Robotica (2025), **43**, pp. 3030–3057
doi:10.1017/S0263574725102099

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Data-driven controller adaptation and parameter estimation in autonomous underwater vehicles by genetic algorithm enhanced unfalsification approach

Tabassum Rasul and Koena Mukherjee

NIT Silchar, Cachar, Assam, India
Corresponding author: Tabassum Rasul; Email: tabassum_rs@ei.nits.ac.in

Received: 1 December 2024; **Revised:** 9 April 2025; **Accepted:** 2 July 2025; **First published online:** 6 August 2025

Keywords: data-driven control; adaptive control; unfalsified control; genetic algorithm; autonomous underwater vehicle

Abstract

The paper presents an enhanced method for unknown parameter estimation and nonlinear controller adaptation that combines the concept of unfalsification with the genetic algorithm (GA). This approach is based on the measured data and employs a bank of nonlinear controllers designed to dynamically adjust to the system's evolving conditions. The controllers in the bank can be switched to meet the system's requirements. This method is applied to an autonomous underwater vehicle (AUV) with uncertain parameters. Using the unfalsification method, these uncertain parameters are estimated, and a suitable controller is selected from the bank to guide the AUV along a desired trajectory. Additionally, an artificial intelligence technique, such as the GA, is employed to update the controller bank, resulting in versatile and optimised candidates. The simulation results obtained in the MATLAB/Simulink environment show that in the environment considered in this paper, the Adaptive Unfalsification algorithm in conjunction with GA estimates the unknown parameter values better than the sole GA-optimised values. Also, the convergence of the actual trajectory of the AUV using the Adaptive Unfalsification algorithm in conjunction with GA is faster and better than the sole GA-optimised algorithm. Furthermore, a survey of experimental results from established literature is included to evaluate the practical implementation of the proposed design, which concludes that within a reasonable time, the Adaptive Unfalsification algorithm in conjunction with GA can be implemented in commercially available processors.

1. Introduction

The rising use of autonomous underwater vehicles (AUVs) in marine industries and naval operations has led to a notable increase in demand for their production. Consequently, the need to develop efficient AUV models with precise control systems has grown in the research field. As is well-known, AUVs must operate in uncertain and unpredictable environmental conditions, making it a challenging task to maintain control over these vehicles. These operations require advanced control systems capable of adapting to changing conditions. Several research studies are being conducted to design and develop new AUV models. Researchers are proposing innovative methods to effectively control AUVs in the presence of disturbances and uncertainties. Various robust control techniques have been employed to improve navigation and communication performance, including those discussed in refs. [1–4]. These studies present trajectory tracking control techniques for underwater vehicles, addressing disturbances, measurement errors, actuator dynamics, and nonlinearities. In ref. [5], the adaptive impedance control method is specifically designed to handle unmodeled uncertainties using function approximation techniques. Additionally, adaptive control methods, as highlighted in ref. [6], observer-based control in ref. [7], and optimal control strategies from ref. [8], have contributed to these advancements. Research has also addressed tracking control for AUVs [9], focusing on scenarios with limited torque and without velocity sensors, thus advancing control strategies for practical AUV applications. The latest developments in this field include

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

intelligent control methods, as explored in refs. [10, 11]. Disturbance observer-based control methods, applied in refs. [12–15], have been used to address environmental disturbance factors in AUV control.

The control systems discussed above have been crucial in improving the performance of AUVs under uncertain conditions. However, these systems often require precise parameter tuning to achieve optimal performance. In recent years, researchers have explored a variety of optimisation techniques to enhance the design and operation of AUVs. One such approach that has gained attention is the use of genetic algorithms (GA), which have been applied in several studies to fine-tune controller parameters and optimise various AUV design elements. The application of GA in AUVs has been explored for tuning controller parameters, as seen in studies such as refs. [16–18]. In these works, model predictive control (MPC) is utilised to follow a specified trajectory while addressing external disturbances and uncertainties in the model. In the literature, GA has also been employed for controller gain optimisation in refs. [19, 20], and for hull design optimisation in ref. [21, 22], as well as path length optimisation in [23] and [11], among others.

However, the methods outlined in these papers often require knowledge of system parameters and their accurate mathematical modelling. This is particularly challenging in the case of AUVs due to the constant changes in hydrodynamic parameters and the added mass of the vehicle. This challenge motivates the design of a control system that is not model-dependent but is instead data-driven and adaptive, capable of handling disturbances effectively. The unfalsified control (UC) method is one such data-driven approach, first devised by Safonov and Tsao [24]. It is a supervisory control method that consists of a bank of controllers, of which the one that satisfies the desired performance criteria gets selected. When the conditions and performance criteria change and are not satisfied by the current controller, switching takes place, and another controller from the bank that fits the criteria gets selected next. This method does not require any information about system parameters. The supervisory controller adapts its actions based on modified conditions derived from measured input and output data. This approach was implemented to control a 2-degree-of-freedom (DOF) robotic arm, as demonstrated in ref. [25]. The mathematical framework and various properties of this method were examined in ref. [26]. Additionally, its robustness was tested in ref. [27], and stability characteristics were analysed in ref. [28]. The concept of unfalsification has also been employed in designing controllers for reference tracking, as discussed in refs. [29, 30], where it was integrated with model reference adaptive control. Furthermore, this control design was applied to a knee joint in neuroprosthetic processes, as noted in ref. [31], and later adapted for temperature control in a reactor system in ref. [32]. In the context of AUVs, it was utilised for dynamic positioning in ref. [33]. Moreover, data-driven methods have been employed for sample collection in oceanographic studies, as shown in refs. [34, 35]. In ref. [36], optimisation of the H_∞ controller was applied within the framework of UC, while ref. [37] introduced the design of a data-driven H_∞ loop shaping controller. Recently, this method has been investigated in the context of nonlinear time-varying systems, as discussed in refs. [38–40].

The literature on UC theory is limited to the fact that the selection of an unfalsified controller is made from a predetermined pool of candidates, which limits the robustness of the unfalsified controller. Thus, an intelligent optimisation-based data-driven method is proposed to vary the candidate set using the concept of GA to improve the scope of the unfalsified technique. Therefore, the improved closed-loop system now not only has access to a diversified candidate controller set but also gets the unfalsified controller selected from an optimised set of values, bearing in mind the requirement of optimisation of energy resources in AUV applications. Therefore, aiming to improve the exploring ability and thereby the performance of the unfalsified controller, the following contributions are made in this paper:

1. A novel unfalsified controller is designed for uncertain parameter estimation and trajectory tracking of an AUV. The unfalsified controller is a data driven, online technique and does not require any model parameters. Moreover, the controller ensures trajectory tracking in the presence of unknown ocean waves and currents.
2. The controller bank is optimised using GA-based method to increase the robustness of the proposed controller. This results in a diversified set of candidate controllers, and the final controller is selected from this modified set of candidates.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

3. The stability analysis of the complete closed-loop system is performed using Lyapunov-based techniques, and the system stability is proved using the feasibility condition. Furthermore, finite switching is also ensured through proof of the hysteresis algorithm.
4. The controller is simulated for both adaptive unfalsified controller with GA-based optimisation and only GA optimisation. The simulation results prove that an adaptive unfalsified controller with GA-based optimisation performs significantly better trajectory tracking in the presence of external disturbances and parametric uncertainty than a sole GA-based technique. Also, the absolute error analysis proves that the estimated values obtained by adaptive falsification in conjunction with GA remain well within the known parametric bounds and show better convergence as compared to those of only GA-optimised values.
5. Furthermore, various aspects of the practical implementation of the adaptive unfalsified controller with GA-based optimisation are discussed to note that this controller design can be implemented in real time using commercially available processors.

The structure of the paper is organised as follows: Section 2 provides a detailed description of the AUV system. In Section 3, the problem is outlined along with the problem statement. Section 4 explains the controller structure with an introduction to the GA and its implementation in the controller design. The stability theorem and proof for the proposed controller structure are presented in Section 5. Section 6 presents the algorithm of unfalsification. Section 7 showcases the results of the simulation and compares the proposed design with the GA optimisation. A survey of the existing literature on the implementation of optimisation techniques such as GAs on AUV hardware is presented in Section 7, thus assessing the feasibility of the proposed design. Finally, Section 8 concludes the paper, followed by the bibliography and the appendix.

Annotations used: The notation $[\,\cdot\,]^{\mathrm{T}}$ indicates the transpose of a matrix, $\|\cdot\|$ represents the Euclidean norm of a signal, and $\int$ denotes the integration of a function. The symbol J refers to a rotation matrix dependent on the chosen Euler angles, while $\mathfrak{J}$ signifies the cost function, which depends on system output, control input, and a fictitious reference. More details on these topics will be discussed in the following sections.

2. AUV system description and mathematical representation

The location and alignment of the AUV are defined in relation to two reference frames: a stationary inertial frame known as the Earth-fixed frame (X_E, Y_E, Z_E) and a dynamic body-fixed frame (X_B, Y_B, Z_B), as illustrated in Figure 1. The mathematical model of the AUV describes its kinematics, which is the geometric aspect of motion, and its dynamics, which examines the forces that drive that motion.

2.1. AUV kinematics

AUVs are capable of moving in 6 DOF. According to SNAME notation [41], these motions are labelled as follows: linear motion along the X-axis is referred to as surge (x), while the corresponding rotational motion is termed roll (φ). For the Y-axis, linear motion is identified as sway (y), with the rotational motion being called pitch (θ). Finally, along the Z-axis, linear motion is designated as heave (z), while the associated rotational motion is known as yaw (ψ).

The vector $\eta = [\eta_1^T, \eta_2^T]^{\mathrm{T}}$ comprises linear displacements $\eta_1 = [x, y, z]^{\mathrm{T}}$ and angular displacements $\eta_2 = [\varphi, \theta, \psi]^{\mathrm{T}}$, all in relation to the Earth-fixed frame.

In the same manner, the vector $\nu = [\nu_1^T, \nu_2^T]^{\mathrm{T}}$ represents both linear and angular velocities in relation to the body-fixed frame, where $\nu_1 = [u, v, w]^{\mathrm{T}}$ and $\nu_2 = [p, q, r]^{\mathrm{T}}$.

Therefore, the kinematics of the AUV can be derived from these definitions as:

$$\dot{\eta} = J(\eta)\nu \tag{1}$$

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

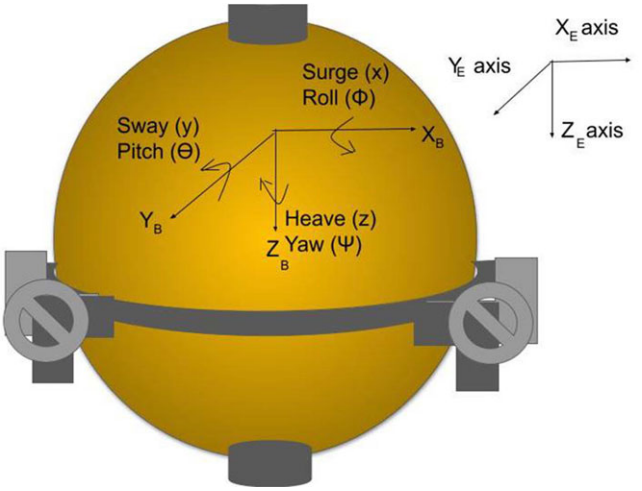


Figure 1. *Reference frames and relative displacements in AUV.*

where $J(\eta)$ is the transformation matrix that converts the velocity from the body-fixed frame to the Earth-fixed frame, expressed as follows:

$$J(\eta) = \begin{bmatrix} J_1(\eta_2) & 0_{3\times3} \\ 0_{3\times3} & J_2(\eta_2) \end{bmatrix} \tag{2}$$

$$J_1(\eta_2) = \begin{bmatrix} c\psi c\theta & -s\psi c\varphi + c\psi s\theta s\varphi & s\psi s\varphi + c\psi c\varphi s\theta \\ s\psi c\theta & c\psi c\varphi + s\varphi s\theta s\psi & -c\psi s\varphi + s\theta s\psi c\varphi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{bmatrix} \tag{3}$$

$$J_2(\eta_2) = \begin{bmatrix} 1 & s\varphi t\theta & c\varphi t\theta \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi/c\theta & c\varphi/c\theta \end{bmatrix} \tag{4}$$

s, c, and t represent the sine, cosine, and tangent of angles, respectively, and will be used as such in all subsequent references throughout the paper.

2.2. AUV dynamics

The hydrodynamic model of an AUV takes into account various factors, including added mass, viscous damping, and gravitational and restoring forces, as well as the forces and moments generated by the thrusters that control the vehicle's movement. The interplay of these forces and the resulting motion of the AUV can be represented by the following nonlinear equation:

$$\mathcal{M}\ddot{\eta} + \mathcal{C}(\eta, \dot{\eta})\dot{\eta} + \mathcal{D}(\eta, \dot{\eta})\dot{\eta} + g(\eta) = \mathcal{T} + \mathcal{T}_E \tag{5}$$

The mass matrix is denoted as $\mathcal{M} \in \mathcal{R}^{6\times6}$, while the Coriolis and centripetal matrix is represented by $\mathcal{C}(\eta, \dot{\eta}) \in \mathcal{R}^{6\times6}$. The damping matrix is indicated as $\mathcal{D}(\eta, \dot{\eta}) \in \mathcal{R}^{6\times6}$, and the gravitational matrix is given by $g(\eta) \in \mathcal{R}^6$. The control input forces $\mathcal{T} \in \mathcal{R}^6$ encompasses the thruster forces and torques, while $\mathcal{T}_E \in \mathcal{R}^6$ accounts for disturbances arising from environmental influences.

2.2.1 Properties

P1: $\mathcal{M}$ is symmetric and strictly positive definite, satisfying the condition:

$$\delta_m I_n \leq \mathcal{M} \leq \delta_M I_n \tag{6}$$

δ_m, δ_M are positive constants, and n is the degree of freedom.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

P2: Since $\mathcal{M} > 0$, therefore, $\mathcal{M}^{-1} \neq 0$. The system parameters are defined as follows [42]:

$$\mathcal{M} = \begin{bmatrix} m_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & m_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & m_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & m_{66} \end{bmatrix} \tag{7}$$

$$\mathcal{C} = \begin{bmatrix} 0_{3\times 3} & \mathcal{C}_{12} \\ \mathcal{C}_{21} & \mathcal{C}_{22} \end{bmatrix} \tag{8}$$

$$\mathcal{C}_{12} = \mathcal{C}_{21} = \begin{bmatrix} 0 & m_{33}\dot{z} & -m_{22}\dot{y} \\ -m_{33}\dot{z} & 0 & m_{11}\dot{x} \\ m_{22}\dot{y} & -m_{11}\dot{x} & 0 \end{bmatrix} \tag{9}$$

$$\mathcal{C}_{22} = \begin{bmatrix} 0 & m_{66}\dot{\psi} & -m_{55}\dot{\theta} \\ -m_{66}\dot{\psi} & 0 & m_{44}\dot{\varphi} \\ m_{55}\dot{\theta} & -m_{44}\dot{\varphi} & 0 \end{bmatrix} \tag{10}$$

$$\mathcal{D} = \begin{bmatrix} d_{11}|\dot{x}| & 0 & 0 & 0 & 0 & 0 \\ 0 & d_{22}|\dot{y}| & 0 & 0 & 0 & 0 \\ 0 & 0 & d_{33}|\dot{z}| & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{44}|\dot{\varphi}| & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{55}|\dot{\theta}| & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{66}|\dot{\psi}| \end{bmatrix} \tag{11}$$

$$g = \begin{bmatrix} (W-B)s\theta \\ -(W-B)c\theta s\varphi \\ -(W-B)c\theta c\varphi \\ -(y_G W - y_B B)c\theta c\varphi + (z_G W - z_B B)c\theta s\varphi \\ (z_G W - z_B B)s\theta + (x_G W - x_B B)c\theta c\varphi \\ -(x_G W - x_B B)c\theta s\varphi - (y_G W - y_B B)s\theta \end{bmatrix}^{\mathrm{T}} \tag{12}$$

where (x_G, y_G, z_G) is the centre of gravity and (x_B, y_B, z_B) is the centre of buoyancy and

$$\mathcal{T} = \begin{bmatrix} \mathcal{T}_x & \mathcal{T}_y & \mathcal{T}_z & \mathcal{T}_\varphi & \mathcal{T}_\theta & \mathcal{T}_\psi \end{bmatrix}^{\mathrm{T}} \tag{13}$$

2.3. Disturbances affecting an AUV

Disturbances can be present in any system, potentially resulting in adverse effects on its operation and overall stability. The existence of these disturbances and uncertainties leads to poor performance of the controller designed to control the system. These disturbances may be internal, such as sensor noise, or external in nature like environmental disturbance. The two main types of environmental disturbances that pose a threat to the control of AUVs are [43]:

- Waves
- Ocean currents

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

In the dynamic equations of AUV (5), the environmental disturbance $\mathcal{T}_E$ can be expressed as

$$\mathcal{T}_E = \mathcal{T}_W + \mathcal{T}_O \tag{14}$$

where $\mathcal{T}_W$ is a periodic signal caused by waves and $\mathcal{T}_O$ is a constant signal caused by ocean currents.

Using Eq. (14) in Eq. (5), we have

$$\mathcal{M}\ddot{\eta} + \mathcal{C}(\eta, \dot{\eta})\dot{\eta} + \mathcal{D}(\eta, \dot{\eta})\dot{\eta} + g(\eta) = \mathcal{T} + \mathcal{T}_W + \mathcal{T}_O \tag{15}$$

2.4. AUV system with unknown parameters

Let the system described in Eq. (15) have uncertain parameters and be expressed in linear parametric form. Thus, the system dynamics becomes

$$Y(\eta, \dot{\eta}, \ddot{\eta})\Phi = \mathcal{T} + \mathcal{T}_W \tag{16}$$

where $Y(\eta, \dot{\eta}, \ddot{\eta}) \in \mathcal{R}^{6\times p}$ is a known regressor matrix and $\Phi = [\phi_1, \ldots, \phi_p]^{\mathrm{T}}$ is a vector of p unknown parameters.

The regressor matrix $Y(\eta, \dot{\eta}, \ddot{\eta})$ is an appended matrix given by:

$$Y(\eta, \dot{\eta}, \ddot{\eta})\Phi = \begin{bmatrix} Y_{MC} & Y_D & Y_g & Y_O \end{bmatrix} \begin{bmatrix} \phi_{MC} \\ \phi_D \\ \phi_g \\ \phi_O \end{bmatrix} \tag{17}$$

where

$$Y_{MC} = \begin{bmatrix} \ddot{x} & -\dot{y}\dot{\psi} & \dot{z}\dot{\theta} & 0 & 0 & 0 \\ \dot{x}\dot{\psi} & \ddot{y} & -\dot{z}\dot{\varphi} & 0 & 0 & 0 \\ -\dot{x}\dot{\theta} & \dot{y}\dot{\varphi} & \ddot{z} & 0 & 0 & 0 \\ 0 & -\dot{y}\dot{z} & \dot{z}\dot{y} & \ddot{\varphi} & -\dot{\theta}\dot{\psi} & -\dot{\psi}\dot{\theta} \\ \dot{x}\dot{z} & 0 & -\dot{z}\dot{x} & \dot{\varphi}\dot{\psi} & \ddot{\theta} & -\dot{\psi}\dot{\varphi} \\ -\dot{x}\dot{y} & \dot{y}\dot{x} & 0 & -\dot{\varphi}\dot{\theta} & \dot{\theta}\dot{\varphi} & \ddot{\psi} \end{bmatrix} \tag{18}$$

$$Y_D = \begin{bmatrix} |\dot{x}|\dot{x} & 0 & 0 & 0 & 0 & 0 \\ 0 & |\dot{y}|\dot{y} & 0 & 0 & 0 & 0 \\ 0 & 0 & |\dot{z}|\dot{z} & 0 & 0 & 0 \\ 0 & 0 & 0 & |\dot{\varphi}|\dot{\varphi} & 0 & 0 \\ 0 & 0 & 0 & 0 & |\dot{\theta}|\dot{\theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & |\dot{\psi}|\dot{\psi} \end{bmatrix} \tag{19}$$

$$Y_g = \begin{bmatrix} -s\theta & 0 & 0 & 0 \\ c\theta s\varphi & 0 & 0 & 0 \\ c\theta c\varphi & 0 & 0 & 0 \\ 0 & 0 & -c\theta c\varphi & c\theta s\varphi \\ 0 & c\theta c\varphi & 0 & s\theta \\ 0 & -c\theta s\varphi & -s\theta & 0 \end{bmatrix} \tag{20}$$

and

$$Y_O = \begin{bmatrix} J_1^T(\eta_2) & 0_{3\times 3} \\ 0_{3\times 3} & J_1^T(\eta_2) \end{bmatrix} \tag{21}$$

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

The unknown parameter matrix ϕ_{MC} consists of the mass parameters given by

$$\phi_{MC} = \begin{bmatrix} m_{11} & m_{22} & m_{33} & m_{44} & m_{55} & m_{66} \end{bmatrix}^{\mathrm{T}} \tag{22}$$

and ϕ_D consists of the damping parameters.

$$\phi_D = \begin{bmatrix} d_{11} & d_{22} & d_{33} & d_{44} & d_{55} & d_{66} \end{bmatrix}^{\mathrm{T}} \tag{23}$$

ϕ_g consists of the gravitational matrix parameters

$$\phi_g = \begin{bmatrix} -(W-B) \\ x_G W - x_B B \\ y_G W - y_B B \\ z_G W - z_B B \end{bmatrix} \tag{24}$$

And lastly, ϕ_O is a constant matrix of ocean current terms.

$$\phi_O = \begin{bmatrix} \mathcal{T}_{o1} & \mathcal{T}_{o2} & \mathcal{T}_{o3} & \mathcal{T}_{o4} & \mathcal{T}_{o5} & \mathcal{T}_{o6} \end{bmatrix}^{\mathrm{T}} \tag{25}$$

3. Problem description

At the onset, some necessary terms are defined to elucidate the problem statement.

x_τ is the signal $x(t)$ truncated over a time interval $\tau \in \mathbb{R}$ defined as:

$$x_\tau = \begin{matrix} x(t), & t \in [0, \tau] \\ 0, & otherwise \end{matrix}$$

$\|x\|_\tau$ is the truncated Euclidean or L_2 norm of the signal $x(t)$ for time interval τ defined as:

$$\|x\|_\tau = \int_0^\tau |x(t)|^2 dt$$

$x_\tau \in \mathfrak{L}_{2e}$, if $\|x\|_\tau$ exists $\forall \tau < \infty$, where $\mathfrak{L}_{2e}$ is the extended $\mathfrak{L}_2$ norm space.

3.1 Assumption on measured data

A1: The system's input-output data, which include the control input torque ($\mathcal{T}$) as well as the position, velocity, and acceleration vectors ($\Upsilon = [\eta, \dot{\eta}, \ddot{\eta}]$), are not only theoretical but also directly measurable using standard sensors and instrumentation available in many practical systems.

Note: The control input torque ($\mathcal{T}$) can be measured through torque sensors or current sensors in electric motors, which directly relate to the torque being applied by the control system. Position, velocity, and acceleration vectors ($\Upsilon = [\eta, \dot{\eta}, \ddot{\eta}]$) can be obtained through position sensors (e.g., encoders, GPS), velocity sensors, and accelerometers, which are widely used in robotic systems like AUVs, automotive applications, and aerospace systems.

3.2 Problem statement:

The AUV system described by Eq. (5) is simulated to generate output data $\Upsilon_T = \Upsilon(t)_{t=0}^{\infty}$ from the input torque $\mathcal{T}_T = \mathcal{T}(t)_{t=0}^{\infty}$. This data set $D_T = \{\mathcal{T}_T, \Upsilon_T\}$ is truncated over the time interval $t \in [0, \tau]$ to form the resulting set $D_\tau = \{\mathcal{T}, \Upsilon\}$ such that $D_\tau \in \mathfrak{L}_{2e}$. This data set is employed to lay out an adaptive controller assembly.

Given that the system (5) contains uncertain parameters, it is reformulated in a linear parametric form as shown in Eq. (16). The objective is to derive an approximation for the unknown parameter vector Φ in Eq. (16). Using this parameter estimate, design a controller bank $\mathcal{C}_B = [\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n]$ consisting of a fixed number of candidate controllers, which can switch based on performance requirements. This controller assembly will guide the AUV along a desired trajectory while accounting for waves and ocean currents. Additionally, to minimise the tracking errors and control torque, update the candidates in the controller bank $\mathcal{C}_B$ using a real-time optimisation technique, to achieve a globally optimal solution.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

4. Controller structure design

4.1. Unfalsified control

UC is a data-driven supervisory approach that combines the strengths of both robust and adaptive control techniques. AUVs, as nonlinear systems, are significantly influenced by parametric uncertainties and environmental disturbances. Therefore, employing an adaptive control approach that effectively manages disturbances and nonlinearities is advantageous and is independent of specific models. In this iterative procedure, an algorithm is devised using the system's measured data to find a controller, from a set of a finite number of candidate controllers, that can adapt to the changing environment and fulfil the desired performance criteria. The selected controller that satisfies these criteria is termed "unfalsified." Conversely, if the available data indicate that a controller would fail to meet performance specifications in a feedback loop, it is considered falsified. The UC structure requires two important constituents – a fictitious reference generator and a cost function involving the fictitious reference. These are defined as follows.

4.1.1. Fictitious reference

A fictitious reference is a virtual signal that would generate specific input-output values for a controller ($\mathcal{C}$), assuming it were part of a closed-loop system. Given the input-output data ($\mathcal{T}, \Upsilon$), the fictitious reference $\hat{\eta}_{des}$ can be determined using the equation:

$$\hat{\eta}_{des} = \mathcal{C}^{-1}\mathcal{T} + \Upsilon.$$

4.1.2. Cost function

In UC, the cost function typically depends on the system's input, output, and fictitious reference, and governs the system's performance and stability. The cost detectability property ensures the stability of the unfalsified controller [44].

Let,

$$\mathfrak{J}^*(\mathcal{T}, \eta, \hat{\eta}_{des}, t) = -\iota_1 + \int_0^t \left\{ \frac{W_1 \|(\hat{\eta}_{des}(\tau) - \eta(\tau))\|^2 + W_2 \|\mathcal{T}(\tau)\|^2}{\iota_2 + W_3 \|\hat{\eta}_{des}(\tau)\|^2} \right\} \mathrm{d}\tau \tag{26}$$

where $\hat{\eta}_{des}(\tau)$, $\eta(\tau)$, and $\mathcal{T}(\tau)$ are, respectively, the fictitious reference, output, and input of the system truncated over a time period τ. ι_1 is the initial value, ι_2 is a small positive number, and W_1, W_2, and W_3 are weighting coefficients.

The supremum of $\mathfrak{J}^*$ over $t \leq \tau$ is given by

$$\sup_{t \leq \tau} \mathfrak{J}^*(\mathcal{T}, \eta, \hat{\eta}_{des}, t) \tag{27}$$

The cost function $\mathfrak{J}$ is the maximum value of $\mathfrak{J}^*$ up to a certain time τ. Therefore,

$$\mathfrak{J}(\mathcal{T}, \eta, \hat{\eta}_{des}, \tau) = \sup_{t \leq \tau} \mathfrak{J}^*(\mathcal{T}, \eta, \hat{\eta}_{des}, t) \tag{28}$$

Hence, $\mathfrak{J}(\mathcal{T}, \eta, \hat{\eta}_{des}, \tau)$ represents the maximum value of the cost function $\mathfrak{J}^*$ evaluated upto any time t within the interval $[0, \tau]$. For mathematical brevity, $\mathfrak{J}(\mathcal{T}, \eta, \hat{\eta}_{des}, \tau)$ shall be represented as $\mathfrak{J}$ in subsequent discussions.

Figure 2 illustrates the block diagram of the controller design. The closed loop structure includes the AUV system, whose parameters are unknown, and a controller bank $\mathcal{C}_B = [\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n]$, consisting of n nonlinear controllers, each designed with parameter values randomly selected from a finite, bounded set. Both the controller input and the system output are available for measurement. The UC method iteratively uses this measured data to identify the most appropriate controller from the bank. An algorithm is employed to switch out controllers that fail to meet the desired criteria and introduce one that does into the closed loop. Fictitious reference signals are generated to mathematically evaluate the candidate controllers without requiring them to be in the closed loop. Ultimately, only the selected controller is integrated into the closed-loop system.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

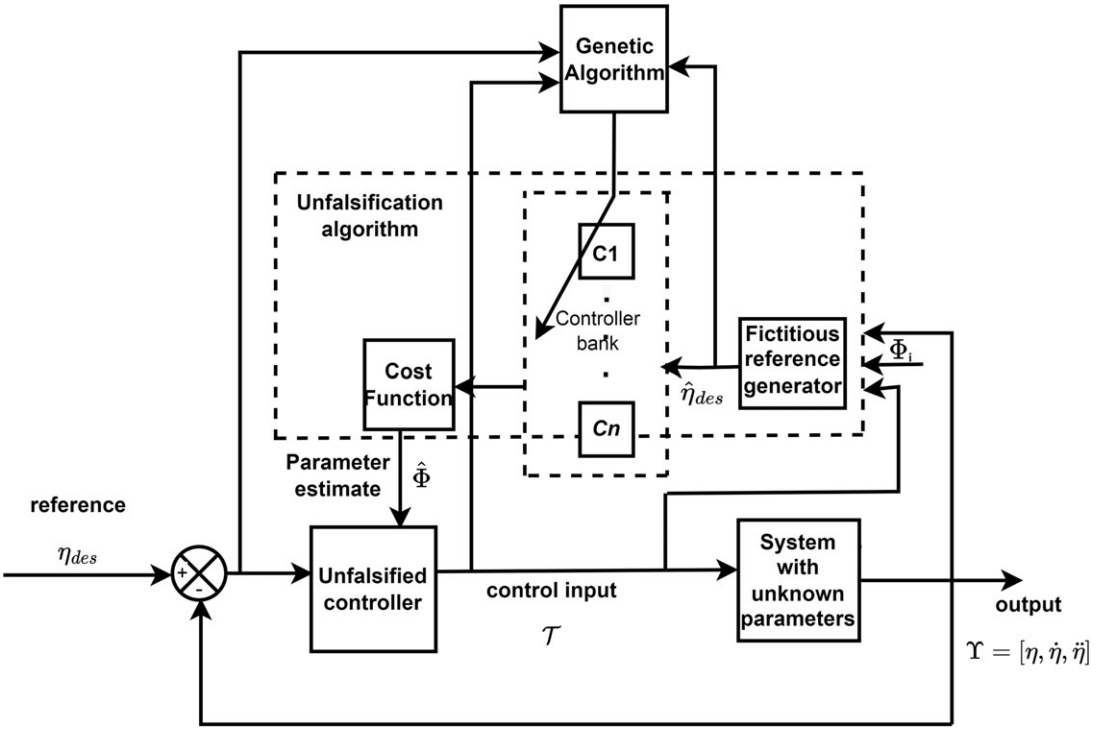


Figure 2. *Adaptive unfalsified control with GA applied to AUV system.*

4.2. Genetic algorithm-based adaptive unfalsified control

The literature on UC theory states that an unfalsified controller can be selected from a predetermined set of candidates or updated online using optimisation techniques. To enhance the capabilities of the UC method, this chapter introduces the concept of the GA into the technique of unfalsification to vary the candidate controller set. GA is a well-known evolutionary optimisation technique grounded in Darwinian principles of "Survival of the Fittest." It is employed to address a broad range of optimisation challenges across various engineering fields [45]. The GA method provides the system with access to a diversified set of candidate controllers and selects the unfalsified controller from an optimised set of values. This approach also considers the optimisation of energy resources in AUV applications. The uncertain parameters of the AUV system are estimated using the method of unfalsification, from which a controller is selected to manage the AUV. This controller enables the AUV to follow a desired trajectory despite disturbances such as waves and ocean currents. Furthermore, GA is applied to update the candidates in the controller bank, generating versatile and optimised options for the controller bank.

4.3. Genetic algorithm implementation

The steps to implement the GA are presented in a flow chart given in Figure A1 of Appendix A.

To initiate the GA, an initial starting population consisting of n potential solutions to the problem is generated using random numbers. A real number representing each solution is referred to as a chromosome. The fitness of the chromosomes is assessed, and a pair of chromosomes is selected as parents by the Roulette wheel selection. These parent chromosomes undergo crossover at randomly chosen positions, where their gene values are exchanged, resulting in the creation of two new chromosomes, referred to as offspring. This process continues until n offspring are produced. Subsequently, a mutation process introduces and maintains diversity within the population by randomly altering values within any chromosome. The operations of crossover, mutation, and fitness evaluation are iteratively performed, with

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

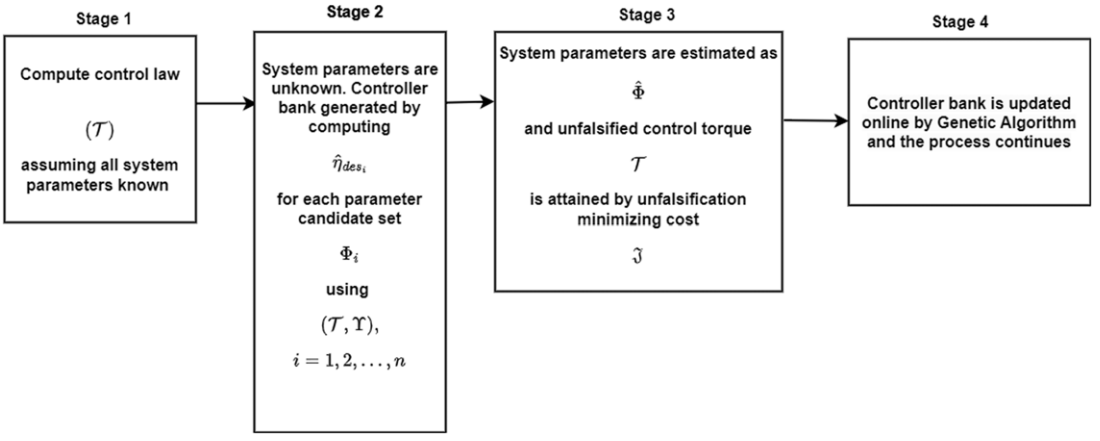


Figure 3. *Flow chart of the adaptive unfalsified control with GA scheme.*

offspring replacing their parent chromosomes after each cycle to form a new population. Due to the random nature of the initial population, each simulation run yields slightly different results, which benefits the proposed design where the controller bank candidates are updated using the GA procedure.

4.4. Adaptive unfalsified controller design with GA

The adaptive unfalsified controller designed for an unknown AUV system is combined with GA to update the controller bank in real time. The nonlinear controllers in the bank adapt to changing performance specifications by switching to the most appropriate candidate within the bank. Incorporating the GA enhances the versatility of the unfalsified controller, thereby improving its efficiency by optimising costs.

The controller design consists of four stages as shown in the flow chart presented in Figure 3.

Stage-1

The first stage involves designing the nonlinear computed torque controller (CTC), presuming the system parameters are known. To achieve this, the error, defined as the difference between the desired and actual positions, is calculated as

$$e_\eta = \eta_{des} - \eta \tag{29}$$

The CTC generates the control torque, given by

$$\mathcal{T} = \mathcal{M}[\ddot{e}_\eta + \Lambda_1 \dot{e}_\eta + \Lambda_2 e_\eta] + \mathcal{M}\ddot{\eta} + \mathcal{C}(\eta, \dot{\eta})\dot{\eta} + \mathcal{D}(\eta, \dot{\eta})\dot{\eta} + g(\eta) \tag{30}$$

Λ_1 and Λ_2 are positive gains that determine the rate at which the tracking error converges to zero. Substituting the value of $\mathcal{T}$ in Eq. (5), we get

$$\begin{aligned} \mathcal{M}\ddot{\eta} + \mathcal{C}(\eta, \dot{\eta})\dot{\eta} + \mathcal{D}(\eta, \dot{\eta})\dot{\eta} + g(\eta) &= \mathcal{M}[\ddot{e}_\eta + \Lambda_1 \dot{e}_\eta + \Lambda_2 e_\eta] \\ &+ \mathcal{M}\ddot{\eta} + \mathcal{C}(\eta, \dot{\eta})\dot{\eta} + \mathcal{D}(\eta, \dot{\eta})\dot{\eta} + g(\eta) + \mathcal{T}_E \end{aligned} \tag{31}$$

Using Eq. (14) and Eq. (16), the system dynamics in Eq. (31) can thus be written as

$$\begin{aligned} \mathcal{M}\ddot{\eta} + \mathcal{C}(\eta, \dot{\eta})\dot{\eta} + \mathcal{D}(\eta, \dot{\eta})\dot{\eta} + g(\eta) &= \mathcal{M}[\ddot{e}_\eta + \Lambda_1 \dot{e}_\eta + \Lambda_2 e_\eta] \\ &+ \mathcal{M}\ddot{\eta} + \mathcal{C}(\eta, \dot{\eta})\dot{\eta} + \mathcal{D}(\eta, \dot{\eta})\dot{\eta} + g(\eta) + \mathcal{T}_O + \mathcal{T}_W \end{aligned} \tag{32}$$

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

Stage-2

Using the linear parametric form of the 6 DoF model as given in Eq. (16), a controller bank is constituted by considering n random values for each of the unknown parameters.

$$Y(\eta, \dot{\eta}, \ddot{\eta})\Phi = \mathcal{M}[\ddot{e}_\eta + \Lambda_1 \dot{e}_\eta + \Lambda_2 e_\eta] + Y(\eta, \dot{\eta}, \ddot{\eta})\Phi_i + \mathcal{T}_W, \qquad i = 1, 2, \ldots, n \tag{33}$$

$$\mathcal{M}[\ddot{e}_\eta + \Lambda_1 \dot{e}_\eta + \Lambda_2 e_\eta] + Y(\eta, \dot{\eta}, \ddot{\eta})(\Phi_i - \Phi) + \mathcal{T}_W = 0 \tag{34}$$

Thus, Φ_i, $i = 1, 2, \ldots, n$ in the Eq. (34) constitutes the controller bank of n candidate controllers.

Stage-3

By the method of unfalsification, near-optimal values are estimated for the unknown parameters, and the CTC with the estimated parameters is inserted into the feedback loop. The CTC takes care of the trajectory tracking by AUV. By property (6), $\mathcal{M}^{-1}$ exists and

$$0 < \mathcal{M}^{-1} \leq \Delta_m \tag{35}$$

where Δ_m is a positive value.

4.5 Assumption on unknown mass matrix

A2: Let the unknown mass matrix be approximated as $\hat{\mathcal{M}}$ and $\hat{\mathcal{M}}^{-1}$ be bounded as

$$0 < \hat{\mathcal{M}}^{-1} \leq \delta_m \tag{36}$$

where δ_m is a positive value.

Note: This assumption is particularly relevant for ensuring the system's stability and feasibility in practical applications. In real-world systems, mass matrices often represent the inertia of mechanical components or robots, and it is crucial for M^{-1} to remain bounded to guarantee the system's physical viability. The condition (36) ensures that the system avoids scenarios where the mass matrix or its inverse could become unbounded, which might lead to numerical instability or ill-conditioning in control algorithms. This is important for maintaining smooth, stable operation in systems like AUVs and other robotic systems.

In many practical applications, the mass matrix is not always known exactly and must be approximated. The boundedness condition reflects real-world constraints, such as the need to limit variations in mass/inertia due to uncertainties or variations in the system components (e.g., wear, temperature changes, etc.). The upper bound represents a safe margin, ensuring that the inverse of the mass matrix stays within a practical range for control purposes. It also allows for the use of standard control techniques that assume a stable and well-behaved mass/inertia matrix, ensuring that the controller can handle real-world dynamics without significant performance degradation due to model inaccuracies or perturbations.

By assumption (36), the unfalsified controller is obtained with the parameter estimate as

$$\mathcal{T} = \hat{\mathcal{M}}[\ddot{e}_\eta + \Lambda_1 \dot{e}_\eta + \Lambda_2 e_\eta] + Y(\eta, \dot{\eta}, \ddot{\eta})\hat{\Phi} + \mathcal{T}_W \tag{37}$$

where $\hat{\Phi}$ is estimated from the controller bank by the unfalsification method. This estimate is obtained by minimising the cost function $\mathfrak{J}$ defined in 4.1.2. Eq. (37) defines the control law. Using Eq. (16) and Eq. (37), the system dynamics thus become

$$Y(\eta, \dot{\eta}, \ddot{\eta})\Phi = \hat{\mathcal{M}}[\ddot{e}_\eta + \Lambda_1 \dot{e}_\eta + \Lambda_2 e_\eta] + Y(\eta, \dot{\eta}, \ddot{\eta})\hat{\Phi} + \mathcal{T}_W \tag{38}$$

which implies

$$\hat{\mathcal{M}}[\ddot{e}_\eta + \Lambda_1 \dot{e}_\eta + \Lambda_2 e_\eta] + Y(\eta, \dot{\eta}, \ddot{\eta})\hat{\Phi} - Y(\eta, \dot{\eta}, \ddot{\eta})\Phi + \mathcal{T}_W = 0 \tag{39}$$

$$\hat{\mathcal{M}}[\ddot{e}_\eta + \Lambda_1 \dot{e}_\eta + \Lambda_2 e_\eta] + Y(\eta, \dot{\eta}, \ddot{\eta})e_\Phi + \mathcal{T}_W = 0 \tag{40}$$

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

where $e_\Phi = \hat{\Phi} - \Phi$ is the estimation error. Further modification gives

$$[\ddot{e}_\eta + \Lambda_1 \dot{e}_\eta + \Lambda_2 e_\eta] + \hat{\mathcal{M}}^{-1} Y(\eta, \dot{\eta}, \ddot{\eta}) e_\Phi + \hat{\mathcal{M}}^{-1} \mathcal{T}_W = 0 \tag{41}$$

$$\ddot{e}_\eta = -\Lambda_1 \dot{e}_\eta - \Lambda_2 e_\eta - \hat{\mathcal{M}}^{-1} Y(\eta, \dot{\eta}, \ddot{\eta}) e_\Phi - \mathcal{M}^{-1} \mathcal{T}_W \tag{42}$$

Eq. (42) in matrix form can be represented as

$$\begin{bmatrix} \dot{e}_\eta \\ \ddot{e}_\eta \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\Lambda_2 & -\Lambda_1 \end{bmatrix} \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix} + \begin{bmatrix} 0 \\ -\hat{\mathcal{M}}^{-1} Y(\eta, \dot{\eta}, \ddot{\eta}) e_\Phi - \hat{\mathcal{M}}^{-1} \mathcal{T}_W \end{bmatrix} \tag{43}$$

Eq. (43) constitutes the error dynamics of the system. Once the control law is designed according to (37), a fictitious reference is computed for each controller of the bank as

$$\begin{aligned} \ddot{\hat{\eta}}_{des} + \Lambda_1 \dot{\hat{\eta}}_{des} + \Lambda_2 \hat{\eta}_{des} = \hat{\mathcal{M}}^{-1} [\mathcal{T} + \hat{\mathcal{M}}(\Lambda_1 \dot{\eta} + \Lambda_2 \eta) \\ - \hat{\mathcal{C}}(\Phi, \dot{\eta}, \eta)\dot{\eta} - \hat{\mathcal{D}}(\Phi, \dot{\eta}, \eta)\dot{\eta} - \hat{g}(\Phi, \eta)] \end{aligned} \tag{44}$$

$\hat{\mathcal{M}}, \hat{\mathcal{C}}, \hat{\mathcal{D}}$, and $\hat{g}$ represent the estimates of mass matrix, Coriolis and centripetal matrix, damping matrix, and gravitational matrix, respectively. This fictitious reference is used to evaluate the cost function that determines the selection of controllers.

Note: The controller inverse, in this case, requires the inverse of the estimated mass matrix, which is invertible and bounded as given in Eq. (36).

Stage-4: GA for candidate set variation

To begin with, the fitness function is considered to be the cost function $\mathfrak{J}$ defined in Section 4.1.2. Next, the search space is initialised by defining a maximum and a minimum value for each of the unknown parameters of the system. A chromosome, therefore, represents a set of real values for as many unknown parameters available. Let the set of minimum values for the unknown parameters be defined as $\Phi_{min} = [\phi_{1min}, \phi_{2min}, \ldots, \phi_{pmin}]$ and the set of maximum values for the unknown parameters be defined as $\Phi_{max} = [\phi_{1max}, \phi_{2max}, \ldots, \phi_{pmax}]$, p being the number of unknown parameters. The population size and total number of iterations are predetermined. Random values within the specified range for each of the unknown parameters in each chromosome are generated to create the initial population. This population forms the starting set of candidate controllers.

Let $\varpi_1 = \{\varpi_{11}, \varpi_{21}, \ldots, \varpi_{p1}\}$ and $\varpi_2 = \{\varpi_{12}, \varpi_{22}, \ldots, \varpi_{p2}\}$ be 2 parent chromosomes selected by Roulette Wheel selection from the initial population. By uniform crossover, a new set of offspring $\varepsilon_1 = \{\varepsilon_{11}, \varepsilon_{21}, \ldots, \varepsilon_{p1}\}$ and $\varepsilon_2 = \{\varepsilon_{12}, \varepsilon_{22}, \ldots, \varepsilon_{p2}\}$ is generated given by

$$\varepsilon_{i1} = \varrho_i \varpi_{i1} + (1 - \varrho_i)\varpi_{i2} \tag{45}$$

$$\varepsilon_{i2} = \varrho_i \varpi_{i2} + (1 - \varrho_i)\varpi_{i1} \tag{46}$$

$\varrho_i \in [-\varsigma, 1 + \varsigma]$ where ς is a small positive number (say 0.1). Next, the mutation is applied to change any of the chromosome sets, say ϖ_j selected at random to ϖ_j'. Thus,

$$\varpi_j' = \varpi_j + \beta \tag{47}$$

where β is a random number having normally distributed probability with a mean of μ and variance of σ. The new set of values forms a new set of candidates for the controller bank. Thus, GA is employed to generate a new bank of candidate controllers, from which the controller that minimises the required cost function is selected.

5. Stability analysis and proof

5.1. Stability theorem

Theorem 5.1. *Given the plant in Eq. (5) and a closed-loop system in Eq. (43) with a bank of controllers* $\mathcal{C}_B = [\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n]$ *that satisfies the feasibility condition of having at least one stabilising controller*

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

as in Eq. (30), the system is said to be stable in closed loop if the unfalsified controller minimises the cost function $\mathfrak{J}$ and if the function $\mathfrak{J}$ satisfies the cost detectability property.

For stability analysis of the adaptive unfalsified controller, the following lemmas with proof are stated:

5.2. Lemma 1

Feasibility: The control problem is considered feasible if at any time for a system $\mathbb{S}_\zeta$ there exists at least one stabilising controller in the bank of candidate controllers $\mathcal{C}_B$.

Proof: Let the system (5) lie in the space $\mathbb{S}_\zeta = \{\zeta \in \mathbb{R}: \|\zeta\| \leq \frac{2\Gamma\|P\|}{\kappa_{min}(Q)}\}$, where $\zeta = \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix}$ represents the error matrix and $\dot{\zeta} = \begin{bmatrix} \dot{e}_\eta \\ \ddot{e}_\eta \end{bmatrix}$.

The error dynamics (43) can be written as

$$\dot{\zeta} = A_C\zeta + \chi \tag{48}$$

where $A_C = \begin{bmatrix} 0 & 1 \\ -\Lambda_2 & -\Lambda_1 \end{bmatrix}$, and $\chi = \begin{bmatrix} 0 \\ -\hat{\mathrm{M}}^{-1}Ye_\Phi - \hat{\mathrm{M}}^{-1}\mathcal{T}_W \end{bmatrix}$.

A quadratic positive Lyapunov function candidate is considered as:

$$\mathrm{V}(\zeta) = \frac{1}{2}\zeta^T P\zeta \tag{49}$$

where P is a symmetric positive definite matrix.

$$\kappa_{min}(P)\|\zeta\|^2 \leq \mathrm{V}(\zeta) \leq \kappa_{max}(P)\|\zeta\|^2 \tag{50}$$

Then

$$\begin{aligned} \dot{\mathrm{V}}(\zeta) &= \frac{1}{2}\zeta^T P\dot{\zeta} + \dot{\zeta}^T P\zeta \\ &= \zeta^T P(A_C\zeta + \chi) + (A_C\zeta + \chi)^T P\zeta \\ &= \zeta^T PA_C\zeta + \zeta^T P\chi + \zeta^T A_C^T P\zeta + \chi^T P\zeta \\ &= \zeta^T(PA_C + A_C^T P)\zeta + 2\chi^T P\zeta \end{aligned} \tag{51}$$

By design, we can ensure that

$$PA_C + A_C^T P = -Q \tag{52}$$

Then Eq. (51) becomes

$$\begin{aligned} \dot{\mathrm{V}} &= -\zeta^T Q\zeta + 2\chi P\zeta \\ &\leq -\kappa_{min}(Q) + 2\|\chi\|\|P\|\|\zeta\| \end{aligned} \tag{53}$$

Now,

$$\begin{aligned} \|\chi\| &= \left\| \begin{bmatrix} 0 \\ -\hat{\mathrm{M}}^{-1}Ye_\Phi - \hat{\mathrm{M}}^{-1}\mathcal{T}_W \end{bmatrix} \right\| \\ &\leq \|\hat{\mathrm{M}}^{-1}\|(\|Ye_\Phi\| + \mathcal{T}_{W_{max}}) \\ &\leq \Gamma(\|Ye_\Phi\| + \mathcal{T}_{W_{max}}) \\ &\leq -\kappa_{min}(Q)\|\zeta\|^2 + 2\Gamma\|P\|\|\zeta\| \\ &\leq -\kappa_{min}(Q)\|\zeta\|(\|\zeta\| - \frac{2\Gamma\|P\|}{\kappa_{min}(Q)}) \end{aligned} \tag{54}$$

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

Hence,

$$\|\zeta\| \leq \frac{2\Gamma\|P\|}{\kappa_{min}(Q)} \tag{55}$$

Next, as stated in the stability Theorem 5.1, the proof of stability requires the knowledge of cost detectability defined as follows:

Therefore, according to Lyapunov stability theory, the system is ultimately bounded within the bounds specified by Eq. (55). This confirms the feasibility (by Definition 5.2) condition of Theorem 5.1.

Next, as stated in the stability Theorem 5.1, the proof of stability requires the knowledge of cost detectability, defined as follows:

5.3. Lemma 2

Cost Detectability: A cost function $\mathfrak{J}$ is considered cost detectable if it fulfils the conditions given as follows [26]:

1. The cost function $\mathfrak{J}_i$ increases monotonically over time t for all $(\mathcal{C}_i, \mathcal{T}, \Upsilon, \hat{\eta}_{des_i})$, $i = 1, 2, 3 \ldots$.
2. $\lim_{t\to\infty} \mathfrak{J}_i$ remains uniformly bounded for all $(\mathcal{C}_i, \mathcal{T}, \Upsilon, \hat{\eta}_{des})$, $i = 1, 2, 3 \ldots$ when $\mathcal{C}_i$ is stabilising.

Proof: The following cost function is considered for minimisation:

$$\mathfrak{J} = \frac{\|\hat{e}_\eta\|^2 + \alpha\|\mathcal{T}\|^2}{\|\hat{\eta}_{des}\|^2 + \rho} \tag{56}$$

$\alpha, \rho > 0$ are positive constants. $\hat{\eta}_{des}$ is the fictitious reference signal and $\hat{e}_\eta = \hat{\eta}_{des} - \eta$ represents the fictitious error, which quantifies the difference between the actual output and the fictitious reference. The cost function $\mathfrak{J}$ features a denominator that includes the norm of the fictitious reference $\hat{\eta}_{des}$. This design choice normalises the numerator, making the cost function independent of variations in the fictitious reference across different controllers. Consequently, it eliminates the need to insert each candidate controller into the loop for input-output data collection, thereby avoiding potentially hazardous experimentation. By utilising fictitious references, data can be gathered without the risks associated with real-time controller deployment. To mitigate potential problems if the fictitious reference causes the denominator to approach zero, a constant ρ is introduced to the denominator.

According to the cost detectability proof, the cost function must be shown to increase over time but remain bounded for all time $t < \infty$. To prove that $\mathfrak{J}$ is monotonically increasing in time, Eq. (56) is differentiated w.r.t time t as follows:

$$\begin{aligned}
\dot{\mathfrak{J}} &= \frac{d}{dt}\left\{\frac{\|\hat{e}_\eta\|^2 + \alpha\|\mathcal{T}\|^2}{\|\hat{\eta}_{des}\|^2 + \rho}\right\} \\
&= \frac{d}{dt}\left\{\frac{\{(\int_0^\infty |\hat{e}_\eta|^2 dt)^{\frac{1}{2}}\}^2 + \alpha\{(\int_0^\infty |\mathcal{T}|^2 dt)^{\frac{1}{2}}\}^2}{\{(\int_0^\infty |\hat{\eta}_{des}|^2 dt)^{\frac{1}{2}}\}^2 + \rho}\right\} \\
&= \frac{d}{dt}\left\{\frac{(\int_0^\infty |\hat{e}_\eta|^2 dt) + \alpha(\int_0^\infty |\mathcal{T}|^2 dt)}{\int_0^\infty |\hat{\eta}_{des}|^2 dt + \rho}\right\} \\
&= \frac{\begin{array}{c}(\int_0^\infty |\hat{\eta}_{des}|^2 dt + \rho)\frac{d}{dt}\{(\int_0^\infty |\hat{e}_\eta|^2 dt \\ + \alpha\int_0^\infty |\mathcal{T}|^2 dt)\} - (\int_0^\infty |\hat{e}_\eta|^2 dt) \\ + \alpha(\int_0^\infty |\mathcal{T}|^2 dt)\frac{d}{dt}(\int_0^\infty |\hat{\eta}_{des}|^2 dt + \rho)\end{array}}{(\int_0^\infty |\hat{\eta}_{des}|^2 dt + \rho)^2}
\end{aligned} \tag{57}$$

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

$$
\begin{aligned}
&= \frac{\begin{array}{c}(\int_0^\infty |\hat{\eta}_{des}|^2 dt + \rho)(|\hat{e}_\eta|^2 + \alpha|\mathcal{T}|^2) \\ -(\int_0^\infty |\hat{e}_\eta|^2 dt + \alpha \int_0^\infty |\mathcal{T}|^2 dt)|\hat{\eta}_{des}|^2\end{array}}{(\int_0^\infty |\hat{\eta}_{des}|^2 dt + \rho)^2} \\
&= \frac{\begin{array}{c}(\|\hat{\eta}_{des}\|^2 + \rho)(|\hat{e}_\eta|^2 + \alpha|\mathcal{T}|^2) \\ -(\|\hat{e}_\eta\|^2 + \alpha\|\mathcal{T}\|^2)|\hat{\eta}_{des}|^2\end{array}}{(\|\hat{\eta}_{des}\|^2 + \rho)^2} \\
&= \frac{(\|\hat{\eta}_{des}\|^2 + \rho)(|\hat{e}_\eta|^2 + \alpha|\mathcal{T}|^2)}{(\|\hat{\eta}_{des}\|^2 + \rho)^2} - \frac{\|\hat{e}_\eta\|^2|\hat{\eta}_{des}|^2}{(\|\hat{\eta}_{des}\|^2 + \rho)^2} - \frac{\alpha\|\mathcal{T}\|^2|\hat{\eta}_{des}|^2}{(\|\hat{\eta}_{des}\|^2 + \rho)^2} \\
&= \frac{|\hat{e}_\eta|^2 + \alpha|\mathcal{T}|^2}{\|\hat{\eta}_{des}\|^2 + \rho} - \frac{\|\hat{e}_\eta\|^2|\hat{\eta}_{des}|^2}{(\|\hat{\eta}_{des}\|^2 + \rho)^2} - \frac{\alpha\|\mathcal{T}\|^2|\hat{\eta}_{des}|^2}{(\|\hat{\eta}_{des}\|^2 + \rho)^2}
\end{aligned} \tag{58}
$$

Since, $\frac{\|\hat{e}_\eta\|^2|\hat{\eta}_{des}|^2}{(\|\hat{\eta}_{des}\|^2+\rho)^2} \geq 0$ and $\frac{\alpha\|\mathcal{T}\|^2|\hat{\eta}_{des}|^2}{(\|\hat{\eta}_{des}\|^2+\rho)^2} \geq 0$.

Therefore,

$$
\dot{\mathfrak{J}} \leq \frac{|\hat{e}_\eta|^2 + \alpha|\mathcal{T}|^2}{\|\hat{\eta}_{des}\|^2 + \rho} \tag{59}
$$

Hence, $0 \leq \dot{\mathfrak{J}} \leq \frac{|\hat{e}_\eta|^2+\alpha|\mathcal{T}|^2}{\|\hat{\eta}_{des}\|^2+\rho}$ for all $(\mathcal{C}_i, \hat{\eta}_{des}, \mathcal{T}, \Upsilon)$, $i = 1, 2, 3 \ldots$.

This confirms the cost detectability of $\mathfrak{J}$, which by Theorem 5.1 fulfils the other condition of stability, besides feasibility. The candidate controllers aim to minimise the cost function, which depends on the system's fictitious error, as well as its input and output.

5.4. Lemma 3

Unfalsified Stability: The stability of the system $\mathbb{S}_\zeta$ is considered falsified by the truncated input-output pair $(\mathcal{T}, \Upsilon)$ if

$$
\sup_{\tau \in R_+, \|\mathcal{T}\| \neq 0} \frac{\|\Upsilon\|_\tau}{\|\mathcal{T}\|_\tau} \longrightarrow \infty \tag{60}
$$

If this condition is not met, then the stability of the system $\mathbb{S}_\zeta$ is regarded as unfalsified by the data $(\mathcal{T}, \Upsilon)$ [28].

At $t = \tau_f$, the controller with minimum $\mathfrak{J}$ remains unfalsified. Thus, $\mathcal{C} = \mathcal{C}_f$ with fictitious reference $\hat{\eta}_{des_f}$ results in

$$
\lim_{t \to \tau_f} \|\hat{\eta}_{des_f} - \eta_{des}\| = 0 \tag{61}
$$

This ensures the convergence of the error. Additionally, the finite set of candidate controllers generated with a specified range for the unknown parameters guarantees the boundedness of the estimates. Hence, by Eq. (61) and all the arguments stated above, it is concluded that the control system is stable.

The switching of controllers depending on the attainment of the minimum cost function is guaranteed to be finite by the following algorithm:

5.5. Lemma 4

Hysteresis Switching Algorithm: Let $\Gamma_k \in \mathbb{R}$ represent a continuous logic input signal, and δ denote the output switching signal, with both δ and k belonging to a normed vector space $\mathbb{K}$. At $t = t_0$, if $\delta(0) =$ be chosen as arg min $_{k\in\mathbb{K}}\{\Gamma_k(0)\}$. Then at some later time $t = t_m$, δ switches to a new value $l \in \mathbb{K}$. In such a case, δ will remain constant for all subsequent time $t_{m+1} > t_m$ provided that $(1 + n)$ min $\{\Gamma_k(t_{m+1})\} \leq \Gamma_l(t_m)$, where n is the hysteresis constant. This process results in a piece-wise constant signal for δ. If Γ_k

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

is uniformly bounded and $\Gamma_k \geq \omega$, where $\omega > 0$, $\forall k \in \mathbb{K}$, and all $t \geq 0$, then the switching remains finite [46].

Proof: Thus, the fact that the set of unfalsified controllers is monotonically decreasing and bounded below by an empty set ensures a finite number of switching amongst controllers. If there are n candidate controllers and there is at least one stable controller that satisfies the desired performance criteria, then by the Definition 5.5 stated above, the convergence of the switching of unfalsified controllers after at most $n-1$ switches can be assured.

6. Algorithm of unfalsification

The steps to be followed for implementing the procedure of unfalsification using the hysteresis algorithm are stated as follows:

(i) Given input and output data $D_\tau = \{\mathcal{T}, \Upsilon\}$ of the system with initial states.
(ii) *For* $i = 1, 2, \ldots, n$, formulate control law using Eq. (32) to constitute the candidates of the controller bank $\mathcal{C}_B = [\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n]$.
(iii) Compute the fictitious reference $\hat{\eta}_{des_i}$ as given in Eq. (44) for $\mathcal{C}_i$.
(iv) Calculate the fictitious error $e_{\hat{\eta}_i} = \hat{\eta}_{des_i} - \eta$ for $\mathcal{C}_i$.
(v) Evaluate the performance of $\mathcal{C}_i$ by minimising the cost function $\mathfrak{J}$ given in Eq. (56). Let, at the truncated time τ, the controller in closed loop be $\mathcal{C} = \mathcal{C}_{initial}$, resulting in a cost of $\mathfrak{J} = J_{initial}$
(vi) *If* for i^{th} iteration $\mathfrak{J}_i < J_{initial}$, then $\mathfrak{J} = J_i$ and $\mathcal{C} = \mathcal{C}_i$ *else* $\mathcal{C} = \mathcal{C}_{initial}$ End *if*
(vii) Increment i.
(viii) End *for*
(ix) Repeat (iv) to (vi) to find the arg min $\mathfrak{J}$.
(x) Terminate when minimum $\mathfrak{J}$ is achieved. $\mathcal{C}$ is the controller that remains unfalsified.

7. Result

The implementation of the proposed controller on the system described in Eq. (16) is evaluated through simulations carried out using MATLAB/SIMULINK. A circular trajectory is considered the reference. The desired position and orientation are defined as follows:

$$\begin{aligned} \eta_{des} &= [x_{des} \quad y_{des} \quad z_{des} \quad \varphi_{des} \quad \theta_{des} \quad \psi_{des}]^{\mathrm{T}} \\ &= \left[10sin(t) \quad 10cos(t) \quad 10 \quad 0 \quad 0 \quad \frac{\Pi}{3}\right]^{\mathrm{T}} \end{aligned} \tag{62}$$

The initial state values are considered

$$\eta(0) = [0.1, 0.1, 0.5, 0, 0, 0.1]^{\mathrm{T}}$$

and

$$\dot{\eta}(0) = [0, 0, 0, 0, 0, 0]^{\mathrm{T}}$$

with a total of $p = 22$ unknown parameters. The controller gains are set at $\Lambda_1 = 10$, $\Lambda_2 = 170$. The parameters used for the system are detailed in Table I.

A controller bank is created by considering an arbitrary set of 5 random values for each unknown parameter $[\phi_1, \phi_2, \ldots, \phi_{22}]$ within the minimum and maximum values specified in Table II. Each random value for each of the unknown parameter $[\phi_1, \phi_2, \ldots, \phi_{22}]$ in the control law (32) constitutes a candidate

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

Table I. *System parameters and their nominal values.*

System parameter	Nominal value	Unit
W	300	N
B	308	N
(x_G, y_G, z_G)	(0, 0, 0.002)	m
(x_B, y_B, z_B)	(−0.061, 0, 0)	m
$m_{11} = m - X_{\dot{u}}$	31.51	kg
$m_{22} = m - Y_{\dot{v}}$	66.08	kg
$m_{33} = m - Z_{\dot{w}}$	66.08	kg
$m_{44} = I_x - K_{\dot{p}}$	0.2474	kgm^2
$m_{55} = I_y - M_{\dot{q}}$	8.33	kgm^2
$m_{66} = I_z - N_{\dot{r}}$	8.33	kgm^2
$d_{11} = -X_{\|u\|u}$	1.62	kg/m
$d_{22} = -Y_{\|v\|v}$	131	kg/m
$d_{33} = -Z_{\|w\|w}$	131	kg/m
$d_{44} = -K_{\|p\|p}$	0.13	kgm^2/rad^2
$d_{55} = -M_{\|q\|q}$	188	kgm^2/rad^2
$d_{66} = -N_{\|r\|r}$	94	kgm^2/rad^2

Table II. *Minimum and maximum values of uncertain parameters.*

Unknown parameter	System parameter	Nominal value	Minimum value	Maximum value
ϕ_1	m_{11}	31.51	20	40
ϕ_2	m_{22}	66.08	56	76
ϕ_3	m_{33}	66.08	56	76
ϕ_4	m_{44}	0.2474	0	1
ϕ_5	m_{55}	8.33	1	20
ϕ_6	m_{66}	8.33	1	20
ϕ_7	d_{11}	1.62	0	3
ϕ_8	d_{22}	131	120	140
ϕ_9	d_{33}	131	120	140
ϕ_{10}	d_{44}	0.13	0	1.5
ϕ_{11}	d_{55}	188	170	200
ϕ_{12}	d_{66}	94	80	110
ϕ_{13}	$-(W - B)$	8	0	20
ϕ_{14}	$x_G W - x_B B$	18.81	1	30
ϕ_{15}	$y_G W - y_B B$	0	0	1
ϕ_{16}	$z_G W - z_B B$	0.58	0	1
ϕ_{17}	$\mathcal{T}_{o1}$	−10	−20	0
ϕ_{18}	$\mathcal{T}_{o2}$	−20	−30	−10
ϕ_{19}	$\mathcal{T}_{o3}$	8	0	20
ϕ_{20}	$\mathcal{T}_{o4}$	5	0	20
ϕ_{21}	$\mathcal{T}_{o5}$	10	0	20
ϕ_{22}	$\mathcal{T}_{o6}$	20	10	30

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

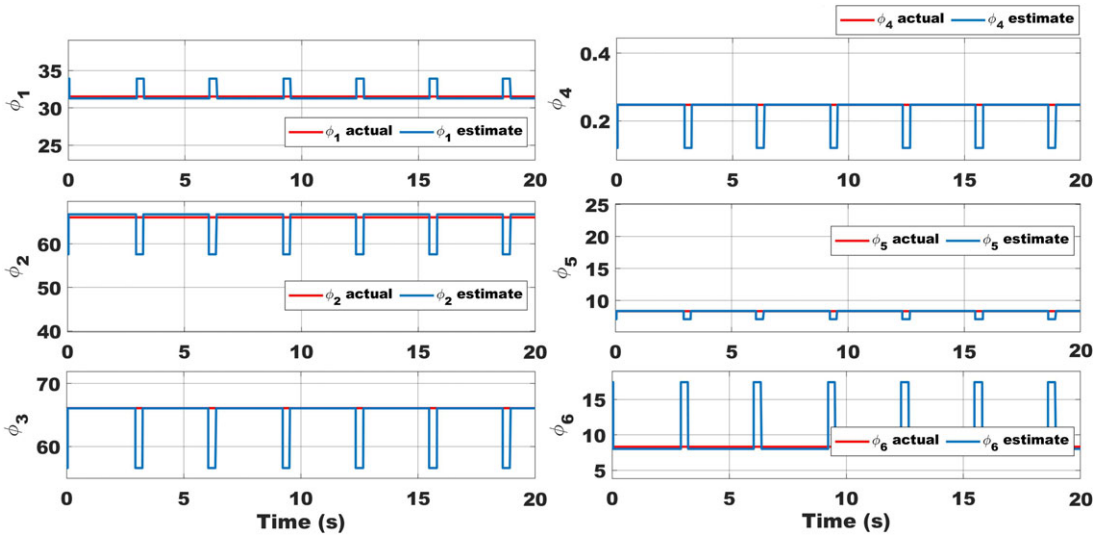


Figure 4. *Switching in mass parameters.*

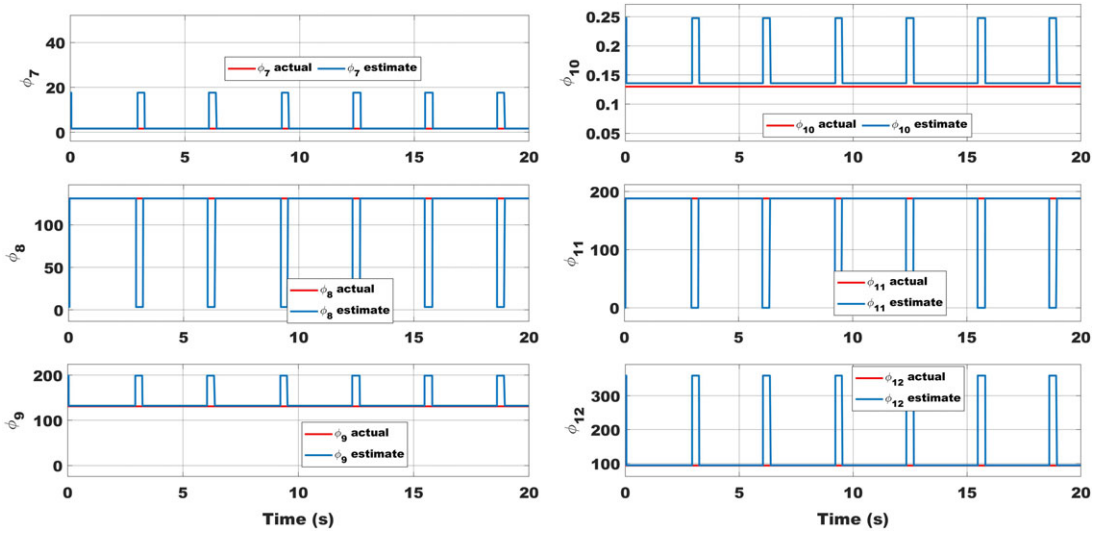


Figure 5. *Switching in damping parameters.*

in the controller bank. Thus there are 5 candidates namely $\mathcal{C}_B = [\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_5]$. The simulation is run with the initial guess of the unknown parameters.

Then the fictitious reference for each candidate controller $\hat{\eta}_{desi}$, $i = 1, 2, \ldots, 5$ is computed using the relation (44). The simulation continues to determine the unfalsified controller for the minimum cost function $\mathfrak{J}$ given in Eq. (56). The unfalsification technique results in the switching of the candidate values to obtain the estimates of unknown parameters that ensure minimum cost. This switching in parameters can be observed in Figures 4, 5, 6, and 7. As the algorithm runs for a complete iteration, the change in plant dynamics results in a change in parameter values, thus causing the switches.

Thereafter, GA is applied to create a new set of candidates. As the simulation runs, the initial candidate set is changed to a new set by the concept of crossover and mutation of GA and the controller that minimises the required cost function gets selected from this new set of candidates. The parameters used in GA according to Eqs. (45)–(47) are considered as follows: $\varrho = 1$, $\varsigma = 0.1$, $\beta = 0.02$, population size of 40 and maximum iterations of 20. Table III presents the parameter estimates obtained using

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

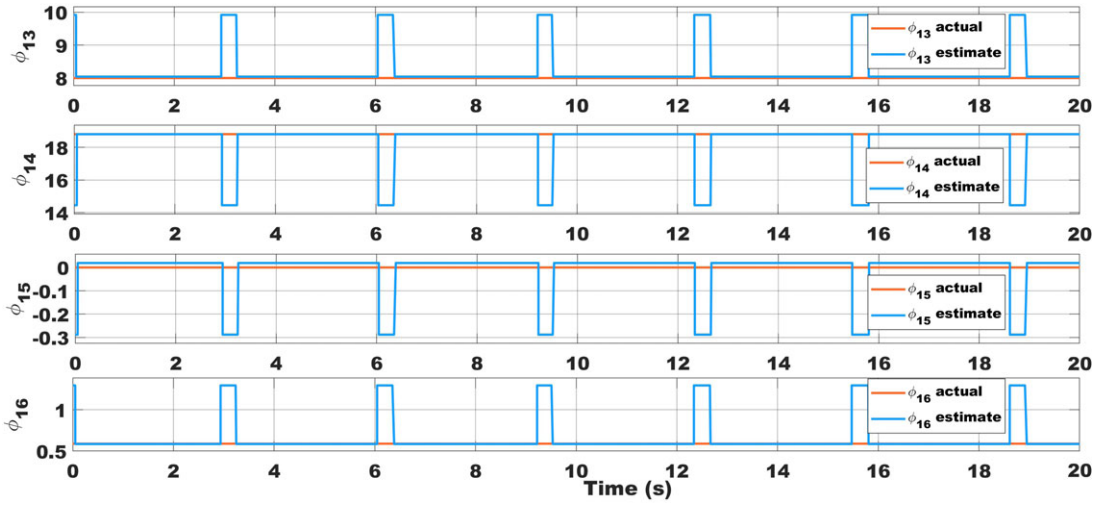


Figure 6. *Switching in gravitational matrix parameters.*

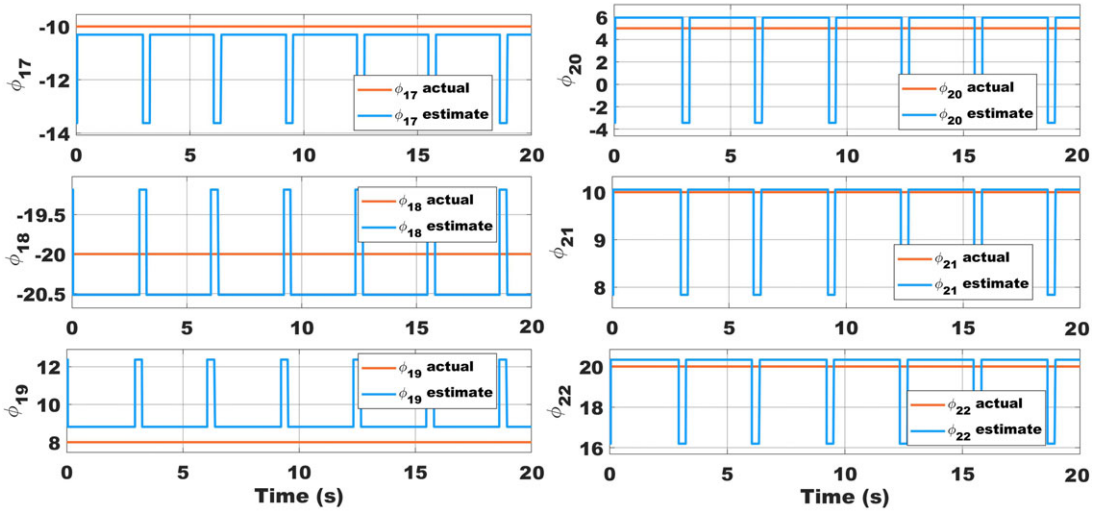


Figure 7. *Switching in ocean current disturbance parameters.*

the Adaptive Unfalsification method with GA. For comparison, estimates obtained solely through the GA technique are also included in the same table. Additionally, the absolute errors for the estimated parameters are calculated and provided in the table to facilitate a comparison of the results. The error results show that even if the range between the lower and upper bound is generously considered, the estimated values obtained by Adaptive Unfalsification in conjunction with GA remain well within the given bounds and show better convergence as compared to that of only GA-optimised values.

The control law computed using the unfalsified estimated values drives the AUV to the desired circular trajectory. A disturbance $\mathcal{T}_w = [0.1cos(10t), 0.1sin(10t), 0, 0, 0, 0]^{\mathrm{T}}$ has been added as wave disturbance. Yet by switching to the appropriate candidate controller, the AUV efficiently tracks the desired trajectory. The position and angular tracking by AUV are shown in Figures 8 and 9. Also, a comparison of the circular trajectory tracking by Adaptive Unfalsification with GA and by GA optimisation is shown in Figures 10, 11. The position error convergence in the two cases is also shown in Figures 12 and 13. As evident from the results, the tracking of reference trajectory is faster and more precise in the case of the Adaptive Unfalsification technique with GA compared to that by using only GA.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

Table III. *Comparison of the estimated values of unknown parameters obtained using both methods.*

Unknown parameter	Nominal value	Estimated value		Absolute error	
		Adaptive Unfalsification with GA	GA optimisation	Adaptive Unfalsification with GA	GA optimisation
ϕ_1	31.51	31.27	27.33	0.24	4.18
ϕ_2	66.08	66.76	69.83	0.68	3.07
ϕ_3	66.08	66.06	69.29	0.02	3.21
ϕ_4	0.247	0.24	0.39	0	0.143
ϕ_5	8.33	8.36	6.57	0.03	1.76
ϕ_6	8.33	8.02	8.93	0.31	0.6
ϕ_7	1.62	1.66	0.62	0.04	1
ϕ_8	131	131.1	128.9	0.1	2.1
ϕ_9	131	132.2	129.3	1.2	1.7
ϕ_{10}	0.13	0.13	0.77	0	0.64
ϕ_{11}	188	188.1	188.4	0.1	0.4
ϕ_{12}	94	94.68	100.3	0.68	6.3
ϕ_{13}	8	8.04	13	0.04	5
ϕ_{14}	18.81	18.81	17.46	0	1.35
ϕ_{15}	0	0.02	0.58	0.02	0.58
ϕ_{16}	0.58	0.58	0.42	0	0.16
ϕ_{17}	−10	−10.31	−4.59	0.31	5.41
ϕ_{18}	−20	−20.52	−23.05	0.52	3.05
ϕ_{19}	8	8.81	5.73	1.19	2.27
ϕ_{20}	5	5.95	8.69	0.95	3.69
ϕ_{21}	10	10.05	11.33	0.05	1.33
ϕ_{22}	20	20.33	18.55	0.33	1.44

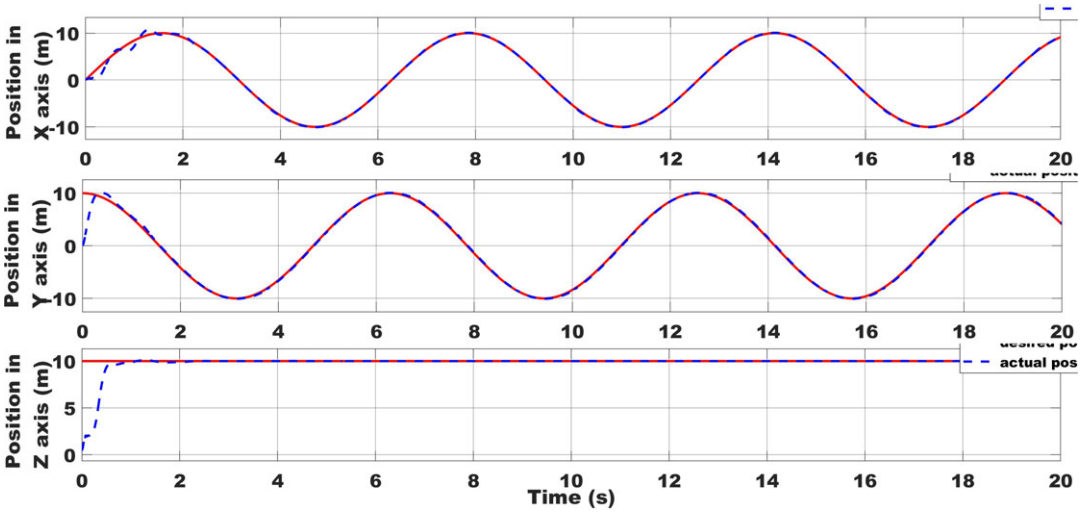


Figure 8. *Linear position tracking using Adaptive Unfalsification with GA.*

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

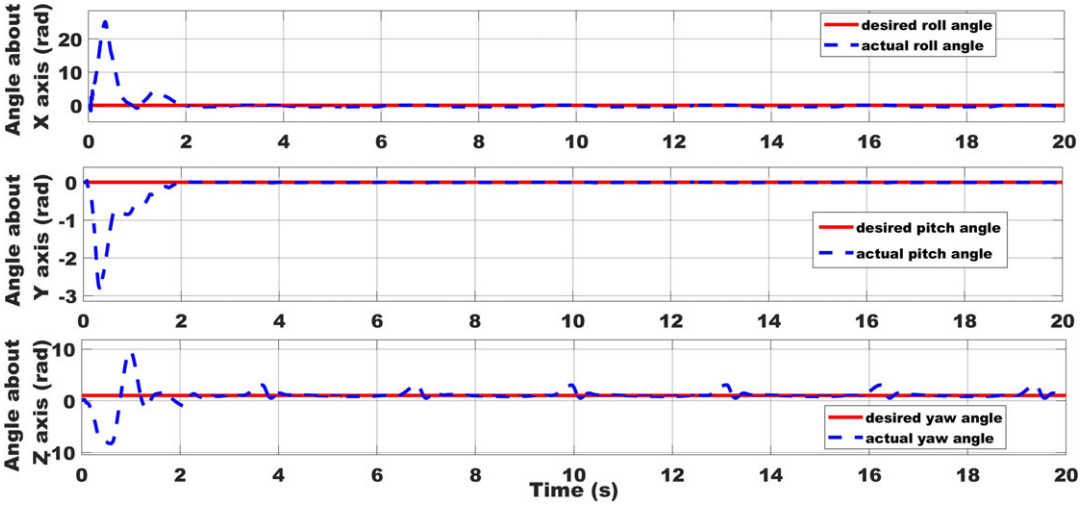


Figure 9. *Angular position tracking using Adaptive Unfalsification with GA.*

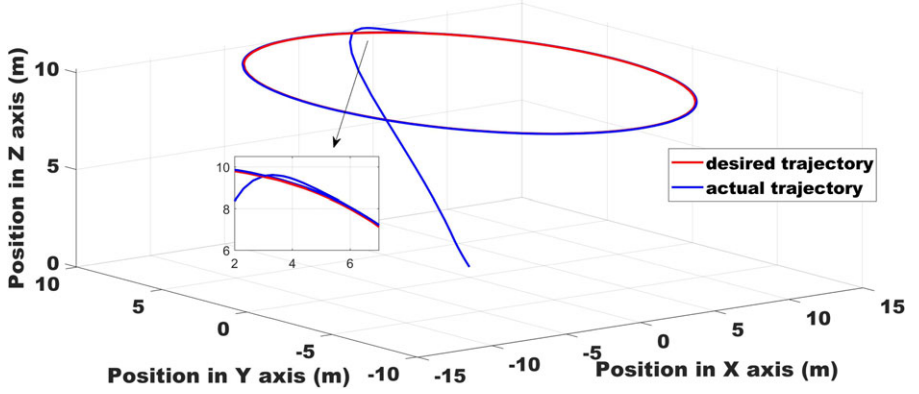


Figure 10. *Circular trajectory tracking using Adaptive Unfalsification with GA.*

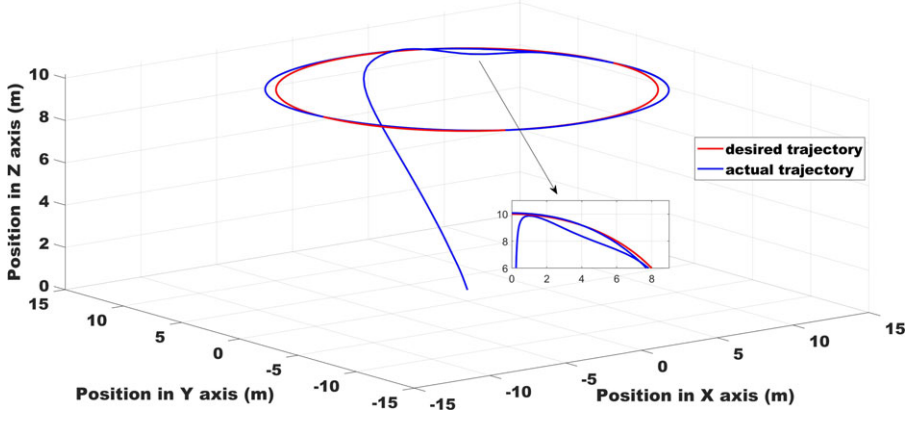


Figure 11. *Circular trajectory tracking using GA optimisation.*

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

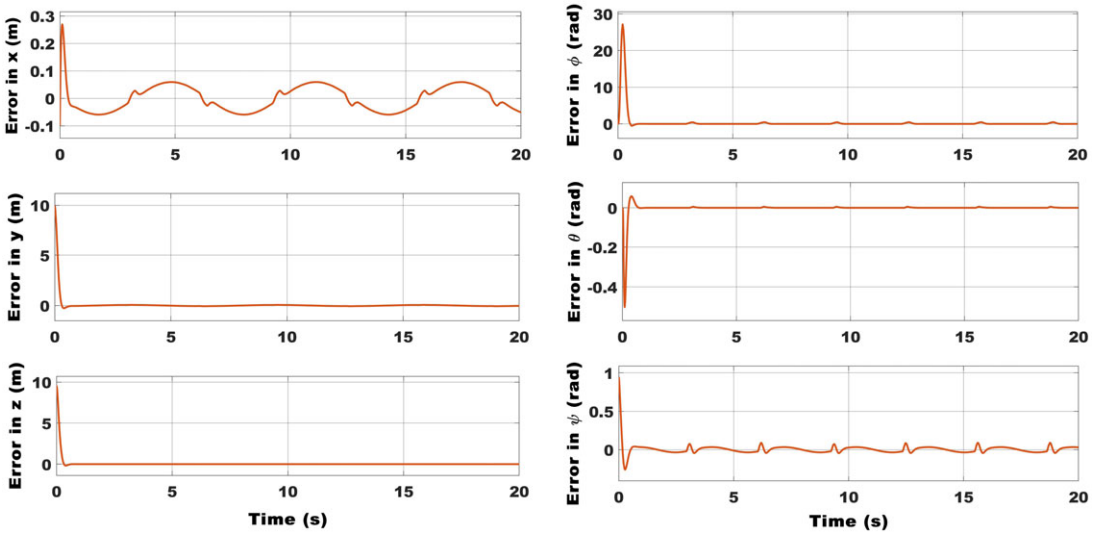


Figure 12. *Illustration of the error convergence in Adaptive Unfalsification with GA. The error between the desired and actual linear position in x, y, z direction and angular position ϕ,θ, ψ is shown in the figure. Error in the x-direction is oscillatory, but the amplitude of error is very less.*

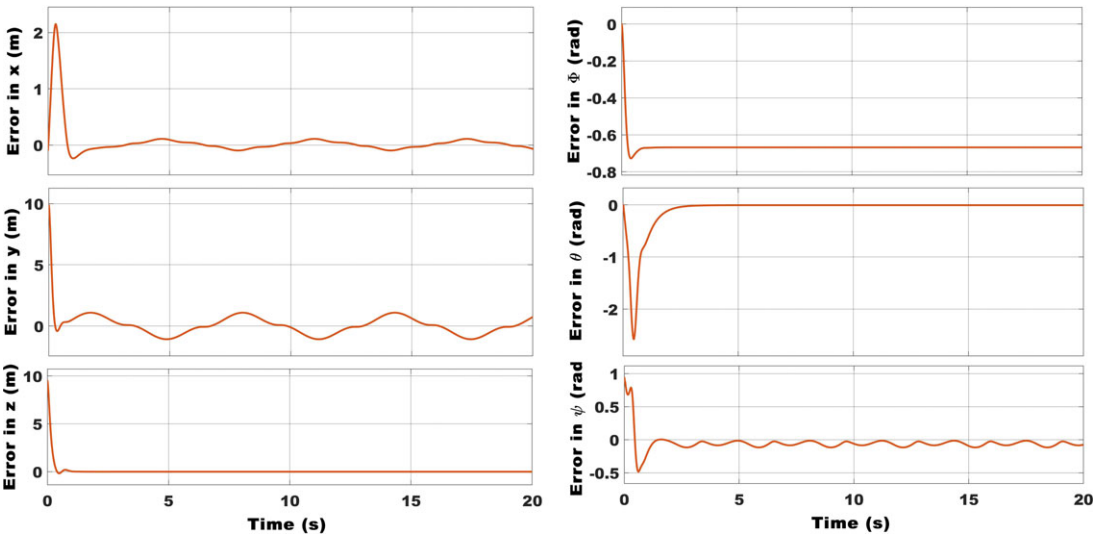


Figure 13. *Illustration of the error convergence in sole GA optimisation. The error between the desired and actual linear position in x,y,z direction and angular position ϕ,θ, ψ is shown in the figure. The error amplitude is greater in the case of linear position in the x direction than in the Adaptive Unfalsification with GA case.*

Moreover, the drawback of the best-fit solution being dependent on the appropriate population size and the total number of iterations in the GA optimisation technique is overcome in the proposed method, as GA is used here solely to assist the unfalsification process. It increases and improves the search space of solutions in terms of optimality and exploration of a global solution. In this, fewer iterations or a smaller population size do not impact the accuracy of estimation results. All of this is obtained with measured information of input-output only and without any knowledge of exact system parameters.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

Remark: The results demonstrate that introducing variability into the candidate set has significantly improved the tracking performance of the controller. This effectively justifies the proposed design to integrate the GA with Adaptive Unfalsification. Furthermore, it establishes the robustness of the proposed control design in relation to parameter variations.

8. Discussion on GA implementation

Developing hardware for an AUV is a highly expensive endeavour and falls beyond the scope of this study. Nonetheless, this section explores the requirements for implementing a controller in AUV hardware and discusses the feasibility of the proposed controller design in terms of execution time by referencing established works in the field.

To implement the autonomous missions of AUV, single board computers (SBCs) are required, which can handle sensor inputs and other aspects of the system. An SBC includes a processor, i/o ports, memory, and additional features similar to those of a usual computer. The selection of a processor is determined by factors such as processing speed requirements, external ports, etc., which are essential for different functions and the successful completion of crucial tasks within specified timeframes.

The processors in SBCs can be built with different internal processing structures, such as the central processing units (CPUs) of Intel architecture, multi-core processing architectures, or graphical processing architectures. The primary distinction between graphical processing units (GPUs) and CPUs lies in their respective capabilities: CPUs excel at swiftly executing sequential tasks, while GPUs employ parallel processing to perform multiple operations with enhanced speed and efficiency concurrently.

Our focus is directed towards controlling via GPUs, given their adeptness at handling specialised computations and executing parallel operations. GPUs prove proficient in accelerating specialised tasks like optimisation, machine learning, data analytics, and other artificial intelligence (AI) applications [47]. GPUs have emerged as a feasible alternative for hardware implementation with minimal investment.

8.1. Implementation of heuristic algorithms using GPUs

The GPU-based architecture can be used for computationally intensive optimisation techniques. This can greatly improve the convergence of the optimisation process and substantially decrease the total execution time. Additionally, GPU computing offers the advantage of cost-effectiveness.

In ref. [48], various optimisation algorithms are introduced and tested across benchmark problems of varying difficulty levels. In the case of the mixed integer nonlinear programming (MINLP) problem, which incorporates both continuous and discrete variables, using the NVIDIA Tesla C2050 GPU results in an overall speed improvement of between 20 and 42 times when compared to the Intel Core i7-920 CPU.

Implementing a GA on a GPU results in approximately 6 times faster execution than implementing on a CPU. Employing GA on a multi-GPU platform achieves a speedup ranging from 8 to 771 times that of the CPU.

The adoption of GA utilising the Compute Unified Device Architecture (CUDA) architecture leads to performance enhancements with a speedup ranging from 40 to 400 times.

Other optimisation methods like CFO and PSO also require significantly less time. Implementing these on an NVIDIA GTX280 can accelerate the computing time by 17 to 41 times.

The GPU proves notably more effective and efficient for evolutionary algorithms, yielding results 20 to 35 times faster than the CPU.

8.1.1. Implementation of genetic algorithm using GPU

GAs stand out as highly adaptable optimisation techniques. Their intrinsic parallelism makes them ideal for implementation on parallel hardware like GPUs.

At the beginning, it was inferred that GA was partially compatible with GPU [49]. The conclusion was made as it was observed that the crossover operator could operate on an entire chromosome rather than an individual gene. This could not be implemented on the preliminary architectures of GPU.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

Table IV. *Comparison of GPU and CPU performance [57].*

Hardware	Time (s)
AMD (6 cores)	5.80
Intel (6 cores)	2.95
Tesla C1060	5.17
GeForce GTX 480	1.0

Table V. *Comparison of speedups in GPUs and CPUs [57].*

Platform	AMD (1)	Intel (1)	AMD (6)	Intel (6)	Tesla C1060
Intel (1)	1.5				
AMD (6)	5.9	3.9			
Intel (6)	9.1	5.9	1.5		
Tesla C1060	637.9	420.7	106.7	70.0	
GF GTX 480	1918.7	1265.6	321.1	210.4	3.0

Later, the idea was revisited by applying GA using a master-worker scheme on GPU [50]. With this approach, a remarkable speedup of around 100 times was achieved.

Other than this scheme, various other schemes were later developed to be realised on GPU as presented in the literature [51–53]. Some of the existing research that focused on implementing evolutionary algorithms (EAs) on older GPUs can be seen in refs. [54–56], most of which have achieved speedups exceeding 1000 compared to standard CPUs. In ref. [57], comprehensive experiments were conducted to evaluate the implementation of GA on GPUs, focusing on NVIDIA's Fermi generation graphics chips. The paper presents experimental results comparing multi-core CPU and GPU performance in terms of the time required for generating 10^9 random single-precision floating-point numbers (see Table IV). A table comparing the speedup in different hardware is also presented Table V for an optimisation problem with a population size of 10,000 and 70 iterations. The figure in parentheses signifies the number of cores used.

8.1.2. Example of GA implementation in AUV

The research discussed in ref. [58] employs GA on a GPU platform for fulfilling a path-following objective in an AUV. In this, a GeForce GTX 1080 Ti GPU and an Intel(R) Core(TM) i7-5930 K CPU 3.50 GHz were employed. The CPU was utilised for managing programme flow, initialisation, and obstacle boundary expansion, while the optimisation by GA was performed on the GPU.

It was observed that the fitness values obtained were nearly identical in the case of both CPU and GPU. However, GPU optimisation could achieve an average speedup of 83 times compared to the CPU, resulting in an average runtime of 12.509 s.

8.2. Inference about the proposed design

The elapsed time for a 20-s simulation run of the proposed controller design on an Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz to track a circular reference trajectory was 61.5 s. For the spiral reference trajectory, the simulation run time was 300 s, and the elapsed time was 425.64 s.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

It is notable that the CPU works much slower compared to a GPU. Thus, based on the referenced literature, it can be inferred that implementing the design approach presented in this chapter in AUV hardware is practically feasible and not a concern, particularly with the latest SBCs equipped with GPUs. Noteworthy GPUs in this context include the NVIDIA Jetson AGX, NVIDIA Jetson Orin, and NVIDIA Jetson Nano, among others.

9. Conclusion

This paper applies the concept of GAs, which are widely used in optimisation, to the model-independent, data-driven adaptive control method of unfalsification. Real-time system data are employed to design a candidate controller bank aimed at achieving the control objective. For each candidate controller, a fictitious reference signal is computed to assess whether it meets the desired performance criteria. The controller that best satisfies these criteria is integrated into the control loop. Should the current controller fail, it is replaced by another from the candidate set. GA is applied to dynamically modify the candidate set online, enabling real-time optimisation and enhancing the capabilities of the unfalsification method.

The proposed design demonstrates the ability to obtain optimal parameter estimates of the unknown AUV system with minimal information. The AUV system considered incorporates the Coriolis and centripetal matrix terms, gravitational matrix terms, as well as mass and damping terms. Additionally, the system dynamics account for wave disturbances and ocean currents. The designed controller successfully guides the AUV to the desired reference trajectory despite various disturbances. Simulation results show that the integration of GA into the unfalsification method achieves faster convergence compared to using GA alone, which depends on the number of iterations and population size for accurate estimation. The parameter estimates obtained through unfalsification remained well within the predefined bounds, ensuring the boundedness of estimation error. The computational efficiency was also analysed and found to be suitable for practical implementation with currently available processors such as GPUs.

However, this proposed research work has some shortcomings:

1. The real-world uncertainties are more complex than periodic waves, and the effect of those disturbances remains to be investigated.
2. Furthermore, actuator uncertainties like hysteresis, lag, and saturation related to actual thrusters of an AUV can lead to delay in controller realisation, which in return can hamper the estimator's performance. The controller with actuator saturation estimation can solve this problem to some extent, and future work can be done in this aspect.

Author contribution. Tabassum Rasul and Koena Mukherjee conceived the design. Tabassum Rasul performed the analysis and proof, conducted the required simulation work, and wrote the article. Koena Mukherjee supervised the study and manuscript preparation.

Financial support. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] M. Jun and M. G. Safonov, "Automatic Pid Tuning: An Application of Unfalsified Control," **In:** *Proceedings of the International Symposium on Computer Aided Control System Design* (1999).

[2] S. B. Williams, P. Newman, J. Rosenblatt, G. Dissanayake and H. Durrant-Whyte, "Autonomous underwater navigation and control," *Robotica* **19**(5), 481–496 (2001). doi: 10.1017/S0263574701003423.

[3] M. Hosseini, A. Ranjbar Noei and S. J. S. Rostami, "Trajectory tracking control of an underwater vehicle in the presence of disturbance, measurement errors, and actuator dynamic and nonlinearity," *Robotica* **41**(10), 3059–3078 (2023). doi: 10.1017/S0263574723000875.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

[4] K. M. Tan, A. Anvar and T. Lu, "Autonomous underwater vehicle (auv) dynamics modelling and performance evaluation," *Int. J. Mech. Mechatron. Eng.* **6**(10), (2012).

[5] A. Izadbakhsh and S. Khorashadizadeh, "Polynomial-based robust adaptive impedance control of electrically driven robots," *Robotica* **39**(7), 1181–1201 (2021). doi: 10.1017/S0263574720001009.

[6] B. K. Sahu and B. Subudhi, "Adaptive tracking control of an autonomous underwater vehicle," *Int. J. Autom. Comput.* **11**(3), 299–307 (2014).

[7] A. Izadbakhsh, S. Khorashadizadeh and P. Kheirkhahan, "Tracking control of electrically driven robots using a model-free observer," *Robotica* **37**(4), 729–755 (2019). doi: 10.1017/S0263574718001303.

[8] Z. H. Ismail and M. W. Dunnigan, "Nonlinear h_∞ optimal control scheme for an underwater vehicle with regional function formulation," *J. Appl. Math.* **2013**(1), 732738 (2013).

[9] K. Shojaei, "Three-dimensional tracking control of autonomous underwater vehicles with limited torque and without velocity sensors," *Robotica* **36**(3), 374–394 (2018). doi: 10.1017/S0263574717000455.

[10] C. Wu, Y. Dai, L. Shan, Z. Zhu and Z. Wu, "Data-driven trajectory tracking control for autonomous underwater vehicle based on iterative extended state observer," *Math. Biosci. Eng.: MBE* **19**(3), 3036–3055 (2022).

[11] S. P. Sahoo, B. Das, B. B. Pati, F. P. Garcia Marquez and I. Segovia Ramirez, "Hybrid path planning using a bionic-inspired optimization algorithm for autonomous underwater vehicles," *J. Mar. Sci. Eng.* **11**(4), 761 (2023).

[12] S. Liu, D. Wang and E. K. Poh, "A Nonlinear Observer for Auvs in Shallow Water Environment," **In:** *IEEE/RSJ International Conference On Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566)*, vol. 2 (IEEE, 2004) pp. 1130–1135.

[13] S. Liu, D. Wang and E. K. Poh, "Nonlinear Adaptive Observer Design for Tracking Control of Auvs in Wave Disturbance Condition," **In:** *OCEANS 2006-Asia Pacific* (IEEE) pp. 1–8.

[14] J. Li, H. Huang, L. Wan, Z. Zhou and Y. Xu, "Hybrid strategy-based coordinate controller for an underwater vehicle manipulator system using nonlinear disturbance observer," *Robotica* **37**(10), 1710–1731 (2019). doi: 10.1017/S0263574719000213.

[15] K. Mukherjee, I. N. Kar and R. K. P. Bhatt, "Ocean waves: A disturbance observer approach for an autonomous underwater vehicle," *OCEANS 2019-Marseille* (IEEE, 2019).

[16] W. Naeem, R. Sutton and S. M. Ahmad, "Pure Pursuit Guidance and Model Predictive Control of an Autonomous Underwater Vehicle for Cable/Pipeline Tracking," **In:** *Proceedings-Institute of Marine Engineering Science and Technology Part C Journal of Marine Science and Environment* (Citeseer) pp. 25–35.

[17] W. Naeem, R. Sutton, J. Chudley, F. R. Dalgleish and S. Tetlow, "A genetic algorithm-based model predictive control autopilot design and its implementation in an autonomous underwater vehicle," *Proc. Inst. Mech. Eng., Part M: J. Eng. Maritime Environ.* **218**(3), 175–188 (2004).

[18] W. Naeem, R. Sutton, J. Chudley, F. R. Dalgleish and S. Tetlow, "An online genetic algorithm based model predictive control autopilot design with experimental verification," *Int. J. Control* **78**(14), 1076–1090 (2005).

[19] Q. Chen, T. Chen and Y. Zhang, "Research of GA-Based PID for AUV Motion Control," **In:** *International Conference on Mechatronics and Automation* (IEEE, 2009) pp. 4446–4451.

[20] E. W. McGookin, "Auv sliding mode autopilot optimisation using genetic algorithms," *IFAC Proc.* **34**(7), 317–322 (2001).

[21] K. Vasudev, R. Sharma and S. Bhattacharyya, "Shape optimisation of an auv with ducted propeller using ga integrated with cfd," *Ships Offshore Struct.* **13**(2), 194–207 (2018).

[22] S. Abbasi, M. Zeinali and P. Nejadabbasi, "Autonomous underwater vehicle hull geometry optimization using a multi-objective algorithm approach," *Int. J. Eng.* **31**(9), 1593–1601 (2018).

[23] N. H. Tran, A. D. Nguyen and T. N. Nguyen, "A Genetic Algorithm Application in Planning Path Using b-Spline Model for Autonomous Underwater Vehicle (auv)," **In:** *Applied Mechanics and Materials*, vol. 902 (Trans Tech Publications, 2020) pp. 54–64.

[24] M. G. Safonov and T. C. Tsao, "The Unfalsified Control Concept: A Direct Path From Experiment to Controller," **In:** *Feedback Control, Nonlinear Systems, and Complexity* (Springer, 1995) pp. 196–214.

[25] T. C. Tsao and M. G. Safonov, "Unfalsified direct adaptive control of a two-link robot arm," *Int. J. Adapt. Control* **15**(3), 319–334 (2001).

[26] M. G. Safonov and S. Y. Cheong, "Unfalsified Control: Theory and Applications," **In:** *IEEE Conf. On Decision and Control, San Diego, CA December*.

[27] M. G. Safonov, Data-Driven Robust Control Design: Unfalsified ControlTechnical report (University of Southern California Los Angeles Dept of Electrical Engineering, 2006),

[28] C. Manuelli, S. G. Cheong, E. Mosca and M. G. Safonov, "Stability of Infalsified Adaptive Control with Non-Scli Controllers and Related Performance Under Different prior knowledge," **In:** *European Control Conference (ECC)* (IEEE, 2007) pp. 702–708.

[29] A. Paul, M. Stefanovic, M. G. Safonov and M. Akar, "Multi-Controller Adaptive Control (MCAC) for a Tracking Problem Using an Unfalsification Approach," **In:** *Proceedings of the 44th IEEE Conference on Decision and Control* (IEEE, 2005) pp. 4815–4820.

[30] F. B. Cabral and M. G. Safonov, "Unfalsified Model Reference Adaptive Control Using the Ellipsoid Algorithm," **In:** *Proceedings of IEEE Conference on Decision and Control* (2013).

[31] F. Previdi, T. Schauer, S. M. Savaresi and K. J. Hunt, "Data-driven control design for neuroprostheses: a virtual reference feedback tuning (vrft) approach," *IEEE Trans. Control Syst. Tech.* **12**(1), 176–182 (2004).

[32] J. Wang, Y. Wang, L. Cao and Q. Jin, "Adaptive iterative learning control based on unfalsified strategy for chylla-haase reactor," *IEEE/CAA J. Autom. Sin.* **1**(4), 347–360 (2014).

[33] V. Hassani, T. F. Onstein and A. M. Pascoal, "Application of data-driven control to dynamic positioning," *IFAC-PapersOnLine* **50**(1), 12392–12397 (2017).

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

[34] J. Das, Fédéric Py, J. B. J. Harvey, J. P. Ryan, A. Gellene, R. Graham, D. A. Caron, K. Rajan and G. S. Sukhatme, "Data-driven robotic sampling for marine ecosystem monitoring," *Int. J. Robot. Res.* **34**(12), 1435–1452 (2015).
[35] T. O. Fossum, Intelligent autonomous underwater vehicles: A review of auv autonomy and data-driven sample strategies, Applied Underwater Robotic Laboratory Norwegian University of Science and Technology (2016).
[36] R. S. Sanchez-Pena, P. Colmegna and F. Bianchi, "Unfalsified control based on the H_∞ controller parameterisation," *Int. J. Syst. Sci.* **4**(1) (2013).
[37] Y. C. Sung, S. V. Patil and M. G. Safonov, "Data-driven h_∞ loop-shaping controller design," *Int. J. Robust Nonlinear* **28**, 3678–3693 (2018).
[38] S. V. Patil, Y. C. Sung and M. G. Safonov, "Nonlinear unfalsified adaptive control with bumpless transfer and reset," *IFAC-PapersOnLine* **49**(18), 1066–1072 (2016).
[39] Y. C. Sung, S. V. Patil and M. G. Safonov, "Robustness of uncertain switching nonlinear feedback systems against large time-variation," *IEEE Trans. Autom. Control* **67**(2), 993–1000 (2022).
[40] S. V. Patil, Y. C. Sung and M. G. Safonov, "Unfalsified adaptive control for nonlinear time-varying plants," *IEEE Trans. Autom. Control* **67**(8), 3892–3904 (2022).
[41] T. I. Fossen. *Guidance and Control of Ocean Vehicles* (John Wiley and Sons, 1994).
[42] C. Ma, Y. Tang, J. Wang, J. Yan, Y. Liao and D. Jiang, Adaptive trajectory tracking control and parameter identification for autonomous underwater vehicles based on contraction theory (2022).
[43] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control* (John Wiley and Sons, 2011).
[44] M. G. Safonov, "Focusing On the Knowable," **In:** *Control Using Logic-Based Switching* (Springer, 1997) pp. 224–233.
[45] A. H. Wright, "Genetic Algorithms for Real Parameter Optimization," **In:** *Foundations of Genetic Algorithms*, vol. 1 (Elsevier, 1991) pp. 205–218.
[46] J. P. Hespanha. *Logic-Based Switching Algorithms in Control* (Yale University, 1998).
[47] Y. Siek, N. Zhurba and E. Khutornaia, "Onboard computing environment of autonomous unmanned underwater vehicles: Possible design technologies and their comparative analysis," *Vibroeng. Procedia* **38**, 62–67 (2021).
[48] S. Singh, J. Kaur and R. S. Sinha, "A Comprehensive Survey on Various Evolutionary Algorithms on GPU," **In:** *International Conference on Communication, Computing & Systems (ICCCS–2014)* pp. 20–21.
[49] M. L. Wong, T. T. Wong and K. L. Fok, "Parallel Evolutionary Algorithms On Graphics Processing Unit," **In:** *IEEE Congress On Evolutionary Computation*, vol. 3 (IEEE, 2005) pp. 2286–2293.
[50] O. Maitre, L. A. Baumes, N. Lachiche, A. Corma and P. Collet, "Coarse grain Parallelization of Evolutionary Algorithms on GPGPU Cards with EASEA," **In:** *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pp. 1403–1410.
[51] S. Tsutsui and N. Fujimoto, "Solving Quadratic Assignment Problems by Genetic Algorithms with GPU Computation: A Case Study," **In:** *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pp. 2523–2530.
[52] P. Pospichal, J. Jaros and J. Schwarz, "Parallel Genetic Algorithm on the Cuda Architecture," **In:** *European Conference on the Applications of Evolutionary Computation* (Springer) pp. 442–451.
[53] Z. Luo and H. Liu. "Cellular Genetic Algorithms and Local Search for 3-Sat Problem on Graphic Hardware," **In:** *IEEE International Conference on Evolutionary Computation* (IEEE, 2006) pp. 2988–2992.
[54] T. Van Luong, N. Melab and E. G. Talbi, "Parallel Hybrid Evolutionary Algorithms On Gpu," **In:** *IEEE Congress On Evolutionary Computation* (IEEE) pp. 1–8.
[55] S. Debattisti, N. Marlat, L. Mussi and S. Cagnoni, "Implementation of a Simple Genetic Algorithm within the Cuda Architecture," **In:** *The Genetic and Evolutionary Computation Conference*.
[56] P. Pospichal and J. Jaros, "Gpu-based acceleration of the genetic algorithm," *GECCO Competition* (2009).
[57] J. Hofmann, S. Limmer and D. Fey, "Performance investigations of genetic algorithms on graphics cards," *Lect. Notes Comput. Sci.* **12**, 33–47 (2013).
[58] J. Ou, S. H. Hong, P. Ziehl and Y. Wang, "Gpu-based global path planning using genetic algorithm with near corner initialization," *J. Intell. Robot. Syst.* **104**(2), 34 (2022).

Appendix A

Genetic algorithm implementation flow chart

Figure A1 presents a flow chart illustrating the different steps of the genetic algorithm.

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press

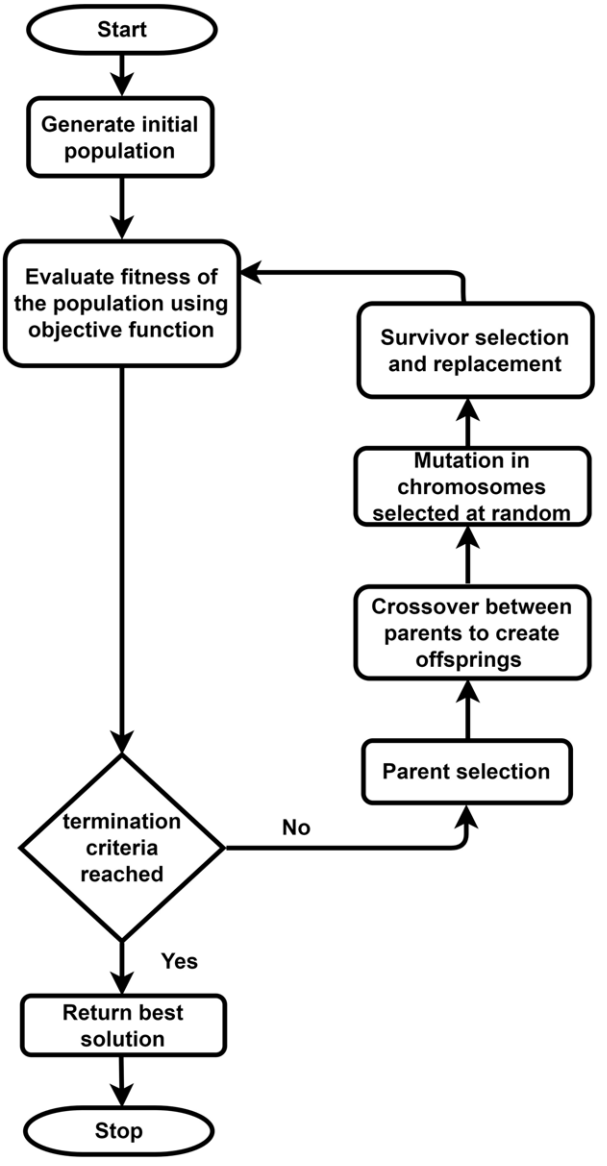


Figure A1. *Flow chart of genetic algorithm.*

Cite this article: T. Rasul and K. Mukherjee (2025). "Data-driven controller adaptation and parameter estimation in autonomous underwater vehicles by genetic algorithm enhanced unfalsification approach", Robotica **43**, 3030–3057. https://doi.org/10.1017/S0263574725102099

https://doi.org/10.1017/S0263574725102099 Published online by Cambridge University Press