Adaptive Model Predictive Control with Online System Identification for an Unmanned Underwater Vehicle

Yang Hu
Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University
Kowloon, Hong Kong, China
yang-h.hu@connect.polyu.hk

Boyang Li
School of Engineering
The University of Newcastle
Callaghan NSW 2308, Australia
boyang.li@newcastle.edu.au

Chih-Yung Wen
Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University
Kowloon, Hong Kong, China
cywen@polyu.edu.hk

OCEANS 2024 - SINGAPORE | 979-8-3503-6207-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/OCEANS51537.2024.10682292

***Abstract*—Unmanned underwater vehicles (UUVs) are increasingly important for various underwater tasks, and achieving autonomy for UUVs is a major focus. This is crucial for enhancing safety, flexibility, extending operational range, and reducing costs. However, designing effective and robust control algorithms for UUVs is challenging due to nonlinear dynamics, uncertainties, constraints, and environmental disturbances. Model predictive control (MPC) is a well-established technique for UUV control, while how to obtain accurate prediction models becomes the key challenge for improving controller performance. This paper proposes an online system identification method based on an extended active observer (EAOB) and the recursive least squares with variable forgetting factor (RLS-VFF) algorithm to estimate environmental disturbances and identify uncertain hydrodynamic parameters. The estimated disturbances and parameters are continuously updated in the MPC's prediction model to generate optimal control inputs based on the real-time environment and vehicle conditions. The proposed approach is validated through the Gazebo and robot operating system (ROS) simulation environment, demonstrating its effectiveness in handling uncertainties and disturbances for UUV control.**

***Index Terms*—Unmanned underwater vehicle, model predictive control, adaptive control, system identification**

I. Introduction

Unmanned underwater vehicles (UUVs) have gained considerable attention and utilization in various demanding underwater tasks. These applications, such as offshore infrastructure inspection [1] [2], geomorphological mapping [3] [4], and underwater operations [5] [6], have demonstrated the effectiveness of UUVs in helping or replacing humans for hazardous and labor-intensive work. Achieving autonomy for UUVs is currently a major development focus, enabling them to make decisions and perform tasks without constant human intervention. In this pursuit, it is crucial to enhance the control capabilities through designing effective and robust control algorithms. The motion control systems design for UUVs poses significant challenges due to the existence of highly nonlinear dynamics, parametric uncertainties, system constraints, and unpredictable external disturbances.

Model predictive control (MPC) has emerged as a powerful technique among control strategies for UUVs [7]. MPC solves an optimal control problem (OCP) iteratively over a finite time horizon, allowing it to handle system constraints and optimize control performance for multiple inputs and multiple outputs (MIMO) systems. This makes it particularly well-suited for marine robotics, as it can handle control limits, control variation bounds, and state constraints effectively. To address the inherent nonlinear dynamics of complex UUV systems, nonlinear model predictive control (NMPC) has been developed as the baseline controller in this work. However, the accuracy of the prediction model greatly influences the performance of MPC. Designing control systems for UUVs poses significant challenges due to parametric uncertainty. This uncertainty arises from the difficulty in precisely identifying hydrodynamic coefficients that capture the complex interactions between the UUV and its surrounding fluid. While previous research has discussed the hydrodynamic properties of UUVs [8] [9], explicitly calculating hydrodynamic coefficients is typically challenging. Additionally, the dynamics of UUVs may change over time, introducing further complications for control tasks. Meanwhile, operating in environments characterized by substantial environmental disturbances, such as unpredictable currents and waves, can also lead to instability and performance degradation of the closed-loop control system. Effectively addressing these environmental disturbances is therefore another critical challenge in UUV control.

Various types of improved MPC have been developed to improve the controller performance. One common method for improving the control system's ability to reject disturbances is to design a disturbance observer (DOB) that estimates and compensates for disturbances. The combination of MPC and DOB has been studied in many research works recently. For instance, a robust MPC based on active disturbance rejection control (ADRC) has been developed in [10], which used a discrete extended state observer (ESO) to estimate the effects

This research was supported by the Research Centre for Unmanned Autonomous Systems, The Hong Kong Polytechnic University.

of model uncertainties and external disturbances. In [11], a nonlinear disturbance observer (NDO) was developed with a distributed MPC to enable safe navigation and motion planning of multiple autonomous surface vehicles (ASVs) in complex coastal environments. A wave predictor has been integrated into the MPC controller, enabling real-time disturbance preview for the controller [12]. Data-driven methods have also been implemented to learn the dynamic residual to provide a data-augmented prediction model for the MPC. For example, a MPC framework with learned residual dynamics using Gaussian Processes has been proposed in [13] to provide an accurate dynamics model for better control performance in high-speed trajectory tracking problems. Additionally, radial basis function neural networks have been employed to compensate for unknown dynamics and disturbances in MPC, enhancing path following performance in surface ships [14]. While these research works have demonstrated enhanced robustness in MPC by addressing external disturbances and model uncertainties, they still have certain limitations. For DOB-based control methods, unmodeled dynamics are simply treated as part of disturbances to be compensated along with external disturbances, which cannot provide an accurate model for MPC. As for data-driven methods, they typically require a lot of data for training, which can lead to expensive operations.

This work builds upon our previous research [15] that utilized an extended active observer (EAOB) to estimate total disturbances, including unmodeled dynamics and environmental disturbances. Instead of treating all sources of disturbances as a single variable in each degree-of-freedom (DOF) using the principle of superposition, this work aims to develop an online system identification module capable of identifying uncertain hydrodynamic parameters based on the overall estimated disturbances provided by the EAOB. To achieve this, a recursive least squares with variable forgetting factor (RLS-VFF) algorithm is employed to iteratively update the estimated parameters in the MPC's prediction model. The RLS-VFF algorithm not only adapts to non-stationary data and time-varying system dynamics but also improves memory efficiency, making the method both feasible and cost-effective. The adjustment of the variable forgetting factor in the algorithm is based on the F-test, which enhances its ability to detect and adapt to changes in the system.

The remaining sections of this paper are structured as follows: Section II presents the dynamic model of the UUV; Section III introduces the incorporation of NMPC with the proposed online system identification module; Section IV showcases the simulation results validated using the robot operating system (ROS) and Gazebo; and Section V summarizes the conclusions and future works of this study.

II. UUV Dynamics Model

The open-frame BlueROV2 vehicle [16] is utilized in this work. The BlueROV2 is capable of actuation in four degrees of freedom (DOFs), including surge, sway, heave, and yaw. For the sake of completeness, the complete six DOFs' dynamic model is presented in this section, but only the disturbances and hydrodynamic parameters in the aforementioned four directions are considered in this work. The overall dynamics model follows Fossen's equations [17], which include both rigid-body dynamics and hydrodynamics. The parameter notations in the UUV's dynamic model are listed in Table I.

TABLE I
NOTATIONS IN THE UUV DYNAMIC MODEL.

	Surge Sway Heave	Roll Pitch Yaw
Position $\boldsymbol{\eta}$	$x\ y\ z$	$\phi\ \theta\ \psi$
Velocity $\boldsymbol{v}$	$u\ v\ w$	$p\ q\ r$
Propulsion Forces and Moments $\boldsymbol{\tau}$	$X\ Y\ Z$	$K\ M\ N$
Control Inputs $\boldsymbol{u}$	$u_1\ u_2\ u_3$	/ / u_4
Disturbances $\boldsymbol{w}$	$X_w\ Y_w\ Z_w$	$K_w\ M_w\ N_w$
Unmodeled dynamics $\boldsymbol{\Delta\tau}$	$\Delta X\ \Delta Y\ \Delta Z$	$\Delta K\ \Delta M\ \Delta N$
Environmental Disturbance $\boldsymbol{\tau}_{\text{env}}$	$X_{\text{env}}\ Y_{\text{env}}\ Z_{\text{env}}$	$K_{\text{env}}\ M_{\text{env}}\ N_{\text{env}}$
Added Mass $\boldsymbol{M_A}$	$X_{\dot{u}}\ Y_{\dot{v}}\ Z_{\dot{w}}$	$K_{\dot{p}}\ M_{\dot{q}}\ N_{\dot{r}}$
Linear Damping $\boldsymbol{D_L}$	$X_u\ Y_v\ Z_w$	$K_p\ M_q\ N_r$
Nonlinear Damping $\boldsymbol{D_{NL}}$	$X_{u\lvert u\rvert}\ Y_{v\lvert v\rvert}\ Z_{w\lvert w\rvert}$	$K_{p\lvert p\rvert}\ M_{q\lvert q\rvert}\ N_{r\lvert r\rvert}$

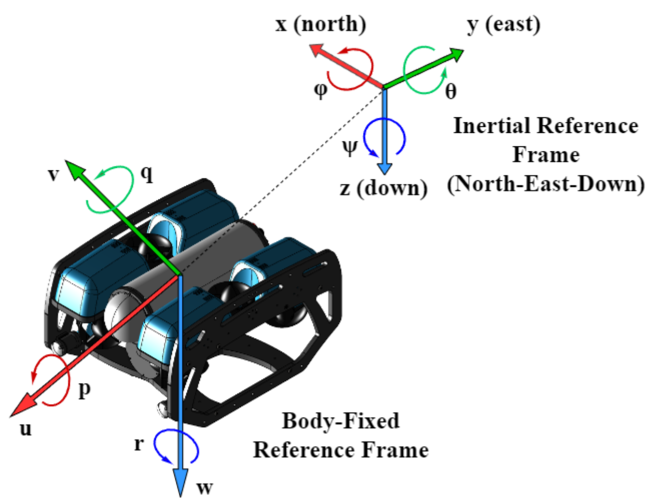

Fig. 1. The body-fixed reference frame and inertial reference frame in the UUV dynamics model.

The system states consist of positions and velocities, denoted as $\boldsymbol{x} = [\boldsymbol{\eta}; \boldsymbol{v}]$. The positions $\boldsymbol{\eta}$ are described in the inertial reference frame (IRF), while the velocities $\boldsymbol{v}$ are described in the body-fixed reference frame (BRF). Figure 1 illustrates the two reference frames, where the IRF adopts the north-east-down (NED) coordinate system in this work. Thus, the nonlinear dynamic equations of the UUV can be described as:

$$\begin{aligned}\dot{\boldsymbol{x}} &= \begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \dot{\boldsymbol{v}} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{v} \\ \boldsymbol{M}^{-1}\left[\boldsymbol{\tau} + \boldsymbol{\tau}_{\text{env}} - \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} - \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} - \boldsymbol{g}(\boldsymbol{\eta})\right] \end{bmatrix}\end{aligned} \tag{1}$$

where $\boldsymbol{M}$ is the sum of the rigid body mass $\boldsymbol{M_{RB}}$ and the hydrodynamic added mass $\boldsymbol{M_A}$, $\boldsymbol{\tau} = K(A\boldsymbol{u})$ represents the combined propulsion forces and moments calculated based on control allocation matrix A and propulsion matrix K, $\boldsymbol{\tau}_{\text{env}}$

denotes environmental disturbances, $\boldsymbol{C}(\boldsymbol{v})$ denotes the rigid-body Coriolis and centripetal forces, $\boldsymbol{D}(\boldsymbol{v})$ represents hydrodynamic damping forces, $\boldsymbol{g}(\boldsymbol{\eta})$ is the hydrostatic restoring forces, and $\boldsymbol{J}(\boldsymbol{\eta})$ represents the rotation matrix to convert linear and angular velocities from body frame to inertial frame. To simplify the dynamic model, three assumptions are made as follows:

1) In many practical applications, such as underwater inspections or slow-speed data collection tasks, the velocities of open-frame UUVs are often less than a few meters per second. In such cases, the effects of Coriolis and centripetal forces caused by added mass are typically considered negligible.
2) The origin of the BRF is positioned at the UUV's geometric center, as well as the center of buoyancy. The UUV exhibits symmetry in both the port–starboard and fore–aft plane.
3) The hydrodynamic damping force is modeled by Fossen damping model [17], which assumes that the UUV is performing noncoupled motions.

Therefore, the dynamics model can be expanded in each DOF as:

$$X = (m - X_{\dot{u}})\dot{u} - mrv + mqw + (-X_u - X_{u|u|}|u|)u + (W - B)\sin\theta - X_{\text{env}}, \tag{2}$$

$$Y = (m - Y_{\dot{v}})\dot{v} + mru - mpw + (-Y_v - y_{v|v|}|v|)v - (W - B)\cos\theta\sin\phi - Y_{\text{env}}, \tag{3}$$

$$Z = (m - Z_{\dot{w}})\dot{w} - mqu + mpv + (-Z_w - Z_{w|w|}|w|)w - (W - B)\cos\theta\cos\phi - Z_{\text{env}}, \tag{4}$$

$$K = (I_x - K_{\dot{p}})\dot{p} + (I_z - I_y)qr + (-K_p - K_{p|p|}|p|)p + z_g W\cos\theta\sin\phi - K_{\text{env}}, \tag{5}$$

$$M = (I_y - M_{\dot{q}})\dot{q} + (I_x - I_z)pr + (-M_q - M_{q|q|}|q|)q + z_g W\sin\theta - M_{\text{env}}, \tag{6}$$

$$N = (I_z - N_{\dot{r}})\dot{r} + (I_y - I_x)pq + (-N_r - N_{r|r|}|r|)r - N_{\text{env}}, \tag{7}$$

where m is the mass of the UUV, I_x, I_y, I_z are inertial moments, W denotes the UUV's weight, B denotes the UUV's buoyancy, and z_g is the distance between the UUV's center of gravity and the origin of the BRF in the z-axis.

While it is possible to calculate the hydrodynamic added mass coefficients of torpedo-shaped UUVs by assuming the vehicle as an ellipsoid, such as the study in [18], it is challenging to directly calculate the added mass coefficients of open-frame UUVs like BlueROV2. Thus, in this work the added mass

$$\boldsymbol{M_A} = -diag[X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}] \tag{8}$$

is estimated by the online system identification. In the meantime, the hydrodynamic damping coefficients are commonly identified through experiment data, which are difficult to be calculated directly. Thus, damping forces with both linear and nonlinear damping coefficients are also treated as variable parameters to be estimated:

$$\begin{aligned} \boldsymbol{D}(\boldsymbol{v}) &= \boldsymbol{D_L} + \boldsymbol{D_{NL}}(\boldsymbol{v}) \\ &= -\text{diag}[X_u, Y_v, Z_w, K_p, M_q, N_r] \\ &\quad - \text{diag}[X_{u|u|}|u|, Y_{v|v|}|v|, Z_{w|w|}|w|, K_{p|p|}|p|, \\ &\qquad M_{q|q|}|q|, N_{r|r|}|r|]. \end{aligned} \tag{9}$$

III. Control Framework

After establishing the nominal UUV dynamics model, the proposed control framework can be achieved by three key steps: 1) The development of an EAOB based on the extended Kalman filter (EKF). This EAOB aims to estimate and provide the total disturbances, encompassing both unmodeled dynamics and external disturbances. These estimated disturbances are subsequently fed into the RLS-VFF algorithm, which constitutes the second step. 2) The RLS-VFF algorithm continuously identifies parameters based on the estimated states and disturbances obtained from the EAOB. 3) The third step focuses on the design and implementation of the adaptive MPC framework. The estimated parameters from the RLS-VFF are updated to the MPC's prediction model to provide an adaptive control law. Thus, the proposed control system can adapt to varying dynamics and unpredictable environmental disturbances, and further enhance the control performance. An optimizer is utilized to solve the OCPs based on a defined cost function. To enable real-time calculation, the OCP is discretized within a prediction horizon and solved using multiple shooting schemes. In this research, the implementation is achieved using the ACADOS open-source software package for real-time control [19].

Figure 2 illustrates the control framework of the adaptive MPC proposed in this work, with the pink box highlights the adaptive mechanism involved in the first and second steps.

A. Observer Design

The EAOB method employed in this research is an enhanced version of the traditional EKF. Unlike most observers that make the simplifying assumption of negligible measurement noise, the EAOB explicitly considers measurement noise as a crucial factor during the estimation process. This realistic approach leads to more accurate state estimation, particularly in underwater scenarios where measurement noise is significant.

The internal disturbance model can be derived from Equation 1 by moving the disturbance term $\boldsymbol{w}$ to the left side. Since $\boldsymbol{w}$ is assumed to be a slow time-varying signal, its derivative is zero. Thus, the disturbance model can be formulated as:

$$\begin{cases} \boldsymbol{w} = \boldsymbol{M_{RB}}\dot{\boldsymbol{v}} + \boldsymbol{C}(v)\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{\eta}) - \boldsymbol{\tau} \\ \dot{\boldsymbol{w}} = 0, \end{cases} \tag{10}$$

where $\boldsymbol{w} = \boldsymbol{\tau}_{\text{env}} + \boldsymbol{\Delta\tau}$ compresses the superposition of both environmental disturbances and unmodeled dynamics.

In order to estimate $\boldsymbol{w}$ by the EKF, $\boldsymbol{w}$ is considered as a system state along with the position $\boldsymbol{\eta}$ and velocity $\boldsymbol{v}$. Thus, the

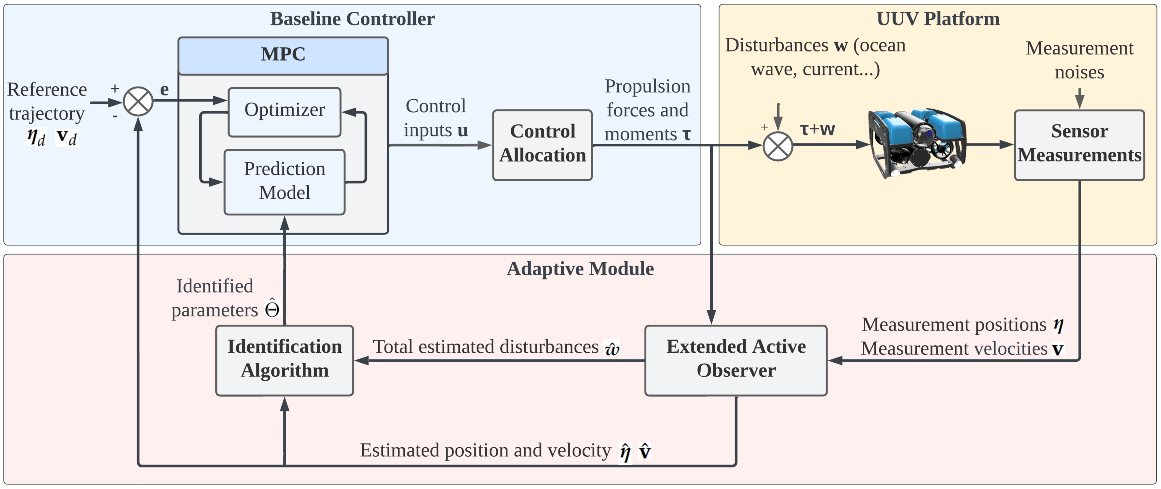

Fig. 2. Block diagram of the proposed adaptive MPC scheme for the UUV, where the blue box illustrates baseline MPC and control allocation, the yellow box shows the UUV platform, and the pink module indicates the adaptive module with an observer and identification algorithm.

system state for the observer is $\boldsymbol{x} = [\boldsymbol{\eta}; \boldsymbol{v}; \boldsymbol{w}]$. Since the $\boldsymbol{w}$ is an unmeasurable state, the measurement state of the observer therefore becomes $\boldsymbol{z} = [\boldsymbol{\eta}; \boldsymbol{v}; \boldsymbol{\tau}]$.

Thus, the system process model can be represented as:

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{\tau}(t)) + W(t) \quad W(t) \sim \mathcal{N}(0, Q(t)) \tag{11}$$

where $f(\boldsymbol{x}(t), \boldsymbol{\tau}(t))$ represents the nonlinear dynamic model, $W(t)$ represents the zero mean Gaussian process noise with covariance $Q(t)$. The system process model is run in continuous time t. The nonlinear function $f(\boldsymbol{x}(t), \boldsymbol{\tau}(t))$ can be derived from the UUV's dynamic model in Equation 1 as:

$$f(\boldsymbol{x}, \boldsymbol{\tau}, \boldsymbol{w}) = \begin{bmatrix} \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{v} \\ (\boldsymbol{M_{RB}})^{-1}[\boldsymbol{\tau} + \boldsymbol{w} - \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} - \boldsymbol{g}(\boldsymbol{\eta})] \\ \dot{\boldsymbol{w}} \end{bmatrix}. \tag{12}$$

Meanwhile, the measurement model $\boldsymbol{z}$ that takes in discrete-time form can be written as:

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + V_k \quad V_k \sim \mathcal{N}(0, R_k) \tag{13}$$

where $h(\boldsymbol{x}_k)$ indicates the nonlinear measurement model, V_k indicates the zero mean Gaussian process noise with covariance R_k. Due to this work is established and validated in the Gazebo simulation platform, it is necessary to further adjust the measurement model of $\boldsymbol{\eta}$ and $\boldsymbol{v}$ based on the specific sensors employed when conducting the real-world experiments. These sensors may include Doppler velocity log (DVL), ultra-short baseline (USBL), barometer, and attitude and heading reference system (AHRS). To estimate $\boldsymbol{w}$, the equation within the measurement model that defines the relationship between $\boldsymbol{w}$ and $\boldsymbol{\tau}$ is:

$$\boldsymbol{\tau} = \boldsymbol{M_{RB}}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{\eta}) - \boldsymbol{w}. \tag{14}$$

Taking the partial derivative of $f(\boldsymbol{x}(t), \boldsymbol{\tau}(t))$ and $h(\boldsymbol{x}_k)$ at point $\boldsymbol{x}_t$ and $\boldsymbol{\tau}_t$, then the linearized state transition matrix $\boldsymbol{F}(t)$ and measurement matrix $\boldsymbol{H}_k$ can be obtained:

$$\boldsymbol{F}(t) = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}(t), \boldsymbol{\tau}(t), \boldsymbol{w}(t)} \tag{15}$$

$$\boldsymbol{H}_k = \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k|k-1}} \tag{16}$$

where $\boldsymbol{F}(t)$ specifies the relationship between the current state and the subsequent predicted state, $\boldsymbol{H}_k$ specifies the relationship between measurement states and the predicted system states.

The EKF estimates the system state recursively using updated measurement state, the system process model, and the measurement model. Prior to the recursive algorithm, the initialization of the system state estimation $\hat{\boldsymbol{x}}$ and the error covariance matrix $\boldsymbol{P}$ is performed based on the initial measurement:

$$\hat{\boldsymbol{x}}(t_0) = E[\boldsymbol{x}(t_0)], \boldsymbol{P}(t_0) = \mathrm{Var}[\boldsymbol{x}(t_0)]. \tag{17}$$

The main process of EKF can be divided into predict part and update part. The predict part of the EKF involves solving these differential equations to obtain the predicted state estimate $\hat{\boldsymbol{x}}_{k|k-1}$ and the predicted error covariance matrix $\boldsymbol{P}_{k|k-1}$ at time t_k, given the previous state estimate $\hat{\boldsymbol{x}}_{k-1|k-1}$ and error covariance matrix $\boldsymbol{P}_{k-1|k-1}$ at time t_{k-1}. The discretization step is necessary in the predict part to obtain discrete-time system dynamics, as measurements are typically taken at discrete time intervals. The numerical integration method used for discretization in this case is the fourth-order

Runge-Kutta (RK4) method. Thus, the predict part can be formulated as:

$$\text{solve} \left\{ \begin{array}{l} \dot{\hat{\boldsymbol{x}}}(t) = f(\hat{\boldsymbol{x}}(t), \boldsymbol{\tau}(t)) \\ \dot{\boldsymbol{P}}(t) = \boldsymbol{F}(t)\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{F}(t)^T + \boldsymbol{Q}(t) \end{array} \right.$$
$$\text{with} \left\{ \begin{array}{l} \hat{\boldsymbol{x}}(t_{k-1}) = \hat{\boldsymbol{x}}_{k-1|k-1} \\ \boldsymbol{P}(t_{k-1}) = \boldsymbol{P}_{k-1|k-1} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \hat{\boldsymbol{x}}_{k|k-1} = \hat{\boldsymbol{x}}(t_k) \\ \boldsymbol{P}_{k|k-1} = \boldsymbol{P}(t_k) \end{array} \right. . \tag{18}$$

The update part of the EKF incorporates new measurement information to refine the state estimate and reduce the uncertainty in the state estimate. The update part involves several steps:

1) Calculate the innovation or measurement residual $\hat{\boldsymbol{y}}_{k|k}$ as the difference between the actual measurement $\boldsymbol{z}_k$ and the predicted measurement $h(\hat{\boldsymbol{x}}_{k|k-1})$.
2) Calculate the Kalman gain $\boldsymbol{K}_k$ using the predicted error covariance matrix $\boldsymbol{P}_{k|k-1}$, the measurement matrix $\boldsymbol{H}_k$, and the measurement noise covariance matrix $\boldsymbol{R}_k$.
3) Update the state estimate $\hat{\boldsymbol{x}}_{k|k-1}$ to obtain the updated state estimate $\hat{\boldsymbol{x}}_{k|k}$ by adding the weighted innovation $\boldsymbol{K}_k\hat{\boldsymbol{y}}_{k|k}$.
4) Update the error covariance matrix $\boldsymbol{P}_{k|k-1}$ to obtain the updated error covariance matrix $\boldsymbol{P}_{k|k}$ using the Kalman gain $\boldsymbol{K}_k$ and the measurement matrix $\boldsymbol{H}_k$.

The above steps can be formulated as:

$$\begin{aligned} &\hat{\boldsymbol{y}}_{k|k} = \left(\boldsymbol{z}_k - h\left(\hat{\boldsymbol{x}}_{k|k-1}\right)\right) \\ &\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T \left(\boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k\right)^{-1} \\ &\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\hat{\boldsymbol{y}}_{k|k} \\ &\boldsymbol{P}_{k|k} = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right)\boldsymbol{P}_{k|k-1}. \end{aligned} \tag{19}$$

Therefore, the state estimation $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{\eta}}; \hat{\boldsymbol{v}}; \hat{\boldsymbol{w}}]$ can be obtained. The stability analysis of the observer has been studied in the previous work [15].

B. Identification Algorithm

The observer's disturbance estimation $\hat{\boldsymbol{w}}$ includes the superposition of the environmental disturbance $\boldsymbol{\tau}_{\text{env}}$ and the unmodeled dynamics $\boldsymbol{\Delta\tau}$. Meanwhile, the unmodeled dynamics encompass the added mass $\boldsymbol{M_A}$ (Equation 8) and linear and nonlinear damping coefficients (Equation 9) as detailed in Section II. Hence, $\hat{\boldsymbol{w}}$ can be expressed as:

$$\begin{aligned} \hat{\boldsymbol{w}} &= \boldsymbol{\tau}_{\text{env}} + \boldsymbol{\Delta\tau} \\ &= \boldsymbol{M_A}\dot{\boldsymbol{v}} + \boldsymbol{D_L}\boldsymbol{v} + \boldsymbol{D}_{\text{NL}}|\boldsymbol{v}|\boldsymbol{v} + \boldsymbol{\tau}_{\text{env}}. \end{aligned} \tag{20}$$

In this section, the surge dynamics in Equation 2 is discussed as an example, while the rest dynamics in Equation 3, 4, and 7 can also be implemented in a similar manner. Therefore, Equation 20 can be reconstructed for the surge dynamics as:

$$\hat{X_w} = \left[\begin{array}{cccc} \dot{u} & u & |u|u & 1 \end{array}\right] \left[\begin{array}{c} X_{\dot{u}} \\ X_u \\ X_{u|u|} \\ X_{\text{ext}} \end{array}\right] \tag{21}$$

where four unknown parameters can be denoted as

$$\Theta = \left[\begin{array}{cccc} X_{\dot{u}} & X_u & X_{u|u|} & X_{\text{ext}} \end{array}\right]^\top. \tag{22}$$

The identification problem involves determining the values of parameters using input/output discrete data while meeting specific goodness-of-fit constraints between predicted data and measurements. Therefore, the Θ can be identified by solving the cost function:

$$\begin{aligned} &\underset{\Theta^\top = [X_{\dot{u}}, X_u, X_{\text{ext}}]}{\arg\min} J(\Theta) = \sqrt{\frac{1}{N}\sum_{k=1}^{k=N}\left[y_k - \Theta^\top\Phi_k\right]^2} \\ &\quad \text{subject to } X_{\dot{u}}, X_u, X_{u|u|}, X_{\text{ext}} \in \Re \end{aligned} \tag{23}$$

where N is the total number of samples available, y_k is the observed output (which is equal to $\hat{X_w}$ in surge dynamics), and

$$\Phi_k = \left[\begin{array}{cccc} \dot{u}_k & u_k & |u|u_k & 1 \end{array}\right]^\top. \tag{24}$$

To find the optimal parameters for a cost function, a common approach is to use the least square (LS) method. LS identifies parameters by minimizing the sum of the squared errors between predicted and observed values. In contrast to LS, which performs regression based on offline collected data, the RLS algorithm operates online. RLS processes data sequentially and updates parameter estimates as new data becomes available. The distinguishing feature of RLS-VFF is the incorporation of a forgetting factor, which enables a trade-off between tracking time-varying parameters and robustness to noise. The forgetting factor determines the weight assigned to past data points relative to recent data points when updating parameter estimates. A forgetting factor close to 1 places more emphasis on past rewards, resulting in low misadjustment but reduced adaptation ability. Conversely, a forgetting factor closer to 0 indicates that the agent prioritizes recent rewards, leading to high adaptation ability but potential susceptibility to outlier data and instability.

In the standard RLS with forgetting factor (RLS-FF) algorithm, the forgetting factor is typically set as a constant value. However, in this research, the forgetting factor is considered as a variable that can be dynamically adjusted based on the outcome of the F-test. The F-test is a statistical test used to compare the variances of two samples. In this context, it is employed to compare the prediction error variance of two windows of past RLS estimation results: a long window and a short window. The purpose of this comparison is to determine whether the variance has increased. The F-test statistic, denoted as F_k, is calculated as follows:

$$F_k = \frac{\sigma_n^2}{\sigma_d^2} = \frac{\frac{1}{n}\sum_{i=k-n}^{n}\left(e_i - \mu_n\right)^2}{\frac{1}{d}\sum_{i=k-d}^{d}\left(e_i - \mu_d\right)^2} \tag{25}$$

where n represents the number of samples in the short window, d represents the number of samples in the long window, σ_n^2 is the prediction error variance with n samples, σ_d^2 is the prediction error variance with d samples, and $d > n \geq 1$. The prediction error e_k can be computed as the difference between the observed output y_k and the parameter estimate at time $k-1$:

$$e_k = y_k - \hat{\Theta}_{k|k-1}^T \Phi_k. \tag{26}$$

By comparing the variance of the prediction errors in the short and long windows, the F-test provides insights into whether the system dynamics have changed significantly. If the F-test statistic exceeds a predefined threshold γ, it indicates that the variance has increased, suggesting a change in the system dynamics. In such cases, the forgetting factor is adjusted to respond to these changes and maintain accurate estimation:

$$\lambda_k = \begin{cases} \lambda_{k-1} + \Delta\lambda & \text{if } F_k < \gamma \\ \lambda_{k-1} - \Delta\lambda & \text{otherwise} \end{cases} \tag{27}$$

where $\Delta\lambda$ represents the adjustment value for the forgetting factor.

This approach of using the F-test to adapt the forgetting factor in RLS-VFF enhances the algorithm's ability to track time-varying system dynamics, resulting in improved estimation performance.

In the initialization stage of the RLS-VFF algorithm, the parameter estimation $\hat{\Theta}$ is initialized to zero. Meanwhile, the initial value of the error covariance matrix P is determined based on the forgetting factor λ:

$$\begin{aligned} \hat{\Theta}_{k=0} &= 0 \\ P_{k=0} &= \frac{1}{\lambda} I \end{aligned} \tag{28}$$

where I is the identity matrix.

The RLS-VFF algorithm operates similarly to the Kalman filters family. At time k, it calculates the Kalman gain K_k using the forgetting factor λ, the error covariance matrix P from time $k-1$, and the regression factor Φ_k. Subsequently, the parameter estimation $\hat{\Theta}_{k|k}$ is updated based on the Kalman gain and the error, and the error covariance matrix $P_{k|k}$ is also updated accordingly. The recursive process can be formulated as:

$$\begin{aligned} K_k &= \frac{P_{k|k-1}\Phi_k}{\lambda + \Phi_k^T P_{k|k-1}\Phi_k} \\ \hat{\Theta}_{k|k} &= \hat{\Theta}_{k|k-1} + K_k e_k \\ P_{k|k} &= \frac{1}{\lambda}\left(P_{k|k-1} - K_k \Phi_k^T P_{k|k-1}\right). \end{aligned} \tag{29}$$

C. Adaptive Model Predictive Control

MPC, also known as receding horizon control, is a widely used control method in various robotic systems, including UUVs. In MPC, a control input sequence, represented as $u^*(s) = \{u_0^*, u_1^*, \ldots, u_{T-1}^*\}$, is computed by solving an online constrained OCP using the current measurements and system constraints at each time step. Subsequently, the first element u_0^* in the control sequence is applied to the system plant.

Solving the OCPs iteratively online can be computationally expensive. To address this issue, an open-source software package called ACADOS is used to generate a fast solver for the nonlinear system [20]. ACADOS can provide efficient optimal control algorithms targeting embedded devices implemented in C, which allows for the real-time deployment of the MPC. The MPC's optimization problem is formulated as:

$$\begin{aligned} \min_{U,X} \int_{t=0}^{T} \|h(x(t),u(t)) - y_{ref}\|_Q^2 \, dt + \|h(x(T)) - y_{N,ref}\|_{Q_N}^2 \\ \text{subject to} \quad \dot{x} = f(x(t),u(t)) \\ u(t) \in \mathbb{U} \\ x(t) \in \mathbb{X} \\ x(0) = x(t_0) \end{aligned} \tag{30}$$

where $h(\cdot)$ represents system output function, $f(\cdot)$ represents system dynamics function, x and u correspond system states and control inputs, T denotes prediction horizon, y_{ref} and $y_{N,ref}$ are reference states and terminal reference states respectively, subsequently, Q and Q_N are state and terminal cost matrices. The control inputs $u(t)$ are constrained to the set of feasible control inputs $\mathbb{U}$, while the states are constrained to $\mathbb{X}$. Additionally, the initial state $x(0)$ is set to the value of the states at time t_0.

The comprehensive adaptive MPC framework is outlined in Algorithm 1.

Algorithm 1 Adaptive MPC with online system identification

Initialization:
  Initialize EAOB, RLS-VFF based on Equation 17, 28
while t≥0 **do**
  Measure $\boldsymbol{z} = [\boldsymbol{\eta}; \boldsymbol{v}; \boldsymbol{\tau}]$
  Estimate $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{\eta}}; \hat{\boldsymbol{v}}; \hat{\boldsymbol{w}}]$ with EAOB by Equation 18, 19
  Calculate F_k by Equation 25
  if $F_k > \gamma$ **then**
    Update forgetting factor $\lambda_k = \lambda_{k-1} - \Delta\lambda$
  else
    Update forgetting factor $\lambda_k = \lambda_{k-1} + \Delta\lambda$
  Estimate $\hat{\Theta} = [X_{\dot{u}} X_u X_{u|u|} X_{\text{ext}}]^T$ with RLS-VFF by Equation 29
  Update MPC control law: $\boldsymbol{\tau} = K(A\boldsymbol{u}) = \boldsymbol{M_{RB}}\dot{\boldsymbol{v}} + C(\boldsymbol{v})\boldsymbol{v} + g(\boldsymbol{\eta}) - \hat{\Theta}^T\Phi$
  Solve the OCP to obtain the optimized control sequence $u^*(s)$ by Equation 30
  Implement the first element u_0^* in the optimized control sequence to the UUV
end while

IV. Results

To perform online system identification, the UUV's motion in each DOF must be captured for training and regression with the RLS-VFF. Therefore, a reference trajectory is specified for training purposes and provided to the MPC before executing other trajectory tracking tasks. The entire training process spans 40 seconds, with the following breakdown: 1) Surge dynamics training: begins at 0 seconds and concludes at 10

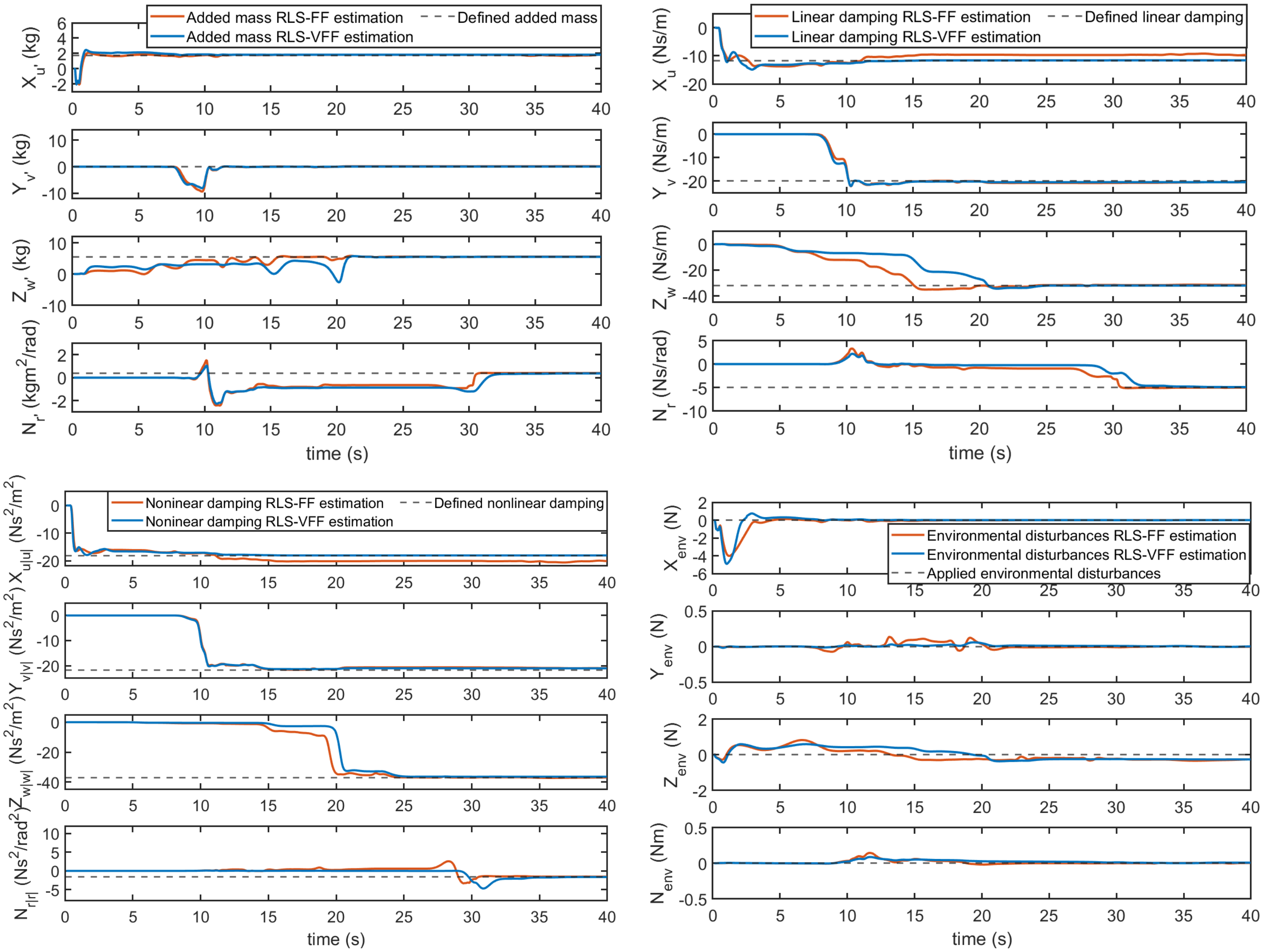

Fig. 3. Online system identification results of added mass, linear damping coefficients, nonlinear damping coefficients, and environmental disturbances using the RLS-FF and the RLS-VFF during the training process without applied environmental disturbances.

seconds; 2) Sway dynamics training: commences at 10 seconds and finishes at 20 seconds; 3) Heave dynamics training: initiates at 20 seconds and terminates at 30 seconds; 4) Yaw dynamics training: starts at 30 seconds and concludes at 40 seconds.

In this study, the number of samples in the short and long windows are set as $n = 10$ and $d = 50$. A smaller n makes the F-test more sensitive to system changes, resulting in a fast adjusting forgetting factor. Additionally, the threshold value γ is set to 0.8. The estimated parameters in $\hat{\Theta}$ obtained using the RLS-VFF algorithm are compared with those obtained using the standard RLS-FF algorithm with a forgetting factor of 0.98. Since the work is conducted in Gazebo, the parameters defined in the Gazebo's unified robot description format (URDF) file are used as a benchmark for comparison with the estimation results.

Figure 3 presents the system identification results in the absence of additional environmental disturbances. In this scenario, the system can be considered as slowly changing. Consequently, the F-test value remains below the threshold γ for most of the time, causing the variable forgetting factor to approach 1. As a result, the estimation results obtained using the RLS-VFF, represented by the blue line, exhibit a slower convergence rate but higher stability, resulting in a smoother line.

On the other hand, Figure 4 demonstrates the system identification results with the introduction of additional environmental disturbances of 5N in the x, y, and z directions in the IRF. Since the environmental disturbances term $\boldsymbol{\tau}_{\text{env}}$ is defined in the BRF in the UUV dynamics model, it has been transformed to the IRF using a rotation matrix for a clearer presentation of the results. After 30 seconds, the orientation of the UUV begins to change to train the yaw dynamics. Consequently, the environmental disturbances acting on the BRF also start to change, leading to a faster changing system. In this situation, the standard RLS-FF struggles to adapt to these changes quickly and stably, resulting in significant chattering in the red line. In contrast, the proposed RLS-VFF still manages to converge to the defined parameters swiftly.

The performance of the proposed adaptive MPC algorithm

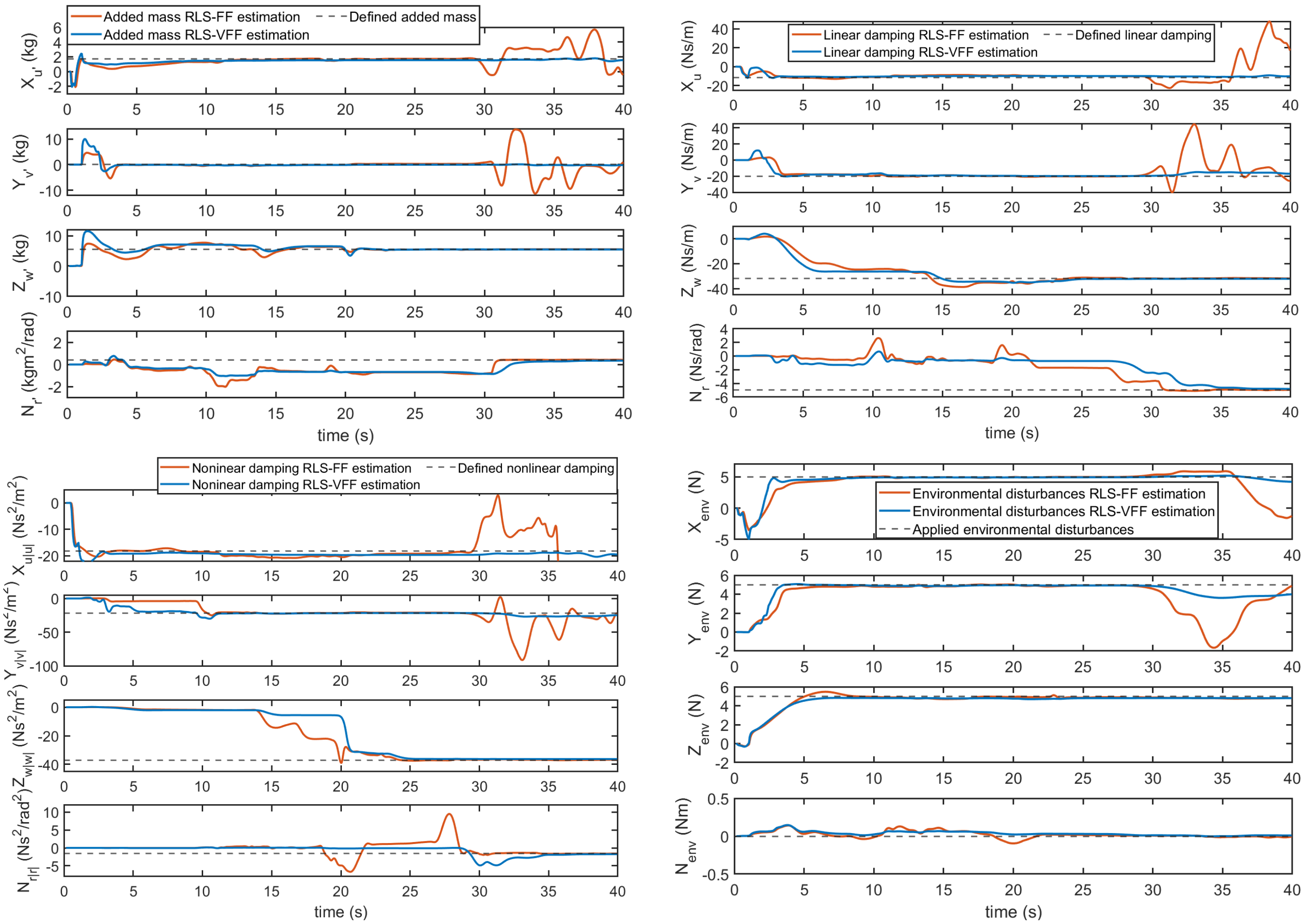

Fig. 4. Online system identification results of added mass, linear damping coefficients, nonlinear damping coefficients, and environmental disturbances using the RLS-FF and the RLS-VFF during the training process with applied environmental disturbances.

is evaluated in comparison to a standard MPC controller and a PID controller. Both the adaptive MPC and the standard MPC employ the same control parameters, as detailed in Table II. Meanwhile, the PID controller's control gains are specified in Table III. A lemniscate trajectory with amplitude of 2 meters is employed as trajectory tracking control problem for these controllers. Furthermore, the additional environmental disturbances of 10N are also applied in the x, y, and z directions in the IRF.

TABLE II
MPC PARAMETERS UTILIZED IN THIS WORK.

Parameters	Value
Prediction horizon	60
Sample time (s)	0.05
Q	[300 300 150 10 10 150 10 10 10 10 10 10 1 1 1 0.5]
Q_N	[300 300 150 10 10 150 10 10 10 10 10 10]
OCP time (ms)	7

Figure 5 presents the control outcomes achieved by the adaptive MPC, standard MPC, and PID controllers in tracking the lemniscate trajectory. The subplots indicate the three-dimensional trajectory tracking results, control inputs, tracking errors, and tracking states, respectively. These results demonstrate the substantial improvement in control performance achieved by employing the proposed adaptive MPC algorithm, even when faced with a highly nonlinear tracking problem and the presence of environmental disturbances.

TABLE III
PID PARAMETERS UTILIZED IN THIS WORK.

Control gain	Surge	Sway	Heave	Yaw
K_p	5	5	5	7
K_i	0.05	0.05	0.05	0.1
K_d	1.2	1.2	1.2	0.6

V. Conclusion

In this study, an effective adaptive control method is proposed by integrating a fast system identification module with MPC for UUVs' motion control in complex underwater environments. Unlike conventional offline system identification,

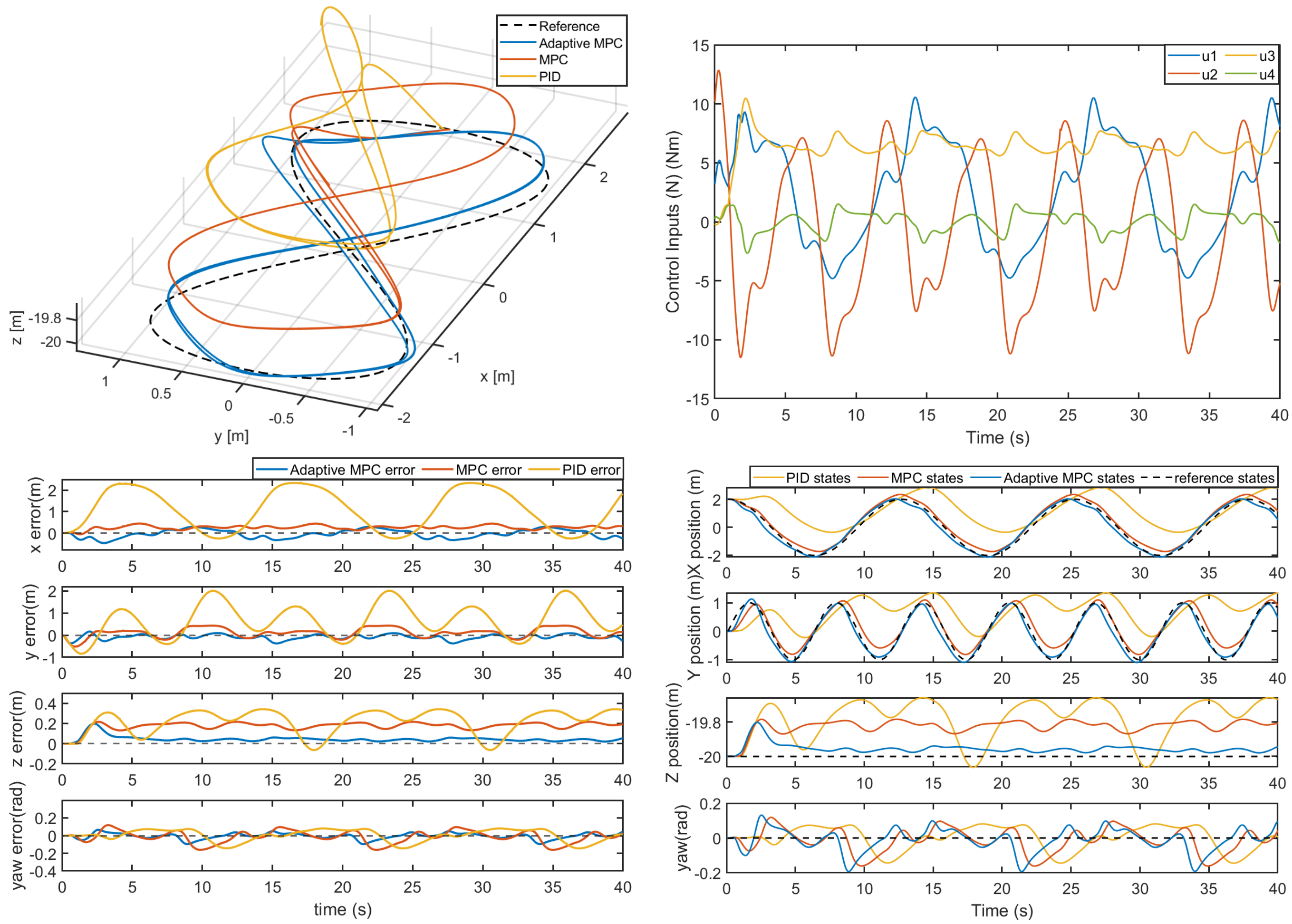

Fig. 5. Control results of lemniscate trajectory tracking using the proposed adaptive MPC, standard MPC, and PID controllers.

the proposed approach utilizes RLS-VFF for online real-time adaptation of the system model when new measurement data is available. This incremental update of model parameters enables continuous learning and tracking of system dynamics. Additionally, RLS-VFF offers computational efficiency by avoiding the need to recompute the regression from scratch for each new data point. By incorporating a variable forgetting factor, the algorithm determines the weight between recent and past data based on the F-test. The F-test assesses if the system has undergone significant changes, and if so, the algorithm gradually reduces the influence of older data to enable rapid and stable adaptation. Another key aspect of the proposed control framework is the utilization of an EAOB for capturing unmodeled dynamics and environmental disturbances. This feature reduces sensitivity to measurement noise and inaccuracies. By improving the accuracy of the prediction model in MPC and compensating for environmental disturbances, the proposed method achieves a reliable controller with the capability of adapting to unknown environments and delivering superior control performance.

REFERENCES

[1] S. Hou, D. Jiao, B. Dong, H. Wang, and G. Wu, "Underwater inspection of bridge substructures using sonar and deep convolutional network," *Advanced Engineering Informatics*, vol. 52, p. 101545, 2022.

[2] S. Hu, R. Feng, Z. Wang, C. Zhu, Z. Wang, Y. Chen, and H. Huang, "Control system of the autonomous underwater helicopter for pipeline inspection," *Ocean Engineering*, vol. 266, p. 113190, 2022.

[3] A. A. Mogstad, Ø. Ødegård, S. M. Nornes, M. Ludvigsen, G. Johnsen, A. J. Sørensen, and J. Berge, "Mapping the historical shipwreck figaro in the high arctic using underwater sensor-carrying robots," *Remote Sensing*, vol. 12, no. 6, p. 997, 2020.

[4] N. Summers, G. Johnsen, A. Mogstad, H. Løvås, G. Fragoso, and J. Berge, "Underwater hyperspectral imaging of arctic macroalgal habitats during the polar night using a novel mini-rov-uhi portable system," *Remote sensing*, vol. 14, no. 6, p. 1325, 2022.

[5] H. Preuß, V. Cherewko, J. Wollstadt, A. Wendt, and H. Renkewitz, "Crawfish goes swimming: Hardware architecture of a crawling skid for underwater maintenance with a bluerov2," in *OCEANS 2022, Hampton Roads*, 2022, pp. 1–5.

[6] J. Mao, G. Song, S. Hao, M. Zhang, and A. Song, "Development of a lightweight underwater manipulator for delicate structural repair operations," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6563–6570, 2023.

[7] H. Wei and Y. Shi, "Mpc-based motion planning and control enables smarter and safer autonomous marine vehicles: perspectives and a tutorial survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 1, pp. 8–24, 2023.

[8] S. B. Gibson and D. J. Stilwell, “Hydrodynamic parameter estimation for autonomous underwater vehicles,” *IEEE Journal of Oceanic Engineering*, vol. 45, no. 2, pp. 385–394, 2020.
[9] K. L. Walker, R. Gabl, S. Aracri, Y. Cao, A. A. Stokes, A. Kiprakis, and F. Giorgio-Serchi, “Experimental validation of wave induced disturbances for predictive station keeping of a remotely operated vehicle,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5421–5428, 2021.
[10] J. Arcos-Legarda and Á. Gutiérrez, “Robust model predictive control based on active disturbance rejection control for a robotic autonomous underwater vehicle,” *Journal of Marine Science and Engineering*, vol. 11, no. 5, p. 929, 2023.
[11] I. Prodan, E. I. Grøtli, L. Lefèvre *et al.*, “Safe navigation in a coastal environment of multiple surface vehicles under uncertainties: A combined use of potential field constructions and nmpc,” *Ocean Engineering*, vol. 216, p. 107706, 2020.
[12] K. L. Walker and F. Giorgio-Serchi, “Disturbance preview for non-linear model predictive trajectory tracking of underwater vehicles in wave dominated environments,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 6169–6176.
[13] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, “Data-driven mpc for quadrotors,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.
[14] R. Li, J. Huang, X. Pan, Q. Hu, and Z. Huang, “Path following of underactuated surface ships based on model predictive control with neural network,” *International Journal of Advanced Robotic Systems*, vol. 17, no. 4, p. 1729881420945956, 2020.
[15] Y. Hu, B. Li, B. Jiang, J. Han, and C.-Y. Wen, “Disturbance observer-based model predictive control for an unmanned underwater vehicle,” *Journal of Marine Science and Engineering*, vol. 12, no. 1, p. 94, 2024.
[16] B. Robotics, “Bluerov2: The world’s most affordable high-performance rov,” *BlueROV2 Datasheet; Blue Robotics: Torrance, CA, USA*, 2016.
[17] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
[18] P. Cardenas and E. A. de Barros, “Estimation of auv hydrodynamic coefficients using analytical and system identification approaches,” *IEEE Journal of Oceanic Engineering*, vol. 45, no. 4, pp. 1157–1176, 2019.
[19] R. Verschueren, G. Frison, D. Kouzoupis, N. van Duijkeren, A. Zanelli, R. Quirynen, and M. Diehl, “Towards a modular software package for embedded optimization,” *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 374–380, 2018.
[20] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, “acados—a modular open-source framework for fast embedded optimal control,” *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2022.