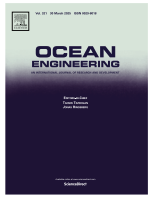
Ocean Engineering 340 (2025) 122137

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng



Research paper

Energy-optimal model predictive control for unmanned underwater vehicles in offshore aquaculture fish net-pen visual inspection

Thein Than Tun [a,b], Loulin Huang [a,*], Mark Anthony Preece [c]

[a] School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand
[b] Blue Economy Cooperative Research Centre, Launceston, Australia
[c] The New Zealand King Salmon Company, 7011, Nelson, New Zealand

ARTICLE INFO

Keywords:
Energy-optimal model predictive control (EO-MPC)
Nonlinear programming (NLP)
Unmanned underwater vehicle (UUV)
Fish net-pen visual inspection (FNVI)
Offshore aquaculture
Robot operating system (ROS)
Gazebo physics engine

ABSTRACT

Unmanned underwater vehicles are deployed to automate the production processes in offshore aquaculture, but the onboard power supply with limited energy capacity constrains the operational range and time. In this paper, a nonlinear energy-optimal Model Predictive Control (EO-MPC) is proposed to perform a 4-degree-of-freedom 3D fish net-pen visual inspection trajectory tracking while minimizing energy consumption. The EO-MPC problem with explicit energy-related terms in the performance index (PI) is transcribed into a nonlinear programming problem (NLP), solved via IPOPT, the open-sourced primal-dual interior point solver. Using the specifications of *Blue Endeavour Project* (the upcoming first-of-its-kind offshore salmon firm in New Zealand) of the New Zealand King Salmon Company and the work-class ROV called RexROV 2, theoretical fundamentals and practical implementation aspects are detailed, and four controllers are tested in high-fidelity simulation using Robot Operating System and Gazebo Physics Engine. In a general constrained operational working environment, the proposed EO-MPC controller saves 3.1 % - 21.4 % more energy than the conventional MPC (CO-MPC) while achieving better or equivalent trajectory tracking performance under different underwater current disturbance speeds (0.0 m/s, 0.5 m/s and 0.9 m/s).

1. Introduction

Inshore and nearshore aquaculture farms encounter both conflicting coastal usages (e.g., tourism, recreation, and conservation), as well as challenging growing conditions (e.g., warm water, low dissolved oxygen) (Chu et al., 2020). To access an improved environment for ideal fish growth, offshore aquaculture offers a cooler (e.g., $12°C - 16°C$) and suitable fish (e.g., King Salmon) farming space, especially in the context of New Zealand and Australia (Preece, 2021). With sustainable land-based farming and offshore aquaculture, the New Zealand aquaculture industry is expected to reach $3 billion market size by 2035 (The New Zealand Government, 2025). Aligning with this trend, the New Zealand King Salmon (NZKS), a leading fish farming company, will start the proof-of-concept phase of its offshore aquaculture project called *Blue Endeavour Project* (the first of its kind in New Zealand) from June 2025 (New Zealand King Salmon, 2024).

Currently, the NZKS company deploys a remotely controlled unmanned underwater vehicle (UUV, also known as ROV) to conduct visual inspection and fish net-pen cleaning tasks. Along with the relocation to offshore, the long-term deployment of workforce in a harsh working environment underscores the safety concern, high operating costs, and the availability of manpower. One way to mitigate those issues is to make UUVs autonomous. Therefore, the autonomous fish net-pen visual inspection (FNVI) with UUVs becomes a research trend (Amundsen et al., 2021; Akram et al., 2022), but to the best of our knowledge, there is currently no fully autonomous solution implemented in the industry. Note: UUV is the generic term for both autonomous underwater vehicle (AUV) and remotely operated underwater vehicle (ROV). AUV is used for the long-range operation without an umbilical cord, whereas ROV is used for the short-range operation with umbilical cord.

To develop the autonomous FNVI with an infrastructure investment (e.g., navigation systems and sensors for UUVs), the UUV must operate without an umbilical cord, which is used for communication and unlimited power supply, to fully benefit from its autonomous capability. The umbilical cord limits the operational range as it can entangle with the mooring lines of the fish farm when the autonomous UUV inspects around the fish net-pen. Without an umbilical cord, the UUV relies on the limited onboard power supply, and thus, an energy-optimal control of the UUV is essential to keep the working range and time required for the autonomous FNVI.

* Corresponding author.
E-mail addresses: theinthan.tun@autuni.ac.nz (T.T. Tun), loulin.huang@aut.ac.nz (L. Huang), mark.preece@kingsalmon.co.nz (M.A. Preece).

https://doi.org/10.1016/j.oceaneng.2025.122137
Received 23 March 2025; Received in revised form 21 June 2025; Accepted 5 July 2025
Available online 24 July 2025

In existing optimal control literature related to UUVs, most of which are developed from general optimal control schemes like Linear Quadratic Regulator (LQR), Linear Quadratic Tracking (LQT) control, and Model Predictive Control (MPC) (Claus and Bachmayer, 2016; Sarkar et al., 2016; Du et al., 2020). LQR is usually used for regulation problems (e.g., station-keeping, setpoint-keeping: at a particular depth), and LQT is used for tracking problems (e.g., trajectory tracking). Both are derived using the necessary and sufficient optimal conditions of Hamilton-Jacobi-Bellman (HJB) equations, and the resulting controller outputs are not explicitly imposed with state and input constraints. Unlike other controllers, Model Predictive Control (MPC) was proposed from the engineering perspective for the use in chemical process industry, and for the initial development of MPC, many have provided credits to the authors of Dynamic Matrix Control (DMC) from Shell and Model Predictive Heuristic Control (MPHC) (Cutler and Ramaker, 1980; Richalet et al., 1978). The present-day MPC is used for a general optimization problem in which a user-defined performance index (PI), subjected to the system dynamics and the constraints on states and control effort (system input), is minimized over the user-defined prediction horizon. Its capability to handle nonlinear dynamics and system constraints is an outstanding feature. Comprehensive reviews on the MPC history, commercial products, and its industrial applications are presented in Köhler et al. (2024), Schwenzer et al. (2021), Qin and Badgwell (2003), Morari and Lee (1999), Henson (1998). Stability and optimality proof for constrained nonlinear and linear MPC can be found in Mayne and Rawlings (2001), Mayne et al. (2000), Zheng and Morari (1995). Stability proofs with and without terminal constraints are presented in Fagiano and Teel (2012), Boccia et al. (2014), Grüne (2012).

MPC has been applied extensively in controlling UUVs, though few MPC schemes are for energy-optimal control. The 3D trajectory tracking control for a torpedo-shaped AUV with dual closed-loop MPC in which the outer-loop performs pose (position and orientation) tracking and the inner-loop performs twist (linear and angular velocity) tracking is demonstrated in Gong et al. (2022). Another work, which formulates the dual closed-loop MPC as multiobjective MPC (MOMPC), is proposed for controlling the rectangular-shaped Saab SeaEye Falcon UUV (Shen et al., 2019). The implementation of Lyapunov-based MPC on the same UUV is presented in Shen et al. (2018). A similar work without the outer-loop for a rectangular-shaped UUV called FlatFish using nonlinear MPC, integral Sliding Mode Control (ISMC) to handle uncertainties and filtered Smith Predictor (SP) for dead-time compensation is reported in Saback et al. (2020). The disturbance observer-based MPC for BlueROV2 (downgraded version of BlueROV2 Heavy Configuration) is demonstrated using Robot Operating System (ROS) and Gazebo Physics Engine (Hu et al., 2024).

Along with the application of MPC, it is important to identify the type of optimization problem, such as linear programming (LP), quadratic programming (QP), or nonlinear programming (NLP). Based on it, the modelling software (e.g., YALMIP, ACADO, and CasADi), considering the compatible programming languages (e.g., Python and C++), can be chosen to assist the formulation of the identified optimization problem and solve it by interfacing with the relevant solvers (e.g., Knitro, Gurobi, and IPOPT) (Drgoňa et al., 2020; Lofberg, 2004, 2025). Theoretical aspects of optimization problem formulation and numerical solvers, which play a crucial role in the actual implementation, are reported in Diehl et al. (2009), Zavala and Biegler (2009), Diehl (2013). One of the widely used open-source NLP solvers based on the primal-dual interior point method is called IPOPT (Wächter and Biegler, 2006). Along with the use of NLP, another important issue is the choice of the transcription (reformulation of MPC into NLP) methods: Direct Single Shooting (DSS), Direct Multiple Shooting (DMS), and Direct Collocation (DC), and other modified variants (Grüne, 2021). In DSS, the discretized control input is taken as the optimization variables, and the optimization is applied over the whole horizon using the initial state. In DMS, both discretized states and control inputs are taken as the optimization variables. At each step over the horizon, the difference between the system model's states and the previously optimized states is treated as the equality constraint, and this treatment is known as the path constraint (also known as the continuity constraint). DC is similar to DMS except that at each step over the horizon, the discretized states are not treated as piecewise-constants but piecewise-polynomials.

Energy-optimal control schemes of a UUV are developed from the optimal control methods presented above. An energy-optimal depth control (achieving a desired depth via hybrid propulsion system either with buoyancy changes with ballast system or lifting surface with thruster) for an autonomous underwater glider (AUG) using linear time-invariant (LTI) model with an infinite time Linear Quadratic Regulator (LQR) is presented in Claus and Bachmayer (2016). Via the adoption of Sliding Mode Control (SMC) output into the finite horizon Linear Quadratic Tracking (LQT) control, a robust energy-efficient trajectory tracking controller is reported for a slender-body UUV (Sarkar et al., 2016). Kinetic energy-optimal 3D motion planning and a Model Predictive Control (MPC) based trajectory tracking control for an AUV is addressed in Du et al. (2020). An energy-optimal AUV depth tracking control: the vertical planar motion (e.g., the sinusoidal forward movement) using MPC is reported in Yao et al. (2019). In all of the above-mentioned control schemes, the energy term on energy consumption is not explicitly defined, and a term on control efforts is used instead in the PI.

In some research, the power consumed by the thrusters (propulsion system) is used to define the term in the PI to reflect the energy consumption (Liang, Haojiao and Li, Huiping and Gao, Jian and Cui, Rongxin and Xu, Demin, 2023; Spangelo and Egeland, 1992; Yang et al., 2018, 2019, 2020, 2024). With the PI defined this way, MPC or NLP with DSS is applied to develop the energy-optimal controller for the ROV (Liang, Haojiao and Li, Huiping and Gao, Jian and Cui, Rongxin and Xu, Demin, 2023; Spangelo and Egeland, 1992). In Yang et al. (2018, 2019, 2020), EO-MPC and proportional-integral-derivative (PID) control are implemented respectively for controlling surge and yaw motions, and pitch and heave motions. Two decentralized MPC (DMPC) are proposed in Yang et al. (2024). In DMPC, the overall optimization problem is decomposed into several individual subproblems with their own PIs, and thus, it is only suitable for system dynamics that have weak coupling among the multi-dimensional motions (e.g., surge for forward-backward horizontal motion, sway for left-right horizontal motion, heave for up-down vertical motion, and Euler angles, roll, pitch and yaw, for rotational motions in the respective axes). In the above-mentioned control schemes, energy consumption by the UUVs to overcome hydrodynamics and the estimated energy required for the UUV to reach the upcoming waypoints, other than the power consumed by the thrusters, is not considered. This deficiency is addressed by the energy-optimal nonlinear robust MPC, solved by an A* -like algorithm, while taking account of the bounded uncertainties on ocean currents, produced using the Regional Ocean Modelling System (ROMS) in the UUV model (Huynh et al., 2015).

Most optimal controllers are based on an infinite horizon unconstrained optimization problem which does not meet the operational requirement of FNVI with limited allowable operational time and constraints on the UUV's state and control (e.g., the hard constraints on the thrust of the thrusters). In addition, in most energy-optimal controllers, the energy-relevant term in the PI is expressed as a quadratic control effort cost or the thruster's power function, which fails to accurately describe the real power consumption in the form of a non-convex function of the system states and the control efforts.

To tackle these issues, the main contributions of this work are summarized as follows.

- **Formulation of a novel EO-MPC problem**: To reflect real power consumption, a system-level power function is adopted in the PI, producing a finite-horizon constrained nonlinear non-convex EO-MPC problem.
- **Non-convex optimization in real-time**: From both theoretical and practical aspects, a detailed step-by-step approach to deal with the

non-convex optimization problem, including convexity test, NLP problem formulation, and NLP solver (e.g., IPOPT), is presented.

- **Investigation into the effects of non-convex system-level power function on energy optimization**: The insights into non-convex system-level power function and how its constraint activation affects energy optimization are discussed.
- **Real-time feasibility tested in high-fidelity simulation**: Using ROS and Gazebo Physics Engine, three EO-MPC controllers and one CO-MPC are simulated under three underwater current disturbance speeds (0.0 m/s, 0.5 m/s, and 0.9 m/s) normally found in the NZKS fish farm.
- **Application-oriented controller design**: The practicality of the controllers is considered by using the specifications of the RexROV 2 with Sub-Fighter 30K's published model parameters and the *Blue Endeavour Project* (the upcoming first-of-its-kind offshore salmon firm in New Zealand) of the New Zealand King Salmon company in high-fidelity simulations.

The remaining part of this paper is organized as follows. The development of a nonlinear state-space model from the kinematic and dynamic model of a 6-DoF UUV is presented in Section 2. Section 3 is on the details of the formulation of fish net-pen visual inspection as a constrained optimal control problem (COCP), followed by the discretization of COCP into a constrained MPC via the fourth-order Runge-Kutta method (RK4). The definitions of energy terms, followed by adoption of those in PIs of EO-MPC, are covered in Section 4. In Section 5, the transcription of EO-MPC into NLP via Direct Multiple Shooting (DMS), the necessary and sufficient conditions for optimality and Karush-Kuhn-Tucker (KKT) conditions, and the primal-dual interior point optimization method are discussed. Section 6 is on the preparation of high-fidelity simulation with ROS and Gazebo. High-fidelity simulation results are presented in Section 7. A summary of the research findings and the future work is presented in Section 8. The real-time implementation of the MPC algorithm as a ROS node is reported in Section A as Appendix.

2. Dynamic model of the UUV

Considering the working environments and the requirements of FNVI as shown in Fig. 1, a highly-maneuverable (6 DoF with 6 thrusters) rectangular-shaped UUV called RexROV 2 is adopted in this study and it generally consists of rigid body dynamics, hydrodynamics, hydrostatics and thruster allocation to fully describe a generic nonlinear UUV dynamic model which will be detailed in this section (Fossen, 2011; Tun et al., 2023).

Suppose Frame $\{b\}$ is assigned at the centre of mass/gravity as shown in Fig. 2, the simplified mass-matrix of the rigid body can be

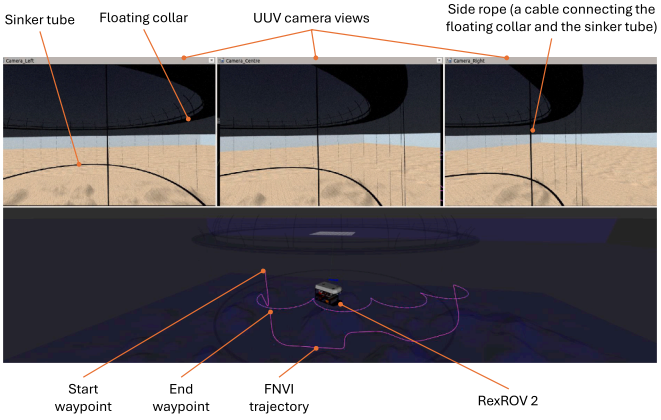


Fig. 1. RexROV 2 conducting FNVI in ROS using the specifications of the *Blue Endeavour Project*.

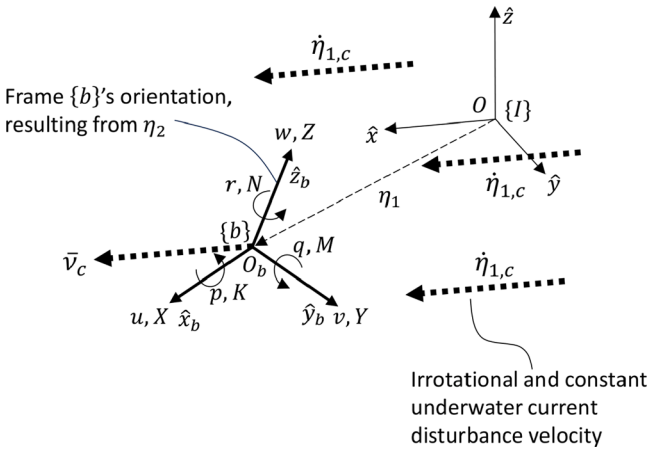


Fig. 2. UUV frames and nomenclature.

represented by

$$M_R = \begin{bmatrix} m_r \mathrm{I}_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_G \end{bmatrix} \tag{1}$$

where m_r is the UUV's mass, $\mathrm{I}_{3\times3} \in \mathbb{R}^{3\times3}$ is the identity matrix and $I_G \in \mathbb{R}^{3\times3}$ is the moment of inertia expressed in Frame $\{b\}$. Similarly, the simplified Coriolis-Centripetal acceleration matrix of the rigid body can be described by

$$C_R(\nu) = \begin{bmatrix} m_r[\nu_2\times] & 0_{3\times3} \\ 0_{3\times3} & -[(I_G\nu_2)\times] \end{bmatrix} \tag{2}$$

where $[(.)\times]$ is the skew-symmetric operator on $(.)$.

Assumption 1. During the operation, the UUV has three planes of symmetry and is completely submerged in the water and operated at low speed.

With Assumption 1, the off-diagonal components of the mass matrix contributed by the added mass are considered negligible.

$$M_A = \mathrm{diag}([X_{\dot{u}}\ Y_{\dot{v}}\ Z_{\dot{w}}\ K_{\dot{p}}\ M_{\dot{q}}\ N_{\dot{r}}]^T) \tag{3}$$

The Coriolis-Centripetal acceleration matrix due to the added mass can be described by

$$C_A(\nu) = \begin{bmatrix} 0_{3\times3} & -\big[(A_{11}\nu_1 + A_{12}\nu_2)\times\big] \\ -\big[(A_{11}\nu_1 + A_{12}\nu_2)\times\big] & -\big[(A_{21}\nu_1 + A_{22}\nu_2)\times\big] \end{bmatrix} \tag{4}$$

where

$$M_A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{5}$$

The damping effects can be approximated by linear and quadratic terms:

$$D_l = \mathrm{diag}\big([X_u\ Y_v\ Z_w\ K_p\ M_q\ N_r]^T\big) \tag{6}$$

$$D_q(\nu) = \mathrm{diag}\Big(\big[X_{u|u|}|u|\ Y_{v|v|}|v|\ Z_{w|w|}|w|\ K_{p|p|}|p| \\ M_{q|q|}|q|\ N_{r|r|}|r|\big]^T\Big) \tag{7}$$

The restoring forces and moments due to the gravitational force and buoyancy can be described by

$$g(\eta) = \begin{bmatrix} R_I^b(\eta)\big(f_G^I + f_B^I\big) \\ r_G^b \times R_I^b(\eta) f_G^I + r_B^b \times R_I^b(\eta) f_B^I \end{bmatrix} \tag{8}$$

where $f_G^I = [0\ 0\ -w_G]^T$, $f_B^I = [0\ 0\ w_B]^T$, w_G is the UUV's weight and w_B is its buoyancy. Suppose the direction vector of the j^{th} thruster with the thrust magnitude of $f_{T,j}$, expressed in Frame $\{T_j\}$ is denoted by

$$d_{T,j} = [x_{T,j}\ y_{T,j}\ z_{T,j}]^T, \tag{9}$$

the j^{th} thrust vector is $d_{T,j} f_{T,j}$.

The wrench propagated by the j^{th} thruster can be described by

$$\tau_{f_{T,j}} = \begin{bmatrix} R^b_{T_j} d_{T,j} f_{T,j} \\ \left[r^b_{T_j}\times\right] R^b_{T_j} d_{T,j} f_{T,j} \end{bmatrix} = B_{T,j} f_{T,j} \tag{10}$$

Therefore, the total wrench propagated by m number of thrusters can be described by

$$\tau = [B_{T,1}\ B_{T,2}\ \dots\ B_{T,m}] f_T = B_T f_T \tag{11}$$

With the terms defined above, the UUV's dynamic model is expressed as Fossen (2011)

$$M\dot{\nu} + (C(\nu) + D(\nu))\nu + g(\eta) = \tau \left(\triangleq B_T f_T\right) \tag{12}$$

where $M = M_R + M_A \in \mathbb{R}^{6\times6}$ is the inertial matrix, $C(\nu) = C_R(\nu) + C_A(\nu) \in \mathbb{R}^{6\times6}$ is the Coriolis-Centripetal acceleration matrix, $D(\nu) = D_l + D_q(\nu) \in \mathbb{R}^{6\times6}$ is the damping matrix, $g(\eta) \in \mathbb{R}^6$ is the vector of gravitational force, buoyancy and the resulting restoring forces and moments. As shown in Fig. 2, $\tau_1 = [X\ Y\ Z]^T$, $\tau_2 = [K\ M\ N]^T$, $\tau = [\tau_1\ \tau_2]^T \in \mathbb{R}^6$ is the wrench — input force and moment mapped from the thrust vector, $f_T \in \mathbb{R}^m$, through thruster control/allocation matrix (TCM/TAM), $B_T \in \mathbb{R}^{6\times m}$, $\eta_1 = [x\ y\ z]^T$ is the position vector of the center of mass (origin of the body frame) of the UUV, $\eta_2 = [\phi\ \theta\ \psi]^T$ is the orientation (Euler angles) vector of the UUV, $\eta = [\eta_1\ \eta_2]^T \in \mathbb{R}^6$ is the pose of the UUV, all of which are described with respect to and expressed in the fixed inertial frame: Frame $\{I\}$, $\nu_1 = [u\ v\ w]^T$, $\nu_2 = [p\ q\ r]^T$, $\nu = [\nu_1\ \nu_2]^T \in \mathbb{R}^6$ is the twist — linear and angular velocity vector of the UUV w.r.t Frame $\{I\}$ expressed in the UUV's body frame: Frame $\{b\}$ which is related to $\dot{\eta}$ by

$$\dot{\eta} = J^I_b(\eta)\nu \tag{13}$$

where $J^I_b(\eta) \in \mathbb{R}^{6\times6}$ is the Jacobian.

$$J^I_b(\eta) = \begin{bmatrix} R^I_b(\eta) & 0_{3\times3} \\ 0_{3\times3} & T^I_b(\eta) \end{bmatrix}$$

$$R^I_b(\eta) = \begin{bmatrix} R^I_{b,c1}(\eta) & R^I_{b,c2}(\eta) & R^I_{b,c3}(\eta) \end{bmatrix},\ R^b_I(\eta) = R^I_b(\eta)^T$$

$$R^I_{b,c1}(\eta) = \begin{bmatrix} \mathrm{c}(\psi)\mathrm{c}(\theta) \\ \mathrm{s}(\psi)\mathrm{c}(\theta) \\ -\mathrm{s}(\theta) \end{bmatrix}$$

$$R^I_{b,c2}(\eta) = \begin{bmatrix} -\mathrm{s}(\psi)\mathrm{c}(\phi) + \mathrm{c}(\psi)\mathrm{s}(\theta)\mathrm{s}(\phi) \\ \mathrm{c}(\psi)\mathrm{c}(\phi) + \mathrm{s}(\psi)\mathrm{s}(\theta)\mathrm{s}(\phi) \\ \mathrm{c}(\theta)\mathrm{s}(\phi) \end{bmatrix}$$

$$R^I_{b,c3}(\eta) = \begin{bmatrix} \mathrm{s}(\psi)\mathrm{s}(\phi) + \mathrm{c}(\psi)\mathrm{s}(\theta)\mathrm{c}(\phi) \\ -\mathrm{c}(\psi)\mathrm{s}(\phi) + \mathrm{s}(\psi)\mathrm{s}(\theta)\mathrm{c}(\phi) \\ \mathrm{c}(\theta)\mathrm{c}(\phi) \end{bmatrix}$$

$$T^I_b(\eta) = \begin{bmatrix} 1 & \mathrm{s}(\phi)\mathrm{t}(\theta) & \mathrm{c}(\phi)\mathrm{t}(\theta) \\ 0 & \mathrm{c}(\phi) & -\mathrm{s}(\phi) \\ 0 & \mathrm{s}(\phi)/\mathrm{c}(\theta) & \mathrm{c}(\phi)/\mathrm{c}(\theta) \end{bmatrix}, \theta \neq \pm\frac{\pi}{2},$$

$$\mathrm{s}(.) = \sin(.), \mathrm{c}(.) = \cos(.), \mathrm{t}(.) = \tan(.)$$

From Eq. (12), we have

$$\dot{\nu} = M^{-1}[-(C(\nu) + D(\nu))\nu - g(\eta)] + M^{-1}B_T f_T \tag{14}$$

From Eqs. (13) and (14), the nonlinear state-space model can be written as follows.

$$\dot{\zeta} = \begin{bmatrix} J^I_b(\zeta_1)\zeta_2 \\ M^{-1}[-(C(\zeta_2) + D(\zeta_2))\zeta_2 - g(\zeta_1)] \end{bmatrix} + \begin{bmatrix} 0_{6\times m} \\ M^{-1}B_T \end{bmatrix} f_T = h(\zeta, f_T) \tag{15}$$

where $\zeta = [\zeta_1\ \zeta_2]^T = [\eta\ \nu]^T \in \mathbb{R}^{12}$.

3. Formulation of constrained model predictive control for trajectory tracking

Visual inspection of a fish net-pen by a UUV can be formulated as a constrained optimal control problem (COCP), described by

$$V^*(\zeta^*, f^*_T) = \min_{f_T(.)\in\mathbb{F}} J(\zeta, f_T) \tag{16}$$

$$\text{subject to}\quad \dot{\zeta}(t) = h(\zeta(t), f_T(t))$$
$$\zeta(t) \in \mathbb{S},\ f_T \in \mathbb{F},$$
$$t \in [0, T].$$

where $J(.) = \int_0^T \ell(\zeta(t), f_T(t))dt$ is the PI, $\ell(.) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^+_0$ is the stage/running cost associated with the fish net trajectory tracking by the UUV. An optimal value or parameter is indicated with the superscript *.

Definition 1. [Admissible state set: $\mathbb{S}$]

$$\mathbb{S} = \{\zeta \in \mathbb{R}^n \mid \zeta_{i,\min} \le \zeta_i \le \zeta_{i,\max},\ \exists\ f_T \in \mathbb{F} \text{ over } N_p\}$$

Definition 2. [Admissible input set: $\mathbb{F}$]

$$\mathbb{F} = \{f_T \in \mathbb{R}^m \mid \quad f_{T,i,\min} \le f_{T,i} \le f_{T,i,\max},$$
$$h(\zeta(k=0), f_T) \in \mathbb{S} \text{ over } N_c\}$$

Such a problem as shown in Eq. (16) can be generally solved by Dynamic Programming (DP) or its continuous form known as Hamilton-Jacobi-Bellman (HJB) (Falcone, 2021), Pontryagin's Minimum Principle using the Calculus of Variations (Vinter, 2021) and numerical methods, namely Direct Single Shooting (DSS), Direct Multiple Shooting (DMS) and Direct Collocation (DC) for transcription and followed by solving the discretized optimal control problem like MPC (Grüne, 2021).

In this paper, the continuous-time COCP will first be transformed into the discrete-time COCP that will be treated as an MPC problem. The resulting discrete-time MPC will be transcribed into NLP via the DMS method and, subsequently, solved by one of the well-known open-sourced NLP solvers called Interior Point Optimizer (IPOPT) based on the primal-dual interior point method.

As the first step to develop the MPC, the COCP is discretized. By applying 4th order Runge-Kutta method, the UUV's state is described in discrete-time as

$$\zeta(i+1) = \zeta(i) + \frac{h_s}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) = h(\zeta(i), f_T(i)) \tag{17}$$

$$k_1 = h(\zeta(i), f_T(i)),$$
$$k_2 = h(\zeta(i) + h_s\frac{k_1}{2}, f_T(i)),$$
$$k_3 = h(\zeta(i) + h_s\frac{k_2}{2}, f_T(i)),$$
$$k_4 = h(\zeta(i) + h_s k_3, f_T(i))$$

where $h_s \in \mathbb{R}^+$ is the step size, $i \in \mathbb{R}^+_0$ is the sampling instant.

Now, the discrete version of COCP in the framework of MPC at the current sampling time instant i can be expressed as follows.

$$\min_{f_T(.)\in\mathbb{F}} J_{N_p}(\zeta(i), f_T) = \ell(\zeta(N_p)) + \sum_{k=0}^{N_p-1} \ell(\zeta(k), f_T(k)) \tag{18}$$

$$\text{subject to } \zeta(k+1) = h(\zeta(k), f_T(k)) \in \mathbb{S}, \forall k \in [0, N_p],$$
$$f_T(k) \in \mathbb{F}, \forall k \in [0, N_c],$$
$$\text{Warm-start} \begin{cases} \zeta_{N_p}(k=0) = \zeta^*_{N_p}(i-1) \in \mathbb{R}^{n\times N_p}, \\ f_{T,N_c}(k=0) = f^*_{T,N_c}(i-1) \in \mathbb{R}^{m\times N_c}. \end{cases}$$

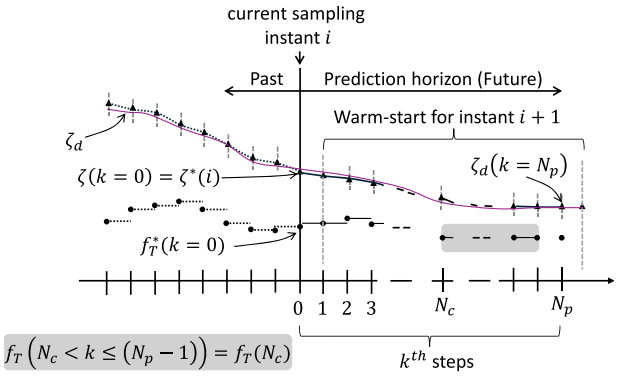


Fig. 3. Illustration of MPC's working principles. Note: the control effort, f_T^*, is under zero-order hold.

where N_p is the prediction horizon, N_c is the control horizon, $N_c < N_p$, $\ell(k)$ is the specified stage/running cost up to N_p, $\ell(N_p) = e^T Q_f e$ is the terminal cost, $Q_f = Q_f^T \in \mathbb{R}^{n\times n} \geq 0$ is the terminal error penalty weighting matrix, $e = \zeta_d - \zeta$ is the tracking error, ζ_d is the desired UUV state or the reference trajectory, $h(\zeta(k), f_T(k))$ is the value of $h(\zeta(i), f_T(i))$ at k^{th} step as shown in Eq. (17) and Fig. 3. $J_{N_p}(\zeta(i), f_T)$ is the summed PI over N_p, initialized using the existing optimal $\zeta^*_{N_p}(i-1)$ and $f^*_{T,N_c}(i-1)$; this method is also called warm-start. In the same fashion, the currently optimized solution $\zeta^*_{N_p}(i)$ and $f^*_{T,N_c}(i)$ will be reused as the warm-start for the time instant $i+1$ after left-shifting operation on the solution matrix. In the complete form, $\zeta(i)$ represents $\zeta(i, k=0,1,\ldots,N_p)$ which is the state at k^{th} step at the current sampling instant i. For ease of readability, $\zeta(i)$ will be used and so are other parameters in the same fashion. For the case $N_c < N_p$, $f_T(k)$ for the horizon between N_c and $N_p - 1$, it can be assigned equivalent to $f_T(N_c)$, resulting in $f_T(N_c < k < (N_p - 1)) = f_T(N_c)$. Therefore, the overall description of the discrete-time MPC process Eq. (18) can be illustrated as shown in Fig. 3.

Alternatively, from the discrete-time MPC perspectives, FNVI can be interpreted as a constrained optimal control problem to maneuver $\zeta(i, k=0)$ to $\zeta(i, k=N_p) = \zeta_d$ searching for $f_T(k=0,\ldots,N_c)$ that minimizes $J_{N_p}(.)$ throughout N_p. This process can be illustrated as shown in Fig. 4.

4. Energy-optimal model predictive control (EO-MPC)

After formulating MPC in discrete-time, the stage/running cost, $\ell(.)$, needs to be defined explicitly. For instance, $\ell(.)$ for the conventional tracking MPC is usually defined as follows (Li et al., 2024).

$$\ell(.)_{\text{CO-MPC1}} = e^T Q e + f_T^T R f_T \tag{19}$$

where $Q = Q^T \in \mathbb{R}^{n\times n} \geq 0$ is the error penalty weighting matrix, $R = R^T \in \mathbb{R}^{m\times m} > 0$ is the control effort penalty weighting matrix. Similarly, the following subsection will detail how to adopt energy-related terms in the stage/running cost, $\ell(.)$, to formulate EO-MPC.

Note: As $\ell(\zeta(N_p))$ mentioned in Eq. (18) is the same for all MPC variants reported in this paper, it will not be repeated in the formulations of different $\ell(.)$.

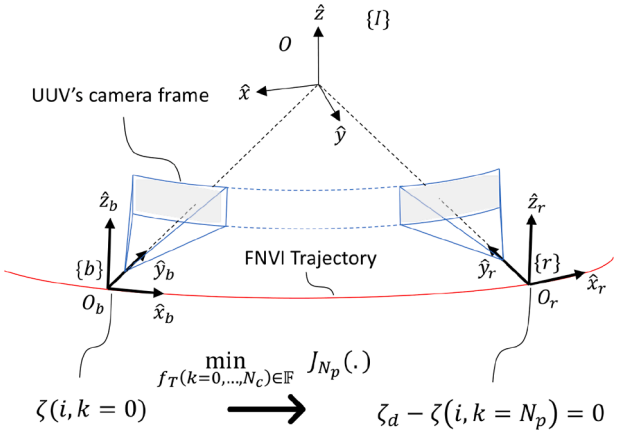


Fig. 4. FNVI from discrete-time MPC perspectives.

4.1. Energy associated with the UUV's motion and thrust

To maneuver the UUV as desired, it requires the power being injected into the system using the onboard power supply (e.g., battery) as follows.

$$p(f_T, v) = f_T^T B_T^T v \tag{20}$$

Eq. (20) must be utilized at all time unless there are no external disturbances and no desired motion.

Therefore, to minimize this power term, it can be added to CO-MPC1, resulting in

$$\ell(.)_{\text{EO-MPC1}} = e^T Q e + f_T^T R f_T + \gamma_1 p(f_T, v) \tag{21}$$

where $p_{\min}(f_T, v) \leq p(f_T, v) \leq p_{\max}(f_T, v)$, $\gamma_1 \in \mathbb{R}^+$ is the penalty weight on power consumption. **Note:** $p_{\max}(f_T, v)$ and $p_{\min}(f_T, v)$ are system-level power and not the total power rating of thrusters. They can be decided by observing Eq. (20) during CO-MPC1 experiment as the baseline maximum and minimum system-level power consumption. Although the first two terms in Eq. (21) are clearly convex functions, the last term needs to be examined of convexity so that the appropriate MPC solver can be chosen. To do so, the convexity test using Hessian Matrix can be conducted as shown below. Using the eigenvalue equation of $\nabla^2 p y = \lambda y$ and $\exists\ \nabla p,\ \nabla^2 p$,

$$\det(\nabla^2 p - \lambda I) = 0 \implies \begin{cases} \text{convex} & \text{if } \lambda \in \mathbb{R}^+ \\ \text{concave} & \text{if } \lambda \in \mathbb{R}^- \\ \text{non-convex} & \text{if } \lambda \in \mathbb{R} \end{cases}$$

where

$$\nabla p = \begin{bmatrix} \frac{\partial p}{\partial f_T} \\ \frac{\partial p}{\partial v} \end{bmatrix} = \begin{bmatrix} v^T B_T \\ f_T^T B_T^T \end{bmatrix}$$

$$\nabla^2 p = \begin{bmatrix} \frac{\partial^2 p}{\partial f_T^2} & \frac{\partial^2 p}{\partial f_T \partial v} \\ \frac{\partial^2 p}{\partial v \partial f_T} & \frac{\partial^2 p}{\partial v^2} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{6\times 6} & B_T^T \\ B_T & \mathbf{0}_{6\times 6} \end{bmatrix}$$

Using the numerical value of B_T provided in Section 6, $p(f_T, v)$ is proven as a non-convex function, which makes Eq. (21) overall non-convex. Therefore, the proposed MPC problem becomes constrained nonlinear non-convex optimisation problem with terminal state constraints.

Note: Generally speaking, convex optimisation yields the global minimum whereas non-convex optimisation produces the local minimum. Regardless, for the trajectory tracking in the field of robotics, the local minimum within the constraints and subjected to the system dynamics is sufficient given the fact that, in real-time, the system dynamics changes swiftly and is imposed with soft (e.g., state) and hard (e.g., input) constraints.

4.2. Energy associated with the large change in control effort

A large change or variation in control effort, Δf_T, in a short period of time can make the resulting motion energy-demanding (Veeraklaew et al., 2005) and its non-smooth (jerky) motion results in poor visual feedback in analysing the fish net-pen's condition. This can be explained as follows. Based on Eq. (12),

$$\Delta\tau = B_T \Delta f_T \tag{22}$$

As real-time control is quite fast enough that by taking limit as time difference approaches zero and for the sake of simplicity, by considering only the rigid body part of Eq. (12), it can be shown that

$$\left.\begin{aligned} \dot{\tau} &= B_T \dot{f}_T \\ \dot{\tau} &= M_R \ddot{\nu} = M_R j \end{aligned}\right\} \quad \dot{f}_T \implies j \tag{23}$$

where M_R is the mass matrix of the rigid body and $j \in \mathbb{R}^6$ is the jerk. Therefore, minimizing the jerk via minimization of variation in control effort should be conducted to achieve energy-optimality and smooth motion profile (Li et al., 2023; Huang et al., 2018). This approach can be described by

$$\ell(.)_{\text{EO-MPC2}} = e^T Q e + f_T^T R f_T + \Delta f_T^T R_\delta \Delta f_T \tag{24}$$

where $R_\delta = R_\delta^T = \gamma_2 R \in \mathbb{R}^{m\times m} > 0$ is the penalty weighting matrix for the large change in control effort, $\gamma_2 \in \mathbb{R}^+$ is the penalty multiplier, $\Delta f_T = f_T(k) - f_T(k-1) \in \mathbb{R}^m$ is the change in control effort between k^{th} and $(k-1)^{th}$ horizon steps. When $\Delta f_T \approx 0$, $\Delta f_T^T R_\delta \Delta f_T \approx 0$. Therefore, Eq. (24) still needs $f_T^T R f_T$ term.

4.3. Incorporation of energy terms

After the discussion on energy associated with the UUV's motion and thrust and energy associated with the large change in control efforts, this subsection will propose the incorporation of those energy terms as each of them is distinct in its formulation — the terms are independent with different perspectives on energy consumption. As proposed,

$$\begin{aligned} \ell(.)_{\text{EO-MPC3}} = e^T Q e + f_T^T R f_T + \gamma_1 p(f_T, \nu) \\ + \Delta f_T^T R_\delta \Delta f_T \end{aligned} \tag{25}$$

This $\ell(.)_{\text{EO-MPC3}}$ is expected to consume energy less than or equal to $\ell(.)_{\text{EO-MPC1}}$ with smooth motion profile.

Now, four different types of the stage/running cost, including the conventional one, are proposed and the following section will detail the transcription of the discrete-time MPC to NLP.

5. Nonlinear programming problem (NLP)

Unlike HJB or the indirect method, direct (numerical) methods take the approach of "discretize first, then optimize". Similar to the discretized PI with its constraints as shown in Eq. (18), the NLP can be written as follows.

$$\min_{\beta \in c_{\text{ineq}}(\beta), c_{\text{eq}}(\beta)} \Phi(\beta) = J_{N_p}(\beta) \tag{26}$$

subject to

$$c_{\text{eq}}(\beta) = \begin{bmatrix} \hat{\zeta}(0) - \zeta(0) \\ h(\zeta(0), f_T(0)) - \zeta(1) \\ \vdots \\ h(\zeta(N_p-2), f_T(N_p-2)) - \zeta(N_p-1) \\ \zeta_d - \zeta(N_p) \end{bmatrix} = 0$$

For Eqs. (19) & (24),

$$c_{\text{ineq}}(\beta) = \begin{bmatrix} c_{\text{ineq}}(\zeta(0), f_T(0)) \\ \vdots \\ c_{\text{ineq}}(\zeta(N_p-1), f_T(N_p-1)) \\ c_{\text{ineq}}(\zeta(N_p)) \end{bmatrix} \leq 0$$

For Eqs. (21) & (25),

$$c_{\text{ineq}}(\beta) = \begin{bmatrix} c_{\text{ineq}}(\zeta(0), f_T(0), p(0)) \\ \vdots \\ c_{\text{ineq}}(\zeta(N_p-1), f_T(N_p-1), p(N_p-1)) \\ c_{\text{ineq}}(\zeta(N_p)) \end{bmatrix} \leq 0$$

where $\beta = [\zeta(0) \dots \zeta(N_p)\ f_T(0) \dots f_T(N_c)]^T \in \mathbb{R}^{(n.N_p + m.N_c)\times 1}$ is the NLP decision variable. Note the notation difference between $f_T(N_c)$ in β and $f_T(N_p-1)$ in $c_{\text{ineq}}(\beta)$ as the former applies exactly up to any $N_c < N_p$ (e.g., $N_c = 5$, $N_p = 15$) for optimization but the latter applies up to $N_p - 1$ for constraints. $\Phi(\beta)$ is the NLP PI. $\hat{\zeta}(0)$ is the initial condition. $c_{\text{ineq}}(\beta)$ consists of inequality constraints of state, control effort, and power consumption. $c_{\text{eq}}(\beta)$ consists of the terminal constraint $(\zeta_d - \zeta(N_p))$ and the path constraint (resulting from DMS), also known as the shooting gap.

5.1. Necessary and sufficient conditions for optimality and karush-kuhn-tucker (KKT) conditions

Just like the necessary and sufficient conditions for optimality in the optimal control using Hamilton-Jacobi-Bellman equation, in NLP, there exists necessary condition for local optimality via KKT condition and sufficient condition for local optimality via the existence of positive Hessian in the feasible direction (Diehl and Gros, 2016; Nocedal and Wright, 2006).

Definition 3. [Feasible set: $\mathbb{B}$]

$$\mathbb{B} = \{\beta \mid c_{\text{ineq}}(\beta) \leq 0,\ c_{\text{eq}}(\beta) = 0\}$$

Definition 4. [Index set of active constraint: $\mathbb{I}$]

$$\mathbb{I}(\beta) = \{i \mid c_{\text{ineq},i}(\beta) = 0\}$$

Definition 5. [Active constraint set: $\mathbb{A}$]

$$\mathbb{A}(\beta) = \{c_{\text{ineq},i}(\beta) = 0,\ \forall i \in \mathbb{I}(\beta)\}$$

Definition 6. [Regular point]

$$\begin{aligned} \beta \in \mathbb{B} \text{ is regular} \iff \exists\ \nabla c_{\text{ineq},\mathbb{A}}(\beta),\ \nabla c_{\text{eq}}(\beta), \\ rank\left[\nabla c_{\text{ineq},\mathbb{A}}(\beta)\ \nabla c_{\text{eq}}(\beta)\right] = N_\beta. \end{aligned}$$

where N_β is the number of equality constraints and active inequality constraint.
In other words, $\beta \in \mathbb{B}$ fulfills the Linear Independence Constraint Qualification (LICQ).

Definition 7. [Feasible direction set: $\mathbb{D}$]

$$\mathbb{D} = \{\delta \mid \nabla c_{\text{eq}}^T(\beta^*)\delta = 0,\ \nabla c_{\text{ineq},i}^T(\beta^*)\delta \leq 0,\ \forall i \in \mathbb{I}(\beta^*)\}$$

5.1.1. KKT conditions

KKT Conditions can be written as follows for the NLP objective function in Eq. (26).

Dual feasibility: $\nabla_\beta \mathcal{L}(\beta^*, \alpha^*) = 0,\ \mu^* \geq 0,$

Primal feasibility: $c_{\text{eq}}(\beta^*) = 0,\ c_{\text{ineq}}(\beta^*) \leq 0,$

Complementarity slackness: $\mu_i^* c_{\text{ineq,i}}(\beta^*) = 0, \forall i \in \mathbb{I}(\beta).$

where $\mathcal{L} = \Phi(\beta) + \lambda^T c_{\text{eq}}(\beta) + \mu^T c_{\text{ineq}}(\beta),\ \alpha = [\lambda\ \mu]^T.$

5.1.2. First-order necessary condition (FONC) for optimality

Based on the above-mentioned KKT conditions, FONC can be described as follows.

$\beta^*$is regular $\implies$ $\exists$ unique α^*s.t. (β^*, α^*) is a KKT point.

5.1.3. Second-order sufficient condition (SOSC) for optimality

Given

- $\Phi(\beta)$, $c_{\text{ineq}}(\beta)$ and $c_{\text{eq}}(\beta)$ are twice differentiable,
- FONC is fulfilled,
- $\mathbb{D}$ as shown in Definition 7 exists.
- $\exists\ a^T \nabla^2_\beta \mathcal{L}(\beta^*, \alpha^*) a > 0, \forall a \neq 0, a \in \mathbb{D}, \forall\ i \in \mathbb{I}(\beta),\ \nabla c^T_{\text{ineq},i}(\beta) a = 0,\ \mu^*_i > 0.$

Then,

β^* is a local minimum.

After the details of necessary and sufficient conditions for local optimality, the following subsection will discuss on restructuring KKT conditions together with slack formulation for primal-dual interior point optimization method.

5.2. Interior point optimization (IPO)

5.2.1. Primal-dual IPO (PD-IPO) method

In a particular PD-IPO method, called IPOPT, the original KKT conditions are restructured to achieve the smoothed complementarity slackness and to ease the step-size selection as follow (Diehl and Gros, 2016; De Schutter et al., 2023).

$$\left.\begin{aligned} \nabla_\beta \mathcal{L}(\beta, \alpha) &= 0, \\ c_{\text{eq}}(\beta) &= 0, \\ c_{\text{ineq}}(\beta) + s &= 0, \\ \mu_i s_i - \kappa &= 0, \\ s > 0,\ \mu &> 0 \end{aligned}\right\} \text{ Slack formulation}$$

where $\nabla_\beta \mathcal{L}(\beta, \alpha) = \nabla\Phi(\beta) + \nabla c_{\text{eq}}(\beta)\lambda + \nabla c_{\text{ineq}}(\beta)\mu$. Let

$$r_\kappa(\beta, \alpha, s) = \begin{bmatrix} \nabla_\beta \mathcal{L}(\beta, \alpha) \\ c_{\text{eq}}(\beta) \\ c_{\text{ineq}}(\beta) + s \\ \mu_i s_i - \kappa \end{bmatrix} = 0 \tag{27}$$

Then, the Newton direction a can be defined by

$$\nabla r^T_\kappa(\beta, \alpha, s) a + r_\kappa(\beta, \alpha, s) = 0 \tag{28}$$

which results in

$$a = -\nabla r^{-1}_\kappa(\beta, \alpha, s) r_\kappa(\beta, \alpha, s) \tag{29}$$

where

$$\nabla r_\kappa(\beta, \alpha, s) = \begin{bmatrix} \mathbf{H} & \nabla c_{\text{eq}}(\beta) & \nabla c_{\text{ineq}}(\beta) & 0 \\ \nabla c_{\text{eq}}(\beta)^T & 0 & 0 & 0 \\ \nabla c_{\text{ineq}}(\beta)^T & 0 & 0 & \text{I} \\ 0 & 0 & \text{diag}(s) & \text{diag}(\mu) \end{bmatrix}$$

$$a = \begin{bmatrix} \Delta\beta \\ \Delta\lambda \\ \Delta\mu \\ \Delta s \end{bmatrix},\ \text{Hessian: } \mathbf{H} = \nabla^2_\beta \mathcal{L}(\beta, \alpha)$$

The Newton step can be taken as

$$\beta_{j+1} = \beta_j + a_\delta \Delta\beta$$
$$\lambda_{j+1} = \lambda_j + a_\delta \Delta\lambda$$
$$\mu_{j+1} = \mu_j + a_\delta \Delta\mu$$
$$s_{j+1} = s_j + a_\delta \Delta s$$

where $a_\delta \in (0, a_{\delta,lb,ub}]$ is computed from the following conditions.

$a_{\delta,lb,ub} \in (0, 1], \epsilon \ll 0$

$s_j + a_{\delta,lb,ub} \Delta s_j \geq \epsilon s$

$\mu_j + a_{\delta,lb,ub} \Delta\mu_j \geq \epsilon\mu$

The Newton direction calculation as shown in Eq. (29) is carried out until the maximum number of iteration is exceeded or $r_\kappa(\beta_{j+1}, \alpha_{j+1}, s_{j+1}) < \epsilon_{\text{set}}$, yielding the optimal NLP decision variable $\beta^* = \beta_{j+1}$. **Note:** Due to the numerial approach, Eq. (27) can never be fulfilled exactly but $r_\kappa(\beta_{j+1}, \alpha_{j+1}, s_{j+1}) < \epsilon_{\text{set}}$, (e.g., $\epsilon_{\text{set}} = 10^{-6}$).

Although Eq. (29) seems reasonably simple conceptually, there involves a lot of implementation work, and the development of efficient PD-IPO solver is not the main topic of this paper. Therefore, for the implementation in this work, one of the open-sourced NLP solvers called IPOPT which uses PD-IPO method along with CasADi is used to solve the NLP version of energy-optimal MPC tracking problem (Wächter and Biegler, 2006; Andersson et al., 2019).

6. High-fidelity simulation with ROS and gazebo

For safety and feasibility of the real-world experiment, it requires the pre-validation of the proposed controllers' performance in high-fidelity simulation using the actual specifications of the offshore aquaculture farm and the UUV as shown in Tables 1 and 2 (Berg, 2012; Tun et al., 2023). In this work, the UUV Simulator is modified to simulate FNVI operations in high-fidelity according to the specifications of the *Blue Endeavour Project* (Manhães et al., 2016; Musa Morena Marcusso Manhães, 2023; Musa Morena Marcusso Manhães, 2023) and the simulation parameters are reported in the followings.

6.1. Trajectory

Based on the specifications of the *Blue endeavour project* site and RexROV 2 with the published model parameters of Sub-Fighter 30K, the trajectory simulation parameters are determined. The trajectory starting and end points are $[x\ y\ z]_s, [x\ y\ z]_e = [33.75\ 0.0\ -1.0]m, [33.75\ 0.0\ -5.0]m$. The path interpolator is the linear interpolator with polynomial blends to generate a 4-DoF (surge, sway, heave and yaw) trajectory with 88 waypoints. The inspection time is approximately 7 min and the maximum forward speed of the trajectory is 0.28 m/s. Detailed explanation on those parameters can be found in the related work Tun et al. (2023).

6.2. Underwater current disturbance

Three underwater current disturbance speeds (0.0 m/s, 0.5 m/s and 0.9 m/s) in the positive y-axis of Frame $\{I\}$, assumed to be irrotational and constant, are considered based on the technical report for the *Blue endeavour project* site (Newcombe et al., 2020).

6.3. Hydrodynamics parameters(Berg, 2012)

Added mass matrix,

$$M_A = \begin{bmatrix} M_{A,11} & M_{A,12} \\ M_{A,21} & M_{A,22} \end{bmatrix}$$

where

$$M_{A,11} = \begin{bmatrix} 779.79 & -6.8773 & -103.32 \\ -6.8773 & 1222.0 & 51.29 \\ -103.32 & 51.29 & 3659.9 \end{bmatrix}$$

Table 1
Specification of *Blue endeavour project* (Tun et al., 2023).

Parameters	Value	Descriptions
D	53.5 m	Fish net-pen internal diameter
d_m	35 mm	Fish net-pen mesh size
h	15 m	Fish net-pen depth
$v_{c,\min}$	0.5 m/s	Mean underwater current
$v_{c,\max}$	0.9 m/s	Surface mean underwater current

Table 2
RexROV 2 with sub-fighter 30K's published model parameters (Berg, 2012).

Parameters	Value	Descriptions
m_r	1862.87 kg	: mass
w_G	18.268 519 kN	: weight
w_B	18.538 264 kN	: buoyancy force
V	1.838 26 m^3	: displaced water volume
w, l, h	1.5 m, 2.6 m, 1.6 m	: width, length, height
T_n	6	: number of thrusters (3 in horizontal, 3 in vertical)
C_f	0.0268	: rotor constant of SE150, 24V thruster
$u_{\max}$	1.543 m/s	: maximum surge speed
$d_{\max}$	1000 m	: standard rated depth
$f_{T,\max}$	1540 N	: maximum forward and reverse thrust of SE150, 24V thruster
r_G^b	[0,0,0] m	: centre of gravity in the respective x, y and z axis of robot frame
r_B^b	[0,0,0.3] m	: centre of buoyancy in the respective x, y and z axis of robot frame
$[I_{xx}, I_{yy}, I_{zz}]$	[525.39, 794.20, 691.23] kg m^2	: moment of inertia expressed in robot frame
$[I_{xy}, I_{xz}, I_{yz}]$	[1.44, 33.41, 2.6] kg m^2	: moment of inertia expressed in robot frame

Note: There is a positive buoyancy force of about $B - W = 270$ N or 27.5 kg f.

$$M_{A,12} = \begin{bmatrix} 8.5426 & -165.54 & -7.8033 \\ 409.44 & -5.8488 & 62.726 \\ 6.1112 & -386.42 & 10.774 \end{bmatrix}$$

$$M_{A,21} = \begin{bmatrix} 8.5426 & 409.44 & 6.1112 \\ -165.54 & -5.8488 & -386.42 \\ -7.8033 & 62.726 & 10.775 \end{bmatrix}$$

$$M_{A,22} = \begin{bmatrix} 534.9 & -10.027 & 21.019 \\ -10.027 & 842.69 & -1.1162 \\ 21.019 & -1.1162 & 224.32 \end{bmatrix}$$

The full matrix of M_A is used for Gazebo Physics Engine but only the diagonal elements are used for the controllers.

Linear damping matrix,

$$D_l = \text{diag}\{74.82,\ 69.48,\ 728.4,\ 268.8,\ 309.77,\ 105\}$$

Quadratic damping matrix,

$$D_q(\nu) = \text{diag}\{748.22,\ 992.53,\ 1821.01,\ 672,\ 774.44,\ 523.27\}$$

Thruster Allocation Matrix (TAM),

$$B_T = \begin{bmatrix} B_{T,11} & B_{T,12} \end{bmatrix}$$

where

$$B_{T,11} = \begin{bmatrix} 0 & 0.2588 & 0 \\ 1 & 0 & 0.3827 \\ 0 & 0.9659 & 0.9239 \\ -0.2373 & 0 & -0.6957 \\ 0 & 0.9461 & -0.7995 \\ 0.4878 & 0 & 0.3312 \end{bmatrix}$$

$$B_{T,12} = \begin{bmatrix} 0 & 0.9063 & 0.9063 \\ 0.3827 & 0.4226 & -0.4226 \\ -0.9239 & 0 & 0 \\ -0.6957 & -0.1016 & 0.1016 \\ 0.7995 & 0.2179 & 0.2179 \\ 0.3312 & -0.7639 & 0.7639 \end{bmatrix}$$

Controller: For the four proposed MPCs, the following tuning parameters are used.

$$Q = 1000 \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}, Q_f = \gamma_3 Q$$

$$Q_1 = \text{diag}([4950, 4950, 4950, 55, 55, 4400]^T)$$

$$Q_2 = \text{diag}([55, 55, 55, 55, 55, 55]^T)$$

$$R = \text{diag}([1.2, 1.2, 1.2, 1.2, 1.2, 1.2]^T)$$

$$\gamma_1 = 2.0,\ \gamma_2 = 2.0,\ \gamma_3 = 0.2,\ h_s = 0.05$$

$$R_c = 5\,\text{Hz},\ N_p = 15\,\text{steps},$$

$$N_c = 5\,\text{steps},\ t_p = \frac{N_p}{R_c} = 3\,\text{s}.$$

As shown above, the relative assignment of $Q_1 > Q_2$ and $Q_{f_1} > Q_{f_2}$ highlights more important role of pose tracking than twist tracking to avoid the scenario where twist tracking error approaches to zero while pose tracking error remains large. These tuning parameters are decided empirically, following the conditions of those tuning parameters as specified in Eqs. (18), (19), (21) and (24). To consistently compare the energy-optimal feature across all controllers, the general guideline of the empirical tuning process is as follows.

- Determine N_p, N_c, R_c and h_s based on the available computing capacity in real-time.
- Tune the control parameters of CO-MPC1 as specified in Eqs. (18) and (19).
- Adjust CO-MPC1's tuning parameters until the tracking errors are smaller. Note: The term "smaller error" here is defined subjectively in this case.
- Once the tuning parameters on CO-MPC1 are fixed, start tuning on EO-MPC1, EO-MPC2 and EO-MPC3 as specified in Eqs. (21) and (24) using the same Q, Q_f and R tuned for CO-MPC1 in the previous step. Note: Q, Q_f and R are kept the same as CO-MPC1 to avoid changing the trajectory tracking tuning parameters and energy-related tuning parameters at the same time.

6.4. Thruster energy consumption

As RexROV 2 is equipped with SE150, 24 V thruster for which thrust to power mapping is not available, angular velocity to thrust and torque respectively reported in Berg (2012)[Appendix C] are used to obtain the formula mapping from thrust to power. SE150, 24V thruster is rated for 8.8 kW and the resulting mapping is close to that value.

Angular velocity to torque and thrust mappings, respectively, can be described as follows (Antonelli, 2018).

$$\tau_{T,j} = C_\tau \omega_j |\omega_j| \tag{30}$$

$$f_{T,j} = C_f \omega_j |\omega_j| \tag{31}$$

$$\omega_j = \text{sign}(f_{T,j}) \sqrt{\frac{|f_{T,j}|}{C_f}} \tag{32}$$

where $j = \{1, \ldots, m\}$, $C_\tau = \rho D^5 K_{\tau_T}$, $C_f = \rho D^4 K_{f_T}$, ρ is the water density, D is the thruster diameter, K_{τ_T}, K_{f_T} are the torque and thrust coefficients, respectively, ω is the thruster's angular velocity.

Thruster power can be described as,

$$P_{T,j} = \tau_{T,j} \omega_j \tag{33}$$

Using the data set from Berg (2012)[Appendix C] and manipulating Eqs. (30)– (32), Eq. (33) can be rewritten as

$$P_{T,j}(f_{T,j}) = C_P \omega_j f_{T,j} = \text{sign}(f_{T,j}) C_p f_{T,j} \sqrt{\frac{|f_{T,j}|}{C_f}} \tag{34}$$

where $C_P = \frac{C_\tau}{C_f}$.

The resulting polynomial regression formulae, their respective R^2 values and the corresponding coefficients ($C_\tau = 0.0006, C_f = 0.0268, C_P = .0224$) are reported as shown in Figs. 5–7.

Subsequently, the total energy consumption of all SE150, 24V thrusters at each control loop is calculated using Fig. (34) as follows.

$$E_{\Delta t} = \sum_{j=1}^{m} P_{T,j}(f_{T,j}) \Delta t \tag{35}$$

The detailed MPC algorithm implementation as a ROS node is explained and discussed in Appendix Section A.

7. Results and discussions

In this section, six main results will be discussed, namely trajectory tracking, pose tracking, twist tracking, energy consumption, effects of $p(f_T, v)$, and tracking & energy consumption. For the error comparison, mean-absolute-error (MAE) as shown in Eq (36) and MAE-ratio with

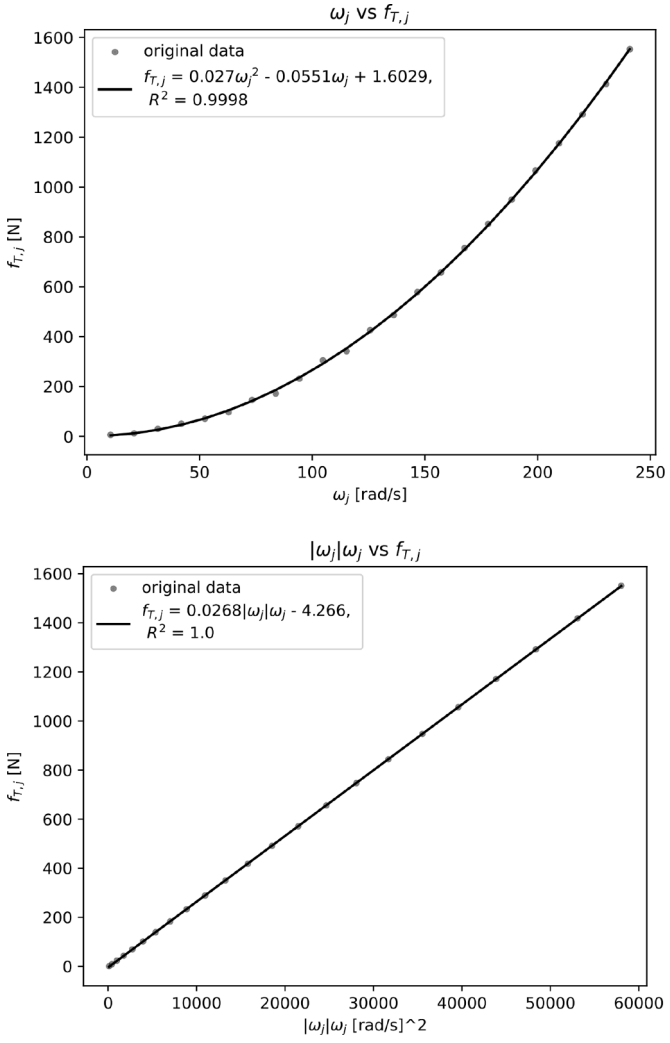


Fig. 5. Relationship between ω_j and $f_{T,j}$ of SE150, 24V thruster.

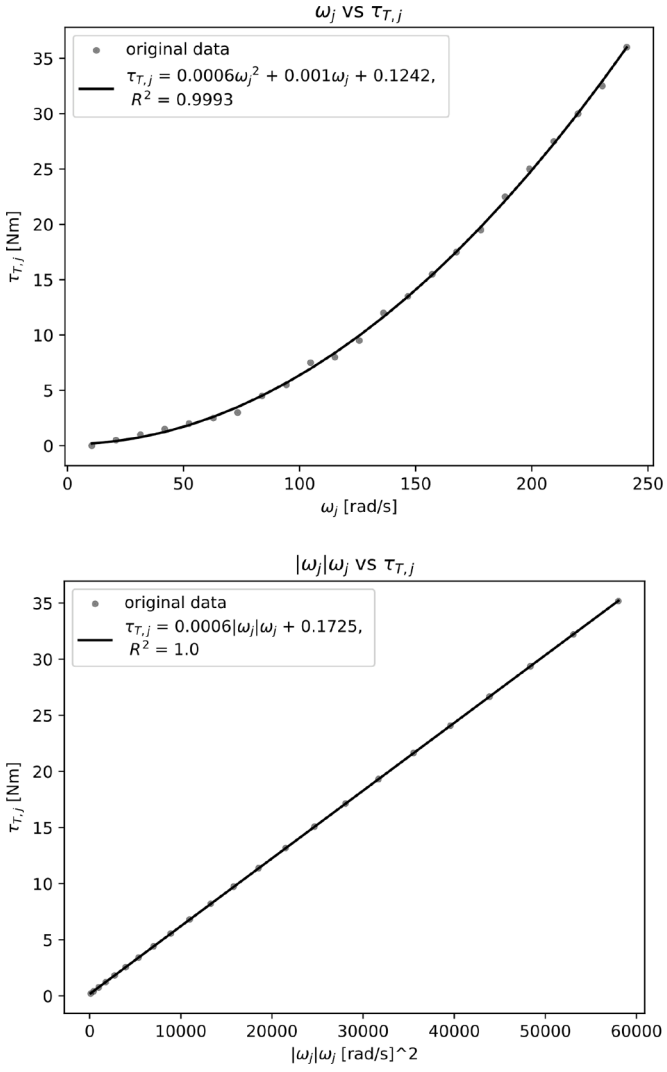


Fig. 6. Relationship between ω_j and $\tau_{T,j}$ of SE150, 24V thruster.

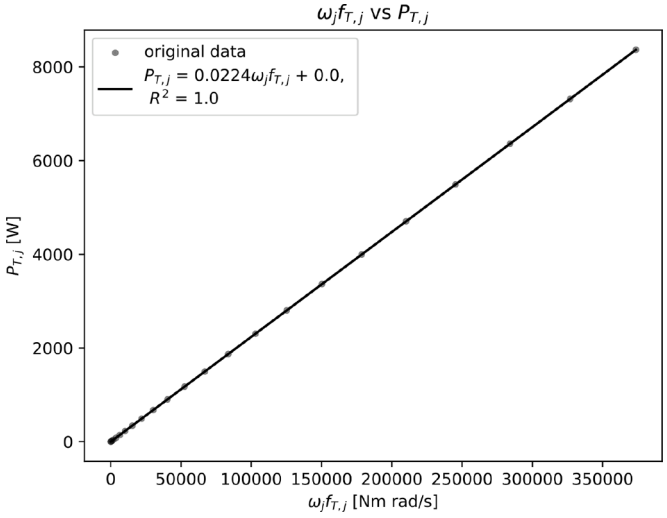


Fig. 7. Relationship between $\omega_j f_{T,j}$ and $P_{T,j}$ of SE150, 24V thruster.

respect to CO-MPC1 (denoted as **1**) as shown in Eq (37) will be used.

$$\text{MAE} = \frac{\Sigma_{i=0}^{N}|e_i|}{N} \tag{36}$$

where N is the number of data recorded, e_i represents i^{th} element of a specific error type (e.g., $e_{x,i}$). For MAE of error norm, e_i represents i^{th} element of a specific norm error type (e.g., $e_{\eta_1,i} = \sqrt{e_{x,i}^2 + e_{y,i}^2 + e_{z,i}^2}$).

$$\text{MAE\_ratio}_i = \frac{(\text{MAE}_1 - \text{MAE}_i)}{\text{MAE}_1} \tag{37}$$

where $i = \{\mathbf{2}, \mathbf{3}, \mathbf{4}\}$ represents EO-MPC1, EO-MPC2 and EO-MPC3, respectively.

The recorded simulation videos are available at https://youtube.com/playlist?list=PLG3nO3TEqwOmaKKJm1raw9A95pzr3EjBr&si=NIt_ZsYYytAnzzYz.

7.1. Trajectory tracking

Figs. 8 and 9 show 3D position plots of the reference and actual trajectories of CO-MPC1, EO-MPC1, EO-MPC2 and EO-MPC3 under three underwater current disturbance speeds 0.0 m/s, 0.5 m/s and 0.9 m/s. The irrotational and constant underwater current disturbance is flowing in the positive y-axis of Frame $\{I\}$. Generally, it can be observed that all controllers perform well except EO-MPC2 and EO-MPC3 at the speed of disturbance 0.9 m/s which drifted substantially in the direction of disturbance.

It is important to note that the tracking behaviors of CO-MPC1 and EO-MPC1 are similar, and so are the tracking behaviors of EO-MPC2 and EO-MPC3. In other words, for CO-MPC1 and EO-MPC1, the conventional tracking behavior is prominent and EO-MPC1 should save more energy as shown in Eqs. (19) and (21). Similarly for EO-MPC2 and EO-MPC3, the tracking behavior with the penalty for a large change or variation in control effort is obvious and EO-MPC3 should save more energy as shown in Eq. (24) and (25). In other words, for EO-MPC2 and EO-MPC3, the penalty for a large change or variation in control effort prevents the large corrective control effort but instead allows the incremental change in control effort, resulting in a larger error especially when the speed of underwater current disturbance is high.

7.2. Pose tracking

Due to the nature of the fish net-pen visual inspection, pose tracking is relatively more important than twist tracking. Pose tracking can be further sub-categorized into position tracking and orientation tracking as shown in Figs. 10 and 11, respectively. From the plot of the position tracking norm as shown in Fig. 10 and Table 3, it can be observed that all controllers exhibit large position tracking errors e_{η_1} resulting from maneuvering the sharp turns and upward/downward slopes of the trajectory. Those tracking errors are more evident when the speed of underwater current disturbance increases. As shown in Fig. 9, the position tracking performance of EO-MPC2 and EO-MPC3 potentially poses the risk of causing physical damage to the aquaculture infrastructure and the UUV especially in a highly constrained operational working environment. Quantitatively, e_{η_1} norm of EO-MPC1 is 0.6 % to 3.9 % lower than that of CO-MPC1 under varying underwater

Table 3
Pose tracking norm — MAE ratio with respect to CO-MPC1 under different simulated underwater current speeds. CO_MPC1, EO_MPC1, EO_MPC2 and EO_MPC3 are denoted by **1**, **2**, **3** and **4**, respectively.

			e_{η_1} norm		
			MAE	MAE Ratio	%
0.0 m/s	MPC Controller Types	**1**	0.795	1.	0.0
		2	0.764	0.039	−3.9
		3	1.699	−1.138	113.8
		4	1.696	−1.134	113.4
0.5 m/s		**1**	0.944	1.	0.0
		2	0.938	0.006	−0.6
		3	2.129	−1.255	125.5
		4	2.084	−1.207	120.7
0.9 m/s		**1**	1.556	1.	0.0
		2	1.533	0.014	−1.4
		3	3.473	−1.232	123.2
		4	3.472	−1.232	123.2

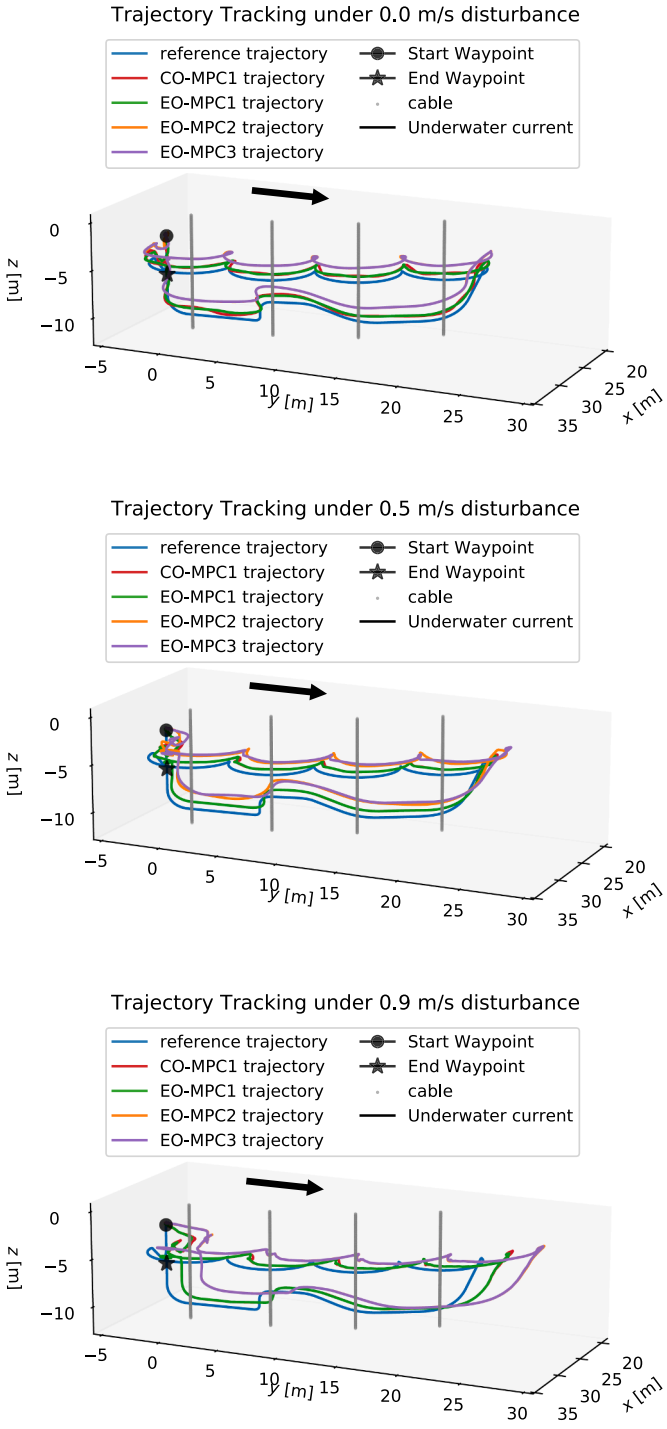


Fig. 8. Trajectory tracking comparison among controllers tested under simulated underwater current speeds. The recorded simulation videos are available at https://youtube.com/playlist?list=PLG3nO3TEqwOmaKKJm1raw9A95pzr3EjBr&si=NIt_ZsYYytAnzzYz.

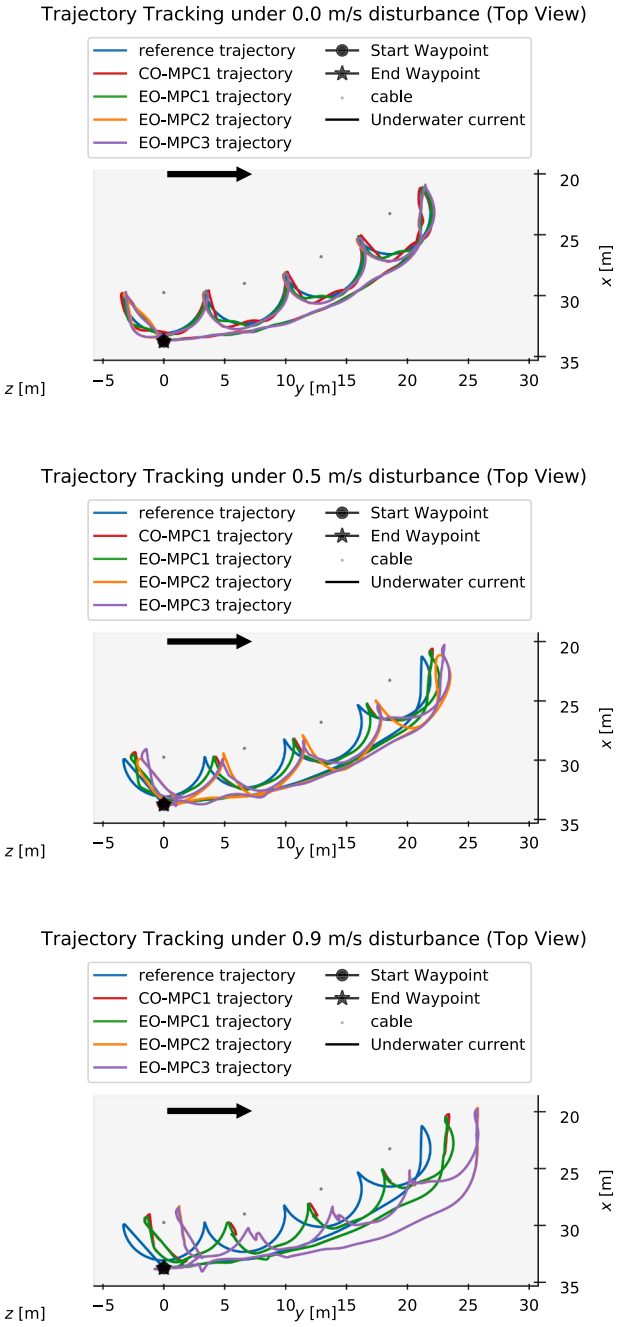


Fig. 9. Trajectory tracking (Top View) comparison among controllers tested under simulated underwater current speeds. The recorded simulation videos are available at https://youtube.com/playlist?list=PLG3nO3TEqwOmaKKJm1raw9A95pzr3EjBr&si=NIt_ZsYYytAnzzYz.

current disturbance speeds, whereas e_{η_1} norm of EO-MPC2 and EO-MPC3 is 113.4 % to 125.5 % higher than that of CO-MPC1. Therefore, EO-MPC1 outperforms other controllers in terms of position tracking.

It is important to note that due to the positive buoyancy force of approximately 270 N, e_{η_1} norm always does not converge to zero. It can be reduced by adding more weight without increasing the volume. However, in real-world application, it is desired to have a slight positive buoyancy as the UUV's safety feature to perform auto-recovery to the surface, but a large positive buoyancy causes the UUV to consume more energy to minimize the depth tracking error, resulting from the positive buoyancy. In this experiment, the original UUV parameters are used, and thus the non-zero e_{η_1} norm is expected Tables 4–6.

Table 4
Orientation tracking norm — MAE ratio with respect to CO-MPC1 under different simulated underwater current speeds. CO_MPC1, EO_MPC1, EO_MPC2 and EO_MPC3 are denoted by **1**, **2**, **3** and **4**, respectively.

			e_{η_2} norm		
			MAE	MAE Ratio	%
0.0 m/s	MPC Controller Types	**1**	6.827	1.0	0.0
		2	6.287	0.079	−7.9
		3	7.293	−0.068	6.8
		4	7.537	−0.104	10.4
0.5 m/s		**1**	7.423	1.0	0.0
		2	7.858	−0.059	5.9
		3	11.148	−0.502	50.2
		4	10.364	−0.396	39.6
0.9 m/s		**1**	12.349	1.0	0.0
		2	12.673	−0.026	2.6
		3	17.56	−0.422	42.2
		4	17.793	−0.441	44.1

Table 5
Linear velocity tracking norm — MAE ratio with respect to CO-MPC1 under different simulated underwater current speeds. CO_MPC1, EO_MPC1, EO_MPC2 and EO_MPC3 are denoted by **1**, **2**, **3** and **4**, respectively.

			e_{ν_1} norm		
			MAE	MAE Ratio	%
0.0 m/s	MPC Controller Types	**1**	0.097	1.	0.0
		2	0.073	0.251	−25.1
		3	0.104	−0.071	7.1
		4	0.103	−0.052	5.2
0.5 m/s		**1**	0.083	1.	0.0
		2	0.08	0.04	−4.0
		3	0.133	−0.603	60.3
		4	0.124	−0.495	49.5
0.9 m/s		**1**	0.124	1.	0.0
		2	0.116	0.06	−6.0
		3	0.174	−0.409	40.9
		4	0.173	−0.402	40.2

Table 6
Angular velocity tracking norm — MAE ratio with respect to CO-MPC1 under different simulated underwater current speeds. CO_MPC1, EO_MPC1, EO_MPC2 and EO_MPC3 are denoted by **1**, **2**, **3** and **4**, respectively.

			e_{ν_2} **norm**		
			MAE	MAE Ratio	%
0.0 m/s	MPC Controller Types	**1**	0.138	1.	0.0
		2	0.138	−0.001	0.1
		3	0.135	0.019	−1.9
		4	0.108	0.215	−2.15
0.5 m/s		**1**	0.135	1.	0.0
		2	0.126	0.07	−7.0
		3	0.151	−0.114	11.4
		4	0.115	0.148	−14.8
0.9 m/s		**1**	0.184	1.	0.0
		2	0.174	0.052	−5.2
		3	0.166	0.099	−9.9
		4	0.166	0.095	−9.5

For orientation tracking, in maneuvering the UUV around the sharp trajectory turns, e_{η_2} norm spikes are more prominent across all controllers. It is interesting to observe that there is no substantial difference in e_{η_2} norm spikes but the change in e_{η_2} norm after the spikes is more obvious as the speed of underwater current disturbance increases. This behavior is observed because the sharp trajectory turn takes a very short duration whereas the smooth turn (e.g., small circular turn) takes

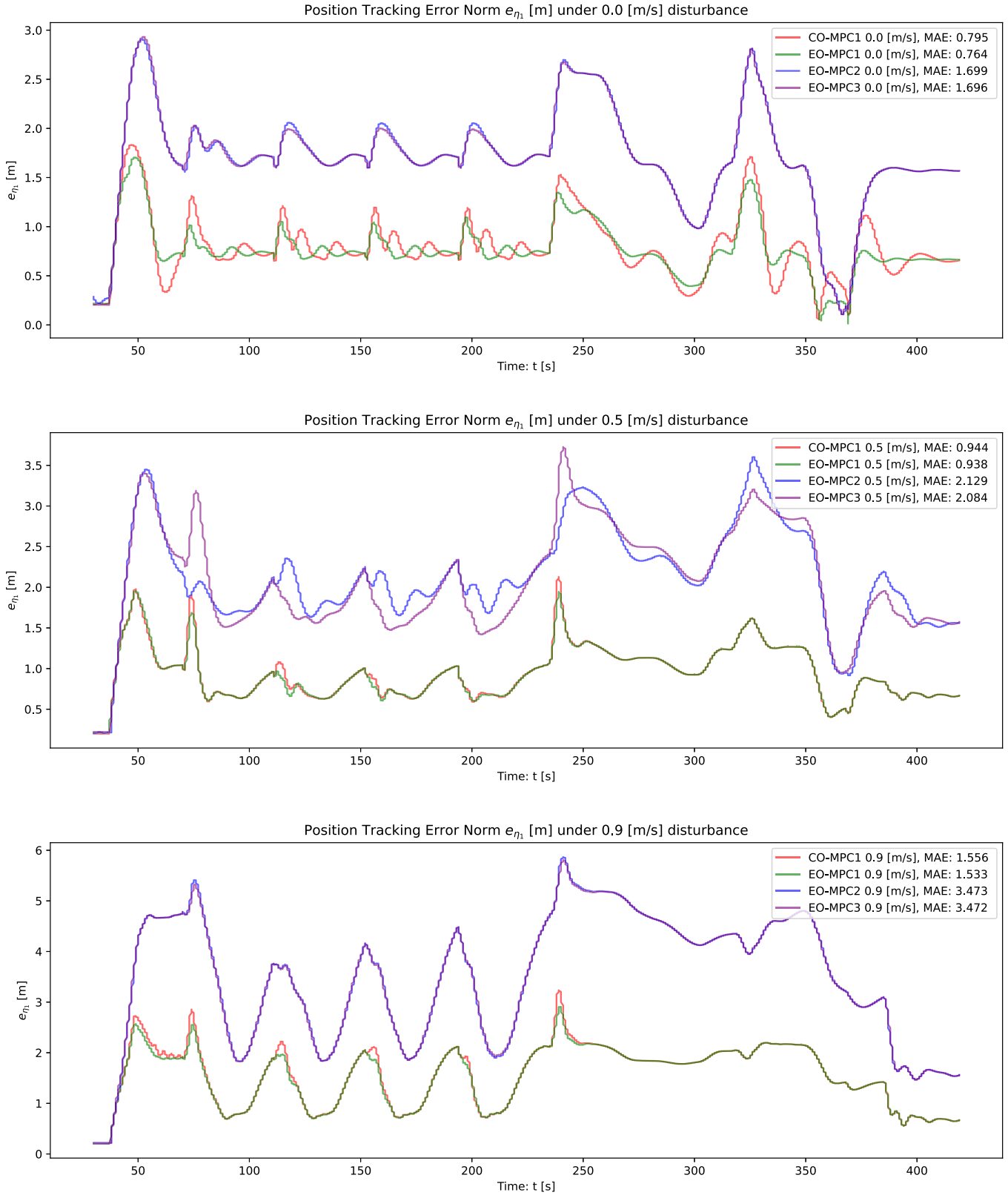


Fig. 10. Position tracking error norm comparison among trajectory tracking controllers tested under simulated underwater current. The recorded simulation videos are available at https://youtube.com/playlist?list=PLG3nO3TEqwOmaKKJm1raw9A95pzr3EjBr&si=NIt_ZsYYytAnzzYz.

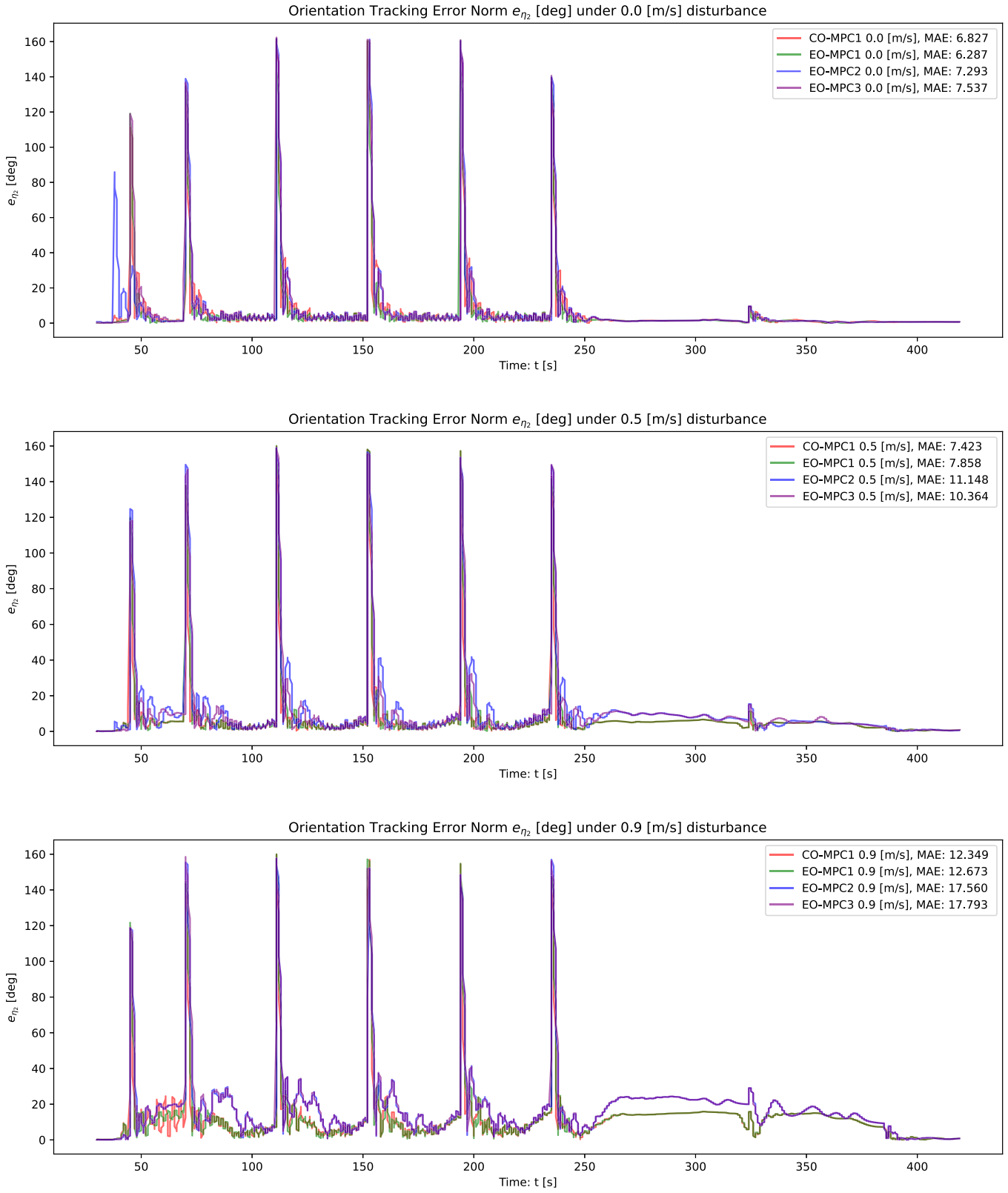


Fig. 11. Orientation tracking error norm comparison among trajectory tracking controllers tested under simulated underwater current. The recorded simulation videos are available at https://youtube.com/playlist?list=PLG3nO3TEqwOmaKKJm1raw9A95pzr3EjBr&si=NIt_ZsYYytAnzzYz.

a longer duration. Therefore, e_{η_2} norm in smooth turn in a longer duration exhibits substantial change, whereas e_{η_2} norm in the sharp trajectory turn in a shorter duration is less obvious. Quantitatively, e_{η_2} norm of EO-MPC1 is 7.9 % lower than that of CO-MPC1 under no disturbance, but it is 2.6 % to 5.9 % higher than that of CO-MPC1 under the disturbance of underwater current. However, under varying underwater current disturbance speeds, e_{η_2} norm of EO-MPC2 and EO-MPC3 is 6.8 % to 50.2 % higher than that of CO-MPC1. Therefore, EO-MPC1's orientation tracking performance is almost equivalent to CO-MPC1's performance but performances of EO-MPC2 and EO-MPC3 are substantially worse.

7.3. Twist tracking

Although twist tracking is relatively less important than pose tracking, it is still necessary so that the visual inspection task is completed in time. Additionally, at station-keeping (the start and end of the trajectory), the velocity reference (e.g., 0.0 m/s, 0.0 rad/s) makes the controllers less sensitive to the pose error, compared to the pose reference only.

For the linear velocity tracking, similar phenomena of pose tracking such as large tracking errors around the sharp turns and upward/downward slopes of the trajectory can be observed as shown in Fig. 12. Quantitatively, e_{ν_1} of EO-MPC1 is 4.0 % - 25.1 % lower than that of CO-MPC1 under any disturbance speeds. On the other hand, e_{ν_1} of EO-MPC2 and EO-MPC3 are 5.2 % - 60.3 % higher than that of CO-MPC1 under different disturbance speeds. Therefore, EO-MPC1's linear velocity tracking performance is better than all other controllers' performance.

For the angular velocity tracking as shown in Fig. 13, sharp spiking phenomena can be observed due to the abrupt turning of the trajectory at the start and end of the half-circle trajectory. Quantitatively, e_{ν_2} of EO-MPC1 is 5.2 % - 7.0 % lower than that of CO-MPC1 under the disturbance, whereas under no disturbance, the former is 0.1 % higher than the latter. Under the varying underwater current disturbance speeds, e_{ν_2} of EO-MPC2 and EO-MPC3 are 1.9 % - 14.8 % lower than that of CO-MPC1 but under 0.5 m/s disturbance speed, EO-MPC2' e_{ν_2} is 11.4 % higher than that of CO-MPC1. Therefore, compared to pose tracking and linear velocity tracking, all controllers perform better than CO-MPC1 in angular velocity tracking except for EO-MPC2 under 0.5 m/s underwater current disturbance speed.

7.4. Energy consumption

Due to the high-power rated thrusters used to maneuver the work-class ROV, it is assumed that the thrusters' power consumption is the highest among all the other electronic components. According to the SE150, 24V thruster specification, its power output at nominal voltage is reported to be 8.8 kW. Therefore, for a total of six thrusters, the observed instantaneous power output cannot exceed 52.8 kW and its peaks should be found around the energy-demanding turns and slope as shown in Fig. 8. This predicted phenomenon can be observed in Figs. 14–16 in which the maximum instantaneous power consumption is around 14 kW. At some peaks, EO-MPC1 exhibits a higher peak in instantaneous power consumption compared to CO-MPC1. However, it can be observed that EO-MPC1 subsequently reduces its power consumption more rapidly than CO-MPC1. On the other hand, both EO-MPC2 and EO-MPC3 consume less instantaneous power than CO-MPC1 and EO-MPC1. This power-consumption profile remains the same under different underwater current disturbance speeds and so does the energy-consumption profile.

To quantitatively compare total energy savings, energy consumption, its ratio, and energy saving percentages are reported in Table 7, taking CO-MPC as the baseline. A comprehensive visualization of the energy savings in percentage terms is illustrated in Fig. 17. All EO-MPC controllers save relatively more energy than CO-MPC. However, it has been observed that the amount of energy savings reduces as the speed of underwater current disturbance increases. This finding highlights that energy optimization becomes less effective under higher underwater current disturbance speeds. It should be noted that the primary goal is to accomplish the designated task such as trajectory tracking, while energy saving is a secondary objective. Inevitably, an increase in control effort is required to compensate for stronger disturbances, resulting in higher energy consumption.

Table 7
Energy consumption — its ratio with respect to CO-MPC1 under different simulated underwater current speeds. CO_MPC1, EO_MPC1, EO_MPC2 and EO_MPC3 are denoted by **1**, **2**, **3** and **4**, respectively.

			Energy consumption [Wh]		
			$E_{\Delta t}$	$E_{\Delta t}$ Ratio	%
0.0 m/s	MPC Controller types	**1**	96.564	1.	0.0
		2	75.934	0.214	21.4
		3	67.268	0.303	30.3
		4	66.117	0.315	31.5
0.5 m/s		**1**	126.719	1.	0.0
		2	118.880	0.062	6.2
		3	110.896	0.125	12.5
		4	106.220	0.162	16.2
0.9 m/s		**1**	300.1102	1.	0.0
		2	290.8249	0.031	3.1
		3	272.2074	0.093	9.3
		4	271.3096	0.096	9.6

In summary, for energy consumption, as shown in Table 7, all EO-MPC controllers exhibit relatively higher energy savings compared to CO-MPC1 in a range of 3.1 % to 31.5 %, depending on the speed of the underwater current disturbance. In general, the energy-optimal feature becomes less effective under higher disturbance speeds. However, the energy-saving percentage alone is not a sufficient indicator of the selection of EO-MPC controllers. Therefore, in real-world application, the selection of EO-MPC controllers depends on trajectory tracking performance, the acceptable trajectory tracking error tolerance and the associated energy-optimality under the varying underwater current disturbance speeds.

Based on the previous analysis on trajectory, pose and twist tracking, EO-MPC1's trajectory tracking performance is equivalent or better than CO-MPC1's performance where other EO-MPC controllers relatively do not perform well in a constrained operational environment. In terms of energy-optimality, the results show that all EO-MPC controllers exhibit superior energy-optimality than CO-MPC1. Therefore, EO-MPC1 is the best candidate in terms of trajectory tracking and energy-optimality. As the main difference in the stage/running cost of CO-MPC1 and EO-MPC1 is the non-convex system level power function $p(f_T, \nu)$ as shown in Eqs. (19) and (21), the effect of $p(f_T, \nu)$ on energy optimization will be discussed in the following subsection.

7.5. Effects of non-convex system-level power function $p(f_T, \nu)$

The non-convexity in $p(f_T, \nu)$ is discussed in Subsection 4.1 and its phenomenon can be observed in Fig. 18. Unlike the thrusters' instantaneous power consumption as shown in Figs. 14–16, a substantial difference can be observed in the constraint-activation of the system-level power function as shown in Fig. 18. EO-MPC1 and EO-MPC3, which incorporate the non-convex system-level power function $p(f_T, \nu)$, strictly enforce $|p_{\min}(f_T, \nu)| = |p_{\max}(f_T, \nu)| = 500$W constraint that is assigned in the algorithm implementation. In contrast, CO-MPC1 and EO-MPC2 permit higher values of $p(f_T, \nu)$, provided that the thrust generated by all individual thrusters remains within the user-assigned bounds. Moreover, once $p(f_T, \nu)$ constraint is activated, EO-MPC1 reduces its system-level power consumption more rapidly compared to the other MPC variants.

From the aforementioned findings, the consideration of $p(f_T, \nu)$ in the MPC stage/running cost (e.g., EO-MPC1) is effective to achieve

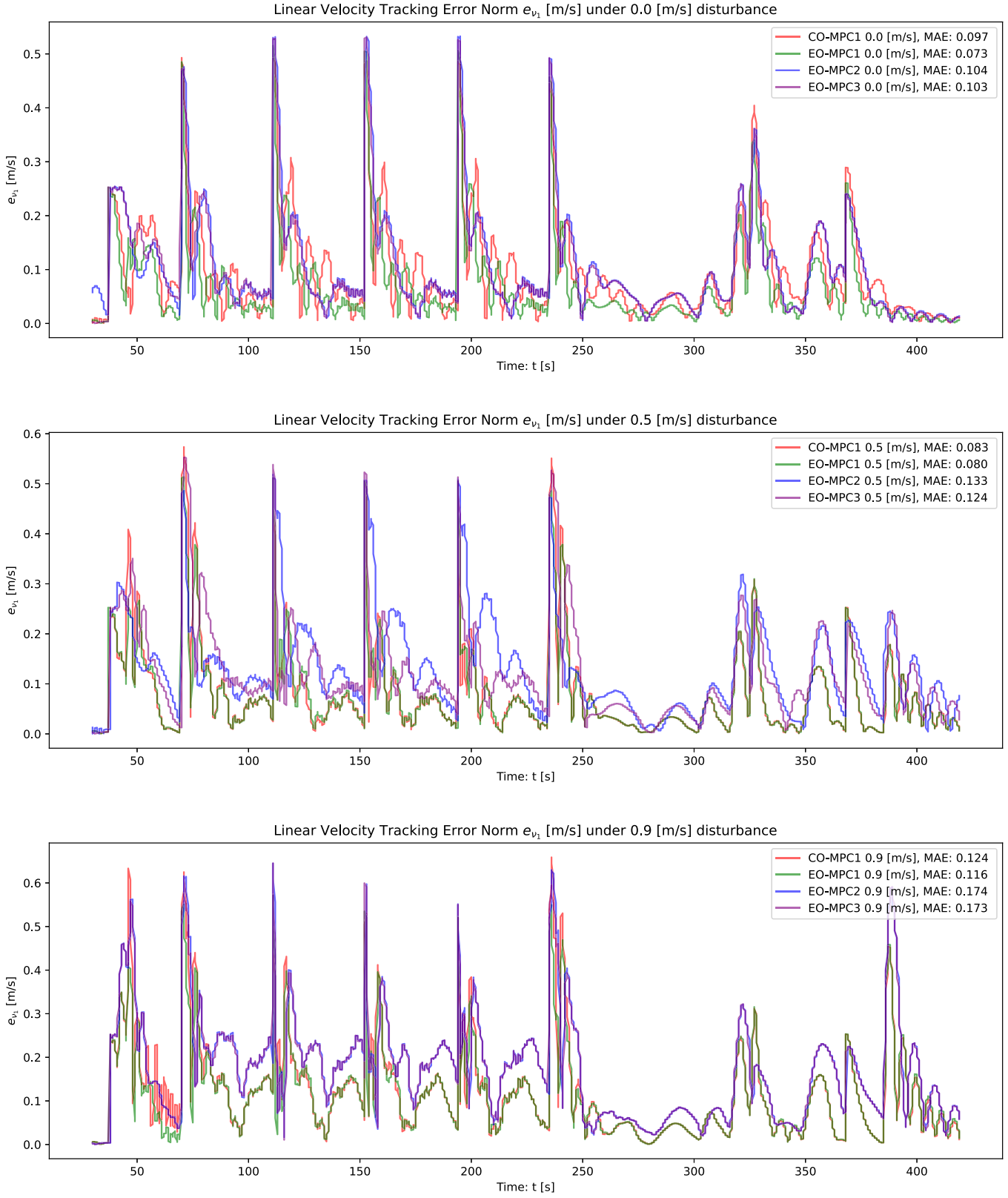


Fig. 12. Linear velocity tracking error norm comparison among trajectory tracking controllers tested under simulated underwater current. The recorded simulation videos are available at https://youtube.com/playlist?list=PLG3nO3TEqwOmaKKJm1raw9A95pzr3EjBr&si=NIt_ZsYYytAnzzYz.

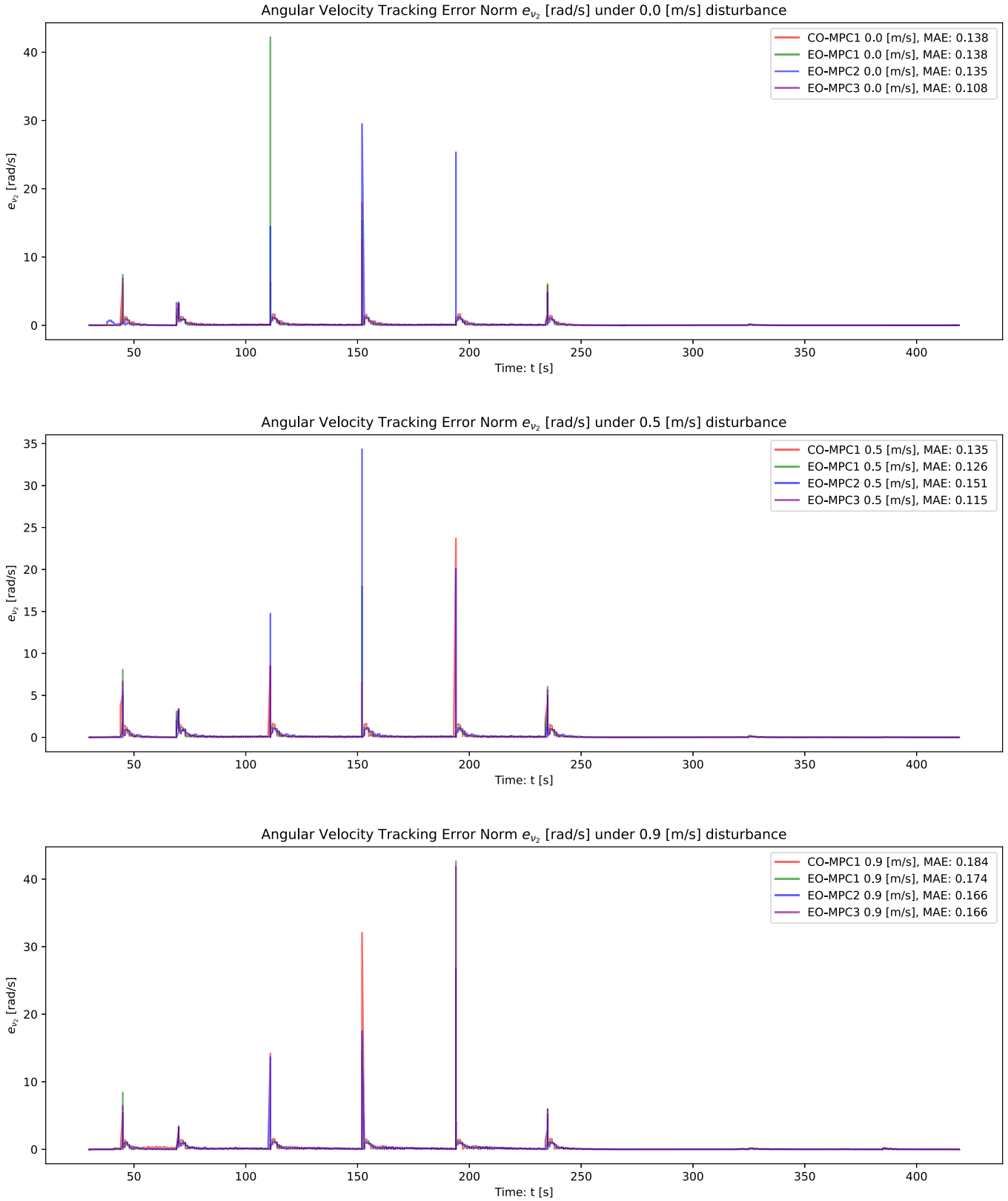


Fig. 13. Angular velocity tracking error norm comparison among trajectory tracking controllers tested under simulated underwater current. The recorded simulation videos are available at https://youtube.com/playlist?list=PLG3nO3TEqwOmaKKJm1raw9A95pzr3EjBr&si=NIt_ZsYYytAnzzYz.

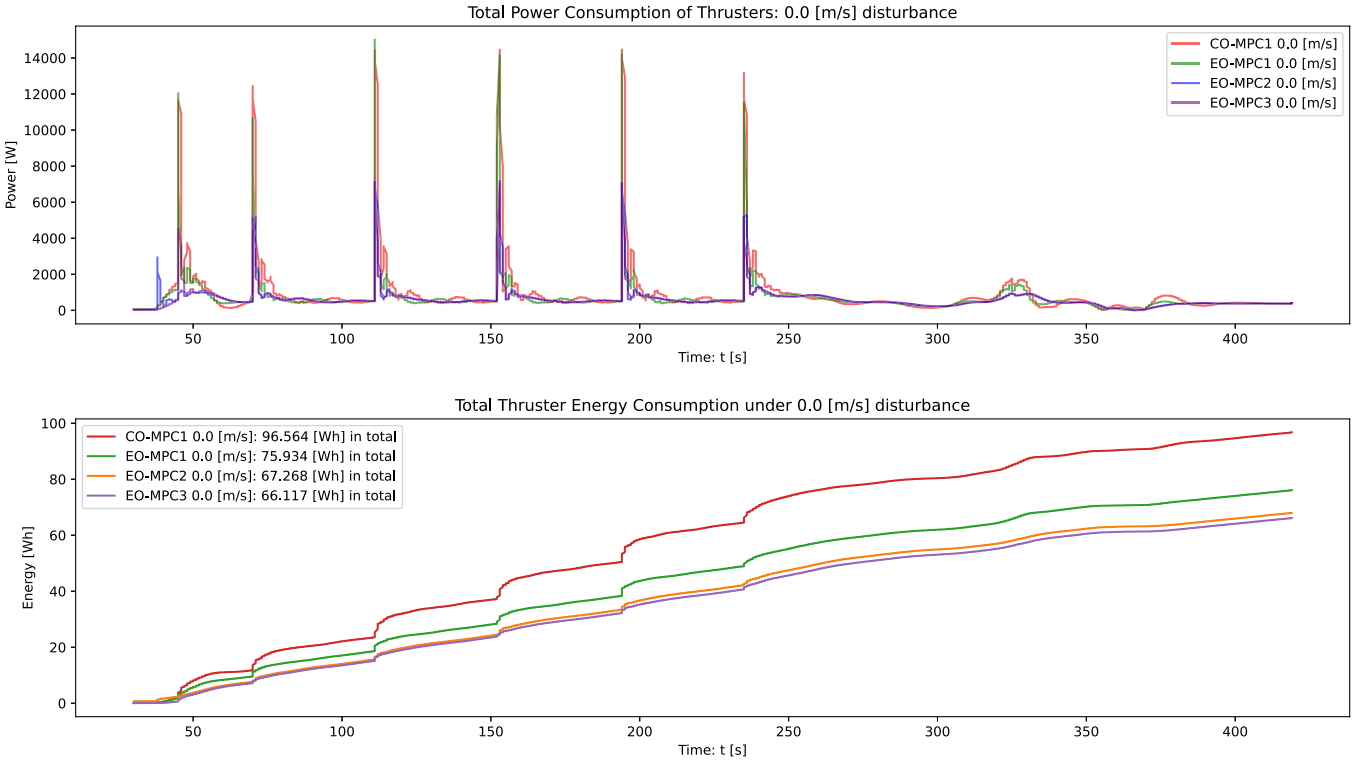


Fig. 14. Instantaneous power and total energy consumption of thrusters during trajectory tracking tested under simulated underwater current speeds of 0.0 m/s. The recorded simulation videos are available at https://youtube.com/playlist?list=PLG3nO3TEqwOmaKKJm1raw9A95pzr3EjBr&si=NIt_ZsYYytAnzzYz.

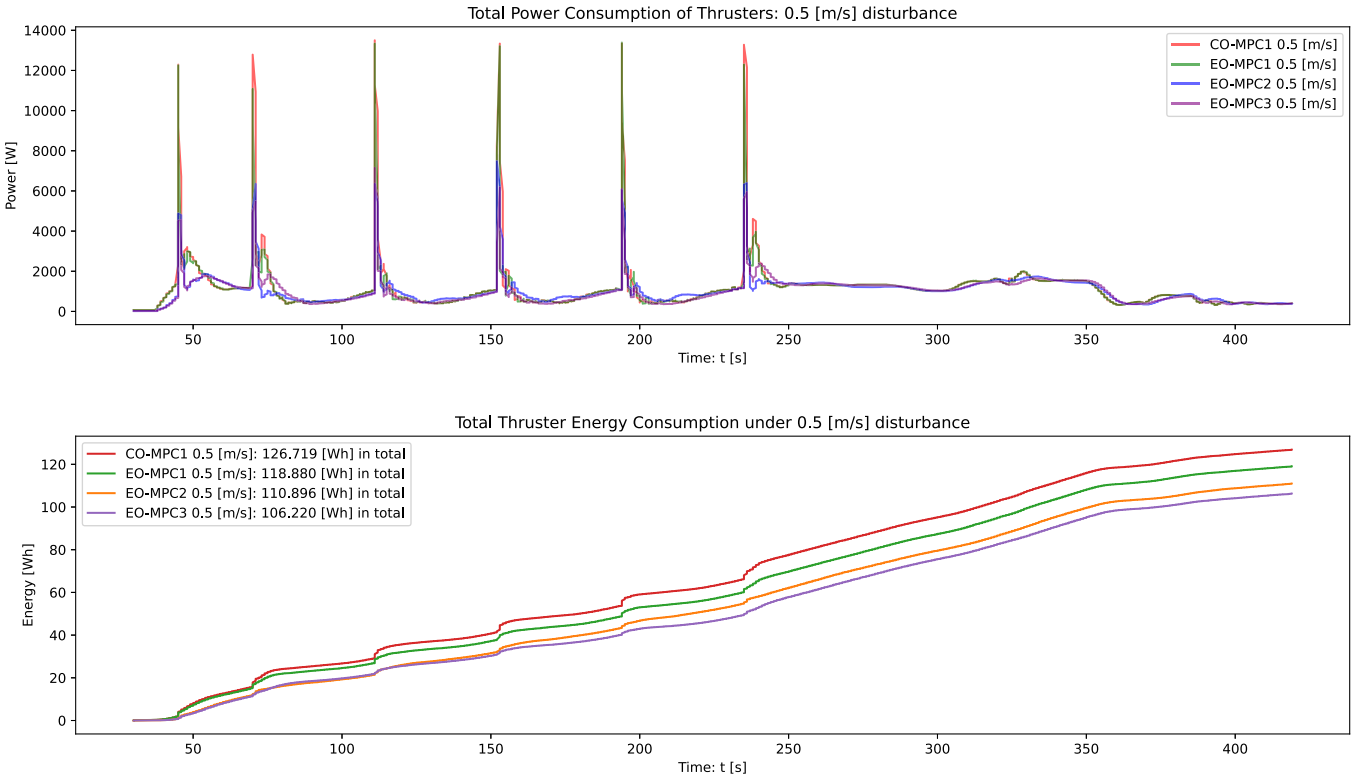


Fig. 15. Instantaneous power and total energy consumption of thrusters during trajectory tracking tested under simulated underwater current speeds of 0.5 m/s. The recorded simulation videos are available at https://youtube.com/playlist?list=PLG3nO3TEqwOmaKKJm1raw9A95pzr3EjBr&si=NIt_ZsYYytAnzzYz.

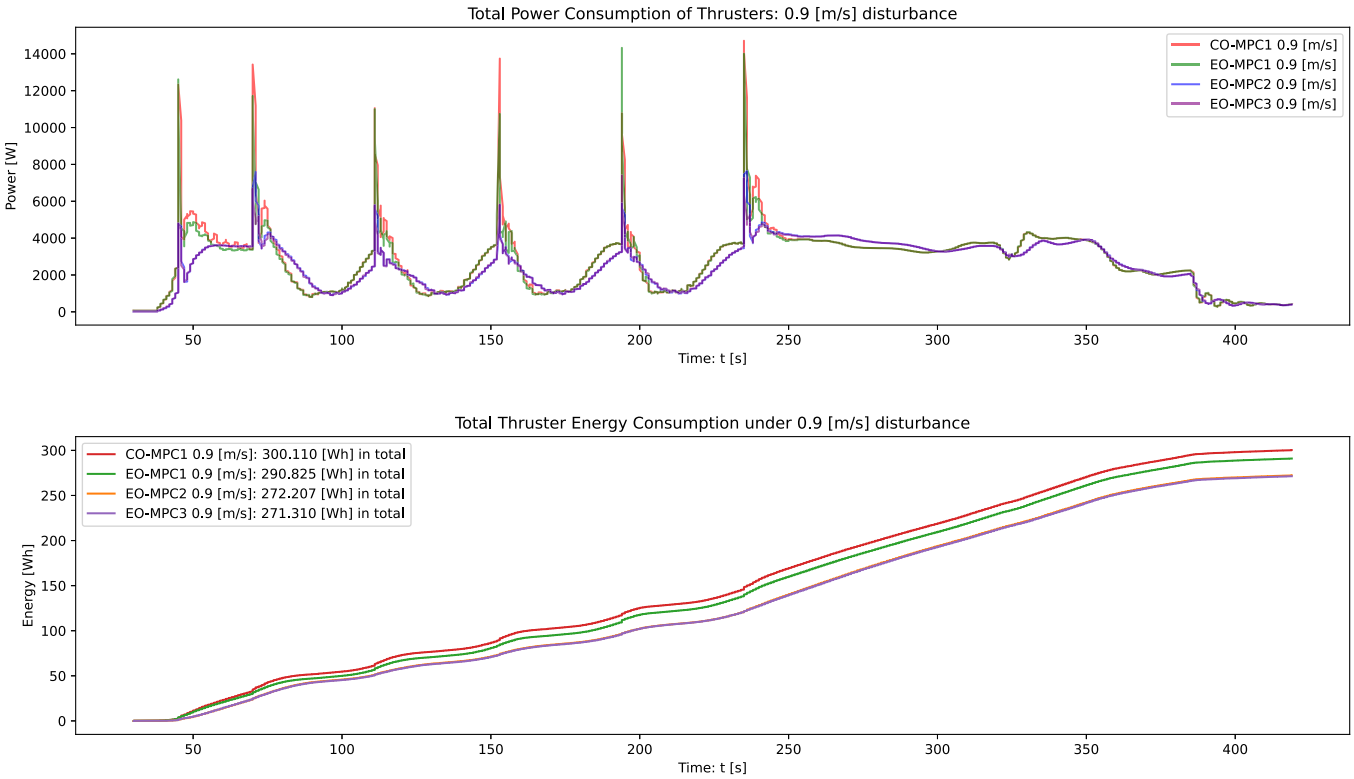


Fig. 16. Instantaneous power and total energy consumption of thrusters during trajectory tracking tested under simulated underwater current speeds of 0.9 m/s. The recorded simulation videos are available at https://youtube.com/playlist?list=PLG3nO3TEqwOmaKKJm1raw9A95pzr3EjBr&si=NIt_ZsYYytAnzzYz.

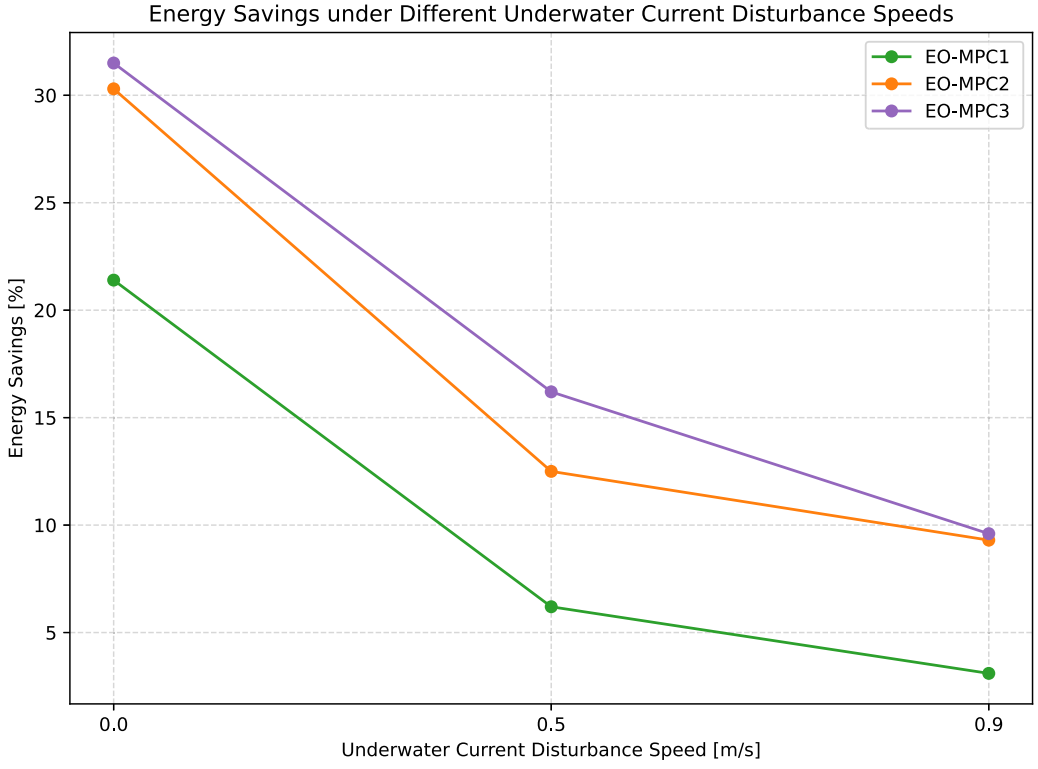


Fig. 17. Effects of underwater current disturbance speed on energy saving.

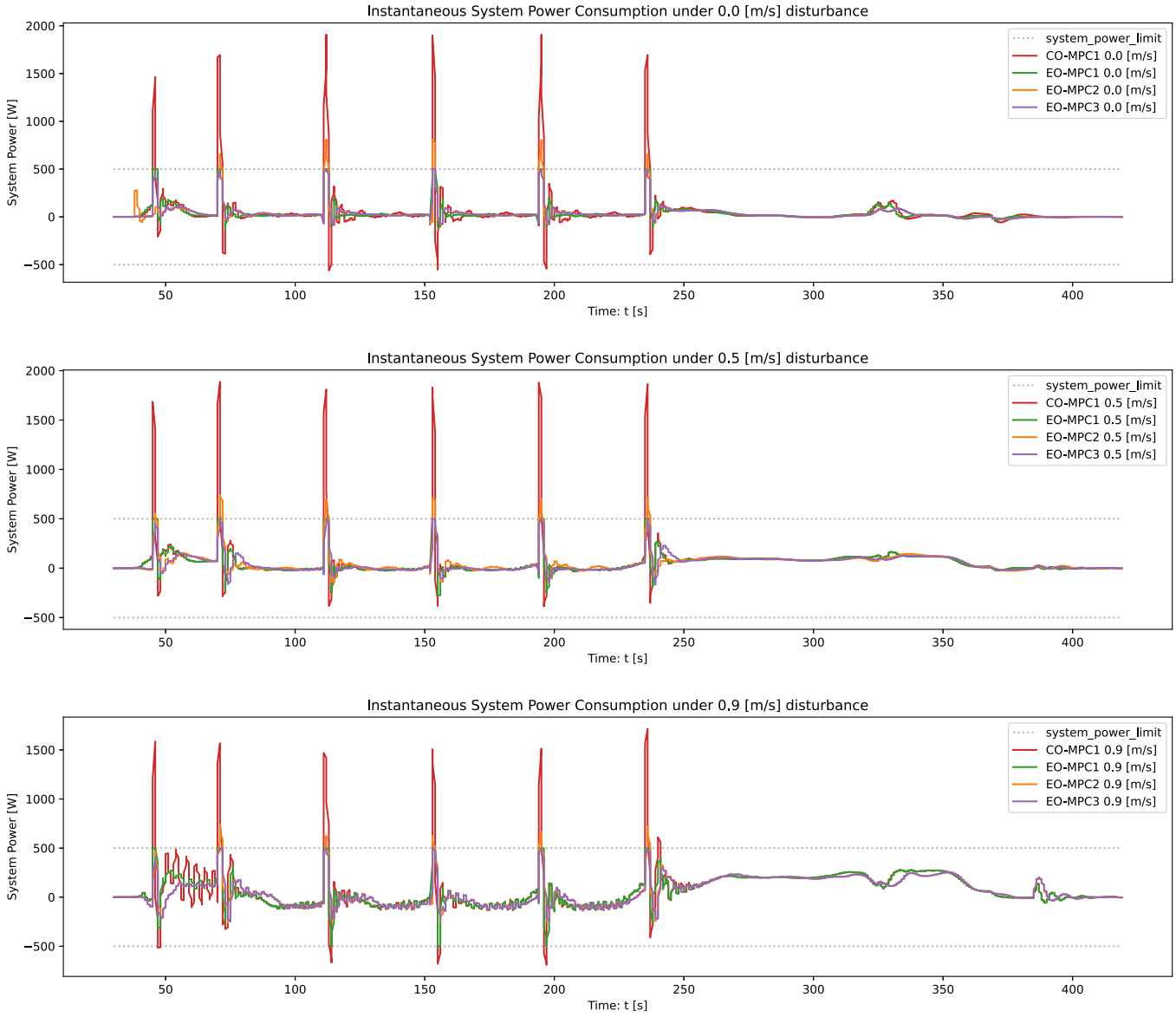


Fig. 18. Effects of non-convex system-level power function in MPC.

superior energy-optimality than CO-MPC1, while maintaining the trajectory tracking performance that is better than or equivalent to CO-MPC1. It is also important to note that the energy-saving feature of EO-MPC1 is evident in the energy-demanding region of the trajectory. The main challenge in the use of $p(f_T, v)$ in MPC is that MPC cannot be solved as a quadratic programming (QP) but as a nonlinear programming (NLP). In fact, $p(f_T, v)$ makes the original finite-horizon constrained nonlinear optimization into a finite-horizon constrained nonlinear non-convex optimization where a NLP solver (e.g., IPOPT) is crucial in handling constraints, nonlinear dynamics and non-convex functions all at the same time in real-time computation.

7.6. Tracking & energy consumption

As mentioned in the introduction, FNVI is the constrained tracking control problem with the need of energy-optimality. So, tracking performance and energy consumption are two main factors that must be considered simultaneously and a complete trade-off between them is not logically acceptable at least from the operation perspectives. Based on this reason, EO-MPC1 is the most suitable energy-optimal controller achieving more energy saving (3.1 % - 21.4 %) than CO-MPC1 under all disturbance speeds (0.0 m/s - 0.9 m/s) while throughout the trajectory, its tracking performance is equivalent to or better than CO-MPC1's performance.

Another important thing to highlight is that the day-to-day operation of FNVI is not necessarily around the helical trajectory in which majority of the region does not have drastic change in twists which demand energy. In real-world applications, it is tedious to design an energy-optimal trajectory in which the drastic change in twist is minimized and the flow of the disturbance is in favor of the UUV's motion. But instead, based on the operational requirement (e.g., a strong underwater disturbance current is flowing against the UUV's motion direction in the area of interest for FNVI), the trajectory cannot be designed to guarantee energy-optimality. In such scenario, EO-MPC1 can perform a more energy-optimal trajectory tracking, compared to CO-MPC1. Therefore, a general FNVI trajectory as shown in Fig. 8 represents a more realistic operational condition of an aquaculture farm.

8. Conclusion

In this paper, the step-by-step transition from the constrained optimal control problem (COCP) to the discrete version of COCP in the

MPC framework is detailed. The formulation of energy-optimal MPC (EO-MPC) with the explicit system-level power function and convexity test is demonstrated so that the proper (convex/ non-convex) optimization solvers can be chosen. The theoretical foundations of nonlinear programming (NLP) and one of the widely used NLP solvers, called IPOPT, to solve convex and non-convex optimization problems are discussed. Based on the experimental results, the insights into the effects of the non-convex system-level power function on energy optimization are also provided. Moreover, implementation aspects, together with the detailed algorithmic structure and exception (e.g., maximum iteration, infeasible problem) handler, are presented. Most importantly, using the actual specifications of the *Blue Endeavour Project* (the upcoming first-of-its-kind offshore salmon firm in New Zealand) of the New Zealand King Salmon Company and RexROV 2 with Sub-Fighter 30K's published model parameters, the proposed controllers are tested in high-fidelity simulation using Robot Operating System (ROS), integrated with Gazebo Physics Engine under three underwater current disturbance speeds (0.0 m/s, 0.5 m/s and 0.9 m/s). Our findings suggest that all three EO-MPCs can achieve energy-optimality under the varying underwater current disturbance speeds. However, only EO-MPC1 can achieve both better or equivalent tracking performance of CO-MPC1 throughout the trajectory and more energy saving (3.1 % - 21.4 %) than CO-MPC1 under all underwater current disturbance speeds (0.0 m/s - 0.9 m/s). Our future work will focus on robust MPC or Tube MPC to further verify energy-optimality under robustness guaranteed.

CRediT authorship contribution statement

Thein Than Tun: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Loulin Huang:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Formal analysis, Conceptualization; **Mark Anthony Preece:** Resources, Project administration, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge the financial support of the Blue Economy Cooperative Research Centre (CRC), established and supported under the Australian Government's CRC Program, grant number CRC-20180101. The CRC Program supports industry-led collaborations between industry, researchers and the community. The authors also acknowledge the graduate research facilities and the Ph.D. fees scholarship of the Auckland University of Technology (AUT). We greatly appreciate the technical support of Mr. Simon Hartley, the senior technician at AUT and Mr. Adam Poloha, the research assistant at Mechatronics Lab, AUT. We also acknowledge that Assoc. Prof. David Wilson's discussions on MPC implementation is insightful.

Appendix A. Appendix

MPC Algorithm with User-defined Stage/Running cost as a ROS node

The following algorithm implementation is based on Mehrez (2021) and substantially modified to work as a ROS node. **Note:** Where applicable, the same symbols as shown in Equations are not used in the following Solver setup as these parameters are CasADi symbolic parameters.

Algorithm: Suppose J_{N_p} is the performance index as shown in Eq. (18) over the prediction horizion, N_p, $X \in \mathbb{R}^{n\times N_p}$, $U \in \mathbb{R}^{m\times N_c}$ are CasADi symbolic matrices for the state and control effort, respectively. $P \in \mathbb{R}^{2n}$ is a CasADi vector comprising the current UUV's states and the reference trajectory, $g \in \mathbb{R}^{n\times N_p}$ or $\in \mathbb{R}^{(n\times N_p)+(N_p-1)}$ is a CasADi vector for equality constraints and F is the CasADi symbolic function of UUV model as shown in Eq. (15). The setup for each MPC PI and DMS can be written as shown in Algorithm 1 and the solver setup can be written as shown in Algorithm 2.

Suppose $const \in \mathbb{R}^n$ is a vector of pose (position: [45.25 45.25 45.25] m, orientation: [∞ ∞ ∞] rad) and twist (linear velocity: [∞ ∞ ∞] m/s

Algorithm 1: Setup for PIs and DMS.

$g \leftarrow X[:0] - P[:n]$;
$c_p \leftarrow$ casadi.SX(N_c, 1);
for i in $N_p - 1$ **do**
  $s \leftarrow X[:,i]$;
  $s_f \leftarrow X[:,-1]$;
  $s_n \leftarrow X[:,i+1]$;
  $s_d \leftarrow P[n:2n]$;
  $c \leftarrow U[:,i]$ **if** $i \le N_c - 1$ **else** $U[:,N_c-1]$;
  $c_p \leftarrow \mathbf{0} \in \mathbb{R}^{N_c}$ **if** $i < 1$ **else** c_p;
  $c_\delta \leftarrow c - c_p$;
  $e \leftarrow s_d - s$;
  $e_{\mathrm{PI}} \leftarrow e^T Q e$;
  $c_{\mathrm{PI}} \leftarrow c^T R c$;
  $c_{\delta,\mathrm{PI}} \leftarrow c_\delta^T R_\delta c_\delta$;
  $p_{\mathrm{PI}} \leftarrow c^T B_T^T s[\frac{n}{2}:n]$;
  if *sol_type* == CO − MPC1 **then**
    $J_{N_p} \leftarrow J_{N_p} + e_{\mathrm{PI}} + c_{\mathrm{PI}}$;
  else if *sol_type* == EO − MPC1 **then**
    $J_{N_p} \leftarrow J_{N_p} + e_{\mathrm{PI}} + c_{\mathrm{PI}} + \gamma_1 p_{\mathrm{PI}}$;
  else if *sol_type* == EO − MPC2 **then**
    $J_{N_p} \leftarrow J_{N_p} + e_{\mathrm{PI}} + c_{\mathrm{PI}} + c_{\delta,\mathrm{PI}}$;
  else if *sol_type* == EO − MPC3 **then**
    $J_{N_p} \leftarrow J_{N_p} + e_{\mathrm{PI}} + c_{\mathrm{PI}} + \gamma_1 p_{\mathrm{PI}} + c_{\delta,\mathrm{PI}}$;
  $s_{\mathrm{RK4}} \leftarrow$ using F and Eqn. (17);
  $g \leftarrow \begin{bmatrix} g \\ s_{\mathrm{RK4}} - s_n \end{bmatrix}$ **if** $i < N_p - 1$;
  $c_p \leftarrow c$;
$e_f \leftarrow s_d - s_f$;
$e_{f,\mathrm{PI}} \leftarrow e_f^T Q_f e_f$;
$J_{N_p} \leftarrow J_{N_p} + e_{f,\mathrm{PI}}$;
$g \leftarrow \begin{bmatrix} g \\ e_f \end{bmatrix}$;
if *sol_type* == EO − MPC1 or EO − MPC3 **then**
  for i in $N_p - 1$ **do**
    $s \leftarrow X[:,i]$;
    $c \leftarrow U[:,i]$ **if** $i \le N_c - 1$ **else** $U[:,N_c-1]$;
    $p_{\mathrm{PI}} \leftarrow c^T B_T^T s[\frac{n}{2}:n]$;
    $g \leftarrow \begin{bmatrix} g \\ p_{\mathrm{PI}} \end{bmatrix}$;

Algorithm 2: Solver setup.

$var \leftarrow \begin{bmatrix} X.\mathrm{reshape}((-1,1)) \\ U.\mathrm{reshape}((-1,1)) \end{bmatrix}$;
$nlp\_prob \leftarrow \{'\mathrm{f}' : J_{N_p}, '\mathrm{x}' : var, '\mathrm{g}' : g, '\mathrm{p}' : P\}$;
$opts \leftarrow \{'\mathrm{ipopt}' : \{'\mathrm{max\_iter}' : \mathrm{max\_}iter\}\}$;
$solver \leftarrow$ casadi.nlpsol('solver','ipopt', $nlp\_prob, opts$);

and angular velocity: [∞ ∞ ∞] rad/s) constraints in order. Other constraints are $f_{T,\text{bound}}$ =800 N, $p_{\min}$ =−500 W and $p_{\max}$ =500 W. As shown in Algorithm 3, they can be set as part of the arguments which will be passed to the solver later.

Algorithm 3: Impose constraints.

$lbx \leftarrow$ casadi.DM.zeros($(n \times N_p) + (m \times N_c)$, 1);
$ubx \leftarrow$ casadi.DM.zeros($(n \times N_p) + (m \times N_c)$, 1);
for i in n **do**
  $lbx[i : (n \times N_p) : n] \leftarrow -const[i]$;
  $ubx[i : (n \times N_p) : n] \leftarrow +const[i]$;
$lbx[(n \times N_p) :] \leftarrow -f_{T,\text{bound}}$;
$ubx[(n \times N_p) :] \leftarrow +f_{T,\text{bound}}$;
if *sol_type* == CO − MPC1 or EO − MPC2 **then**
  $lbg \leftarrow$ casadi.DM.zeros($(n \times N_p)$, 1);
  $ubg \leftarrow$ casadi.DM.zeros($(n \times N_p)$, 1);
else if *sol_type* == EO − MPC1 or EO − MPC3 **then**
  $lbg \leftarrow$ casadi.DM.zeros($(n \times N_p) + (N_p - 1)$, 1);
  $ubg \leftarrow$ casadi.DM.zeros($(n \times N_p) + (N_p - 1)$, 1);
  for i in range($N_p - 1$) **do**
    $i = (n \times N_p)) + i$;
    $lbg[i] \leftarrow p_{\min}$;
    $ubg[i] \leftarrow p_{\max}$;

Suppose $s_{\text{init}} \in \mathbb{R}^n$ and $s_{\text{tar}} \in \mathbb{R}^n$ are the initial and target states of the UUV, respectively. s_{init} is assigned by the state-estimation ROS node and s_{tar} is assigned by the reference trajectory. The solver needs the real-time update of solver parameters as arguments and it can be designed as shown in Algorithm 4.

Algorithm 4: Call solver.

$$args \leftarrow \left\{ \begin{array}{c} \text{'x}_0\text{'} : \begin{bmatrix} S.\text{reshape}(n \times N_p, 1) \\ C.\text{reshape}(m \times N_c, 1) \end{bmatrix} \\ \text{'lbx'} : lbx, \\ \text{'ubx'} : ubx, \\ \text{'lbg'} : lbg, \\ \text{'ubg'} : ubg, \\ \text{'p'} : \begin{bmatrix} s_{\text{init}} \\ s_{\text{tar}} \end{bmatrix} \end{array} \right\};$$

$sol \leftarrow solver(args)$

In addition to the implementation steps based on the theoretical work explained in NLP, the mitigation-steps to handle the exceptions (e.g., infeasible problem, maximum iteration steps) thrown from the NLP solvers must be considered in the implementation process for the simulation or field trials. The details of those exceptions are out of scope of this paper and it can be referred from Fagiano and Teel (2012), Boccia et al. (2014), Grüne (2012), Ipopt Documentation (2024). Brief explanations on those exceptions and mitigation-steps are as follows.

Infeasible problem: In general, the trajectory tracking problem based on NLP only works given a trajectory in which the desired pose and twist are within the system dynamics's constraints — $\zeta_d \in \mathbb{B}$. Suppose such problem is encountered together with the terminal constraint, it can be assumed feasible such that $\zeta^*, f_T^* \in \mathbb{B}$ over the horizon N_p as shown in Definition 3. However, if the trajectory violates the constraints or there is an external disturbance that causes NLP problem to produce $\zeta, f_T \notin \mathbb{B}$, the infeasible or recursive infeasible problem arises.

Algorithm 5: Solver with exception handler.

if *sol_status* fails **then**
  $i \leftarrow i + 1$;
  if *max_iter exception* **then**
    $m_i \leftarrow i \times 100$ **if** $m_i \le m_{i,\max}$ **else** $m_{i,\max}$; **Algorithm 2**
  else if *infeasible problem exception* **then**
    $s_c \leftarrow s_c \times 1.05$ **if** $s_c \le s_{c_{\max}}$ **else** $s_{c_{\max}}$;
    $c_c \leftarrow c_c \times 1.05$ **if** $c_c \le c_{c_{\max}}$ **else** $c_{c_{\max}}$;
    $const_{\text{new}} \leftarrow [s_c, c_c]$;
    Algorithm 3 with $const_{\text{new}}$;
  $S \leftarrow S \in \mathbb{R}^{n \times N_p}$;
  $C \leftarrow \mathbf{0} \in \mathbb{R}^{m \times N_c}$;
  $\zeta \leftarrow S[:, 0]$;
  $f_T \leftarrow C[:, 0]$;
else
  $S^* \leftarrow$ reshape($sol[\text{'x'}][: (n \times N_p)]) \in \mathbb{R}^{n \times N_p}$;
  $C^* \leftarrow$ reshape($sol[\text{'x'}][(n \times N_p) :]) \in \mathbb{R}^{m \times N_c}$;
  if $i \neq 0$ **then**
    $i \leftarrow 0$;
    Algorithm 3 with original *const*;

Maximum iteration steps: Due to the nature of the numerical solver in NLP, the user has to define max_iter parameter (3000 by default) during which the solver tries to reach acceptable convergence tolerance ($1e{-}8$ by default). Generally, it is better to set max_iter value higher than the iteration steps which the solver usually requires, and then, increase max_iter when the exception regarding max_iter occurs.

Suppose m_i and $m_{i,\max} = 2000$ are max_iter and max. max_iter. s_c and $s_{c,\max}$ are state constraint and max. state constraint. For each failed_solver_count, i, m_i is increased by 100 steps. The max. position constraint of $s_{c,\max}$ is set as $1.2s_c$. c_c =800 N and $c_{c,\max}$ =1540 N are control constraint and max. control constraint. When infeasible problem arises, state and control constraints are relaxed by 5 %. The solver with exception handler can be described as shown in Algorithm 5.

Algorithm 6: Controller.

Algorithm 4 ▷ Call Solver;
Algorithm 5 ▷ Exception Handler;
$\zeta^* \leftarrow S^*[:, 0]$;
$f_T^* \leftarrow C^*[:, 0]$;
$S \leftarrow$ Left Shifting on S^*;
$C \leftarrow$ Left Shifting on C^*;

The controller mainly imports the solver with the relevant arguments, checks if it encounters any exception and handles it accordingly. Subsequently, the optimal state, ζ^*, and control effort, f_T^*, are extracted from the optimal solution matrices, S^*, C^*, and horizon-shifting operation is performed on those matrices so that they can be used for "warm-start". The overall structure of such implementation is shown in Algorithm 6.

Finally, all the above Python functions can be clustered as parts of Python Class which will be used a ROS node. The detailed implementation using $R_c = 15$ is shown in Algorithm 7.

Algorithm 7: ROS controller node.

```
Class ROV_MPC_Controller():
    Function Initialization():
        init_node('rov_mpc_controller');
        N_p ← N_p;
        N_c ← N_c;
        ctrl_c ← False;
        R_c ← R_c;
        rate ← rospy.Rate(R_c);
        X ← casadi.SX.sym(n × N_p);
        U ← casadi.SX.sym(m × N_c);
        P ← casadi.SX.sym(2n);
        γ1, γ2 ← γ1, γ2 ∈ ℝ_0^+;
        Q ← Q ∈ ℝ^{n×n};
        R ← R ∈ ℝ^{m×m};
        Q_f ← γ3 Q ∈ ℝ^{n×n};
        Algorithm 1                ▷ PIs and DMS;
        Algorithm 2                ▷ Solver setup;
        Algorithm 3                ▷ Impose constraints;
        S ← 0 ∈ ℝ^{n×N_p};
        C ← 0 ∈ ℝ^{m×N_c};
        f*_{T,prev} ← 0 ∈ ℝ^m;
        E ← 0                      ▷ Total energy consumption = 0;
        ROS Subscribers ...;
        ROS Publishers ...;
        t_start ← ROS_time;
    Function ROS_Callbacks():
        Handle s_init;
        Handle s_tar;
        Handle others;
    Function Call_Controller():
        Algorithm 6;
        Δt ← ROS_time − t_start;
        t_start ← ROS_time;
        E ← E+ Eqn. (35) as f(Δt, f*_{T,prev});
        Publish data;
        f*_{T,prev} ← f*_T;
    Function Shut_Down():
        Publish f*_T ← 0;
        ctrl_c ← True;
if _name_ == '_main_' then
    try:
        mpc_control_obj ← ROV_MPC_Controller();
        while not rospy.is_shutdown() do
            mpc_control_obj.Call_Controller();
            mpc_control_obj.rate.sleep();
    except rospy.ROSInterruptException:
        mpc_control_obj.Shut_down();
        print("Exception. Shut Down.");
```

References

Akram, W., Casavola, A., Kapetanović, N., Mišković, N., 2022. A visual servoing scheme for autonomous aquaculture net pens inspection using ROV. Sensors 22 (9), 3525. https://doi.org/10.3390/s22093525

Amundsen, H.B., Caharija, W., Pettersen, K.Y., 2021. Autonomous ROV inspections of aquaculture net pens using DVL. IEEE J. Oceanic Eng. 47 (1), 1–19. https://doi.org/10.1109/JOE.2021.3105285

Andersson, J. A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M., 2019. CasADi – a software framework for nonlinear optimization and optimal control. Math. Prog. Comput. 11 (1), 1–36. https://doi.org/10.1007/s12532-018-0139-4

Antonelli, G., 2018. Underwater Robots. Springer International Publishing AG.

Berg, V., 2012. Development and Commissioning of a DP System for ROV SF 30k. Master's thesis. Norwegian University of Science and Technology. https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/239258. [Online, Accessed 20 November 2024].

Boccia, A., Grüne, L., Worthmann, K., 2014. Stability and feasibility of state constrained MPC without stabilizing terminal constraints. Syst. Control Lett. 72, 14–21. https://doi.org/10.1016/j.sysconle.2014.08.002

Chu, Y.I., Wang, C.M., Park, J.C., Lader, P.F., 2020. Review of cage and containment tank designs for offshore fish farming. Aquaculture 519, 734928. https://doi.org/10.1016/j.aquaculture.2020.734928

Claus, B., Bachmayer, R., 2016. Energy optimal depth control for long range underwater vehicles with applications to a hybrid underwater glider. Auton. Robots 40 (7), 1307–1320. https://doi.org/10.1007/s10514-016-9555-3

Cutler, C.R., Ramaker, B.L., 1980. Dynamic matrix control - a computer control algorithm. In: Proceedings of the Joint Automatic Control Conference.

De Schutter, J., Leuthold, R., Bronnenmeyer, T., Malz, E., Gros, S., Diehl, M., 2023. AWE-box: an optimal control framework for single- and multi-aircraft airborne wind energy systems. Energies 16 (4). https://doi.org/10.3390/en16041900

Diehl, M., 2013. Optimization Algorithms for Model Predictive Control. Springer London, London. pp. 1–11. https://doi.org/10.1007/978-1-4471-5102-9_9-1

Diehl, M., Ferreau, H.J., Haverbeke, N., 2009. Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 391–417. https://doi.org/10.1007/978-3-642-01094-1_32

Diehl, M., Gros, S., 2016. Constrained Nonlinear Optimization [Online, Accessed 20 September 2024]. https://www.syscop.de/teaching/ws2016/winter-school-numerical-optimal-control-differential-algebraic-equations.

Drgoňa, J., Arroyo, J., Cupeiro Figueroa, I., Blum, D., Arendt, K., Kim, D., Ollé, E.P., Oravec, J., Wetter, M., Vrabie, D.L., Helsen, L., 2020. All you need to know about model predictive control for buildings. Annu. Rev. Control 50, 190–232. https://doi.org/https://doi.org/10.1016/j.arcontrol.2020.09.001

Du, X., Chen, D., Yan, Z., 2020. Research on control method of unmanned underwater vehicle dynamic positioning based on energy consumption optimization. Int. J. Adv. Rob. Syst. 17 (5), 1729881420933835.

Fagiano, L., Teel, A.R., 2012. Model predictive control with generalized terminal state constraint. IFAC Proc. 45 (17), 299–304. 4th IFAC Conference on Nonlinear Model Predictive Control. https://doi.org/10.3182/20120823-5-NL-3013.00063

Falcone, M., 2021. Optimal Control and the Dynamic Programming Principle. Springer International Publishing, Cham. pp. 1585–1590. https://doi.org/10.1007/978-3-030-44184-5_209

Fossen, T.I., 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons.

Gong, P., Yan, Z., Zhang, W., Tang, J., 2022. Trajectory tracking control for autonomous underwater vehicles based on dual closed-loop of MPC with uncertain dynamics. Ocean Eng. 265, 112697. https://doi.org/https://doi.org/10.1016/j.oceaneng.2022.112697

Grüne, L., 2021. Numerical Methods for Nonlinear Optimal Control Problems. Springer International Publishing, Cham. pp. 1540–1547. https://doi.org/10.1007/978-3-030-44184-5_208

Grüne, L., 2012. NMPC without terminal constraints. IFAC Proc. 45 (17), 1–13. 4th IFAC Conference on Nonlinear Model Predictive Control. https://doi.org/10.3182/20120823-5-NL-3013.00030

Henson, M.A., 1998. Nonlinear model predictive control: current status and future directions. Comput. Chem. Eng. 23 (2), 187–202. https://doi.org/10.1016/S0098-1354(98)00260-9

Hu, Y., Li, B., Jiang, B., Han, J., Wen, C.-Y., 2024. Disturbance observer-based model predictive control for an unmanned underwater vehicle. J. Mar. Sci. Eng. 12 (1). https://doi.org/10.3390/jmse12010094

Huang, J., Hu, P., Wu, K., Zeng, M., 2018. Optimal time-jerk trajectory planning for industrial robots. Mech. Mach. Theory 121, 530–544. https://doi.org/10.1016/j.mechmachtheory.2017.11.006

Huynh, V.T., Dunbabin, M., Smith, R.N., 2015. Predictive motion planning for AUVs subject to strong time-varying currents and forecasting uncertainties. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1144–1151. https://doi.org/10.1109/ICRA.2015.7139335

Ipopt Documentation, 2024. Ipopt Options [Online, Accessed 12 September 2024]. https://coin-or.github.io/Ipopt/OPTIONS.html.

Köhler, J., Müller, M.A., Allgöwer, F., 2024. Analysis and design of model predictive control frameworks for dynamic operation-an overview. Annu. Rev. Control 57, 100929. https://doi.org/https://doi.org/10.1016/j.arcontrol.2023.100929

Li, X., Yu, S., Gao, X.-z., Yan, Y., Zhao, Y., 2024. Path planning and obstacle avoidance control of UUV based on an enhanced a* algorithm and MPC in dynamic environment. Ocean Eng. 302, 117584. https://doi.org/https://doi.org/10.1016/j.oceaneng.2024.117584

Li, Y., Wang, Z., Yang, H., Zhang, H., Wei, Y., 2023. Energy-optimal planning of robot trajectory based on dynamics. Arab. J. Sci. Eng. 48 (3), 3523–3536. https://doi.org/10.1007/s13369-022-07185-7

Liang, Haojiao and Li, Huiping and Gao, Jian and Cui, Rongxin and Xu, Demin, 2023. Economic MPC-based planning for marine vehicles: tuning safety and energy efficiency. IEEE Trans. Ind. Electron. 70 (10), 10546–10556. https://doi.org/10.1109/TIE.2022.3220876

Lofberg, J., 2004. Yalmip : a toolbox for modeling and optimization in matlab. In: 2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508), pp. 284–289. https://doi.org/10.1109/CACSD.2004.1393890

Lofberg, J., 2025. YALMIP Solvers [Online,Accessed 19 July 2024]. https://yalmip.github.io/allsolvers/

Manhães, M. M.M., Scherer, S.A., Voss, M., Douat, L.R., Rauschenbach, T., 2016. UUV simulator: a Gazebo-based package for underwater intervention and multi-robot

simulation. In: OCEANS 2016 MTS/IEEE Monterey. IEEE. https://doi.org/10.1109/oceans.2016.7761080
Mayne, D.Q., Rawlings, J.B., 2001. Correction to "constrained model predictive control: stability and optimality. Automatica 37 (3), 483. https://doi.org/10.1016/S0005-1098(00)00173-4
Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M., 2000. Constrained model predictive control: stability and optimality. Automatica 36 (6), 789–814. https://doi.org/10.1016/S0005-1098(99)00214-9
Mehrez, M.W., [2017]. MPC and-MHE Implementation in MATLAB using Casadi [Online, Accessed 10 July 2024]. https://github.com/MMehrez/MPC-and-MHE-implementation-in-MATLAB-using-Casadi/tree/master/workshop_github.
Morari, M., Lee, J.H., 1999. Model predictive control: past, present and future. Comput. Chem. Eng. 23 (4–5), 667–682. https://doi.org/10.1016/S0098-1354(98)00301-9
Musa Morena Marcusso Manhães, [2023]. RexROV 2 [Online, Accessed 17 October 2024]. https://github.com/uuvsimulator/rexrov2.
Musa Morena Marcusso Manhães, [2023]. Uuv_simulator: Unmanned Underwater Vehicle (UUV) Simulation with Gazebo [Online, Accessed 20 November 2024]. https://github.com/uuvsimulator/uuv_simulator.
New Zealand King Salmon, [2024]. Blue Endeavour [Online, Accessed 25 September 2024]. https://www.kingsalmon.co.nz/open-ocean-blue-endeavour/.
Newcombe, E., Knight, B., Smeaton, M., Bennett, H., Mackenzie, L., Scheel, M., Vennell, R., Campos, C., 2020. Water column assessment for a proposed salmon farm offshore of the marlborough sounds. Prepared for The New Zealand King Salmon Co. Limited. Cawthron Report No. 3313 , 96.
Nocedal, J., Wright, S.J., 2006. Numerical Optimization. Springer New York. https://doi.org/10.1007/978-0-387-40065-5
Preece, M.A., 2021. Evidence of mark anthory preece on behalf of the applicant. Applicant's Evidence - 022 - M Preece - NZKS Operational Matters [Online]. https://eservices.marlborough.govt.nz/programmes/ListProgrammeEvents?id=3516198.
Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control Eng. Pract. 11 (7), 733–764. https://doi.org/10.1016/S0967-0661(02)00186-7
Richalet, J., Rault, A., Testud, J.L., Papon, J., 1978. Model predictive heuristic control: applications to industrial processes. Automatica 14 (5), 413–428. https://doi.org/10.1016/0005-1098(78)90001-8
Saback, R.M., Conceicao, A. G.S., Santos, T. L.M., Albiez, J., Reis, M., 2020. Nonlinear model predictive control applied to an autonomous underwater vehicle. IEEE J. Oceanic Eng. 45 (3), 799–812. https://doi.org/10.1109/JOE.2019.2919860
Sarkar, M., Nandy, S., Vadali, S. R.K., Roy, S., Shome, S.N., 2016. Modelling and simulation of a robust energy efficient AUV controller. Math. Comput. Simul. 121, 34–47. https://doi.org/10.1016/j.matcom.2015.08.021
Schwenzer, M., Ay, M., Bergs, T., Abel, D., 2021. Review on model predictive control: an engineering perspective. Int. J. Adv. Manuf. Technol. 117, 1327–1349. https://doi.org/10.1007/s00170-021-07682-3
Shen, C., Shi, Y., Buckham, B., 2018. Trajectory tracking control of an autonomous underwater vehicle using lyapunov-based model predictive control. IEEE Trans. Ind. Electron. 65 (7), 5796–5805. https://doi.org/10.1109/TIE.2017.2779442
Shen, C., Shi, Y., Buckham, B., 2019. Path-following control of an AUV: a multiobjective model predictive control approach. IEEE Trans. Control Syst. Technol. 27 (3), 1334–1342. https://doi.org/10.1109/TCST.2018.2789440
Spangelo, I., Egeland, O., 1992. Computing energy-optimal trajectories for an autonomous underwater vehicle using direct shooting. Model. Identif. Control J. 13 (3), 163–174. https://doi.org/10.4173/mic.1992.3.4
The New Zealand Government, 2025. Aquaculture strategy [Online, Accessed 05 June 2024]. https://www.mpi.govt.nz/fishing-aquaculture/aquaculture-fish-and-shellfish-farming/aquaculture-strategy-for-new-zealand/.
Tun, T.T., Huang, L., Preece, M.A., 2023. Development and high-Fidelity simulation of trajectory tracking control schemes of a UUV for fish net-Pen visual inspection in offshore aquaculture. IEEE Access 11, 135764–135787. https://doi.org/10.1109/ACCESS.2023.3337872
Veeraklaew, T., Piromsopa, P., Chirungsarpsook, K., Pattaravarangkur, C., 2005. A study on the comparison between minimum jerk and minimum energy of dynamic systems. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). Vol. 1, pp. 523–528. https://doi.org/10.1109/CIMCA.2005.1631317
Vinter, R., 2021. Optimal Control and Pontryagin's Maximum Principle. Springer International Publishing, Cham. pp. 1578–1584. https://doi.org/10.1007/978-3-030-44184-5_200
Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106, 25–57. https://doi.org/10.1007/s10107-004-0559-y
Yang, N., Chang, D., Amini, M.R., Johnson-Robersor, M., Sun, J., 2019. Energy management for autonomous underwater vehicles using economic model predictive control. In: 2019 American Control Conference (ACC), pp. 2639–2644. https://doi.org/10.23919/ACC.2019.8815106
Yang, N., Chang, D., Johnson-Roberson, M., Sun, J., 2020. Robust energy-optimal path following control for autonomous underwater vehicles in ocean currents. In: 2020 American Control Conference (ACC), pp. 5119–5124. https://doi.org/10.23919/ACC45564.2020.9147322
Yang, N., Reza Amini, M., Johnson-Roberson, M., Sun, J., 2018. Real-time model predictive control for energy management in autonomous underwater vehicle. In: 2018 IEEE Conference on Decision and Control (CDC), pp. 4321–4326. https://doi.org/10.1109/CDC.2018.8619844
Yang, N., Shen, C., Song, Z., Johnson-Roberson, M., Sun, J., 2024. Robust energy-optimal control for 3-D path-following of autonomous underwater vehicles under ocean currents. IEEE Trans. Control Syst. Technol. 32 (2), 680–687. https://doi.org/10.1109/TCST.2023.3315602
Yao, F., Yang, C., Zhang, M., Wang, Y., 2019. Optimization of the energy consumption of depth tracking control based on model predictive control for autonomous underwater vehicles. Sensors 19 (1), 162. https://doi.org/10.3390/s19010162
Zavala, V.M., Biegler, L.T., 2009. Nonlinear Programming Strategies for State Estimation and Model Predictive Control. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 419–432. https://doi.org/10.1007/978-3-642-01094-1_33
Zheng, A., Morari, M., 1995. Stability of model predictive control with mixed constraints. IEEE Trans. Automat. Contr. 40 (10), 1818–1823. https://doi.org/10.1109/9.467664

Thein Than Tun is a Ph.D. candidate from Auckland University of Technology (AUT), New Zealand. Before joining AUT, he worked in the robotics industry and academic research laboratory at the Singapore University of Technology and Design (SUTD). He completed B.E in Engineering Product Development (EPD) at SUTD in 2016 and a Diploma in Mechatronic Engineering from Ngee Ann Polytechnic, Singapore in 2012.

Dr. Loulin Huang is a Professor of Mechatronics in the Department of Mechanical Engineering of Auckland University of Technology. He obtained the Bachelor and Master degrees in Mechanical Engineering from Huazhong University of Science and Technology (China), Master degree in Electrical and Electronic Engineering from Nanyang Technology University (Singapore) and PhD degree from National University of Singapore. He has about 35 years' experience in the research and teaching in the areas of mechatronics and robotics with a distinctively interdisciplinary flavour. He has completed more than thirty industrial and government-funded projects in a wide range of industrial applications and has (co)authored more than 120 research publications.

Mark Anthony Preece is the Seawater manager for the New Zealand King Salmon Company. He obtained his Masters degree in Marine Science from the University of Otago (Aotearoa / New Zealand), and post graduate Diploma in Business Studies Through Massey University (Aotearoa / New Zealand). He has around 30 years' of practical experience in the Aquaculture industry, mainly focussed on salmonids. The experience has involved managing and running salmon farming operations, undertaking research on nutritional aspects of farmed salmon and abalone, as well as applications and managing the practical design process for open ocean aquaculture farms. He has (co)authored more than 15 research publications.