error band after approximately 6–10 s. The remaining Circle mission required

173.30 s to enter this error band. Subsequent leader state packets allowed the framework to correct this temporary departure from the planned trajectory, although recovery time and execution performance varied across missions.

Table 8. Communication–performance comparison under task-prior mismatch.

Mission	Strategy	ADE/FDE (m)	Position Tracking RMSE (m)	PeakPosition Tracking Error (m)	Packet Count	Mean Inter-Event Interval (s)	Packet Reduction Relative to 0.5 s (%)
Circle	Periodic communication (0.5 s)	0.06104/0.12822	0.12446	1.04043	641.0	0.50000	0.00
Circle	Periodic communication (2 s)	0.09497/0.17029	0.15452	1.24137	164.0	2.00000	74.41
Circle	ET	0.08222/0.15106	0.13571	0.93913	170.0	1.93366	73.48
S-bend	Periodic communication (0.5 s)	0.05413/0.10231	0.04853	0.17652	361.0	0.50000	0.00
S-bend	Periodic communication (2 s)	0.06241/0.09553	0.04613	0.20559	94.0	2.00000	73.96
S-bend	ET	0.05948/0.09295	0.04758	0.22721	95.0	1.97837	73.68

5.5. Parameter Sensitivity and Computational Cost

5.5.1. Adaptation Parameters and Connection Duration

Seven parameter configurations for event-synchronized residual adaptation were compared on one Circle and one S-bend development mission. Each configuration changed one of the following: the FB/EP fusion weights, the position weight β_{EP} in the endpoint error feedback, or the stable/burst update gains. All other parameters, predictor weights, and leader state packet sequences remained unchanged. The baseline settings were 0.40/0.60, 0.50, and 0.300/0.800, respectively. FB and EP denote the force-balance correction estimate and endpoint error feedback. Table 9 reports the 2 s ADE and FDE.

Table 9. One-factor sensitivity of residual adaptation parameters.

Configuration	Circle ADE (m)	Circle FDE (m)	S-Bend ADE (m)	S-Bend FDE (m)
Baseline configuration	0.03119	0.04350	0.03227	0.04657
FB/EP 0.20/0.80	0.03156	0.04482	0.03387	0.05186
FB/EP 0.60/0.40	0.03149	0.04444	0.03154	0.04407
βEP 0.25	0.03145	0.04446	0.03253	0.04745
βEP 0.75	0.03139	0.04392	0.03319	0.04941
Gain 0.225/0.600	0.03164	0.04489	0.03247	0.04719
Gain 0.3375/0.900	0.03146	0.04425	0.03313	0.04955

Relative to the baseline, the maximum increases in ADE were 1.43% for Circle and 4.95% for S-bend, while the maximum increase in S-bend FDE was 11.36%. Across the parameter ranges tested, ADE changed only modestly, whereas FDE was more sensitive to parameter variation.

To examine sensitivity to the quintic connection duration, durations of 1.2, 1.8 (baseline), and 1.95 s were compared on the same two development missions. All groups used the same quintic connection method, with all other settings held fixed. The results are reported in Table 10.

Table 10. Sensitivity to quintic connection duration.

Mission	Connection Duration (s)	Position Tracking RMSE (m)	RMSE of Actual Leader–Follower Distances (m)	Actual Formation Position RMSE (m)	Normalized Total Control Variation
Circle	1.20	0.03393	0.03267	0.05334	127.57
Circle	1.80	0.03284	0.03488	0.05690	88.58
Circle	1.95	0.03254	0.03554	0.05788	81.64
S-bend	1.20	0.03611	0.03346	0.05502	75.64
S-bend	1.80	0.03465	0.03683	0.06039	49.20
S-bend	1.95	0.03440	0.03787	0.06200	44.76

Increasing the connection duration reduced the normalized total control variation and slightly lowered the position tracking RMSE, but increased both the RMSE of actual leader–follower distances and the actual formation position RMSE. All three durations maintained similar tracking performance; the main difference was the trade-off between control smoothness and actual formation accuracy. The 1.8 s duration produced less control variation than 1.2 s and lower actual formation errors than 1.95 s, and was therefore adopted as the compromise setting.

5.5.2. Runtime

The 2 s short-horizon implementation with quintic connection at every update was tested on an AMD Ryzen 9 8940HX CPU using MATLAB R2024b Update 8, with the connection duration fixed at 1.8 s. The model generated only a 2 s leader trajectory prediction. The terminal extension required for control preview and quintic connection propagated the terminal velocity and yaw rate and was not treated as additional model prediction. Both development missions completed continuous execution in simulations that did not account for update computation delays. The position tracking RMSE was 0.03295 m for Circle and 0.03462 m for S-bend.

Table 11 reports the post-warm-up mean, 95th percentile (P95), and measured maximum runtime. Each of the 248 updates from the two missions was timed twice. Complete event processing includes adaptation, prediction, and reference trajectory processing. The adaptation timing includes a separate 2.05 s dynamics rollout for checking candidate online correction states, distinct from the 2 s leader trajectory prediction. The serial timing in Table 11 excludes packet transmission and the follower control loops.

Table 11 shows that leader trajectory prediction and event-synchronized residual adaptation account for most of the computational cost, whereas quintic connection has a mean runtime of 1.32 ms. The mean, P95, and measured maximum runtime for complete event processing are 52.87, 77.13, and 112.01 ms, respectively. Of the 496 calls, 237 exceeded the 50 ms monitoring step. The current serial implementation therefore cannot guarantee completion of a full update within a single monitoring step.

Appendix C examines computation-induced waiting using a standalone asynchronous prototype that was not integrated into the main program or used in the primary experiments.

Table 11. Post-warm-up runtime for 2 s prediction with quintic connection at every update.

Operation	Number of Calls	Mean (ms)	P95 (ms)	Measured Maximum (ms)	Calls Exceeding 50 ms
Paired PINN-X and PINN-D calls	500	1.367	1.692	2.779	0
2 s leader trajectory prediction generation	496	34.132	43.139	55.256	7
Event-synchronized residual adaptation (including candidate generation and acceptance check)	496	16.763	37.927	56.562	4
Reference trajectory generation (including terminal extension)	496	0.113	0.170	0.636	0
Quintic connection processing	496	1.324	1.854	2.485	0
Quintic polynomial construction	496	0.084	0.119	0.818	0
Complete event processing	496	52.866	77.129	112.013	237

5.6. Overall Discussion and Scope of Applicability

The experiments show that the PINN-D force and moment correction and event-synchronized residual adaptation improve 2 s leader trajectory prediction, while quintic connection at every update reduces control input changes. Reference smoothing must be balanced against actual formation position accuracy, and the communication savings and recovery from maneuvers depend on the noise, channel, and mission conditions.

These conclusions apply to the Circle and S-bend missions with preshared global formation trajectory information and to the tested speed, noise, and channel conditions. Spatially varying disturbances were represented by equivalent loads and were not calibrated against measured wind, wave, or current data. The planar 3-DOF USV model does not describe heave, roll, or pitch; when these motions cannot be neglected, a higher-order model and physical-USV validation are required. Appendix E provides conditional local boundedness results, while the simulations in this chapter evaluate prediction and execution under the tested conditions. Neither establishes global closed-loop stability or general adaptation convergence. The present timing results do not establish a 50 ms real-time guarantee for the complete system.

6. Conclusions

This paper proposes an event-domain prediction–adaptation–replanning framework for formation trajectory generation under intermittent, noisy leader state updates. In a formation of three underactuated 3-DOF USVs comprising one leader and two followers, the framework uses irregularly transmitted leader state packets to generate follower reference trajectories that are propagated locally between packet arrivals. In the state–residual PINN predictor, PINN-X first constructs an auxiliary state trajectory from the nominal trajectory. PINN-D then estimates the surge force, sway force, and yaw moment omitted by the nominal 3-DOF model along this auxiliary trajectory, providing the PINN-D force and moment correction. Incorporating this correction into the dynamics rollout produces the final leader trajectory prediction. When a new state packet arrives, event-synchronized residual adaptation updates the six-dimensional online correction state to correct the prediction over the next inter-

event interval. At each valid reference update, a fixed-duration quintic connection gradually joins the current reference trajectory to the updated reference trajectory. It smooths reference updates but not solving a dynamics- or actuator-constrained planning problem.

The simulations used the multiseed evaluation set of 50 Circle and 50 S-bend test missions. Without online adaptation, the frozen state–residual PINN predictor reduced the 0–2 s ADE and 2 s FDE relative to the nominal dynamics rollout. Component analysis showed that the PINN-D force and moment correction provided most of the improvement in position prediction. PINN-X improved the prediction of selected state components, particularly heading, sway velocity, and yaw rate, while the additional discrete physics-consistency terms improved auxiliary trajectory consistency. With residual adaptation, the residual-adapted predictor further reduced ADE and FDE by 17.78% and 31.86%, respectively, relative to the frozen predictor. Across all 223 prespecified disturbance-region traversals, the residual-adapted predictor achieved 189 recoveries over the complete disturbance region, compared with 100 for the frozen state–residual PINN predictor. Relative to Direct replacement, quintic connection reduced normalized total control variation in all 100 missions by 69.70% on average across trajectories. Tests with increased measurement noise, nonideal channels, and task-prior mismatch showed that the balance among transmission load, prediction accuracy, and tracking performance changed with the operating condition. The nonideal-channel results describe prediction and tracking with locally differing event histories, without establishing exact predictor resynchronization after packet loss. The reported improvements are empirical results for these simulation conditions, not guarantees of closed-loop stability or adaptation convergence.

Appendix E establishes a local bound on formation position error near straight-line and constant-turn equilibria under the stated assumptions. It also bounds the online correction state under the specified adaptation conditions, but does not establish convergence of the adaptation algorithm. Future work will integrate asynchronous prediction and adaptation into the main implementation and validate the framework on physical USVs under broader operating and communication conditions.

Author Contributions: Conceptualization, Y.Y.; methodology, Y.Y.; software, Y.Y. and J.W.; validation, Y.Y., Z.Y. (Zhenyu Yuan) and Y.H.; formal analysis, Y.Y., Y.H. and Z.Y. (Zhenxia Ye); investigation, Y.Y. and J.W.; resources, Y.Y.; data curation, Y.Y. and Z.Y. (Zhenyu Yuan); writing—original draft preparation, Y.Y.; writing—review and editing, Y.Y., Z.Y. (Zhenyu Yuan), Y.H., J.W., Z.Y. (Zhenxia Ye) and G.T.; visualization, Y.Y. and J.W.; supervision, G.T.; project administration, Y.Y. and G.T.; funding acquisition, G.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Interdisciplinary Research Program of HUST, grant number 2024JCYJ027.

Data Availability Statement: The data presented in this paper are available after contacting the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Baseline Predictor Designs and Extended-Horizon Prediction Results

Appendix A.1. Data-Driven GRU Baseline

The GRU uses the six most recent leader state packets and does not access actual states or control inputs after the prediction origin. Each packet contributes an eight-dimensional feature vector comprising relative position, the sine and cosine of relative heading, three velocity components, and relative packet time. This vector is concatenated with the six-dimensional public mission descriptor to form a 14-dimensional input; position and heading are expressed in the latest body-fixed frame. The encoder contains 96 GRU units. Its final hidden state is concatenated with the local desired state from the public plan, the feedforward command, and the normalized query time. The resulting vector passes through a 64-unit fully connected layer, a tanh activation, and a six-dimensional output layer to generate the correction. The correction is multiplied by $s^2/(1+s^2)$ and added to a constant-velocity continuation from the latest measured state, ensuring agreement with the measurement at $s = 0$. The GRU uses neither the USV dynamics nor PINN-D. Training and validation use 200 and 40 missions, respectively, corresponding to 10,282 and 1998 windows. The initial curriculum training runs for 14 epochs, followed by up to 24 continuation epochs with a learning rate of 2.5×10^{-4} and a batch size of 256. Training stops if the prespecified improvement criterion is not met for eight consecutive epochs.

Appendix A.2. Six-State EKF Baseline

The six-state EKF uses the state $\boldsymbol{z} = [x, y, \psi, u, v, r]^{\mathsf{T}}$ and observation matrix $\boldsymbol{I}_6$. It propagates the nominal 3-DOF USV model using the same public leader policy and RK4 scheme, without a learned residual or online correction state. Measurement updates occur only when leader state packets arrive, and the posterior state and covariance are retained between packets.

The measurement covariance is $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{\sigma}^2)$, with the standard deviations listed in Table 1 and the heading standard deviation converted to radians. The first packet measurement initializes the state, and $\boldsymbol{P}_0 = \boldsymbol{R}$. The process-noise covariance rate is fixed at $\boldsymbol{Q} = (1000/10)\boldsymbol{R}$, and $\boldsymbol{Q}\Delta t$ is added during propagation. This setting retains the pre-evaluation value Qscale = 1000, with no tuning on the evaluation set.

The covariance-propagation substep is at most 0.05 s. Jacobians are computed by central finite differences with perturbation $10^{-5}\max(1, |z_j|)$. The heading innovation uses the shortest equivalent angular difference, and the covariance is updated in Joseph form and symmetrized. Starting from the posterior state at the common prediction origin, the EKF prediction advances with a 0.025 s step and does not use subsequent measurements.

Appendix A.3. Displacement Errors over Extended Prediction Horizons

Table A1 uses the same 100 missions. The 5 and 8 s evaluations contain 12,238 and 12,086 complete prediction segments, respectively. ADE is the mean position error over the corresponding horizon, whereas FDE is the terminal position error. Segment-level errors are first aggregated within each trajectory, and the 95% CIs are computed by bootstrapping complete trajectories as described in Section 4.3. The different prediction horizons do not increase the number of independent missions.

Table A1. Prediction performance over extended 5 and 8 s horizons with shared event histories.

Prediction Horizon (s)	Predictor Configuration	ADE (m) Mean [95% CI]	FDE (m) Mean [95% CI]
5	Nominal dynamics rollout	0.16812 [0.16147, 0.17468]	0.34859 [0.33201, 0.36595]
5	PINN-X only	0.11632 [0.11097, 0.12191]	0.20280 [0.19005, 0.21833]
5	PINN-D only	0.10426 [0.09914, 0.10936]	0.21064 [0.19790, 0.22463]
5	Frozen state–residual PINN predictor	0.10393 [0.09872, 0.10907]	0.21136 [0.19833, 0.22565]
5	Frozen state–residual PINN predictor (additional discrete physics-consistency terms disabled)	0.10379 [0.09858, 0.10894]	0.21098 [0.19797, 0.22524]
5	PINN-D only + residual adaptation	0.06810 [0.06524, 0.07150]	0.13022 [0.12219, 0.14051]
5	Residual-adapted predictor	0.06778 [0.06490, 0.07127]	0.13019 [0.12183, 0.14113]
5	GRU	0.11261 [0.09669, 0.13634]	0.16433 [0.13295, 0.21122]
5	Six-state EKF	0.16897 [0.16223, 0.17562]	0.34929 [0.33266, 0.36677]
8	Nominal dynamics rollout	0.26048 [0.24730, 0.27447]	0.46712 [0.43440, 0.50508]
8	PINN-X only	0.16366 [0.15343, 0.17577]	0.31959 [0.29270, 0.35361]
8	PINN-D only	0.16253 [0.15259, 0.17433]	0.30086 [0.27496, 0.33513]
8	Frozen state–residual PINN predictor	0.16395 [0.15368, 0.17592]	0.30977 [0.28352, 0.34460]
8	Frozen state–residual PINN predictor (additional discrete physics-consistency terms disabled)	0.16367 [0.15340, 0.17562]	0.30914 [0.28292, 0.34394]
8	PINN-D only + residual adaptation	0.10725 [0.10043, 0.11622]	0.21509 [0.19454, 0.24360]
8	Residual-adapted predictor	0.10797 [0.10067, 0.11778]	0.22096 [0.19794, 0.25291]
8	GRU	0.15081 [0.12403, 0.19043]	0.29723 [0.23903, 0.38224]
8	Six-state EKF	0.26114 [0.24801, 0.27516]	0.46715 [0.43460, 0.50439]

Residual adaptation reduced displacement errors at both extended horizons. PINN-X and the additional discrete physics-consistency terms, however, did not improve position accuracy at every horizon.

Appendix B. Complete Records and Extended Results for Reference Trajectory Connection

This appendix reports the complete execution results for the 100 missions analyzed in Section 5.3. Both strategies use the same residual-adapted predictor, leader state packet sequence, and control configuration with limits of 500 N and 250 N·m.

Appendix B.1. All Updates and Connection Status

Table A2 includes all 12,388 updates without filtering by connection outcome. Completion, interruption, and insufficient observation at mission end are mutually exclusive categories; forward and backward seams are classified from the pre-update geometry in the quintic-connection run.

An interruption denotes a restarted connection following the arrival of a new packet. Each local pair shares the pre-update follower states, controller memory, and previous control inputs, and no subsequent packets are received during the observation window; continuous execution, by contrast, processes all subsequent packets. Records with less than 2 s remaining at mission end are retained over their available duration. Appendix B.5 describes the aggregation procedure.

Table A2. Connection status for all reference trajectory updates.

Statistic	Count
All post-initialization updates/quintic-connection calls	12,388
Reached the scheduled connection endpoint	12,057
Interrupted and restarted upon arrival of a subsequent leader state packet	245
Insufficient observation at mission end	86
Insufficient cache coverage; current reference trajectory retained	0
Forward seams	11,257
Backward seams	1131
Same-state local pairs with a complete 2 s observation window	12,288
Same-state local pairs with less than 2 s remaining at mission end	100
Local pairs with simultaneous reductions in surge force and yaw moment changes	12,062
Local pairs without simultaneous reductions in both channels	326

Appendix B.2. Execution Results by Mission Type

Table A3 reports continuous execution results by mission type.

Table A3. Continuous execution results by mission type.

Mission Type	Strategy	Mean/Median Position Tracking RMSE (m)	Mean/Median Actual Formation Position RMSE (m)
Circle (50 missions)	Direct replacement	0.36648/0.04510	0.37398/0.05416
Circle (50 missions)	Quintic connection	0.24911/0.03202	0.27521/0.06386
S-bend (50 missions)	Direct replacement	0.14012/0.04603	0.14577/0.05321
S-bend (50 missions)	Quintic connection	0.04867/0.03337	0.07601/0.06292

With quintic connection, position tracking RMSE ranged from 0.02684 to 2.27411 m. Circle032 had a peak position tracking error of 9.80860 m, indicating that large tracking errors remained in some missions. All 100 missions are included in the overall statistics.

Appendix B.3. Input Demands and Input Governing

Table A4 distinguishes raw requests, filtered pre-governor demands, and applied inputs. Peaks are the maxima across all missions. Time proportions are first aggregated within each trajectory as described in Appendix B.5 and then averaged with equal trajectory weights. "Near the limit" denotes at least 95% of the corresponding limit, and the longest continuous duration is the maximum for any single USV and input channel.

Table A4. Full-trajectory control input demands and input-governor statistics.

Metric	Direct Replacement	Quintic Connection at Every Update
Peak applied surge force (N)	497.797	497.794
Peak applied yaw moment (N·m)	248.898	248.898
Peak raw surge force request (N)	3223.917	3192.844
Peak raw yaw moment request (N·m)	9106.170	9127.067
Peak filtered surge force demand before the soft governor (N)	2627.245	2602.971
Peak filtered yaw moment demand before the soft governor (N·m)	7309.355	7317.727
Mean proportion of time near the input limits (%)	8.824	2.057
Median proportion of time near the input limits (%)	4.812	0.061
Mean proportion of time above the input limits before governing (%)	9.235	2.117
Mean proportion of time with inputs modified by soft or hard governing (%)	10.637	2.370
Longest continuous time near the input limits (s)	21.20	18.00
Longest continuous time above the input limits before governing (s)	21.20	18.05
Proportion of time under hard clipping after soft governing (%)	0	0

Quintic connection reduced the time spent near the limits and above the pre-governor limits, but it did not reduce the worst-case yaw moment request. Compliance of the applied inputs with the 500 N and 250 N·m limits depends on the common input governor; a zero hard-clipping proportion does not imply that the raw requests never exceeded the limits.

Appendix B.4. Reference-Trajectory and Actual-Motion Diagnostics

Table A5 reports peak reference and actual motion quantities across all missions. Numerical clipping of body-fixed velocities and accelerations is disabled, whereas desired-state saturation, filtering, and input governing are retained. Finite-difference reference acceleration is not actual acceleration, and the reported peaks are not hardware capability limits.

Table A5. Full-trajectory peak reference and actual motions.

Metric	Direct Replacement	Quintic Connection at Every Update
Reference-trajectory planar speed (m/s)	5.7773	1.9635
Reference-trajectory finite-difference acceleration (m/s^2)	91.2169	1.8057
Reference-trajectory discrete curvature (m^{-1})	83.4197	515.2364
Actual planar speed (m/s)	2.5041	2.4987
Actual inertial acceleration (m/s^2)	3.0252	3.1357
Actual surge velocity u (m/s)	2.5023	2.4969
Actual sway velocity v (m/s)	1.3276	1.3268
Actual yaw rate r (rad/s)	1.5211	1.5211
Actual du/dt (m/s^2)	4.0281	3.8585
Actual dv/dt (m/s^2)	1.8661	1.8661
Actual dr/dt (rad/s^2)	3.4719	3.3959

Quintic connection reduced peak reference speed and finite-difference acceleration but increased peak discrete curvature to 515.2364 m^{-1} in Circle044. Reference smoothing therefore does not impose dynamic constraints on every motion quantity.

Appendix B.5. Paired Statistics and Uncertainty

For each input channel at each event, the local comparison first takes the larger change across the two followers and then averages these event values within each trajectory. Time proportions are computed within each trajectory across both followers, both channels, and all samples. Grouped statistics include only trajectories containing the corresponding event type; missing groups are not imputed as zero. Complete trajectories are the statistical units and receive equal weight; events and followers are not treated as independent replicates.

Table A6 uses the stratified paired bootstrap described in Section 4.3, with $n = 100$: 50 Circle and 50 S-bend missions. The procedure resamples complete trajectories 10,000 times while preserving the strategy pairing. Relative reductions are calculated from the ratio of the trajectory-level means for the two strategies. The 95% CIs reflect between-mission variation for the fixed models.

Both total control variation and local input changes decreased, whereas the paired-difference intervals for both position RMSE metrics included zero. Quintic connection tracked its own reference trajectory more accurately in 94 missions but increased actual formation position RMSE in 78. These metrics measure different quantities: the results establish a control-smoothing effect, not a general position-accuracy advantage.

Table A6. Paired effects and uncertainty across all trajectories.

Metric	Mean, Direct Replacement	Mean, Quintic Connection	Mean Difference, Direct Replacement Minus Quintic Connection [95% CI]	Relative Reduction in the Mean [95% CI] (%)
Position tracking RMSE (m)	0.25330	0.14889	0.10441 [−0.01062, 0.22684]	41.22 [−5.65, 70.88]
Actual formation position RMSE (m)	0.25988	0.17561	0.08427 [−0.02965, 0.20585]	32.43 [−15.46, 62.16]
Complete-mission normalized total control variation (TV)	240.7328	72.9409	167.7918 [163.9268, 171.3459]	69.70 [68.34, 70.88]
Local change in surge force (N)	112.1722	9.0104	103.1618 [100.1091, 106.0731]	91.97 [91.26, 92.57]
Local change in yaw moment (N·m)	211.1682	12.3018	198.8665 [196.7275, 200.9706]	94.17 [94.02, 94.32]
Normalized total control variation in the local observation window (TV)	1.4014	0.5627	0.8387 [0.8193, 0.8566]	59.85 [58.36, 61.13]

Appendix B.6. Same-State Local Comparisons for Two Reference-Trajectory Update Cases

Table A7 corresponds to the prediction-error update and the backward seam produced by a maximum-interval update in Figure 7. A backward seam is identified when the pre-update signed along-track step in the quintic-connection run is less than -10^{-10} m. The two strategies share the same pre-update states, and the statistical procedure follows Appendix B.5. The two event types can overlap, so their counts are not additive.

Table A7. Same-state local comparisons for the two update cases in Figure 7.

Group	Missions Containing This Event Type	Updates	Change in Surge Force, Direct Replacement/ Quintic Connection (N)	Change in Yaw Moment, Direct Replacement/ Quintic Connection (N·m)	Reduction in Surge Force/ Yaw Moment Changes (%)
Prediction-error update	90	369	291.8626/23.6215	261.6655/28.5623	91.91/89.08
Backward seam produced by a maximum-interval update	100	1018	239.4510/12.0174	216.4048/12.5938	94.98/94.18

The changes in both channels were reduced in 331/369 and 1008/1018 pairs for the two update cases, respectively. Pairs without simultaneous improvement are retained, and no event-level significance inference is made.

Appendix C. Supplementary Deployment Validation with a Standalone Asynchronous Prototype

This appendix tests whether background computation prevents update processing from blocking control in two development missions. The standalone prototype is not integrated into the main implementation; its hardware, software, frozen model, packet stream, and control configuration match those in Section 5.5.2.

Adaptation and the 2 s leader trajectory prediction run in the background while the main loop continues to execute the current reference trajectory. Once available, the result is advanced from the measurement time to the execution time and incorporated through quintic connection. The model is preloaded and warmed up. A kinematic continuation supplies the reference tail required during the wait and is not counted as a model prediction; the numerical plant advances with a fixed 50 ms step.

Table A8 reports runtimes after warm-up. Main-loop runtime includes reference trajectory processing, follower control, and integration, but excludes background computation.

All 9902 main-loop iterations executed in less than 50 ms, with a maximum runtime of 13.88 ms, and no update was queued. Software-timer jitter increased the maximum actual interval between consecutive calls to 70.88 ms.

Table A8. Runtimes for background updates and the main control loop in the standalone asynchronous prototype.

Operation or Timing Metric	Circle Mean (ms)	Circle P95 (ms)	Circle Measured Maximum (ms)	S-Bend Mean (ms)	S-Bend P95 (ms)	S-Bend Measured Maximum (ms)
Background adaptation and prediction computation	47.766	70.092	95.643	50.248	82.603	112.856
Wall-clock wait from dispatch to result retrieval	75.351	110.586	139.474	77.856	112.568	148.624
Main-loop execution time	1.530	2.720	13.884	1.591	2.572	5.175
Actual interval between consecutive control calls	50.001	61.573	69.530	50.004	61.747	70.881

Table A9 compares the full-mission results for the two execution modes. Follower states are not reset at updates.

Table A9. Continuous-mission results for the standalone asynchronous prototype and zero-wait baseline.

Metric	Circle, Zero Wait	Circle, Background Execution	S-Bend, Zero Wait	S-Bend, Background Execution
Position tracking RMSE (m)	0.03295	0.03354	0.03462	0.03519
Increase in position tracking RMSE relative to zero wait (%)	—	1.76	—	1.65
RMSE of actual leader–follower distances (m)	0.03488	0.03620	0.03664	0.03828
Actual formation position RMSE (m)	0.05673	0.05857	0.06006	0.06260
Peak position tracking error (m)	0.16204	0.16737	0.17775	0.18521

The standalone asynchronous prototype separated update computation from control execution in the two tested missions. All main-loop iterations required less than 50 ms, while position tracking RMSE increased by less than 2% relative to the zero-wait baseline. Thus, complete updates can span multiple monitoring steps in the background without blocking control, while retaining tracking accuracy close to the baseline under the tested conditions. These measurements support multirate software deployment within a 50 ms main-loop computation budget.

Appendix D. Follower Controllers and Execution Settings

The follower USVs use fixed-parameter backstepping tracking controllers. Circle and S-bend missions each use a fixed parameter set shared by Direct replacement and quintic connection.

Appendix D.1. Control Law

Let e_x and e_y denote the body-fixed errors from the follower position to the reference position, and let e_ψ denote the reference-to-actual heading difference wrapped to $[-\pi, \pi)$. Finite differences in the follower reference-trajectory preview give the reference velocities u_0, v_0, and r_0, with the first two expressed in the reference body-fixed frame. The desired quantities are

$$\begin{aligned} u_d &= \mathrm{sat}_{u_{\max}}(u_0 \cos e_\psi - v_0 \sin e_\psi + k_1 e_x), \\ v_d &= \mathrm{sat}_{v_{\max}}(u_0 \sin e_\psi + v_0 \cos e_\psi + k_2 e_y), \\ r_d &= \mathrm{sat}_{r_{\max}}[r_0 + k_4 e_\psi + k_y k_2 e_y + k_v(v - v_d)]. \end{aligned} \tag{A1}$$

sat denotes symmetric saturation. The variable v_d is an internal auxiliary quantity; no sway actuator is used. The raw control input demands are written as

$$\begin{aligned} \tau_u^{\mathrm{raw}} &= m_{11} k_3 (u_d - u) + H_u(\boldsymbol{\nu}), \\ \tau_r^{\mathrm{raw}} &= m_{66} k_5 (r_d - r) + H_r(\boldsymbol{\nu}), \end{aligned} \tag{A2}$$

where $\boldsymbol{\nu} = [u, v, r]^{\mathsf{T}}$, and H_u and H_r are the corresponding Coriolis and damping compensation terms in the nominal model. Both channels then undergo first-order filtering, smooth input governing, and hard saturation, written compactly as

$$\tau_{c,n} = \mathrm{sat}_{L_c}\{G_c[(1-\alpha_c)\tau_{c,n-1} + \alpha_c \tau_{c,n}^{\mathrm{raw}}]\}, \qquad \alpha_c = \frac{h}{T_c + h}. \tag{A3}$$

Here, $c \in \{u, r\}$, $h = 0.05$ s, and $\tau_{c,n-1}$ is the previous applied output, initialized to zero. The governor G_c remains linear for input magnitudes up to $\ell f L_c$, after which it smoothly compresses the input toward the magnitude $f L_c$. The initial state is set from the first available reference pose and desired velocity and is not reset at subsequent region entries. Acceleration feedforward and turn slowdown are disabled.

Appendix D.2. Active Parameters and Input Limits

Table A10 lists the principal parameters. Values are rounded for display, whereas the calculations use the full precision of the frozen configuration.

Leader trajectory generation uses limits of 180 N and 85 N·m, whereas the follower execution tests above use limits of 500 N and 250 N·m. The former generate the leader motion, and the latter evaluate tracking of the updated reference trajectory. These are separate simulation input bounds and do not represent two hardware ratings for the same physical USV. Table 1 reports the corresponding values.

Table A10. Follower controller parameters.

Parameter	Circle	S-Bend
Discrete preview time (s)	0.25	0.85
$(k_1, k_2, k_3, k_4, k_5)$	(8.427, 5.275, 2.724, 1.182, 21.771)	(1.963, 9.882, 3.675, 2.467, 11.447)
(k_y, k_v)	(0.197, −1.175)	(0.662, −0.397)
$(u_{\max}, v_{\max}, r_{\max})$ (m/s, m/s, rad/s)	(3.0, 2.411, 1.5)	(3.0, 0.960, 1.5)
Filter time constants (T_u, T_r) (s)	(0.0140, 0.0123)	(0.0688, 0.0156)
Governor ratios (f, ℓ)	(0.9937, 0.780)	(0.9956, 0.821)

Appendix E. Conditional Boundedness of Formation Position Error Under Straight-Line Motion and Constant-Rate Turning

This appendix establishes a conditional link between upper-layer reference generation and lower-layer control execution. The analysis first bounds the deviations introduced

by leader trajectory prediction, quintic connection, and changes in the final reference trajectory. It then uses the local stability of the implemented controller to derive a disturbed tracking bound and obtains a formation position error bound through the leader–follower reference mapping.

Appendix E.1. Analysis Conditions

Consider the control sampling instants $\vartheta_n = nh_c$, where $h_c = 0.05$ s. The reference speed $V > 0$ and reference yaw rate Ω are constant: $\Omega = 0$ corresponds to straight-line motion, whereas $\Omega \neq 0$ corresponds to constant-rate turning. The controller retains input filtering, the input governor, and the 500 N/250 N·m input limits.

Assume that the prediction cache in use is valid, the age of information is bounded, and the leader position prediction satisfies the pointwise bound $\|\widehat{\boldsymbol{p}}_L - \boldsymbol{p}_L\| \leq \varepsilon_p$. The leader–follower reference mapping $\mathcal{F}_i$ has Lipschitz constant L_i over the relevant operating domain, with its direction determined by the global formation trajectory information. After quintic connection, the difference between the executed and updated reference positions satisfies $\|\boldsymbol{p}_i^{\mathrm{exec}} - \boldsymbol{p}_i^{\mathrm{upd}}\| \leq \varepsilon_{j,i}$, where $\boldsymbol{p}_i^{\mathrm{upd}} = \mathcal{F}_i(\widehat{\boldsymbol{p}}_L)$. These error bounds and the reference-variation conditions below must be verified separately; the confined inter-event timing rule alone does not guarantee them.

In error coordinates that move with the reference, define the normalized eight-dimensional state $\boldsymbol{\zeta}_i$ from the body-fixed position error, heading error, body-fixed velocity, and preceding applied control input, with position scale $s_p = 1$ m. Assume that the nominal sampled-data closed-loop map $\mathcal{G}_i$ has a relative equilibrium $\overline{\boldsymbol{\zeta}}_i$ whose local Jacobian $\boldsymbol{A}_i$ satisfies $\rho(\boldsymbol{A}_i) < 1$, and that desired-value saturation and input-governor compression are inactive at the equilibrium. Denote the steady-state position offset by $\varepsilon_{\mathrm{ss},i} = \|\overline{\boldsymbol{e}}_{p,i}\|$. For the nominal straight-line equilibrium, one may take $\varepsilon_{\mathrm{ss},i} = 0$; for constant-rate turning, this offset is retained. The normalized input $\boldsymbol{w}_{i,n}$ collects deviations, relative to the operating point, in the preview speed, heading variation, and adjacent-sample increments of the final follower reference trajectory, together with model and load deviations. Its admissible magnitude is restricted by the local conditions in the next section.

Under these assumptions, if the initial deviation and input perturbation satisfy the local conditions below, the formation position error is bounded at the control sampling instants.

Appendix E.2. Formation Position Error Bound

Let $\boldsymbol{\xi}_{i,n} = \boldsymbol{\zeta}_{i,n} - \overline{\boldsymbol{\zeta}}_i$. If the local sampled-data map is continuously differentiable and uniformly Lipschitz with respect to its input, then

$$\begin{aligned} \boldsymbol{\xi}_{i,n+1} &= \boldsymbol{A}_i\boldsymbol{\xi}_{i,n} + \boldsymbol{f}_i(\boldsymbol{\xi}_{i,n}) + \boldsymbol{g}_i(\boldsymbol{\xi}_{i,n}, \boldsymbol{w}_{i,n}), \\ \|\boldsymbol{f}_i(\boldsymbol{\xi})\| &= o(\|\boldsymbol{\xi}\|), \quad \|\boldsymbol{g}_i\| \leq b_i^w\|\boldsymbol{w}_i\|. \end{aligned} \tag{A4}$$

Because $\boldsymbol{A}_i$ is Schur stable, there exists $\boldsymbol{P}_i > 0$ satisfying $\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{P}_i\boldsymbol{A}_i - \boldsymbol{P}_i = -\boldsymbol{I}$. Let $m_i = \lambda_{\min}(\boldsymbol{P}_i)$, $p_i = \lambda_{\max}(\boldsymbol{P}_i)$, and $\mathcal{V}_{i,n} = \boldsymbol{\xi}_{i,n}^{\mathsf{T}}\boldsymbol{P}_i\boldsymbol{\xi}_{i,n}$. Restricting the state and input neighborhoods to $\|\boldsymbol{\xi}_i\| \leq \delta_i$ and $\|\boldsymbol{w}_i\| \leq w_i^0$, respectively, the remainder bound and Young's inequality give

$$\begin{aligned} \mathcal{V}_{i,n+1} - \mathcal{V}_{i,n} &\leq -\frac{1}{4}\|\boldsymbol{\xi}_{i,n}\|^2 + C_i\|\boldsymbol{w}_{i,n}\|^2 \\ &\leq -(1-\mu_i)\mathcal{V}_{i,n} + C_i\|\boldsymbol{w}_{i,n}\|^2. \end{aligned} \tag{A5}$$

where $\mu_i = 1 - 1/(4p_i) \in (0,1)$. Suppose that $\|\boldsymbol{w}_{i,n}\| \le \overline{w}_i$ and

$$\mathcal{V}_{i,0} \le m_i\delta_i^2, \qquad \overline{w}_i \le w_i^0, \qquad C_i\overline{w}_i^2 \le (1-\mu_i)m_i\delta_i^2, \tag{A6}$$

Induction on the Lyapunov difference inequality then shows that the sublevel set $\mathcal{V}_i \le m_i\delta_i^2$ is positively invariant, so the local estimates remain valid. Further iteration gives

$$\|\boldsymbol{\xi}_{i,n}\| \le \sqrt{\frac{p_i}{m_i}}\mu_i^{n/2}\|\boldsymbol{\xi}_{i,0}\| + \sqrt{\frac{C_i}{(1-\mu_i)m_i}}\,\overline{w}_i. \tag{A7}$$

Let $\boldsymbol{p}_i^* = \mathcal{F}_i(\boldsymbol{p}_L)$ be the desired formation position corresponding to the true leader position, and define $E_{i,n} = \|\boldsymbol{p}_i(\vartheta_n) - \boldsymbol{p}_i^*(\vartheta_n)\|$. At the same sampling instant, the formation position error is bounded by the control tracking error, the connection deviation, and the mapped effect of the prediction error. Rotation between the body-fixed and inertial frames preserves the position-error norm, so the tracking term satisfies $\|\boldsymbol{p}_i - \boldsymbol{p}_i^{\mathrm{exec}}\| \le \varepsilon_{\mathrm{ss},i} + s_p\|\boldsymbol{\xi}_i\|$. The triangle inequality and the Lipschitz bound on the reference mapping give

$$\begin{aligned} E_{i,n} &\le \|\boldsymbol{p}_i - \boldsymbol{p}_i^{\mathrm{exec}}\| + \|\boldsymbol{p}_i^{\mathrm{exec}} - \boldsymbol{p}_i^{\mathrm{upd}}\| + \|\boldsymbol{p}_i^{\mathrm{upd}} - \boldsymbol{p}_i^*\| \\ &\le \varepsilon_{\mathrm{ss},i} + s_p\sqrt{\frac{p_i}{m_i}}\mu_i^{n/2}\|\boldsymbol{\xi}_{i,0}\| \\ &\quad + s_p\sqrt{\frac{C_i}{(1-\mu_i)m_i}}\,\overline{w}_i + \varepsilon_{j,i} + L_i\varepsilon_p. \end{aligned} \tag{A8}$$

The initial-deviation term decays with the number of samples, and the remaining terms give the ultimate error bound. Applying this bound separately to each follower establishes boundedness of the formation position error.

Appendix E.3. Boundedness of the Online Correction State

During event-synchronized residual adaptation, the network weights remain frozen and the online correction state is updated from newly received leader state packets to compensate for local prediction mismatch. As disturbances, motion states, and measurement noise vary over time, the online correction state may continue to change. The method does not treat this state as an estimate of a fixed set of true physical parameters.

Partition the current affine online correction state into a three-dimensional constant term $\boldsymbol{a}_k^{(0)}$ and a three-dimensional slope term $\boldsymbol{a}_k^{(1)}$. Suppose that the effective inter-event interval used for state propagation satisfies $0 < h_k \le H$ and that the slope-retention factor satisfies $0 \le \lambda_k \le \overline{\lambda} < 1$. Pure propagation is given by

$$\begin{aligned} a_{j,k+1}^{(0),-} &= a_{j,k}^{(0)} + h_k a_{j,k}^{(1)}, \\ a_{j,k+1}^{(1),-} &= \lambda_k a_{j,k}^{(1)}. \end{aligned} \tag{A9}$$

where $j = 1,2,3$. Every candidate, whether committed directly or held for validation, undergoes finite-value checks and clipping; its constant and slope terms are bounded by B_j and G_j, respectively.

Let $c_a = H/(1-\overline{\lambda})$ and define $W_j(\boldsymbol{a}) = |a_j^{(0)}| + c_a|a_j^{(1)}|$. For a pure propagation step,

$$\begin{aligned} W_j(\boldsymbol{a}_{k+1}^-) &\le |a_{j,k}^{(0)}| + (h_k + c_a\lambda_k)|a_{j,k}^{(1)}| \\ &\le |a_{j,k}^{(0)}| + c_a|a_{j,k}^{(1)}| = W_j(\boldsymbol{a}_k). \end{aligned} \tag{A10}$$

When a candidate is committed, W_j is placed within $B_j + c_a G_j$. Propagation under the implemented rules, both before and after validation, does not increase this bound. If validation rejects a candidate, no new candidate is generated, or the safety checks reject an update, the retained propagated state satisfies the same propagation inequality. Therefore, if the initial state satisfies $W_j(\boldsymbol{a}_0) \le B_j + c_a G_j$ and $|a_{j,0}^{(1)}| \le G_j$, induction gives

$$|a_{j,k}^{(0)}| + c_a |a_{j,k}^{(1)}| \le B_j + c_a G_j, \qquad |a_{j,k}^{(1)}| \le G_j, \quad \forall k. \tag{A11}$$

Zero initialization satisfies these initial conditions. For a receiver subject to packet loss, H must bound the effective update interval at that receiver; the maximum inter-event interval at the sender cannot be used directly.

This state bound covers candidate acceptance, candidate rejection, and the absence of a new candidate. It neither requires the state to stop changing nor implies asymptotic convergence, identification of true physical parameters, or convergence of the prediction error to zero.

Appendix E.4. Theoretical Result and Scope of Applicability

Under the stated conditions, the upper layer provides a reference satisfying the prescribed bounds, and the controller tracks it near the corresponding straight-line or constant-turn equilibrium. Together, these properties give an upper bound on the formation position error. The result is limited to the local region specified in Appendix E.2. Verification at discrete operating points does not certify an entire range of turn rates or establish global stability under arbitrary curvature or sustained saturation.

The theoretical objective is to establish bounded formation position error under time-varying mismatch and intermittent communication; it does not require the online correction state to converge to a fixed value. The derivation in Appendix E.2 permits the reference and model deviations to remain time varying. The tracking error remains bounded provided that the reference-error bounds, local control conditions, and bounds on the initial deviation and input magnitude all hold; asymptotic convergence of $\boldsymbol{a}_k$ is not required. For this purpose, the state bound in Appendix E.3 rules out unbounded accumulation of the online correction state and, together with the other conditions, supports the stated result without imposing convergence of that state as an additional assumption. Boundedness of the online correction state alone guarantees neither prediction accuracy nor global closed-loop stability.

References

1. Liu, Z.; Zhang, Y.; Yu, X.; Yuan, C. Unmanned surface vehicles: An overview of developments and challenges. *Annu. Rev. Control* **2016**, *41*, 71–93. [CrossRef]
2. Zereik, E.; Bibuli, M.; Mišković, N.; Ridao, P.; Pascoal, A. Challenges and future trends in marine robotics. *Annu. Rev. Control* **2018**, *46*, 350–368. [CrossRef]
3. Peng, Z.; Wang, J.; Wang, D.; Han, Q.L. An Overview of Recent Advances in Coordinated Control of Multiple Autonomous Surface Vehicles. *IEEE Trans. Ind. Inform.* **2021**, *17*, 732–745. [CrossRef]
4. Peng, Z.; Wang, D.; Li, T.; Han, M. Output-Feedback Cooperative Formation Maneuvering of Autonomous Surface Vehicles with Connectivity Preservation and Collision Avoidance. *IEEE Trans. Cybern.* **2020**, *50*, 2527–2535. [CrossRef]
5. Zheng, H.; Liu, C. An overview of Unmanned Surface Vehicles: Methods, practices, and applications. *Control Eng. Pract.* **2025**, *164*, 106479. [CrossRef]
6. Lu, Y.; Zhang, G.; Sun, Z.; Zhang, W. Adaptive cooperative formation control of autonomous surface vessels with uncertain dynamics and external disturbances. *Ocean Eng.* **2018**, *167*, 36–44. [CrossRef]
7. Yuan, P.; Zhang, Z.; Li, Y.; Cui, J. Leader-follower control and APF for Multi-USV coordination and obstacle avoidance. *Ocean Eng.* **2024**, *313*, 119487. [CrossRef]

8. Mai, Q.; Zeng, J.; Tian, X.; Yu, Z.; Liu, H. Optimal formation control of underactuated multi-AUVs based on artificial potential field and prescribed-time reinforcement learning with intermittent communication. *Ocean Eng.* **2026**, *355*, 125203. [CrossRef]
9. Yang, D.; Ren, W.; Liu, X.; Chen, W. Decentralized event-triggered consensus for linear multi-agent systems under general directed graphs. *Automatica* **2016**, *69*, 242–249. [CrossRef]
10. Cheng, B.; Wu, Z.; Li, Z. Distributed Edge-Based Event-Triggered Formation Control. *IEEE Trans. Cybern.* **2021**, *51*, 1241–1252. [CrossRef]
11. Dai, S.; Wu, Z.; Zhang, P.; Tan, M.; Yu, J. Distributed Formation Control for a Multirobotic Fish System with Model-Based Event-Triggered Communication Mechanism. *IEEE Trans. Ind. Electron.* **2023**, *70*, 11433–11442. [CrossRef]
12. Zhou, B.; Huang, B.; Mao, L.; Zhu, C.; Su, Y. Distributed observer based event-triggered affine formation maneuver control for underactuated surface vessels with positive minimum inter-event times. *Int. J. Robust Nonlinear Control* **2022**, *32*, 7712–7732. [CrossRef]
13. Liu, T.; Li, J.; Zhang, G.; Su, Z.; Wang, X. Distributed observer position-based event-triggered formation control for unmanned surface vessels with confined inter-event times. *Ocean Eng.* **2024**, *296*, 116884. [CrossRef]
14. Huang, C.; Zhou, B.; Hao, Y.; Li, J. Distributed switching formation maneuver control for multiple autonomous underwater vehicles with interval extended edge-based event-triggered communications. *Ocean Eng.* **2026**, *357*, 125569. [CrossRef]
15. Cai, W.; Ju, Y.; Liu, Z.; Zhang, M.; Lv, S. Event-triggered leader-follower communication based formation control with dual-predictor for multiple AUVs. *Ocean Eng.* **2025**, *334*, 121552. [CrossRef]
16. Ding, M.; Wang, L.; Wang, Y.; Hu, J.; Yang, P. Hydrodynamics identification of a dynamic positioning vessel with uncertainties: A rapid online physics-informed approach. *Ocean Eng.* **2025**, *342*, 122972. [CrossRef]
17. Raissi, M.; Perdikaris, P.; Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
18. Xu, P.F.; Han, C.B.; Cheng, H.X.; Cheng, C.; Ge, T. A Physics-Informed Neural Network for the Prediction of Unmanned Surface Vehicle Dynamics. *J. Mar. Sci. Eng.* **2022**, *10*, 148. [CrossRef]
19. Cheng, Y.; Yu, J.; Fan, H.; He, F.; Li, Q.; Huang, Q. Motion Trend Prediction of Unmanned Surface Vessels Based on Physics-Informed Neural Network. In Proceedings of the 2025 4th Conference on Fully Actuated System Theory and Applications (FASTA), Nanjing, China, 4–6 July 2025; pp. 1278–1283. [CrossRef]
20. Tian, K.; Wen, Y.; Wang, S.; Huang, Y.; Huang, L. Physics-Informed Neural Network Model for Trajectory Tracking Control of Unmanned Surface Vehicles. *IEEE Internet Things J.* **2026**, *13*, 16924–16938. [CrossRef]
21. Jiahao, T.Z.; Chee, K.Y.; Hsieh, M.A. Online Dynamics Learning for Predictive Control with an Application to Aerial Robots. In Proceedings of the 6th Conference on Robot Learning, PMLR, Auckland, New Zealand, 14–16 December 2023; Volume 205, pp. 2251–2261.
22. O'Connell, M.; Shi, G.; Shi, X.; Azizzadenesheli, K.; Anandkumar, A.; Yue, Y.; Chung, S.J. Neural-Fly enables rapid learning for agile flight in strong winds. *Sci. Robot.* **2022**, *7*, eabm6597. [CrossRef]
23. Liu, C.; Hu, Q.; Wang, X.; Yin, J. Event-triggered-based nonlinear model predictive control for trajectory tracking of underactuated ship with multi-obstacle avoidance. *Ocean Eng.* **2022**, *253*, 111278. [CrossRef]
24. Dileep, K.; Mija, S.J.; Arun, N.K. Event-triggered MPC design for manipulator trajectory planning using physics-informed neural networks. *Int. J. Intell. Robot. Appl.* **2026**, *10*, 1139–1157. [CrossRef]
25. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: Chichester, UK, 2011. [CrossRef]
26. Chen, Z.; Lai, S.K.; Yang, Z.; Ni, Y.Q.; Yang, Z.; Cheung, K.C. AT-PINN-HC: A refined time-sequential method incorporating hard-constraint strategies for predicting structural behavior under dynamic loads. *Comput. Methods Appl. Mech. Eng.* **2025**, *436*, 117691. [CrossRef]
27. Djeumou, F.; Neary, C.; Goubault, E.; Putot, S.; Topcu, U. Neural Networks with Physics-Informed Architectures and Constraints for Dynamical Systems Modeling. In Proceedings of the 4th Annual Learning for Dynamics and Control Conference, PMLR, Stanford, CA, USA, 23–24 June 2022; Volume 168, pp. 263–277.
28. Wang, S.; Teng, Y.; Perdikaris, P. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *Siam J. Sci. Comput.* **2021**, *43*, A3055–A3081. [CrossRef]
29. Levy, J.; Gibson, J.; Vlahov, B.; Tevere, E.; Theodorou, E.; Fridovich-Keil, D.; Spieler, P. Meta-Learning Online Dynamics Model Adaptation in Off-Road Autonomous Driving. In Proceedings of the Robotics: Science and Systems XXI, Los Angeles, CA, USA, 21–25 June 2025. [CrossRef]
30. Zhao, J.; Cai, C.; Liu, Y. Barrier Lyapunov Function-based adaptive prescribed-time extended state observers design for unmanned surface vehicles subject to unknown disturbances. *Ocean Eng.* **2023**, *270*, 113671. [CrossRef]
31. Qu, X.; Zhang, Q.; Liu, G.; Wang, Y.; Zhang, R. Euclidean distance-based cooperative target encircling of uncrewed surface vehicles in GPS-denied environments. *Ocean Eng.* **2026**, *357*, 125681. [CrossRef]

32. Wang, R.; Liu, K.; Lyu, S.; Liu, Y. Event-triggered global uniform asymptotic stabilization of under-actuated surface vessels with input saturation and actuator faults. *Ocean Eng.* **2026**, *353*, 124724. [CrossRef]
33. Krishnapriyan, A.S.; Gholami, A.; Zhe, S.; Kirby, R.M.; Mahoney, M.W. Characterizing Possible Failure Modes in Physics-Informed Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 26548–26560.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.