Journal of Advanced Marine Engineering and Technology, Vol. 48, No. 4, pp. 207~218, 2024
J. Advanced Marine Engineering and Technology (JAMET)
https://doi.org/10.5916/jamet.2024.48.4.207

ISSN 2234-7925 (Print)
ISSN 2765-4796 (Online)
Original Paper

Computationally efficient trajectory tracking control of AUVs with nonlinear model predictive control using neural-based dynamics modeling

Ajay Gautam†

(Received July 7, 2024 ; Revised July 30, 2024 ; Accepted July 30, 2024)

Abstract: A computationally efficient nonlinear model predictive control (MPC) scheme is presented for the problem of trajectory tracking control, in the 3D space, of autonomous underwater vehicles (AUVs) under constrained speeds and thruster forces, and possibly in the presence of disturbances and actuator limitations. The proposed scheme considers AUV dynamics with four degrees of maneuverability and uses an accurate modeling of the discrete-time dynamics of the system using a suitable neural network that enables efficient state propagations and MPC cost gradient computations owing to the parallel computation structure of the network, thereby allowing a more efficient solution of the constrained nonlinear MPC optimization problem using a sequential quadratic programming-based approach. This also paves way for MPC optimizations over larger time horizons which may be necessary under certain situations. The effectiveness of the proposed scheme is verified with extensive simulations covering various scenarios including the ones that deal with the presence of random and non-random time-varying disturbances and, additionally, the condition of underactuation of the vehicle due to the failure of an actuator.

Keywords: Trajectory tracking, Nonlinear model predictive control, Autonomous underwater vehicles, Underactuated AUV, Computationally efficient NMPC

1. Introduction

Over the past several years, there has been an increasing academic, technological and commercial interest in unmanned autonomous systems operating in various environments – on ground and water, in air and space and under the sea **[1]**. An autonomous underwater vehicle (AUV) is a computer controlled robotic vehicle equipped with a propulsion system enabling its maneuvering in the three-dimensional (3D) space under the sea. Like other unmanned autonomous systems, AUVs have extensive scientific, military and commercial applications such as mapping and imaging of underwater landscapes, deep-sea exploration of marine life, inspection of underwater infrastructure, detection and clearance of pollutants and mines, undersea search and rescue and so on (see, e.g., **[2]**-**[5]**). Behind the successful design and operation of AUVs are various enabling technologies in the field of sensing, propulsion, communications, guidance, navigation and control **[6]**.

Target or trajectory tracking ability is very crucial for most AUV applications and this requires sophisticated navigation and control schemes since an AUV and the environment that it operates in are characterized by high-dimensional nonlinear dynamics, often uncertain, constrained and underactuated, and usually unpredictable oceanic disturbances. Research on reliable control of AUVs is extensive and continuing **[7]**. A survey of early works on AUV control can be found in **[8]**. Traditional control design for AUVs includes various analytical design approaches employing nonlinear feedback **[9]**, gain-scheduled or parameterized linear feedback **[10]**, input-output feedback linearization **[11]**, and so on. To deal with control challenges owing to the highly nonlinear dynamics involving parameter uncertainties and disturbances, techniques such as sliding mode control **[12]** and adaptive control **[13]** have been tried and continue to be explored **[14]**-**[16]**. Alternative approaches that have been extensively explored include the more computation-oriented techniques such as those using fuzzy logic **[17]**, and neural networks **[18]**-**[19]**.

Control approaches explored in the works mentioned above do not directly deal with state and input constraints whose satisfaction is usually crucial in trajectory tracking applications. Model

† Corresponding Author (ORCID: http://orcid.org/0000-0002-9647-626X): Assistant Professor, School of Electrical, Electronics and Communication Engineering, Korea University of Technology and Education, 1600 Chungjeol-ro, Byeongcheon-myeon, Cheonan-si, Chungcheongnam-do 31253, Republic of Korea, E-mail: agautam@koreatech.ac.kr, Tel: 041-560-1425

predictive control (MPC) is a promising computation-oriented control approach that handles constraints in a systematic way **[20]**. This approach is based on solving an open-loop control optimization problem online in real time and, with the advancements in computing hardware speed, the technology has become applicable for an increasing variety of linear and nonlinear control applications including trajectory planning and tracking applications. Theoretical analysis of the method has also matured over the years **[21]**. MPC has been used for trajectory tracking control of marine vehicles in several earlier works **[22]**-**[28]**. Nonlinear MPC is used together with an unscented Kalman filter for positioning of ships in **[22]**. In **[23]**, the authors present a combination of MPC and genetic algorithms to achieve dynamic trajectory tracking. Similarly, a Lyapunov-based MPC is explored for robust trajectory tracking of AUVs in **[24]**. An event-triggered nonlinear MPC is presented in **[25]** for trajectory tracking by an underactuated ship with reduced computational burden and a similar approach is used for AUVs in the presence of disturbances in **[26]**. In these schemes, the MPC optimization is carried out only when the vehicle's deviation from the desired state is outside a chosen bound. In **[27]**, a robust nonlinear MPC is designed to achieve robust trajectory tracking of underactuated AUVs in the presence of disturbance inputs.

While MPC offers a remarkable control performance, it is computationally demanding, which is particularly significant for nonlinear systems. So, a fast and reliable algorithm to solve the nonlinear MPC problem is always a necessity **[28]**. Since MPC is primarily envisaged in the discrete-time (DT) domain, an accurate DT model of the system is required, and is not readily obtained for a nonlinear system. Most MPC schemes use DT models obtained using a simple Euler or Runge-Kutta (RK)-based discretization, which requires a very small sampling period to be considered. This limits the horizon length over which the MPC optimization can be carried out in real time, and it can be a drawback in certain situations in trajectory tracking.

In this work, we present a trajectory tracking nonlinear MPC scheme that is based on an accurate modeling of the DT dynamics of the AUV and therefore allows an MPC optimization over a larger time horizon. In particular, we consider an AUV with four degrees of maneuverability and model its DT dynamics for a suitable sampling time with a feedforward neural network (NN). A network with a single hidden layer is chosen so that nested or sequential computations can be avoided during cost gradient computations. Such a model can be obtained for sampling intervals significantly larger than those that can be used with the approximate methods. Following **[29]**, we present a sequential quadratic programming (sQP) algorithm to solve the resulting NMPC problem for trajectory tracking. The sQP algorithm ensures feasibility after each iteration and uses a simple trust region constraint to achieve convergence.

We verify the performance of the proposed tracking control scheme through extensive nonlinear simulations for an AUV mission in ideal and non-ideal scenarios. Non-ideal conditions include the presence of external disturbances and/or the failure of the control component along the transverse direction. Simulations show the effectiveness of the proposed approach.

Notations: I_n denotes an identity matrix of size n, $\mathbf{0}_{m\times n}$ denotes an $m \times n$ matrix of all zeros, and $\mathbf{1}_n$ represents a n-dimensional vector of all ones. If any subscript is omitted, the dimension should be clear from the context. For a vector $\mathbf{v}$, diag($\mathbf{v}$) represents a diagonal matrix with the elements of $\mathbf{v}$ along the diagonal, and with entities $\Delta_1, \Delta_2, \dots$, $\mathrm{diag}(\Delta_1, \Delta_2, \dots)$ represents a (block) diagonal matrix with $\Delta_1, \Delta_2, \dots$ along the diagonal. For a vector $\mathbf{v}$, $\|\mathbf{v}\|_Q^2$ with a symmetric matrix Q represents the quadratic form $\mathbf{v}^T Q\mathbf{v}$. For a DT signal $x(k)$, $x(k+i|k)$ denotes the future signal value $x(k+i)$ predicted at time k.

2. System Description

2.1 Mathematical Model of AUV Motion

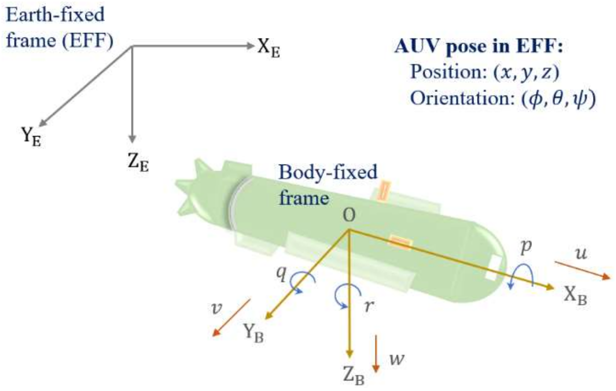


Figure 1: Coordinate frames for describing AUV motion

The motion of an AUV, like that of any marine vehicle, is conveniently described in 6 degrees of freedom (DOF) with its pose specified by its position $\mathbf{p} = (x, y, z)$ and its orientation $\boldsymbol{\theta} = (\phi, \theta, \psi)$ (in terms of Euler angles) in an Earth-fixed coordinate frame X_E-Y_E-Z_E and its linear and angular velocities $\mathbf{v} = (u, v, w)$ and $\boldsymbol{\omega} = (p, q, r)$ specified in a body-fixed reference frame X_B-Y_B-Z_B as illustrated in **Figure 1**. The origin O of the body-fixed frame is chosen at the vehicle's center of gravity (CG) and its axes are chosen to coincide with the three principal axes of inertia – longitudinal, transverse and normal. So, the linear velocity components u, v and w correspond to surge, sway and heave speeds and the angular velocity components p, q and r correspond to roll, pitch and yaw speeds.

The kinematic relationship between the pose vectors and the velocity vectors are given by

$$\dot{\mathbf{p}} = \boldsymbol{J_1}(\boldsymbol{\theta})\mathbf{v}, \qquad \dot{\boldsymbol{\theta}} = \boldsymbol{J_2}(\boldsymbol{\theta})\boldsymbol{\omega}$$

where $\boldsymbol{J_1}(\boldsymbol{\theta})$ and $\boldsymbol{J_2}(\boldsymbol{\theta})$ are 3×3 transformation matrices with elements depending on ϕ, θ, ψ (see, e.g., **[30]** for the details).

In this paper, we consider underwater vehicles equipped with metacentric restoring forces preventing roll and pitch motions so that the roll and pitch angles ϕ and θ and the angular speeds p and q are negligible and close to zero. Under this assumption, the reduced-order kinematic relationship is described by

$$\dot{\mathbf{p}} = \boldsymbol{J_1}(\psi)\mathbf{v} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}\mathbf{v}, \qquad \dot{\psi} = r \tag{1}$$

Further, the dynamics of the vehicle are specified in terms of the combined velocity vector $\boldsymbol{\upsilon} = (\mathbf{v}, r)$ by the relationship

$$\boldsymbol{M}\dot{\boldsymbol{\upsilon}} + \boldsymbol{C}(\boldsymbol{\upsilon})\boldsymbol{\upsilon} + \boldsymbol{D}(\boldsymbol{\upsilon})\boldsymbol{\upsilon} + \mathbf{g} = \boldsymbol{\tau} + \mathbf{d} \tag{2}$$

where $\boldsymbol{M}$ is the matrix of mass/inertia components (including the added mass/inertia), $\boldsymbol{C}(\boldsymbol{\upsilon})$ is the matrix of Coriolis terms (including that of added mass/inertia), $\boldsymbol{D}(\boldsymbol{\upsilon})$ is the hydrodynamic damping matrix, $\boldsymbol{g}$ is the vector of gravitational forces and moments, $\boldsymbol{\tau}$ is the control input vector comprising forces and torques and $\mathbf{d}$ is the vector of disturbance forces/torques, mainly due to water currents. Assuming the AUV body to be symmetric about the x-y and x-z planes, and that it has a slightly positive buoyancy, we have,

$$\boldsymbol{M} = \mathrm{diag}(m_X, m_Y, m_Z, I_Z),$$

$$\boldsymbol{C}(\boldsymbol{\upsilon}) = \begin{bmatrix} 0 & 0 & 0 & -m_Y v \\ 0 & 0 & 0 & m_X u \\ 0 & 0 & 0 & 0 \\ m_Y v & -m_X u & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{D}(\boldsymbol{v}) = -\mathrm{diag}(X_u, Y_v, Z_w, N_r) - \mathrm{diag}\big(X_{u|u|}|u|, Y_{v|v|}|v|, Z_{w|w|}|w|, N_{r|r|}|r|\big)$$

$$\mathbf{g} = [0 \quad 0 \quad B - W \quad 0]^T$$

$$\boldsymbol{\tau} = [\tau_X \quad \tau_Y \quad \tau_Z \quad \tau_N]^T$$

$$\mathbf{d} = [d_X \quad d_Y \quad d_Z \quad d_N]^T$$

where W and B are the weight and the buoyancy of the vehicle, and, m_X, m_Y, m_Z, I_Z are mass/inertia terms (including added mass/inertia), X_u, Y_v, Z_w, N_r and $X_{u|u|}, Y_{v|v|}, Z_{w|w|}, N_{r|r|}$ are hydrodynamic damping terms, $\tau_X, \tau_Y, \tau_Z, \tau_N$ are the control forces/torque generated by the thrusters and d_X, d_Y, d_Z, d_N are disturbance forces/torque effective along/about the surge, sway, heave and yaw directions.

Thus, combining **Equations (1)** and **(2)**, we can write the mathematical description of the AUV motion in the 3D space in the form of the following nonlinear state-space model

$$\dot{\mathbf{x}} = \boldsymbol{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}), \qquad \mathbf{y} = C\,\mathbf{x} \tag{3}$$

where $\mathbf{x} = [x \ y \ z \ \psi \ u \ v \ w \ r]^T \in \mathbb{R}^8$ is the state vector, $\mathbf{u} = \boldsymbol{\tau} \in \mathbb{R}^4$ is the control input vector, $\mathbf{d} \in \mathbb{R}^4$ is the disturbance input vector, and

$$\boldsymbol{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) = \begin{bmatrix} u\cos\psi - v\sin\psi \\ u\sin\psi + v\cos\psi \\ w \\ r \\ \frac{1}{m_X}(m_Y vr + (X_u + X_{u|u|}|u|)u + \tau_X + d_X) \\ \frac{1}{m_Y}(-m_X ur + (Y_v + Y_{v|v|}|v|)v + \tau_Y + d_Y) \\ \frac{1}{m_Z}(W - B + (Z_w + Z_{w|w|}|w|)w + \tau_Z + d_Z) \\ \frac{1}{I_Z}((m_X - m_Y)uv + (N_r + N_{r|r|}|r|)r + \tau_N + d_N) \end{bmatrix}$$

Further, the output vector $\mathbf{y} = C\,\mathbf{x} = [x \ y \ z \ \psi]^T \in \mathbb{R}^4$ so that $C = [I_4 \quad \mathbf{0}_{4\times4}]$.

Remark 1: The motion of an AVU described by **(3)** has 4 DOF and is supported by control inputs along the 4 directions. The AUV can still be maneuvered if the sway input component τ_Y is unavailable or the corresponding actuator is faulty. A vehicle in such a situation is considered to be underactuated. Several works on AUV trajectory tracking control have focused on underactuated AUVs (e.g., **[27]**)

2.2 Problem Description

We wish to control the motion of an AUV for missions that require tracking a constant target point or tracking a specified time-varying reference trajectory in some optimal way while satisfying all requirements including system constraints.

2.2.1 System Constraints

The motion of an AUV is usually constrained by limitations on the values of control inputs and state components; that is, the limits on the thruster forces and torque that can be generated and the limits on the velocities that are tolerated in each direction. Considering that the control input components and the velocity components are bounded in their absolute values, we express the constraints in collective forms mentioned below:

$$\mathbf{u} = \boldsymbol{\tau} \in \mathbb{U} = \left\{ \mathbf{u} \in \mathbb{R}^4 \;\middle|\; \begin{matrix} |u_1| \leq \hat{\tau}_X, |u_2| \leq \hat{\tau}_Y, \\ |u_3| \leq \hat{\tau}_Z, |u_4| \leq \hat{\tau}_N \end{matrix} \right\} \tag{4a}$$

$$\mathbf{x} \in \mathbb{X} = \left\{ \mathbf{x} \in \mathbb{R}^8 \;\middle|\; \begin{matrix} |x_5| \leq \hat{u}, |x_6| \leq \hat{v}, \\ |x_7| \leq \hat{w}, |x_8| \leq \hat{r} \end{matrix} \right\} \tag{4b}$$

Constraints on the position components of the state may exist but they mostly depend on the mission and the environment and may be specified dynamically during the mission.

2.2.2 Target Reference Point or Trajectory

When a target reference point is given, it is the desired final position $\mathbf{p}_d$ which the vehicle is supposed to eventually reach. The reference trajectory and the final time may not be specified in this scenario.

When a target reference trajectory is given, it is specified in the form of a time-varying position vector $\mathbf{p}_d(t) = (x_d(t), y_d(t), z_d(t))$ which is a smooth and continuous function of time. The other components of the desired state $\mathbf{x}_d(t)$ can be readily obtained by first finding $\psi_d(t) = \mathrm{atan}\left(\frac{y_d(t)}{x_d(t)}\right)$, and then finding the derivatives: $\mathbf{v}_d(t) = \boldsymbol{J}_1^{-1}(\psi)\dot{\mathbf{p}}_d(t)$ and $r_d(t) = \dot{\psi}_d(t)$.

Here, we assume that the target reference point or reference trajectory is feasible in the sense that it can be tracked by the AUV while satisfying all constraints. This is theoretically so if the desired position trajectory is a part of a feasible solution to the system model **(3)** satisfying all constraints.

2.2.3 Control Objectives

Given a constant target position $\mathbf{p}_d(t) = \mathbf{p}_d$ or a time-varying reference trajectory $\mathbf{p}_d(t)$, the primary control objective is to drive the AUV as close as possible to the reference point or trajectory in some best possible way, particularly in the sense of minimizing a quadratic function of the position error

$$\int_{t=t_0}^{t_f} \boldsymbol{\Delta}\mathbf{p}^T(t)\, Q_p\, \boldsymbol{\Delta}\mathbf{p}(t) dt, \quad \boldsymbol{\Delta}\mathbf{p}(t) = \mathbf{p}(t) - \mathbf{p}_d(t) \tag{5}$$

over the specified or some reasonable time horizon $[t_0, t_f]$. The control scheme should ensure that all system constraints are satisfied at all times. Further, it is also expected to show a reasonably good performance even when there are disturbances and parameter uncertainties or some minor faults in the system model.

2.2.4 Other Considerations

A control scheme designed to achieve the above-mentioned control objective needs to be aware of the desired reference trajectory and the vehicle state at all times. It is assumed that the AUV receives tracking mission information through suitable communications from some external agency or it is capable of generating the reference trajectory itself. Further, it is supported by an appropriate navigation system with several sensors whose readings can be used to estimate the actual state of the AUV. Navigation schemes for autonomously operating systems have been well researched and newer and more promising techniques continue to be explored (see, e.g., **[31]**, **[32]**).

3. AUV Trajectory Tracking with NMPC using Neural-Modeled System Dynamics

3.1 Basic NMPC Algorithm

Among various control approaches, MPC is perhaps one of the best suited control schemes to achieve the control goal mentioned in Section 2.2.3 for a constrained AUV system. MPC uses the system dynamics model, usually in the DT framework, to numerically optimize the control input over some time horizon so as to minimize a suitably defined cost function over that horizon. This optimization is done regularly in real time, usually at every discrete time step to update the control input to be applied to the system.

Consider a DT version of model **(3)**, viz.,

$$\mathbf{x}(k+1) = \boldsymbol{\phi}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k)), \qquad \mathbf{y}(k) = C\mathbf{x}(k) \tag{6}$$

Here, $\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k)$ and $\mathbf{y}(k)$ represent the values of the signals at the discrete sampling time $kT, k \in \mathbb{Z}$, where T is the sampling period. An MPC scheme meant to achieve the desired trajectory tracking by the AUV uses a cost function of the form

$$J(\mathbf{x}(k), U(k), k) = \sum_{i=0}^{N-1} \ell(\mathbf{x}(k+i|k), \mathbf{u}(k+i|k), k+i) \tag{7}$$

with $\ell(\mathbf{x}, \mathbf{u}, k) = ||\mathbf{x} - \mathbf{x}_d(k)||_Q^2 + ||\mathbf{u} - \mathbf{u}_d(k)||_R^2$

where Q and R are positive (semi-)definite matrices defining the stage cost function $\ell(.,.,.)$ and

$$U(k) = [\mathbf{u}(k|k)^T \;\; \mathbf{u}(k+1|k)^T \ldots\ldots \mathbf{u}(k+N-1|k)^T\,]^T$$

is the stacked vector of inputs to be optimized. The MPC optimization problem to be solved at each time step k is stated as

$$\underset{U(k)}{\text{minimize}} \; J(\mathbf{x}(k), U(k), k) \tag{8}$$

such that

$$\mathbf{x}(k|k) = \mathbf{x}(k)$$

$$\mathbf{x}(k+i+1|k) = \boldsymbol{\phi}\big(\mathbf{x}(k+i|k), \mathbf{u}(k+i|k), \mathbf{d}(k+i|k)\big), \quad i = 1,..,N-1$$

$$M_{\mathbf{x}}\,\mathbf{x}(k+i|k) \le \mathbf{1}, \qquad i = 1,2,..,N$$

$$M_{\mathbf{u}}\,\mathbf{u}(k+i|k) \le \mathbf{1}, \qquad i = 0,1,..,N-1$$

where matrices $M_{\mathbf{x}}$ and $M_{\mathbf{u}}$ in the last two constraints are so chosen that they represent $\mathbf{x}(k+i|k) \in \mathbb{X}$ and $\mathbf{u}(k+i|k) \in \mathbb{U}$.
A basic MPC algorithm based on problem **(8)** is outlined below.

Algorithm 1: Online MPC algorithm: At each time step k:

i) Measure or estimate the state of the system, $\mathbf{x}(k)$.
ii) Solve problem **(8)** to obtain an optimal $U^*(k)$.
iii) Apply the control input $\mathbf{u}^*(k|k)$ to the AUV.

The cost function **(7)** used in the MPC algorithm can be made to approximate **(5)** with the choices $Q = C^T C$ and $R = 0$. Still, there are various difficulties in successfully achieving the control objective by implementing Algorithm 1. These include theoretical and practical concerns related to stability, robustness and implementation.

3.2 Stability

If the time horizon $[t_0, t_f]$ of the reference trajectory is significant, covering it entirely in **(7)** may be computationally prohibitive. It is desired that a repeated solution of **(8)** with a modest value of N in **(7)** as outlined in Algorithm 1 ensures the asymptotic or exponential convergence of the tracking error $\Delta\mathbf{x}(k) = \mathbf{x}(k) - \mathbf{x}_d(k)$ to the origin or to a set around it.

Let us consider the disturbance-free dynamics

$$\mathbf{x}(k+1) = \boldsymbol{\phi_o}\big(\mathbf{x}(k), \mathbf{u}(k)\big) \tag{9}$$

where $\boldsymbol{\phi_o}(\mathbf{x}, \mathbf{u}) = \boldsymbol{\phi}(\mathbf{x}, \mathbf{u}, \mathbf{0})$.We make the following assumption about the reference trajectory.

Assumption 1: The desired reference trajectory $(\mathbf{x}_d(k), \mathbf{u}_d(k))$ satisfies the DT state dynamics in **(9)** :

$$\mathbf{x}_d(k+1) = \boldsymbol{\phi_o}\big(\mathbf{x}_d(k), \mathbf{u}_d(k)\big)$$

Under Assumption 1, the tracking error dynamics are given by

$$\Delta\mathbf{x}(k+1) = \boldsymbol{\phi_o}\big(\mathbf{x}_d(k) + \Delta\mathbf{x}(k), \mathbf{u}_d(k) + \Delta\mathbf{u}(k)\big) - \boldsymbol{\phi_o}\big(\mathbf{x}_d(k), \mathbf{u}_d(k)\big)$$

where $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}_d(k)$. A local linearization of the error dynamics about a reference point $\mathbf{r} = (\mathbf{x}_d(k), \mathbf{u}_d(k))$ in the desired trajectory results in

$$\Delta\mathbf{x}(k+1) = A_{\mathbf{r}}\,\Delta\mathbf{x}(k) + B_{\mathbf{r}}\,\Delta\mathbf{u}(k) + \vartheta\big(\Delta\mathbf{x}(k), \Delta\mathbf{u}(k)\big)$$

where $A_{\mathbf{r}} = \left.\frac{\partial\phi_o}{\partial\mathbf{x}}\right|_{\mathbf{r}}$, $B_{\mathbf{r}} = \left.\frac{\partial\phi_o}{\partial\mathbf{u}}\right|_{\mathbf{r}}$ and $\vartheta(.,.)$ represents the higher order terms.

If $(A_{\mathbf{r}}, B_{\mathbf{r}})$ is stabilizable, for a $\Delta\mathbf{x}(k)$ sufficiently close to the origin, there exists a state feedback gain matrix $K_{\mathbf{r}}$ such that with $\mathbf{u}(k) = \mathbf{u}_d(k) + K_{\mathbf{r}}\Delta\mathbf{x}(k)$, the tracking error satisfies the contractivity property

$$||\Delta\mathbf{x}(k+1)||^2_{P_{\mathbf{r}}} < \rho||\Delta\mathbf{x}(k)||^2_{P_{\mathbf{r}}} \tag{10}$$

with $\rho \in (0,1)$ for some positive definite Lyapunov matrix $P_{\mathbf{r}}$ **[21]**. This fact can be used to construct terminal state feedback controllers together with associated invariant sets to be used as terminal state constraints in the online MPC optimization problem to guarantee stability. For a class of tracking problems, a parameterized state feedback gain and the associated invariant set may be computed and used for a range of desired state-input pairs **[29]**. However, computing terminal state-feedback gains and the associated invariant sets is usually computationally burdensome. As explored in **[21]**, under the local stabilizability condition (which the AUV system satisfies), Algorithm 1 with a positive definite matrix Q in **(7)** and a sufficiently large horizon length N ensures that the tracking error exponentially converges to the origin. However, the theoretical bound on the required horizon length N can be quite conservative.

3.3 Handling Disturbances and Uncertainties

The AUV motion model includes the disturbance term $\mathbf{d}(t)$ which is used to mainly model the effect of ocean currents. This creates a difficulty in the evaluation of the cost function and in ensuring the constraints in **(8)**. Since the bounds on the disturbance components can be estimated, we can use a robust approach to cost evaluation and constraint satisfaction. Potential robust approaches include the tube-based approach **[33]**, disturbance-feedback-based approach (e.g., **[34]**), and so on. The latter deals with linear time-varying systems, and, as we shall see in Section 3.4, when problem **(8)** is solved using the sQP approach, nonlinear dynamics constraints are linearized about the existing state-input trajectories effectively resulting in an MPC problem for a linear-time-varying system.

The robust approaches, however, usually lead to conservative results while significantly increasing the computational burden. In our problem, the disturbance input is an input disturbance that enters the system in the same way as the control input. It can be estimated using an appropriate filter and its effect countered effectively using the control input. Since the disturbance is slowly time-varying, it can be assumed to remain constant through the prediction horizon. Slightly tightened input constraints can be used to account for potential variations in the disturbance values.

Since the system model that we use is based on several simplifying assumptions and the model uses a number of parameters whose values may only be approximately known, there are other potential sources of uncertainties. However, the inherent robustness of the MPC approach due to its receding horizon implementation can be trusted to handle the effects of these uncertainties.

3.4 Numerical Implementation

The desired performance of the control scheme outlined in Algorithm 1 is achieved only under the condition that the DT dynamics model is accurate and that the nonlinear optimization problem can be numerically solved within a fraction of the sampling period. An exact equivalent DT model of a continuous-time nonlinear system is not readily available. Numerical approximations using Euler difference methods or RK methods are often used in applications. However, these numerical approximations are accurate only for small sampling intervals. Since we envisage the possibility of considering optimizations over longer horizon lengths, we consider a more accurate modeling of the DT dynamics using alternative approximations. A suitable approximator in this context is a single-layer feedforward neural-network (ff-NN) which does not involve nested nonlinear function evaluations. This simplifies and speeds up future state computations and also provides a simple analytical expression for gradient evaluations when solving the numerical optimization problem.

3.4.1 Neural Modeling of DT Dynamics

In the continuous-time state equation, the function $\boldsymbol{f}(\mathbf{x},\mathbf{u},\mathbf{d})$ is such that the input vectors $\mathbf{u}$ and $\mathbf{d}$ always appear together and none of the component functions depend on position variables x,y and z. So, we can assume a single input vector $\mathbf{u}$ representing the combined input and also omit x,y,z as variable inputs to the DT nonlinear dynamics function. Under these considerations, defining a vector $\boldsymbol{\chi} = [\psi \;\; u \;\; v \;\; w \;\; r]^T$, the DT state equation can be written as

$$\mathbf{x}(k+1) = \bar{A}\,\mathbf{x}(k) + \boldsymbol{\phi}_{\boldsymbol{n}}\big(\boldsymbol{\chi}(k),\mathbf{u}(k)\big) \tag{11}$$

where $\bar{A} \;= \mathrm{diag}(I_3, \mathbf{0}_{5\times 5})$ and $\boldsymbol{\phi}_{\boldsymbol{n}}\big(\boldsymbol{\chi}(k),\mathbf{u}(k)\big)$ represents the nonlinear part of DT dynamics. Clearly, $\boldsymbol{\phi}_{\boldsymbol{n}}(\boldsymbol{\chi}\,,\mathbf{u})$ is a continuous function of its arguments and it can be approximated to any desired accuracy by a ff-NN with a single hidden layer of a sufficiently large number of neurons. Let it be approximated by a network with η neurons in the hidden layer:

$$\boldsymbol{\phi}_{\boldsymbol{n}}(\boldsymbol{\chi},\mathbf{u}) = W_o\boldsymbol{\varphi}\big(W_\chi\boldsymbol{\chi} + W_u\mathbf{u} + \mathbf{b}\big) + \mathbf{e}(\boldsymbol{\chi},\mathbf{u})$$

where $W_\chi \in \mathbb{R}^{\eta\times 5}$ and $W_u \in \mathbb{R}^{\eta\times 4}$ are the hidden layer weights for network inputs $\boldsymbol{\chi}$ and $\mathbf{u}$, $\mathbf{b} \in \mathbb{R}^{\eta}$ is the bias vector and $W_o \in \mathbb{R}^{8\times\eta}$ is the output layer weight matrix. The vector function $\boldsymbol{\varphi}$: $\mathbb{R}^{\eta} \to \mathbb{R}^{\eta}$ is a diagonal operator that applies an identical continuous nonlinearity $\varphi_i(.) = \varphi(.)$, usually a sigmoidal function (e.g., tanh(.) function) to each component of the argument of $\boldsymbol{\varphi}(.)$. $\mathbf{e}(\boldsymbol{\chi},\mathbf{u})$ represents the approximation error and is assumed to be small.

The ff-NN representing the nonlinear DT dynamics function $\boldsymbol{\phi}_{\boldsymbol{n}}(\boldsymbol{\chi},\mathbf{u})$ needs to be suitably trained by generating a rich set of input-output data samples through numerical simulations. Input data samples are chosen randomly from within the bounds considered and the corresponding output samples are computed using highly accurate numerical integration.

3.4.2 Numerical Solution of the MPC Optimization Problem

Having obtained a sufficiently accurate neural-based model of nonlinear DT dynamics, we follow the sQP-based algorithm mentioned in **[29]** to numerically solve the MPC optimization problem **(8)** where we replace the nonlinear dynamics constraint by the neural-based model

$$\mathbf{x}(k+i+1|k) = \bar{A}\mathbf{x}(k+i|k) + W_o\boldsymbol{\varphi}\big(\mathbf{z}(k+i|k)\big) \tag{12}$$

where $\mathbf{z}(k+i|k) = W_\chi G\mathbf{x}(k+i|k) + W_u\mathbf{u}(k+i|k) + \mathbf{b}$ with $G = [\mathbf{0}_{5\times 3} \;\; I_5]$. Here, we have assumed that the approximation error is negligible. The algorithm is based on the linearization of the nonlinear equation in **(12)** about the existing feasible state and input trajectories. Given an existing feasible input sequence

$$\bar{U}(k) = [\bar{\mathbf{u}}(k|k)^T \;\; \bar{\mathbf{u}}(k+1|k)^T \ldots\ldots \bar{\mathbf{u}}(k+N-1|k)^T\;]^T$$

the corresponding state sequence

$$\bar{X}(k) = [\;\bar{\mathbf{x}}(k|k)^T \;\; \bar{\mathbf{x}}(k+1|k)^T \ldots\ldots \bar{\mathbf{x}}(k+N|k)^{\boldsymbol{T}}\;]^T$$

can be immediately obtained using **(12)**. Linearizing **(12)** using the fist-order Taylor expansion of the function $\boldsymbol{\varphi}(.)$ about the existing state-input trajectories gives us

$$\begin{aligned}\mathbf{x}(k+i+1|k) = A(k+i|k)\,\mathbf{x}(k+i|k)\\ +B(k+i|k)\,\mathbf{u}(k+i|k) + \mathbf{h}(k+i|k) \qquad (13)\end{aligned}$$

where

$$A(k+i|k) = \bar{A} + W_o \operatorname{diag}\left(\boldsymbol{\varphi}'\big(\bar{\mathbf{z}}(k+i|k)\big)\right) W_\chi G$$

$$B(k+i|k) = W_o \operatorname{diag}\left(\boldsymbol{\varphi}'\big(\bar{\mathbf{z}}(k+i|k)\big)\right) W_u$$

$$\begin{aligned}\mathbf{h}(k+i|k) = W_o\boldsymbol{\varphi}\big(\bar{\mathbf{z}}(k+i|k)\big) - \big(A(k+i|k) - \bar{A}\big) \times\\ \bar{\mathbf{x}}(k+i|k) - B(k+i|k)\bar{\mathbf{u}}(k+i|k)\end{aligned}$$

Here, $\boldsymbol{\varphi}'(.) = \left[\varphi_1'(\bar{z}_1) \ \dots \ \varphi_\eta'(\bar{z}_\eta)\right]^T$ is a vector with component-wise derivatives.

With the linearized version **(13)** of the dynamics constraint for $i = 0,1,..,N-1$, the optimization problem **(8)**, can be expressed as a QP problem

$$\underset{U(k)}{\text{minimize}} \quad \|[\mathbf{x}(k)^T \quad U(k)^T]^T\|_{\mathbf{H}}^2 \qquad (14)$$

$$\text{such that} \quad \mathbf{M}[\mathbf{x}(k)^T \quad U(k)^T]^T \leq \mathbf{1}$$

with appropriately defined matrices **H** and **M** (See, e.g., **[29]**,**[34]** for the details.

In the following, we briefly outline the sQP-based algorithm to solve problem **(14)**.

Algorithm 2: sQP-based solution of neural-modeled NMPC problem

```
i)   If $k = 0$, choose a feasible initial decision vector
         $\bar{U}(k) = [\bar{\mathbf{u}}(k|k)^T \ \bar{\mathbf{u}}(k+1|k)^T \dots \bar{\mathbf{u}}(k+N-1|k)^T\,]^T$
     Else, set $\bar{\mathbf{u}}(k+i|k) = \mathbf{u}(k+i|k-1), i = 0,..,N-2$
     and $\bar{\mathbf{u}}(k+N-1|k) = \mathbf{u}(k+N-2|k-1)$.
     Also, using (12), find the corresponding state sequence
     $\{\bar{\mathbf{x}}(k+i|k\}_{i=0}^N$.
ii)  Compute $A(k+i|k), B(k+i|k)$ and $\mathbf{h}(k+i|k)$ for $i = 0,\dots,N-1$ as mentioned in (13).
iii) Imposing an additional constraint $U(k) - \bar{U}(k) \in \mathcal{U}$ for
     some suitably chosen set $\mathcal{U}$, solve (14) to obtain $U^*(k)$.
iv)  If $\|U^*(k) - \bar{U}(k)\| \leq \epsilon$, stop.
     Else
       • Compute the state sequence $\{\mathbf{x}^*(k+i|k\}_{i=1}^N$ corresponding to the optimal solution $U^*(k)$.
       • If $\mathbf{x}^*(k+i|k) \in \mathbb{X}, i = 1,\dots,N$, set $\bar{U}(k) = U^*(k)$
         and go to Step (ii).
         Else, update the set $\mathcal{U}$ and go to Step (iii).
```

The approach outlined in Algorithm 2 is also referred to as feasibility-perturbed sequential QP approach **[35]** since the feasibility is ensured through nonlinear propagation after each iteration. The additional constraint used in Step (iii) is a trust region constraint imposed to ensure that the new computed vector $U(k)$ does not deviate too much from $\bar{U}(k)$ so that the convergence of the solution can be ensured. A proper initialization procedure starting with a small N is usually required to obtain a feasible solution at time $k = 0$. With a proper initialization and a suitable trust region constraint, the algorithm gives a near-optimal solution within a few iterations.

4. Numerical Simulations

4.1 AUV Details

Extensive simulations are carried out to assess the performance of the proposed approach. Parameters specifying the dynamics of the AUV and the constraints applicable on speeds and thruster actions are mentioned in **Table 1**.

Table 1: AUV dynamics parameters and constraints

Mass / Inertia		Hydrodynamic damping Coefficients				Speed limits		Thruster limits	
m_X	40	X_u	-70	$X_{u\|u\|}$	-100	$\hat{u}$	1	$\hat{\tau}_X$	200
m_Y	51	Y_v	-100	$Y_{v\|v\|}$	-200	$\hat{v}$	0.12	$\hat{\tau}_Y$	25
m_Z	51	Z_w	-100	$Z_{w\|w\|}$	-100	$\hat{w}$	0.75	$\hat{\tau}_Z$	180
I_Z	15	N_r	-50	$N_{r\|r\|}$	-100	$\hat{r}$	$\pi/4$	$\hat{\tau}_N$	100
Restoring forces: $W = 343.5, B = 345$									

4.2 AUV DT Dynamics Modeling with NNs

We obtain two ff-NNs – NN-1 and NN-2 to represent nonlinear DT dynamics for two different sampling periods – $T_1 = 1$s and $T_2 = 0.25$s. Both are trained with about 10000 samples of input-output data. Input data samples are selected randomly from the set $1.1([-\pi,\pi] \times G\mathbb{X} \times \mathbb{U})$ where the set $[-\pi,\pi]$ is the range chosen for the yaw angle ψ. The factor 1.1 is used to provide some margin at the constraint boundaries. Outputs corresponding to input samples are obtained using the ode45 solver in MATLAB. Networks NN-1 and NN-2 are designed with $\eta_1 = 48$ and $\eta_2 = 40$ hidden layer neurons which use the tanh() activation function. After sufficient training, their mean squared error (MSE) performances are found to be of the order 10^{-5} and 10^{-6} respectively.

Table 2 gives a comparison, in terms of accuracy and computational efficiency, of model NN-2 with the popular RK-4 method (which is used, e.g., in **[22]**, [**26]**). Here, the model accuracy is indicated by the MSE over all training and test samples, and, the computational efficiency is expressed in terms of the ratio of the average model function evaluation time over the average time taken by the accurate ode45 method in the same machine. While NN-2 is expectedly more accurate for this step size, what is significant is its comparative computational advantage which should remain significant (for both function evaluation and gradient evaluation) even for small step sizes for which the accuracy advantage may not be significant. Note that the RK-4 method is not stable with the time step size $h = 1$.

Table 2: Comparison of approximations with NN-2 and RK-4

	NN-2 ($T = 0.25$)	RK-4 ($h = 0.25$)
MSE	2×10^{-6}	0.036
$\bar{t}_c/\bar{t}_{ode45}$	0.004	0.032

4.3 AUV Mission Simulation with NMPC

For tracking control performance evaluation, we consider an AUV mission comprising predefined time-varying trajectory tracking and free set-point tracking segments. In particular, the following phases are considered in the mission:

1) Descending phase: Trajectory tracking phase 1 (100s)

$$x_d(t) = 2 + 0.2t, \qquad y_d(t) = 1 + 0.05t,$$

$$z_d(t) = 22 - 20e^{-\frac{(x_d(t)-2)^2+(y_d(t)-1)^2}{25}}$$

2) Free set-point tracking phase (No fixed target time)

$$(x_d, y_d, z_d) = (60,10,30)$$

3) Object inspection phase (Trajectory tracking phase 2 (800s))

$$x_d(t) = 71 - 10\cos(0.05(t - t_2)) + 0.01(t - t_2)$$

$$y_d(t) = 10 + 8\sin(0.05(t - t_2)),$$

$$z_d(t) = 30 - 0.025(t - t_2)$$

where t_2 is the time when Phase 2 is completed.

We assume that the AUV is initially at rest at (0,0,0).

4.3.1 Performance in Disturbance-Free Scenario

We first consider the case without disturbances in the simulation model. In the MPC cost function, we use the cost matrix $Q = \text{diag}(I_4, 0.1I_4)$ in all phases, and since the desired inputs are not specified, we mildly penalize the actual inputs with $R = 10^{-5}I_4$ in Phases 1 and 3 and $R = 10^{-3}I_4$ in Phase 2. We also look for a horizon length N that is sufficient to ensure that the predicted terminal state is close to the desired pose throughout the mission. It is found that with model NN-1 ($T_1 = 1$s), a horizon length of $N = 10$ is generally sufficient when the initial state is not very far from the corresponding reference point. In Phase 2, the target point is initially not close but to avoid using a large horizon length, we consider a virtual reference converging to the target point (60,10,30) and update the virtual reference in real time.

Figure 2 shows the reference and actual AUV position trajectories that we achieve with the MPC scheme using model NN-1 in the disturbance-free scenario. It can be seen that the MPC is able to drive the AUV closely along the desired trajectory.

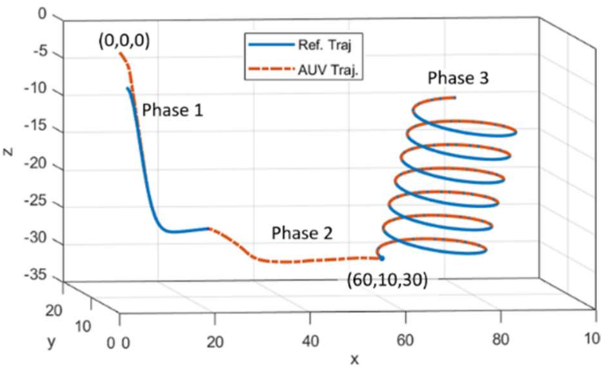


Figure 2: AUV trajectory obtained with the MPC scheme in disturbance-free scenario, alongside the reference trajectory

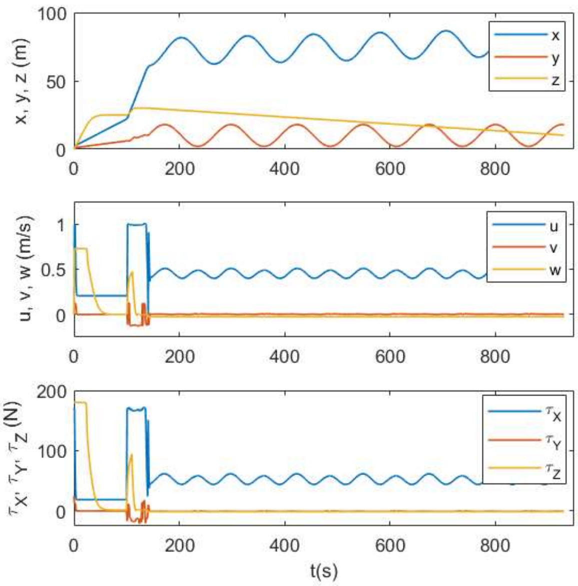
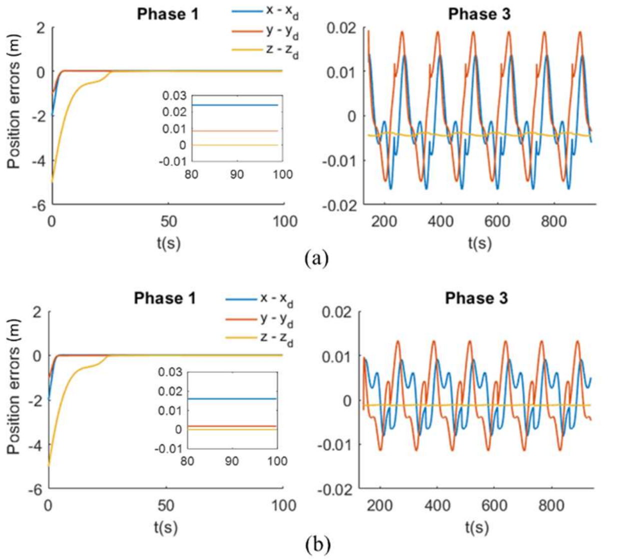


Figure 3: Position, velocity and thruster force components during AUV mission in disturbance-free scenario

Figure 3 shows the position and velocity components of the AUV state and the thruster forces in x-,y-,z- directions. It can be seen that the z-direction thruster reaches the limit during Phase 1 when the decent is steep. Similarly, the surge and sway speed limits are reached during Phase 2. **Figure 4(a)** shows the position tracking errors during Phase 1 and 3, and it can be seen from the figure that the tracking error is within ± 0.025m in each dimension. Note that the tracking error is not relevant in Phase 2.

In order to assess the impact of the accuracy of the dynamics model, we also simulate the performance of the MPC scheme with model NN-2. Since NN-2 uses a sampling time of 0.25s, a horizon length of 10s would require $N = 40$. However, since the input can be updated quickly, we consider $N = 10$ as earlier. The tracking response is similar to the one obtained with NN-1. However, as we can see in **Figure 4(b)**, which presents the position tracking errors during Phase 1 and Phase 3, the tracking errors are smaller with NN-2 than with NN-1.



(a)

(b)

Figure 4: Position tracking errors in Phases 1 and 3 in disturbance-free scenario: (a) with model NN-1, (b) with model NN-2

This can be expected since model NN-2 is at least one order of magnitude more accurate (in terms of MSE values) than NN-1 and it also updates the control actions 4 times more frequently.

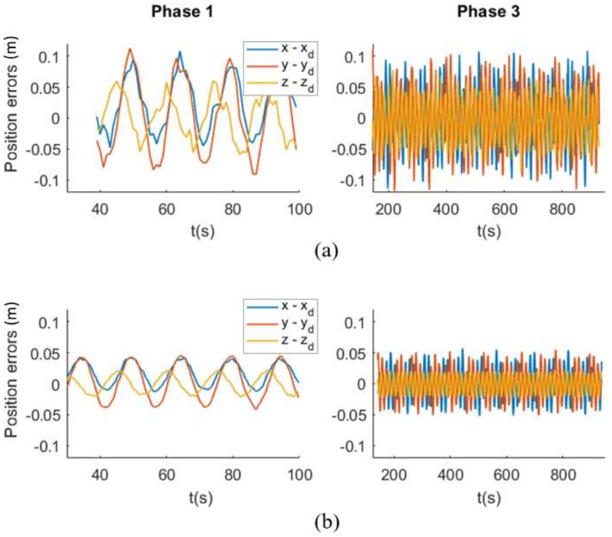
4.3.2 Performance in the Presence of Disturbances

We next consider a scenario with disturbance inputs. Disturbances are mainly due to water currents inside the sea and are slowing varying in time but there can be other sources too. We consider a disturbance vector of the following form:

$$\mathbf{d} = \begin{bmatrix} 8\sin(0.1t) + 2v_1 \\ 8\cos(0.1t - 0.3) + 2v_2 \\ 8\cos(0.1t + 0.8) + 2v_3 \\ 2\sin(0.1t + 0.5) + 0.05v_4 \end{bmatrix}$$

where v_1, v_2, v_3, v_4 are zero-mean Gaussian random variables with unit variance.

Figure 5 shows the tracking errors in Phases 1 and 3 in the presence of disturbance inputs. The errors in Part (a) are for the scheme with NN1 and those in Part (b) are for that with NN-2. Note that the transient phase is not shown. Evidently, the error magnitudes in the presence of disturbances are larger by a factor of about 5 (compare with **Figure 4**). Also, the error magnitudes are lower by about 50% with NN-2 than with NN-1 for reasons mentioned above. Nevertheless, the MPC scheme is able to limit the tracking errors within about 10cm in each direction when NN-1 is used and within about 5cm when NN-2 is used.



(a)

(b)

Figure 5: Position tracking errors in Phases 1 and 3 in the presence of disturbances: (a) with model NN-1, (b) with model NN-2

4.3.3 Performance in the Absence of Sway Control

In this part, we explore the performance of the MPC scheme in a scenario with a faulty sway (transverse direction) actuator. Even without a fault, the sway control considered in our scheme is limited because of the limitation in sway speed and sway-direction thrust. When sway control is completely absent, the AUV is considered to be underactuated and its maneuvering ability may be restricted in some situations. We consider this situation in the presence of disturbances as mentioned in Section 4.3.2. **Figure 6** shows the state and control components evolving under the MPC scheme (using NN-1) in this scenario. It can be observed that while the control inputs (except τ_Y, which is zero) and the velocities react and fluctuate to correct the effect of the disturbances that affect the system, position components vary rather smoothly with time.

The plots in **Figure 7** show 3D position errors ($e = \sqrt{(x - x_d)^2 + (y - y_d)^2 + (z - z_d)^2}$) during Phase 3 in the three different scenarios considered (in Sections 4.3.1~4.3.3).

The errors in the first figure are obtained with model NN-1 and those in the second figure are obtained with model NN-2. Clearly, the presence of disturbances increases the tracking error and the actuator fault further increases it. And, because of higher accuracy and smaller sampling time, with model NN-2, errors are smaller by a factor of about 2.

4.3.4 Computational Requirements

Solving the nonlinear MPC problem is computationally demanding. Algorithm 2 requires a repeated solution of a QP problem at every time step. It is found that, except at the first step of every phase, the algorithm converges in up to 4 or 5 steps. The initial step requires the search for an initial feasible solution which may take 5 to 10 or more steps depending on the horizon length. We use the QP solver qpOASES **[36]** in MATLAB in a Windows machine with Intel i7 1.8 GHz processor and 24 GB RAM for the computations. It is found that the computations (QP solving and other preparatory computations) to be made after the measurement of the state at each step to obtain the optimal solution take, on the average, about 19ms when NN-1 is used and about 15ms when NN-2 is used for a horizon length of $N = 10$. In the absence of sway control, since we have a fewer number of optimization variables, the corresponding average computation times are slightly smaller – 15ms and 12ms respectively. Clearly, these computation times are small fractions of the respective sampling times and therefore they do not adversely affect the implementability of the control scheme.

5. Conclusion

The problem of 3D trajectory tracking of AUVs under 4 or 3 degrees of maneuverability was addressed with an effective and efficient nonlinear MPC scheme that uses a suitable modeling of the DT dynamics of the system using a ff-NN. An accurate NN-based DT dynamics model simplifies the online state propagation and cost gradient computations when solving the nonlinear MPC optimization problem with a sequence of QPs. Realistic

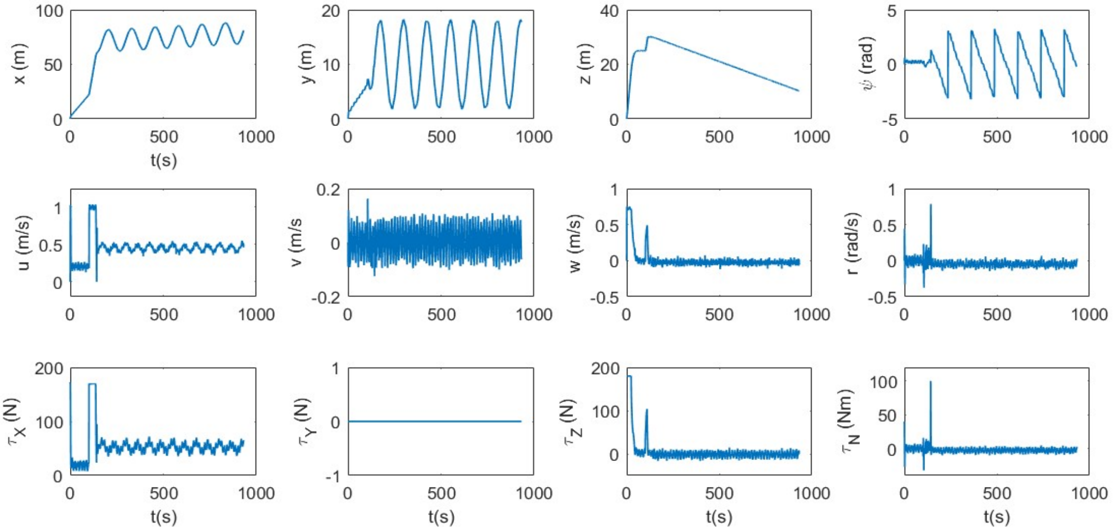


Figure 6: The evolution of state and control components during the mission without sway control in the presence of disturbances

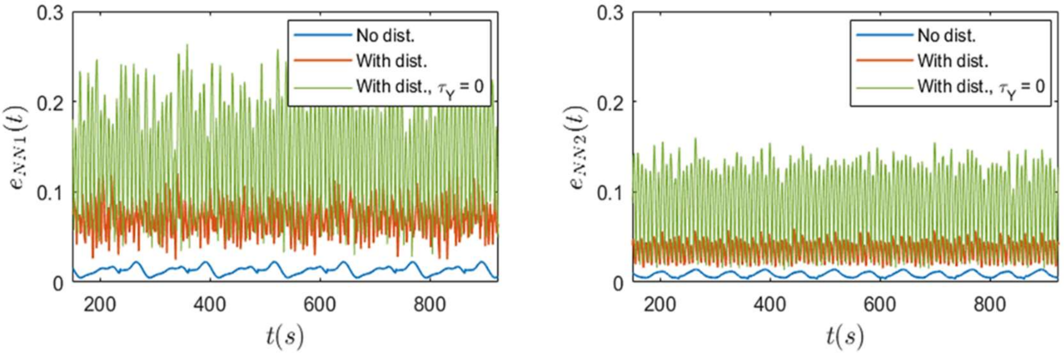


Figure 7: 3D position tracking errors in various scenarios with dynamics models NN-1 and NN2

numerical simulations have shown the effectiveness of the approach in various situations including the presence of random and non-random disturbances and/or the lack of maneuverability along the sway direction.

Acknowledgements

This paper was supported by Education and Research promotion program of KOREATECH.

Author Contributions

All relevant contributions are by the Corresponding author.

References

[1] T. Fossen, K. Y. Pettersen, and H. Nijmeijer, Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles, Lecture Notes in Control and Information Sciences, vol. 474, 2017.

[2] H. Singh, C. Roman, O. Pizarro, R. Eustice, and A. Can, "Towards high-resolution imaging from underwater vehicles," The International Journal of Robotics Research, vol. 26, no. 1, pp. 55-74, 2007.

[3] D. Sward, J. Monk, and N. Barrett, "A systematic review of remotely operated vehicle surveys for visually assessing fish assemblages," Frontiers in Marine Science, vol. 6, 2019.

[4] L. Wang, D. Zhu, W. Pang, and Y. Zhang, "A survey of underwater search for multi-target using multi-AUV: Task allocation, path planning, and formation control," Ocean Engineering, vol. 278, 114393, 2023.

[5] R. B. Wynn, et al., "Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience," Marine Geology, vol. 352, pp. 451-468, 2014.

[6] A. Wibisono, M. J. Piran, H. -K. Song, and B. M. Lee, "A survey on unmanned underwater vehicles: Challenges, enabling technologies, and future research directions," Sensors, vol. 23, no. 17, p. 7321, 2023.

[7] A. Bashir, S. Khan, N. Iqbal, S. Bashmal, S. Ullah, Fayyaz, and M. Usman, "A review of the various control algorithms for trajectory control of unmanned underwater vehicles," Sustainability, vol. 15, no. 20, p. 14691, 2023.

[8] J. Yuh, "Design and control of autonomous underwater robots: A survey," Autonomous Robots, vol. 8, pp. 7-24, 2000.

[9] K. Y. Pettersen and O. Egeland, "Time-varying exponential stabilization of the position and attitude of an underactuated autonomous underwater vehicle," IEEE Transactions on Automatic Control, vol. 44, no. 1, pp. 112-115, 1999.

[10] C. Silvestre and A. Pascoal, "Control of the INFANTE AUV using gain scheduled static output feedback," Control Engineering Practice, vol. 12, no. 12, pp. 1501-1509, 2004.

[11] C. Paliotta, E. Lefeber, K. Y. Pettersen, J. Pinto, M. Costa and J. T. de F. B. de Sousa, "Trajectory tracking and path following for underactuated marine vehicles," IEEE Transactions on Control Systems Technology, vol. 27, no. 4, pp. 1423-1437, 2019.

[12] M. Mat-Noh, R. Mohd-Mokhtar, M. R. Arshad, Z. M. Zain, and Q. Khan, "Review of sliding mode control application in autonomous underwater vehicles," Indian Journal of Geo Marine Sciences, vol. 48, no. 7, pp. 973-984, 2019.

[13] J. Nie, J. Yuh, E. Kardash, and T. I. Fossen, "On-board sensor-based adaptive control of small UUVs in very shallow water," International Journal of Adaptive Control and Signal Processing, vol. 14, no. 4, pp. 441-452, 2000.

[14] J. Guerrero, J. Torres, V. Creuze, and A. Chemori, "Trajectory tracking for autonomous underwater vehicle: An adaptive approach," Ocean Engineering, vol. 172, pp. 511-522, 2019.

[15] Z. Yan, M. Wang, and J. Xu, "Robust adaptive sliding mode control of underactuated autonomous underwater vehicles with uncertain dynamics," Ocean Engineering, vol. 173, pp. 802-809, 2019.

[16] L. Qiao and W. Zhang, "Trajectory tracking control of AUVs via adaptive fast nonsingular integral terminal sliding mode control," IEEE Transactions on Industrial Informatics, vol. 16, no. 2, pp. 1248-1258, 2020.

[17] B. Sun, D. Zhu, and S. Yang, "An optimized fuzzy control algorithm for three-dimensional AUV path planning," International Journal of Fuzzy Systems, vol. 20, no. 5, pp. 1-14, 2017.

[18] B. S. Park, "Neural network-based tracking control of underactuated autonomous underwater vehicles with model uncertainties," Journal of Dynamic Systems, Measurement, and Control, vol. 137, no. 2, p. 021004, 2015.

[19] J. Li, J. Du, and C. L. P. Chen, “Command-filtered robust adaptive NN control with the prescribed performance for the 3-D trajectory tracking of underactuated AUVs,” IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 11, pp. 6545-6557, 2022.

[20] J. M. Maciejowski, Predictive Control with Constraints, Prentice Hall, Harlow, England, 2002.

[21] J. Köhler, M. A. Müller, and F. Allgöwer, “Nonlinear reference tracking: An economic model predictive control perspective,” IEEE Transactions on Automatic Control, vol. 64, no. 1, pp. 254-269, 2019.

[22] A. Jayasiri, A. Nandan, S. Imtiaz, D. Spencer, S. Islam, and S. Ahmed, “Dynamic positioning of vessels using a UKF-based observer and an NMPC-based controller,” IEEE Transactions on Automation Science and Engineering, vol. 14, no. 4, pp. 1778-1785, 2017.

[23] W. Gan, D. Zhu, and D. Ji, “QPSO-model predictive control-based approach to dynamic trajectory tracking control for unmanned underwater vehicles,” Ocean Engineering, vol. 158, pp. 208-220, 2018.

[24] C. Shen, Y. Shi, and B. Buckham, “Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control,” IEEE Transactions on Industrial Electronics, vol. 65, no. 7, pp. 5796-5805, 2018.

[25] C. Liu, Q. Hu, X. Wang, and J. Yin, “Event-triggered-based nonlinear model predictive control for trajectory tracking of underactuated ship with multi-obstacle avoidance,” Ocean Engineering, vol. 253, 111278, 2022.

[26] W. Zhang, Q. Wang, W. Wu, X. Du, Y. Zhang, and P. Han, “Event-trigger NMPC for 3-D trajectory tracking of UUV with external disturbances,” Ocean Engineering, vol. 283, 115050, 2023.

[27] S. Heshmati-Alamdari, A. Nikou, and D. V. Dimarogonas, “Robust trajectory tracking control for underactuated autonomous underwater vehicles in uncertain environments,” IEEE Transactions on Automation Science and Engineering, vol. 18, no. 3, pp. 1288-1301, 2021.

[28] C. Shen, B. Buckham, and Y. Shi, “Modified C/GMRES algorithm for fast nonlinear model predictive tracking control of AUVs,” IEEE Transactions on Control Systems Technology, vol. 25, no. 5, pp. 1896-1904, 2017.

[29] A. Gautam and Y. C. Soh, “Stabilizing model predictive control using parameter-dependent dynamic policy for nonlinear systems modeled with neural networks,” Journal of Process Control, vol. 36, pp. 11-21, 2015.

[30] T. I. Fossen, Guidance and Control of Ocean Vehicles, John Willey & Sons, 1994.

[31] P. A. Miller, J. A. Farrell, Y. Zhao, and V. Djapic, “Autonomous underwater vehicle navigation,” IEEE Journal of Oceanic Engineering, vol. 35, no. 3, pp. 663-678, 2010.

[32] Y. -S. Han, M. -J. Kim, H. -I. Seo, and D. -H. Seo, “Direct orientation estimation through inertial odometry based on a deep transformer model,” Journal of Advanced Marine Engineering and Technology, vol. 48, no. 2, pp. 96-106, 2024.

[33] D. Q. Mayne, E. C. Kerrigan, E. V. Wyk, and P. Falugi, “Tube-based robust nonlinear model predictive control,” International Journal of Robust and Nonlinear Control, vol. 21, no. 11, pp. 1341-1353, 2011.

[34] A. Gautam, Y. -C. Chu, and Y. C. Soh, “Optimized dynamic policy for receding horizon control of linear time-varying systems with bounded disturbances,” IEEE Transactions on Automatic Control, vol. 57, no. 4, pp. 973-988, 2012.

[35] S. J. Wright and M. J. Tenny, “A feasible trust-region sequential quadratic programming algorithm,” SIAM Journal of Optimization, vol. 14, no. 4, pp. 1074-1105, 2004.

[36] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, “A parametric active-set algorithm for quadratic programming,” Mathematical Programming Computation, vol. 6, no. 4, pp. 327-363, 2014.