Ocean Engineering 266 (2022) 112836

Contents lists available at ScienceDirect

Ocean Engineering

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

High fidelity modelling and GA optimized control of yaw dynamics of a custom built remotely operated unmanned underwater vehicle

Ahsan Tanveer, Sarvat M. Ahmad, PhD *

Faculty of Mechanical Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, 23460, Pakistan

ARTICLE INFO

Keywords:
Underwater vehicle
Robust control
External disturbance
PI
ROV modelling
micro-ROV
Linear system identification
Genetic algorithm (GA)
Linear quadratic regulator (LQR)
Marine predators algorithm (MPA)

ABSTRACT

In this paper, a custom built, underactuated, inspection class micro Remotely Operated Unmanned Underwater Vehicle (ROV) is employed as a testbed to investigate control and modelling problems related to underwater vehicles in shallow waters and pools. Dynamic model for yaw is obtained via mathematical modelling and system identification techniques. To instil confidence in the identified model, residuals and cross-validation tests are carried out to obtain high fidelity vehicle model for subsequent stabilizing closed-loop control design. Following the modelling exercise, design, real-time implementation, and analysis of a GA optimized PI controller for yaw is carried out. The performance of the GA optimized controller is benchmarked against the experimental results of a multi-parameter root-locus tuned PI controller and simulated responses of a standard linear quadratic regulator (LQR) controller. In addition, the efficacy of the GA-PI controller is gauged employing recently developed marine predator algorithm (MPA). The need for a controller with optimized performance motivated the utilization of GA and MPA optimization techniques. The results from real-time pool experiments indicate substantially enhanced performance of GA optimized controller, outperforming other controllers by as much as 22% in performance indicators such as settling time and maximum overshoot. Furthermore, the GA optimized controller demonstrated far better robustness and disturbance rejection capabilities.

1. Introduction

Research in the field of underwater vehicles started in the 1950s (Antonelli et al., 2008) and the first unmanned vehicle, called Cable-controlled Underwater Recovery Vehicle (CURV) (Wernli, 2001) came out in the 1960s. Nowadays with the advent of technological development, underwater robotic systems are extensively employed for ocean floor mapping (Tena, 2011), studying marine life (Smolowitz et al., 2015), reconnaissance (Midtgaard et al., 2000), wreckage inspection (Ura et al., 2005), undersea cable inspection (Jacobi and Karimanzira, 2013), search and rescue (Venkatesan, 2016), and water condition monitoring (Li et al., 2017). Underwater robotic systems can be classified into (i) manned submersible vehicles (MSVs), (ii) tethered Remotely Operated Vehicles (ROVs), and (iii) cordless Autonomous Underwater Vehicles (AUVs) (Jebelli and Yagoub, 2016).

ROVs are effectively miniaturized submarines controlled by a trained pilot operator from a ground station or a surface ship. Nowadays, there is a wide range of ROVs that exist and are employed in subsea settings. Capocci et al. (2017), based on their weight, have classified underwater vehicles into inspection class, and intervention class vehicles. Micro-ROVs, the focus of this paper, fall under the category of inspection class vehicles and can weigh up to 20 kg (Capocci et al., 2017). The purpose behind the use of such vehicles is to cut operational costs and system intricacies, allowing job completion in an effective fashion. These vehicles are particularly designed to function as baby ships, supplementing the capabilities of larger inspection class motherships, and tend to be in an open-frame configuration, allowing for additional sensors to be installed.

Generally speaking, most ROVs are manually operated without any autonomous capabilities. Automation, however, can reduce the time and expense of operation (Tena, 2011). The development of an automated control strategy requires understanding of the dynamics of the vehicle. The dynamics of any physical system can be best described by mathematical modelling, also referred to as first-principle modelling. Parameters associated with mathematical models directly influence the system's physical properties. The first principle-based mathematical models are best suited for systems that are relatively simple in nature. However for complex systems, such as underwater vehicles, with their

* Corresponding author.
E-mail addresses: smahmad@giki.edu.pk, cop97sma@gmail.com (S.M. Ahmad).

https://doi.org/10.1016/j.oceaneng.2022.112836
Received 22 March 2022; Received in revised form 11 September 2022; Accepted 6 October 2022

intricate mechanical, electronic, electromechanical subsystems and tether, all interacting with each other, using this conventional modelling approach is imprecise. Furthermore, specialist knowledge of each subsystem and its relationship with other subsystems is needed.

System identification (SI), on the other hand, is an experimental approach that is generic in nature and easy to employ. Using this approach, a valid system model can be predicted, in an iterative manner, using experimental data alone. It provides reliable and accurate models in a short amount of time and that too without any prior knowledge of the system. However, there is an issue with experimentation involving underwater vehicles: the prohibitively expensive nature of repeated testing. Therefore, it is not possible to write off the significance of mathematical modelling altogether vis a vis SI modelling. A crucial benefit of acquiring high fidelity models is the elimination of unwieldy tuning of controllers. Therefore, both these modelling techniques are investigated in this paper in an attempt to weigh their pros and cons for underwater applications.

Unlike aerial vehicles, dynamic equations for underwater vehicles are not that easy to establish from the first principle. Although, Fossen (2014) has developed 6-DoF equations of motion for underwater vehicles, linearization and then parameter estimation from these equations is a tedious task and, in some cases, impractical. This is because a significant number of mutually interacting parameters, such as buoyancy, drag and added mass affect the behaviour of underwater vehicles. The effect of these parameters is even more pronounced for micro-ROVs. Mai et al. (Christian et al., 2017), Wang et al. (Wang and Clark, 2006), and Enevoldsen et al. (2018) have already explored the first principle method for modelling underwater vehicles. However, most of these studies have used off-the-shelf ROVs. Also, lack of rigorous validation of the resulting model was observed in these studies. Likewise, modelling using system identification approaches have been investigated by Ahmad et al. (Ahmad and Sutton, 2003), Tinker et al. (1979) and Goheen et al. (Goheen and Jefferys, 1990). But these studies use non-linear mathematical models as a representative underwater vehicle for modelling data generation. However for this work, real-time pool environment data of a custom built micro-ROV is utilized for parameter estimation and model generation. The mathematical model for dynamic characterization of yaw motion is then followed by dynamic model obtained using linear black-box system identification techniques. The acquired models are then validated using residual and cross-validation test. High fidelity models thus obtained are employed for closed-loop control design.

For closed-loop control design, PID controller is by far the most basic yet powerful algorithm. It has been employed in process industries (McMillan, 2012), in aerial vehicles (Noshahri and Kharrati, 2014), for solar panel tracking (Arif et al., 2018), to name a few. Although PID has been extensively used for ROV control, however, most of the reported studies are carried out in simulation environment only. For instance, Folcher et al. (Folcher and Rendas, 2001) evaluated the performance of a classical root locus heave controller for a medium-sized inspection class vehicle in a numerical simulation environment. Walker et al. (2020) utilized models of a mini-ROV to develop a PID controller for station keeping in an attempt to demonstrate the disturbance rejection characteristics in simulations. Santiago et al. (Rúa and Vásquez, 2016) simulated PID controller tuned using heuristic methods for a medium-sized inspection class ROV. Zanoli et al. (Zanoli and Conte, 2000) simulated PID fuzzy controller in order to reduce overshoot in depth for an ROV. All the aforementioned works utilize off-the-shelf medium-sized ROVs and carry out the high-fidelity modelling exercises in simulated environments. But no matter how complicated or non-linear a model is, it does represent every aspect of the ROV's motion. This makes real-time testing inevitable. For this very reason, the design and real-time implementation of the controller onto the vehicle is extensively investigated in this work.

Nonetheless, the behaviour of the controller after implementation onto a vehicle is influenced by how the controller is tuned. Traditional tuning approaches, such as Ziegler Nicholas and root-locus are an excellent place to start when designing a closed-loop controller since they provide ballpark estimates for controller gains. Some of the works related to conventional controller tuning are that of Abidin et al. (2016) and Vahid et al. (Vahid and Javanmard, 2016). The former implemented Ziegler Nicholas tuned PID controller for depth control while the latter used conventional root-locus for the design of PD controller for pitch control. Though classical tuning methods are simple and intuitive, the gains obtained using these methods have less parametric robustness and cannot be regarded as optimal, as demonstrated by Ming-Tzu et al. (Ho et al., 1997).

In this regard, meta-heuristic tuning algorithms provide a potential alternative to the conventional methods. The two main categories of meta-heuristic algorithms are swarm intelligence (C and Li, 2008) and evolutionary (Bäck, 1996) methods. The principles of natural evolution are imitated by evolutionary algorithms. The Genetic Algorithm (GA) (DE and Holland, 1988) is the most powerful and widely known algorithm in this class. In GA, a collection of random solutions to a specific problem are used to initiate the optimization process. It alters the variables of the solutions based on their fitness value after analysing the solutions using the objective function. Because the best candidates are more likely to be involved in improving the solution, the randomly generated initial solutions are very likely to experience improvements (Mirjalili et al., 2017). This is why a genetic algorithm is being investigated for this study. Furthermore, as demonstrated by Piotrowski et al. (2017), the performance of GA is fairly superior to counterparts in a vast majority of real-world problems. Also, since the number of iterations it takes GA to converge to global minima strictly depends on the initial population (Deng et al., 2015), knowing the ballpark estimates for an optimal solution may significantly minimize the compute resources required to carry out GA optimization. The study under consideration, hence, takes into account the estimates obtained from the root-locus control design technique in generating initial population for GA optimization. Nonetheless, researchers have studied numerous strategies for ROV control. Chen et al. (2020) investigated the design of a PSO fuzzy PID controller. Hernandez et al. (Hernandez-Alvarado et al., 2016), on the other hand, implemented a backpropagation neural network on an underactuated ROV. Likewise, genetic algorithm (GA) has been employed by many researchers. Moghaddam et al. (Javadi-Moghaddam and Bagheri, 2010) proposed GA for adaptive neuro-fuzzy sliding mode controller for ROVs. Xu et al. (Xu and Wang, 2018) used GA to tune fuzzy PID controller for a medium-sized inspection class ROV in the SIMULINK simulation environment. Similarly, Kartal et al. (Kartal et al. Leblebicioğlu) simulated the performance of optimal PID controller tuned using GA for a 3-thruster survey vehicle "SAGA". As noted, most of the work found in the open literature on the control design of ROVs using GA is in the simulation environment. However, this work investigates and evaluates the performance of a GA optimized PI controller for yaw motion of the micro-ROV in real-time. The optimized controller is not only implemented onto the micro-ROV in real-time, but its performance is benchmarked against a conventional root-locus and LQR controllers. In addition, GA-PI is evaluated against a fairly recent meta heuristic called marine predators algorithm or MPA. It is worth noting that, to the best of the authors' knowledge, no evidence of MPA being used for ROV control exists in the literature. Nonetheless, the aim is to show the supremacy of the GA optimized controller as opposed to a conventionally tuned one. The key contributions of this paper include demonstrating the effectiveness of the modelling methodology for custom-built micro ROVs, and real-time implementation of the multi parameter root-locus tuned and GA optimized controllers for comparison with the simulated results of LQR and MPA.

The paper is organized as follows: In Section 2 a brief description of the vehicle is provided. Section 3 discusses the development of first principle model and parameter estimation for yaw motion. A detailed description of dynamic model obtained using system identification is provided in Section 4. Section 5 deals with the design, real-time

implementation, and analysis of a root-locus and GA tuned PI controller for yaw. A detailed discussion outlining comparison of GA with LQR and MPA also forms part of Section 5. The paper is concluded in Section 6.

2. Vehicle description

The development of an ROV and its subsystems, including electronic, mechanical and communication sub-system is a highly iterative task, and cannot be considered mutually exclusive. Once the design and selection of components for individual subsystems is complete, each subsystem is tested before integrating all three sub-systems together to create an operational micro-ROV ready for outdoor testing, as seen in Fig. 1. Fig. 2 depicts the interaction of various sub-systems of the micro-ROV.

As evident from Fig. 2, the micro-ROV has a water-tight hull that houses the electronic subsystem, which includes a microcontroller, two motor drivers, a turbidity sensor and an inertial measurement unit (IMU). IMU used in the vehicle is MPU-9250 which is a System in Package (SiP) that combines two chips: AK8963, a 3-axis compass; and MPU6500, a 6-axis accelerometer and gyroscope. IMU is used to obtain the vehicle's orientation. A complimentary filter, shown in equation (1), is used to convert accelerometer and gyroscope data to orientation angles.

$$\theta_n = (1-\alpha) * (\theta_{n-1} + gyr * dt) + (\alpha * acc) \tag{1}$$

$$\alpha = t_c / (t_c + 1/f_s) \tag{2}$$

Where, θ_n is the current output, θ_{n-1} is the previous output, gyr is the gyroscope data, acc is the accelerometer data, t_c is time constant and f_s is the sampling frequency.

One of the key components of the mechanical subsystem of the ROV is the thrusters; one downward-facing thruster and two rear-facing thrusters making the ROV an under-actuated vehicle. The downward-facing thruster is responsible for sinking and surfacing. The vehicle is safe for operation up to 10-m of depth. The maximum downward speed is approximately 0.2 m/s. Power and control signal from the ground-station-based PC is provided through the tether where the program for on-board computer is written in C++ in Arduino development environment. Surge, heave, and yaw are the only directly controllable DoFs with the current vehicle configuration. However for this article, yaw motion will be investigated because the idea behind the development of a micro-ROV is to demonstrate its control capabilities for inspection of underwater structures. Control over yaw will allow the pilot to lock onto a target at all times while maintaining the desired vehicle orientation

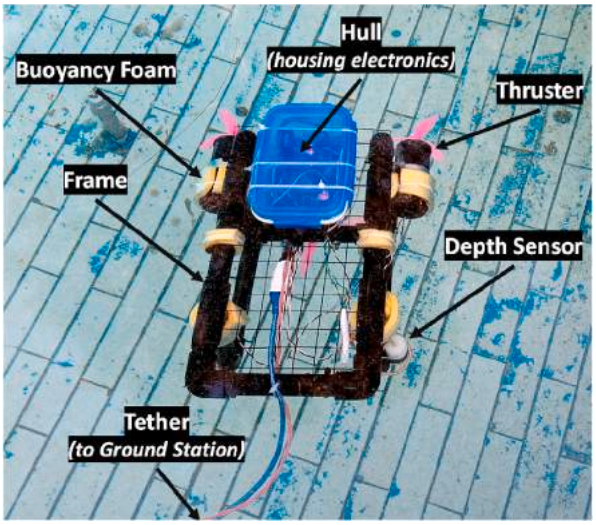

Fig. 1. Fully functional indigenous micro-ROV.

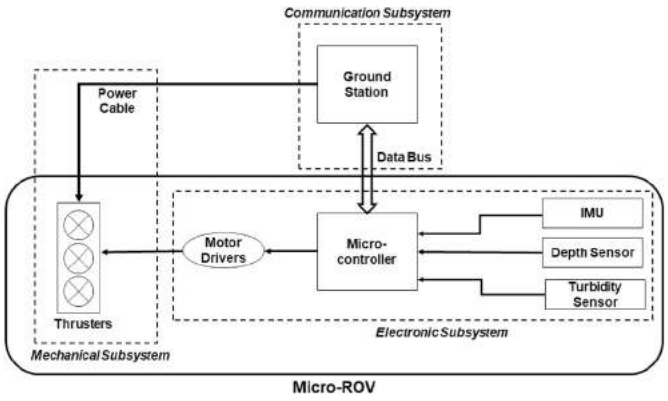

Fig. 2. Interacting sub-systems of the micro-ROV.

underwater. For complete specifications of the micro-ROV see Table 1.

3. Mathematical modelling

Just like aerial vehicles, underwater vehicles have six degrees of freedom: three translation and three rotational, as shown in Fig. 3. All the forces and moments acting on the vehicle are calculated about the assumed reference coordinate axis. The x-axis always points in the forward direction of the vehicle's motion. The y-axis points towards the starboard side, and the z-axis defines the depth. Z-axis is zero at the surface and increases with depth. The vehicle also undergoes angular motion in each of the axes namely φ, θ, and ψ.

The standard 6 DOF notation for an underwater vehicle can be seen in Table 2.

With respect to a local body-fixed frame of reference, the yaw dynamics of an ROV can be expressed by the following non-linear equation of motion (Fossen, 2011).

$$\begin{aligned} I_z\dot{r} + (I_y - I_x)pq - I_{xy}(p^2 - q^2) - I_{yz}(pr - \dot{q}) + I_{xz}(qr - \dot{p}) \\ + m[x_G(\dot{v} + ur - wp) - y_G(\dot{u} - vr + wq)] = N \end{aligned} \tag{3}$$

Where N represents the moments about c.g, while x_G and y_G are rectangular components of the vector linking body-frame to world-frame. I_z represents moment of inertia of the ROV in the z-plane. While I_{xy}, I_{yz}, I_{xz} refer to moment of inertia induced in one axis due to motion in another axis. Apart from the first term in equation (1), every other term refers to the cross-coupling of yaw with other DoFs.

Utilizing the condition that the body-fixed frame coincides with c.g of the vehicle equation is simplified:

$$I_z\dot{r} + (I_y - I_x)pq - I_{xy}(p^2 - q^2) - I_{yz}(pr - \dot{q}) + I_{xz}(qr - \dot{p}) = N \tag{4}$$

Equation (4) represents simplified non-linear equation of motion for yaw. Ideally, a linear and SISO model is preferred for a closed-loop control design because of its intuitive nature and ease of parameter estimation. Therefore, it is desirable to linearize Equation 2 for modelling and closed-loop control design purposes.

Rearranging equation 4, we get:

Table 1
Micro-ROV specifications.

Parameter	Value
Weight of the Hull	0.26 kg
Weight of the ROV	1.68 kg
Weight of the ROV (SolidWorks)	1.67 kg
Weight of Eq. Water (SolidWorks)	0.34 kg
Yaw Moment of Inertia (SolidWorks)	0.02 kgm^2
Yaw Moment of Inertia (Experimental)	0.019 kgm^2
Distance between center of mass and thruster	0.11 m

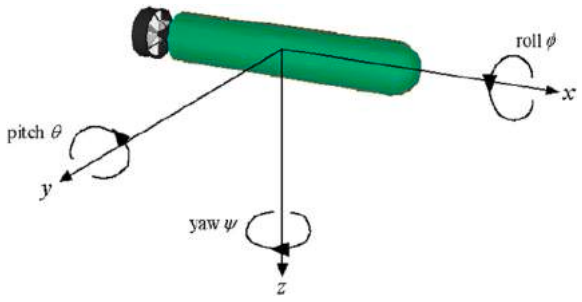

Fig. 3. Underwater vehicle coordinate system.

Table 2
6-DOF standard underwater vehicle notation.

DOF	Motions	Forces & Moments	Linear & Angular Velocities	Positions and Euler Angles
1	Surge	X	u	x
2	Sway	Y	v	y
3	Heave	Z	w	z
4	Roll	L	p	φ
5	Pitch	M	q	θ
6	Yaw	N	r	ψ

$$I_z\dot{r} = N + h(p,q,r) \tag{5}$$

Where $h(p,q,r)$ represent coupled dynamics affecting yaw and is a function of angular velocities. Enevoldsen et al. (2018) suggest ignoring the coupled dynamics if the following assumptions hold true: the ROV is symmetric in all three axes; the vehicle operates at low velocities. Both assumptions are true for the vehicle under consideration. Therefore, ignoring the cross-coupling terms and rewriting equation 5, we get:

$$I_z\dot{r} = N \tag{6}$$

Where N represent net moment acting on the ROV and is represented as:

$$N = \tau_{rear} - \tau_d^r \tag{7}$$

Substituting equation 7 in equation (6). By rearranging the resulting equations, we get:

$$I_z\dot{r} + \tau_d^r = \tau_{rear} \tag{8}$$

Where τ_d^r represents yaw dependent drag torque, and τ_{rear} rear thruster. Yaw dependent drag torque is directly proportional to the angular velocity squared. Incorporating the drag equation, as it is, into Equation (8) will result in a non-linear equation. However, the non-linear nature of drag torque can be assumed linear provided that the ROV operates only in a small range of available velocity assortment. In doing so, only the slope of the velocity-vs-drag curve is now required to determine drag torque, and the relationship of drag torque with respect to the angular velocity becomes:

$$\tau_d^r = b_{yaw}r \tag{9}$$

Where b_{yaw} represents drag coefficients in yaw. Using equation 9 in equation 8 and at the same time replacing r with $\dot{\psi}$, we get:

$$I_z\ddot{\psi} + b_{yaw}\dot{\psi} = \tau_{rear} \tag{10}$$

Equation 10 represents yaw in terms of angular position. Where τ_{rear} is rear thrust torque and is equal to differential force generated by thruster times the moment arm (l) between thruster and c.g of ROV. Using $\tau_{rear} = 2lF_{diff}$ in equation 10.

$$I_z\ddot{\psi} + b_{yaw}\dot{\psi} = 2lF_{diff} \tag{11}$$

Equation (11) represents simplified linear equation of motion for an ROV. By applying the Laplace transform we can obtain transfer functions, as shown below.

$$\frac{\psi(s)}{F_{diff}(s)} = \frac{2l}{I_z s^2 + b_{yaw}s} \tag{12}$$

The unknowns in equation 12 are drag coefficient and thruster gain. Both these unknowns need to be determined experimentally. The following subsections deal with thruster modelling and the determination of thruster gain, followed by a discussion on the determination of drag coefficients.

3.1. Thruster model

The force represented in equation 10 is provided to the system by the thrusters, and in order to control the ROV, the dynamics of thrusters must be known. Incorporating thruster dynamics with the plant model will present a holistic view of the micro-ROV's dynamics. Understanding the dynamics of any thruster requires static as well as dynamic characterisation.

According to Quan (2017), the dynamics of a thruster can be represented by a generic low pass filter. A general first-order transfer function representing thruster dynamics is:

$$\frac{F(s)}{Input\ Signal} = \frac{gain}{\tau s + 1} \tag{13}$$

Where τ represents time constant. From Fig. 4, which represents the step response of Micro-ROV's thruster to a known input, τ comes out to be 0.3 s.

As evident from Fig. 5, which shows the thrust curve for the thrusters, the thrusters are inherently non-linear. However, in the small operating region closer to the dead-zone the static force-PWM relationship can be described by a straight line. A straight-line curve fit to the data will resolve the nonlinearity issue. The slope of the straight-line provides the gain for the thrusters and is equal to 0.006.

Using $\tau = 0.3$ and $gain = 0.006\ N/pwm$ in equation 13, we get the transfer function for the thrusters:

$$\frac{F(s)}{Input\ Signal} = \frac{0.006}{0.3s + 1} \tag{14}$$

It is important to note that adding complete dynamics of the thrusters with the plant will increase the overall order of the forward loop transfer function thereby making it difficult to use conventional second-order control design techniques. Therefore, the decision as to whether

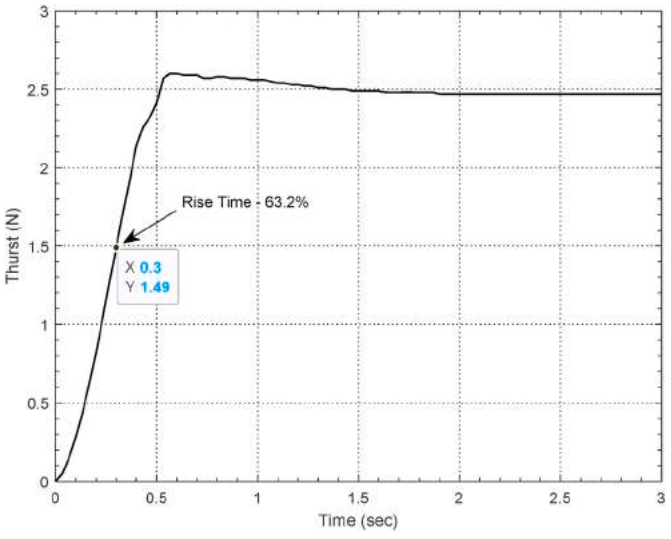

Fig. 4. Thruster step response curve.

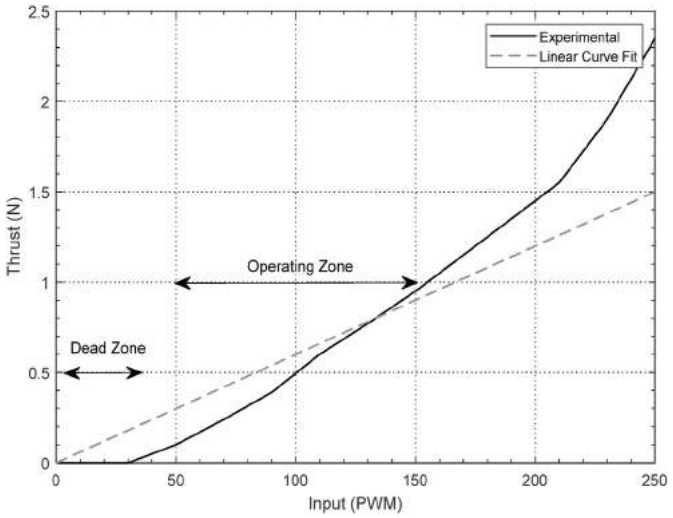

Fig. 5. Thrust calibration curve.

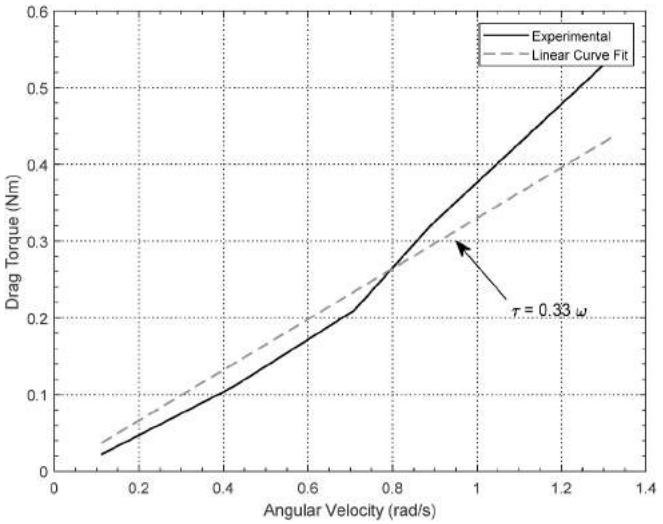

Fig. 6. Torque-angular velocity curve for yaw.

incorporate complete thruster dynamics or just the gains eventually depend on the bandwidth of the system. ROVs, unlike aerial vehicles, are often characterized by their low bandwidth. The thruster dynamics is much faster than the micro-ROV speed of response and hence can be ignored. Moreover, ignoring these dynamics is unlikely to alter the overall vehicle model fidelity. Nonetheless, it will simplify the control design process. Therefore, equation (14) becomes:

$$\frac{F(s)}{pwm} = 0.006 \tag{15}$$

Where $F(s)$ represents the force produced by the thruster and *pwm* refers to the input signal provided to the thruster.

3.2. Parameter estimation/experimentation

The objective of experiments in yaw is to obtain yaw angle data against rear thruster differential torque. The obtained data would help not only in performing System Identification but also to determine drag coefficient for mathematical model. All the experiments were conducted in a controlled pool environment with a maximum depth of 1-m and a diameter of 3-m. To excite the system, each thruster is powered in opposite direction thereby inducing a rotational motion. Due to ROV's symmetry, it is assumed that the angular velocity obtained clockwise, and anticlockwise direction is the same. Onboard gyroscope data is integrated to obtain yaw angle in radians. Most of the experiments showed twisting and tangling of tether due to rotation of the vehicle which adds strain to the vehicle. The effect of the twisted tether is analogous to a coiled spring. The data, however, is without any noise and is still quite useful as it provides a reasonable estimate of the yaw motion. The obtained data showing relationship between the drag or applied torque and angular velocity can be seen in Fig. 6. A linear curve fit to the slope in Fig. 6 provides the drag coefficient i.e., $b_{yaw} = 0.33 \frac{Nm}{rad-sec}$. Using the values of b in equation 9 gives:

$$\tau_d^r = 0.33r \tag{16}$$

Equation 16 represents yaw-dependent drag torque acting on the micro-ROV.

3.3. Experimental validation of the linear model

The yaw transfer functions for the ROV with complete information of parameters is:

$$\frac{\psi(s)}{pwm_{yaw}(s)} = \frac{0.0632}{s^2 + 17.37s} \tag{17}$$

Where ψ represents yaw angle (in rad) as the output and PWM signal from ground-PC as the input to the system. Both the poles for the system are real with one pole lying at the imaginary axis making the system unstable. A schematic of the system yaw model, with corresponding inputs and outputs, can be seen in Fig. 7.

Once parameter estimation is done, the resulting transfer function is simulated in MATLAB SIMULINK. Open-loop response of the system obtained from simulations is then validated with the experimental data, as seen in Fig. 8. The model output overlaps with the experimental one with slight deviation once the input is removed. This is because of the twisting of the tether as explained in the preceding paragraphs, unwinding of tether forces the ROV to assume its starting position. Experimental results reveals that the model captures the dynamics of the system reasonably well. However, the winding effect of tether needs to be incorporated in the model, and that is where system identification comes into play. Further explanation of system identification technique and its inherent advantages over first principle model are be explained in next section.

4. System identification

System Identification (SI) is a black box modelling technique that is used extensively in control system engineering. For this research, system identification is performed using MATLAB System Identification Toolbox. Following the identification process, model validation is carried out to instil confidence in the identified model. Two types of validation tests are: the residual test, and cross-validation test.

For the purpose of model identification, yaw experimental data is fed into the toolbox. The data was found to be without bias or noise, therefore, no pre-processing of the data was necessary. As for as the selection of model structure is concerned, a comprehensive array of linear model structures is available. The simplest of the linear model structures are the ARX and transfer function models. Based on the un-

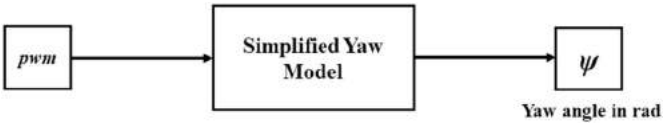

Fig. 7. A schematic of the SISO transfer function model.

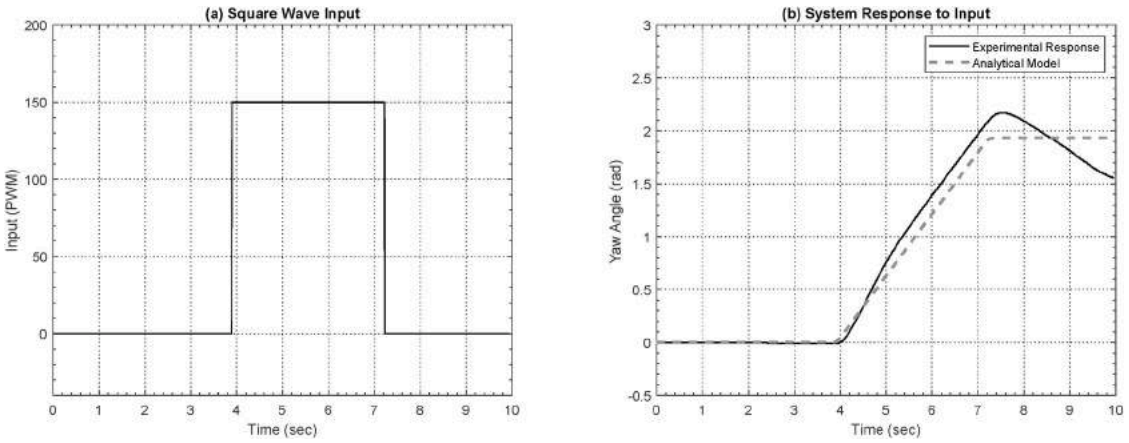

Fig. 8. Input-output signal used for analytical model validation.

derstanding of the physical system and to maintain continuity with the analytical model, a transfer function model is selected as a starting point. The resulting yaw transfer function is:

$$\frac{\psi(s)}{pwm_{yaw}(s)} = \frac{0.01394}{s^2 + 2.08s + 0.4681} \tag{18}$$

From the analysis of yaw transfer function, expresses in Equation (18), the system has two real poles on the left-hand side of the imaginary axis unlike analytical model that has a pole at the origin. Nonetheless, the position of dominant poles is the same for both the models, therefore, the response would be identical. Once identification exercise was completed, validation of the identified model was carried out. It was found that autocorrelation of residuals for TF model is outside the 99% confidence band, while CCF lies well within the confidence band. A reason for that could be that transfer function models are characterized by lack of an element to model noise or coupled dynamics. For this very reason, a polynomial model is needed for estimation that can account for noise and external disturbances, such as ARX and ARMAX model structures. The source of noise or disturbance in here is believed to be the tether, which has its own dynamics. This makes ARMAX model structure an ideal choice unlike ARX which does not model the coloured noise (refer Fig. 9).

4.1. Yaw ARMAX model

ARMAX model structure falls under the linear polynomial model category and a discrete-time ARMAX model is of the form:

$$(z)y(k) = B(z)u(k-n) + C(z)e(k) \tag{19}$$

Where $y(k)$ is the system output, $u(k)$ represents the input to the system, n is the system delay and $e(k)$ represents disturbance. $A(z)$, $B(z)$, and $C(z)$ are the system polynomials.

For excitation of the system, a doublet square wave input signal is used. The obtained system polynomials are then transformed to the continuous domain for transfer function development. The resulting yaw transfer function is:

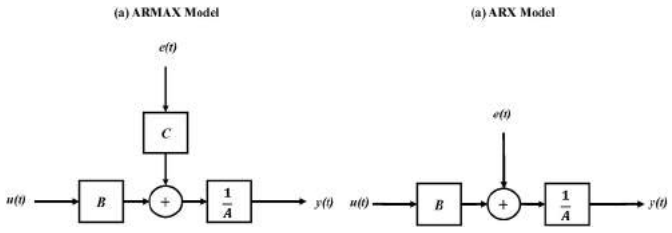

Fig. 9. ARX and ARMAX model signal flow schematic.

$$\frac{\psi(s)}{pwm_{yaw}(s)} = \frac{6.653e^{-5}s + 0.01068}{s^2 + 1.563s + 0.3512} \tag{20}$$

A comparison of the transfer function obtained using analytical, transfer function and ARMAX model reveals that ARMAX model has its poles in close proximity to those of TF model. The only major difference is the introduction of zero in the ARMAX model which can be attributed to additional inertia induced by the tether thereby delaying the system response.

5. Yaw ARMAX model validation

Fig. 10 shows the results of residual test for ARMAX model. As evident from the figure, both ACF and CCF lie within the confidence interval suggesting that the dynamics have been captured accurately.

Residual test is followed by cross-validation test. A doublet square wave input is used to excite the system to obtain cross-validation data. The response of the system response alongside ARMAX model can be seen in Fig. 11. The fitness of the estimated model is 59% which is although low owing to uncertainties in recording data, but still acceptable. The agreement of experimental data with ARMAX model demonstrates the capability of the ARMAX Model structure to accurately capture the dynamics of the system. However, it is important to note that the TF model cannot be completely disregarded as it can be used in the control design process to obtain ballpark gains. This is because CCF of the TF model lie within the confidence bounds, and it is recommended that for any model estimated using IV technique, the decision regarding

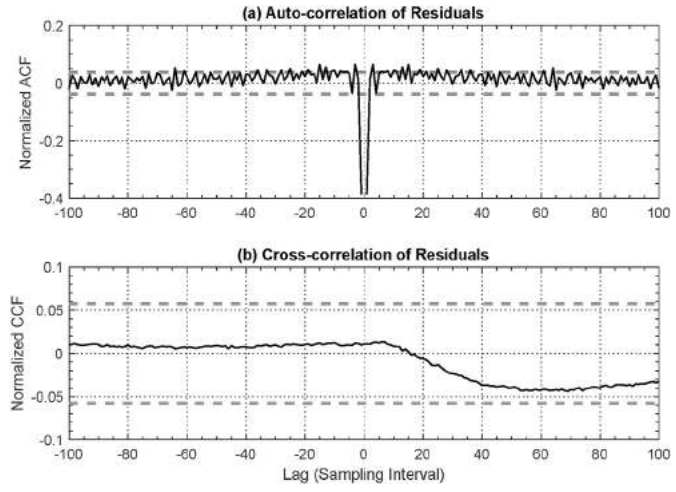

Fig. 10. Residual test for yaw ARMAX model.

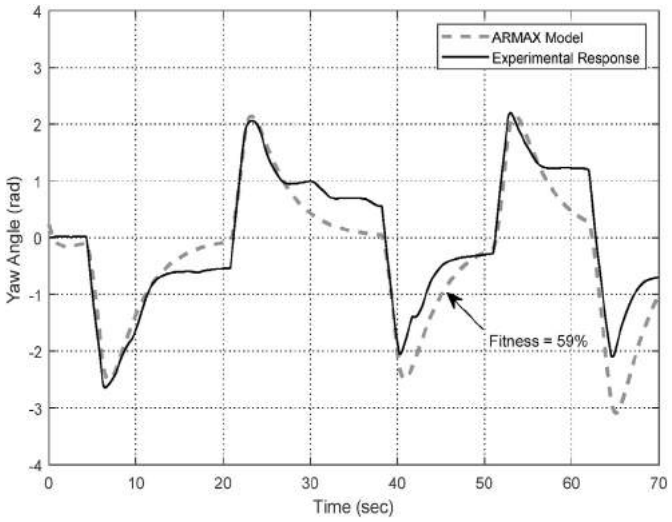

Fig. 11. A comparison of the system response with ARMAX model in cross-validation.

the effectiveness of the model should be made on the analysis of CCF alone. However, for closed-loop simulation and analysis complex ARMAX model can be utilized. With modelling exercise completed, the following section deals with the controller design.

5. Control design

The idea behind controller design is to enable a system to achieve desired performance. For that, it is essential to set the performance criteria well before the controller design. The effectiveness and performance of the resulting controller is assessed by its ability to meet the pre-set performance criteria. For vehicle under consideration, time-domain analysis is pursued. Conservative performance requirements are set based on the ROV's actuation capabilities. Pre-set performance requirements for yaw motion are:

- The closed-loop system is stable.
- Peak time should not be more than 3 s.
- The maximum allowable overshoot is 20%.
- Settling time should be less than 5 s.

The process of controller design starts with the evaluation of open-loop response, which is essential for two reasons. First to determine if feedback control is even needed in the first place. Secondly, to decide which controller configuration must be implemented. Experimental open-loop responses for yaw led to the selection of a Proportional-Integral (PI) controller. The mathematical expression expressing the PI controller is as follows.

$$u(t) = k_p e(t) + k_i \int e(t) \tag{21}$$

Where $u(t)$ is the control signal, $e(t)$ is the system's error, and k_p and k_i represent gains for Proportional, and Integral modes.

The process of PI controller design is mere determination of k_p and k_i. Various methods can be employed for the selection of these gains. These methods range from heuristic methods such as Ziegler-Nicholas to model-based methods such as root locus to AI-based methods such as ant-colony, artificial neural network, and genetic algorithm. Root-locus is a fairly simple and intuitive method. It can provide ballpark gains for tuning of PID controllers and therefore can be used as a starting point in controller design. However, controllers tuned using root-locus are neither optimal nor extensively robust. Therefore, this paper deals with the design of a PI controller optimized using Genetic Algorithm. The performance of the optimized controller is then compared with controllers tuned using multi-parameter root-locus, LQR, and MPA to demonstrate the effectiveness of GA in optimizing the performance of a controller for underwater applications.

6. Root-locus based design of a benchmark PI controller

Root locus provides a deterministic way of determining PI controller gains. It works on utilizing the information of poles of the plant. But first, it is essential to identify the desired pole location and it depends on the pre-set performance requirements. In order to pinpoint the desired pole location on an s-plane, information of desired response damping and closed-loop natural frequency is needed. Equation (22) expresses the damping coefficient in terms of percentage overshoot. Similarly, Equation (23) relates natural frequency and damping coefficient with settling time for 5% overshoot requirement (Nise, 2019).

$$\zeta = -\frac{\ln\left(\%_{\frac{OS}{100}}\right)}{\sqrt{(\pi)^2 + \ln\left(\%_{\frac{OS}{100}}\right)^2}} \tag{22}$$

$$T_s = \frac{3}{\zeta \omega_n} \tag{23}$$

Using the performance requirements of maximum overshoot of 20% and settling time of 5 s in Equation 22 and 23, we get

$$\zeta = 0.45\ , \omega_n = 1.75\ rad/s \tag{24}$$

Where ζ is the damping coefficient and ω_n is the desired closed-loop natural frequency.

Equation 24, when plotted on an s-plane, give the exact location from where the root locus of the system should pass in order to meet the performance requirements.

Using the yaw model for the micro-ROV from Equation (18), the closed-loop yaw transfer function becomes:

$$G_{cl}(s) = \frac{0.01394k_p s + 0.01394k_i}{s^3 + 2.08s^2 + \left(0.4681 + 0.01394k_p\right)s + 0.01394k_i} \tag{25}$$

Extracting and rearranging denominator polynomial, we get:

$$s^3 + 2.08s^2 + \left(0.4681 + 0.01394k_p\right)s + 0.01394k_i = 0 \tag{26}$$

Writing Equation 26 in terms of integral gain by separating k_i:

$$1 + k_i\left(\frac{0.01394}{s^3 + 2.08s^2 + \left(0.4681 + 0.01394k_p\right)s}\right) = 0 \tag{27}$$

Equation 27 is used for multi-parameter root locus design. Numerous values of k_p (200, 230, and 250) are inserted into the equation and as a result a contour plot is obtained, as shown in Fig. 12. Each individual line on the plot represents a constant k_p while k_i changes along each constant k_p line. Desired proportional and integral gains are obtained from the curve that passes through $\zeta = 0.45$ and $\omega_n = 1.75$.

From the multi-parameter plot, the curve with $k_p = 230$ and point $k_i = 90$ coincides with $\zeta = 0.45$ and $\omega_n = 1.75$. As a result, these gains are chosen for simulation and subsequently for implementation on the actual system. Root locus predicts an overshoot of 20.2%, damping of 0.453, and frequency of 1.75 for the selected gains. It is worth mentioning here that the root locus technique provides gains in a ball-park region and some manual tuning of the controller would still be needed. Fig. 13 depicts the closed-loop response of the yaw ARMAX model for the gains obtained using the root locus approach, i.e., $k_p = 230$ and $k_i = 90$.

From the analysis of closed-loop system response in Fig. 13, the peak time of the system is found to be 2 s, with 22% overshoot and a settling time of 4.3 s. The response of the system has slightly deteriorated than predicted by the root locus. This can be attributed to the fact that multi-parameter root locus analysis is performed on a simpler TF model but for

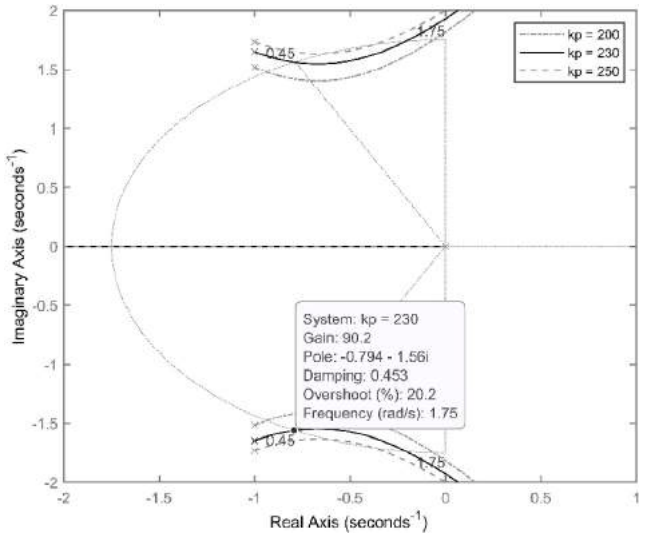

Fig. 12. Yaw multi-parameter root-locus contour plot.

simulations, the high-fidelity ARMAX model is used which is much closer to system dynamics. Furthermore, the response shows that the control effort is well within the actuator limit.

7. Robustness analysis of root-locus tuned controller

ROVs operate in the unforgiving underwater environment, therefore, a robust controller is a must for their smooth operation. The robustness of a controller is analysed in both the time and frequency domain. In the frequency domain, gain and phase margins provide a measure of robustness while in the time domain the robustness is verified by changing system parameters by some percentage and then drawing a comparison of the responses with the original system.

7.1. Frequency domain analysis

Systems with higher gain and phase margins can withstand greater uncertainties parameters before going unstable. Although no standard exists in terms of minimum gain and phase margin requirements for underwater vehicles, Safonov et al. (Safonov and Athans, 1977) and Lehtomaki et al. (1981) have suggested that ROVs must have:

- 60° of phase margin in each DoF
- Infinite gain margin

Utilizing Equation 25 with root locus gains ($k_p = 230$ and $k_i = 90$), the closed-loop transfer function becomes:

$$G_{cl}(s) = \frac{3.136s + 1.533}{s^3 + 2.08s^2 + 3.605s + 1.533} \tag{28}$$

The gain and phase margin plot for Equation (28) can be seen in Fig. 14.

As depicted in Fig. 14, the gain margin is infinite while the phase margin is 73.4°. Both the gain and phase margins are much higher than required.

7.2. Time-domain analysis

The open-loop transfer function for yaw reveals two parameters on which the system's response is dependent (i) inertia of the micro-ROV, and (ii) yaw drag. Changing the inertia of the vehicle will mimic a real-world scenario in which an additional payload is placed inside the hull which alters the original inertia of the vehicle. The controller designed for the original vehicle is deployed and tested for the system with modified inertia in Simulink. If the vehicle performance is not degraded by much, the controller would be deemed appropriate and no new control design would be needed. Therefore, for robustness analysis in yaw, the system is simulated by varying inertia of the ROV by ± 50%. From Fig. 15, it is clear that even with a 50% change in the inertia of the ROV the response of the system is exactly the same. Moreover, control efforts are within the actuator limit. Hence, it can be said that the designed PI controller is robust enough to overcome parameter variation.

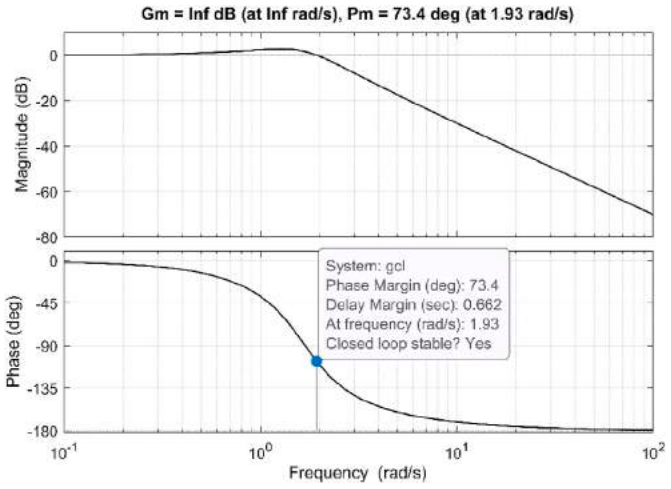

Fig. 14. Bode plot for root locus tuned yaw model.

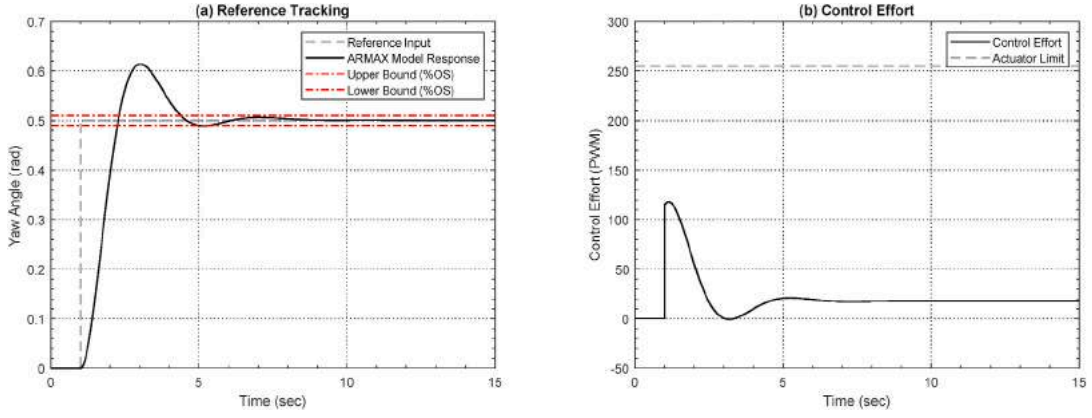

Fig. 13. Closed-loop response of the yaw ARMAX model, (a) reference tracking, (b) control effort.

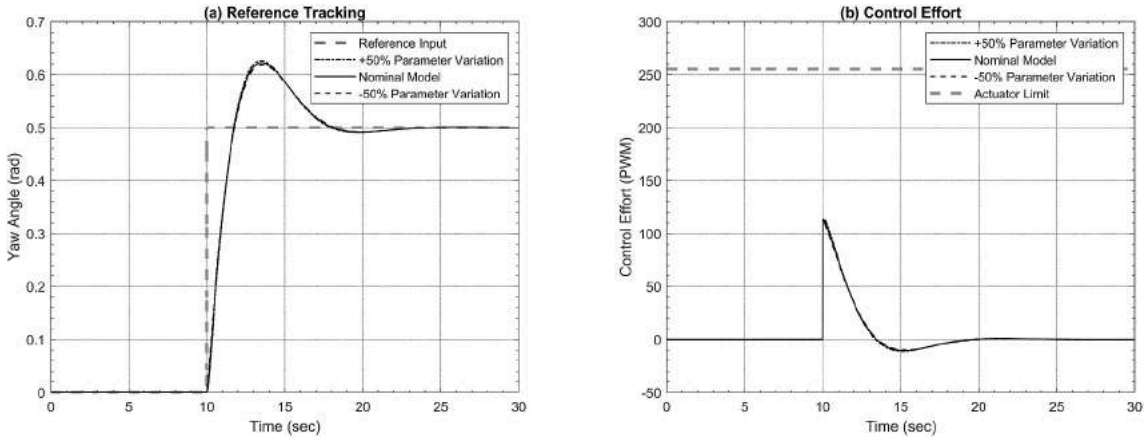

Fig. 15. Yaw step-response for nominal and varied root locus tuned models.

7.3. Real-time implementation

Once the controller is tested in simulation environment, it is implemented onto the vehicle to test its performance in real-time . Implementation of the controller on the micro-ROV in real-time is a servo control problem. Yaw control is achieved by controlling the speed of the side-mounted motor. A control signal is provided to the thrusters by the PI controller. Fig. 16 shows the closed-loop schematic diagram for yaw control. Where $r_y\ (t)$ and $\psi(t)$ represent reference input and output of the system in radians, $e(t)$ and $u(t)$ represent error and control signals, respectively.

At first, gains obtained using the root locus approach were used in the real-time implementation. Then, iteratively gains were adjusted and fine-tuned. Finely, $k_p = 225$ and $k_i = 110$ were selected as they provided the desired response.

A positive step input of 0.5 radians is provided and the output of the vehicle is recorded. The designed controller demonstrated good reference tracking capability, as shown in Fig. 17. Also, the control effort is well within the actuator limit. The vehicle is found to achieve the desired performance requirements with a peak time of 1.34 s, settling time of 4.52 s, and 18% maximum overshoot. A quantitative comparison of simulated and experimental heave controllers is provided in Table 3.

The controller underperforms in simulations due to model uncertainty, however, upon implementation on the vehicle, the controller demonstrates considerable performance improvement. These results show that the r-locus tuned controller delivers on its promise by meeting the requirements in real-time, therefore can serve as a benchmark controller for comparison with the one tuned using AI-based techniques.

8. Genetic algorithm problem setup

The Genetic Algorithm (GA)—a search-based optimization technique based on the Darwinian theory of genetic evolution—has been extensively employed to find optimal solutions to complex problems, such as the one under consideration. The performance of GA, however, is dependent on the selection of an appropriate cost function. As the goal behind every control design is to minimize the error between the reference input and the system's response, therefore, the error function

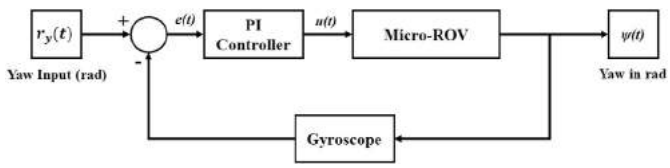

Fig. 16. Closed-loop schematic for yaw control.

is used as a cost function for optimization.

Now for control system performance assessment, three basic types of error are extensively used, (i) Integral of Time-weighted Absolute Error (ITAE) (ii) Integral of the Square of the Error (ISE), and (iii) Integral of Absolute Error (IAE). All three error functions can be represented by mathematical expressions shown below.

$$ITAE = \int_0^\infty t.|e(t)|.dt \tag{29}$$

$$ISE = \int_0^\infty e^2(t).dt \tag{30}$$

$$IAE = \int_0^\infty |e(t)|.dt \tag{31}$$

Where $e(t)$ represents error signal.

All of the three error functions are employed in this study for optimization of PI controller. Proportional and integral gains for the controller are found by minimizing the error function. Fig. 18 demonstrates the sequence of steps involved in setting up genetic algorithm for PI controller optimization. A summary of various parameter used in setting up the problem can be seen in Table 4.

Once setup is complete, the algorithm is executed several times for each cost function, and the results are displayed on a scatter plotter, as shown in Fig. 19. The outcome of GA optimization is the minimal value of the cost function for that specific iteration, which is subsequently converted into proportional and integral gains. From Fig. 19, it can be seen that the intra-cost function values are tightly packed while inter-cost function values are loosely spread. Therefore, selection of a single optimal region is not practical in this case. So, small optimal regions within each cost function spread, having the highest density, are identified. Gains are chosen from each smaller optimal region and simulated on the vehicle's model. The representative gains for each cost function are selected for simulation. From the analysis of the system response, ITAE representative response ($k_p = 260$, and $k_i = 70$) is shortlisted for real-time implementation owing to its fast peak and settling time. The response of the system demonstrated performance within the desired performance range with peak time of 1.9 s, settling time of 4.3 s with maximum overshoot of 18%. Also, controller effort is well within the actuator limit. Therefore, the controller is deemed fit for real-time implementation following robustness analysis.

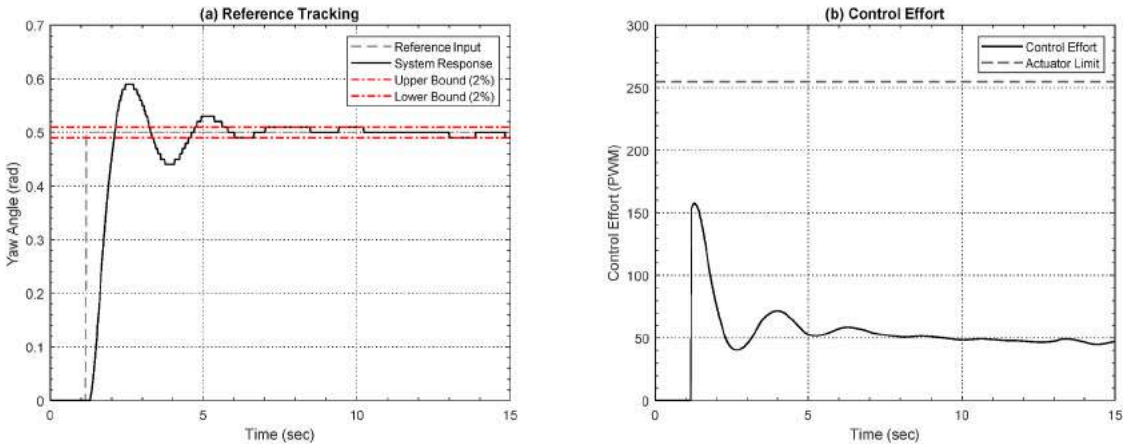

Fig. 17. R-locus tuned controller response to step input, (a) ref. tracking, (b) control effort.

Table 3
A quantitative comparison of simulated and experimental r-locus tuned yaw controller.

Parameters	Simulated Controller (kp = 230, ki = 90)	Experimental Controller (kp = 225, ki = 110)	% Improvement in Experimental Controller
Peak Time	2 s	1.34 s	33%
Settling Time	4.3 s	4.52 s	−5%
% Overshoot	22%	18%	18.2%

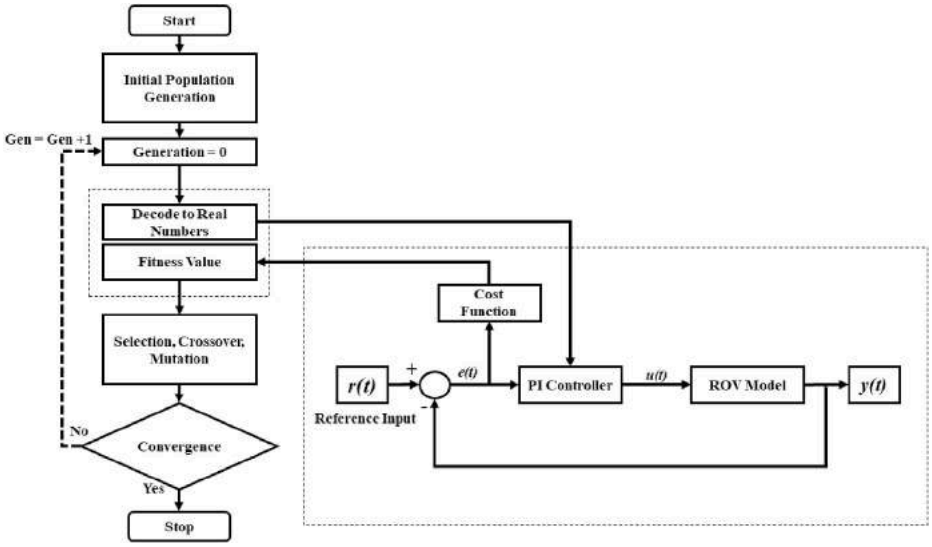

Fig. 18. Flowchart of genetic algorithm.

Table 4
Genetic Algorithm setup parameters.

Parameter	Value
Bounds for kp	50 to 300
Bounds for ki	0 to 200
Convergence function tolerance	10^{-3}
Stall generation	50
Creation Function	Uniform
Selection Function	Stochastic Selection Function
Crossover	Intermediate
Mutation	Adaptive Feasible

9. Robustness analysis of GA optimized controller

9.1.1. Frequency domain analysis

GA optimized closed-loop yaw transfer function is given as:

$$G_{cl}(s) = \frac{3.624s + 0.9758}{s^3 + 2.08s^2 + 4.092s + 0.9758} \tag{32}$$

The gain and phase margin plot for Equation 32, depicted in Fig. 20, show that both the gain and phase margin of GA optimized controller are comfortably above the set requirements. This shows that GA optimized PI controller is robust enough to handle parameter variation.

9.1.2. Time-domain analysis

For robustness analysis in yaw, the system is simulated by varying inertia of the ROV by ± 50% and the response of the system is shown in Fig. 21. As it is clear from Fig. 21 that even with a 50% change in the

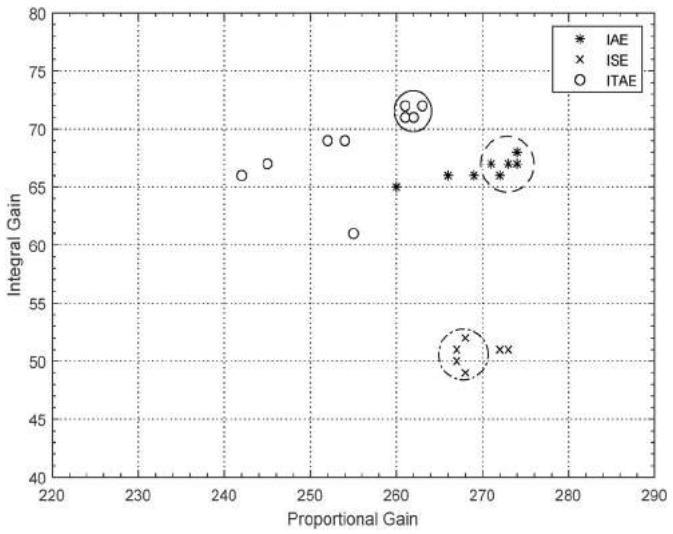

Fig. 19. Yaw PI gains for various cost functions.

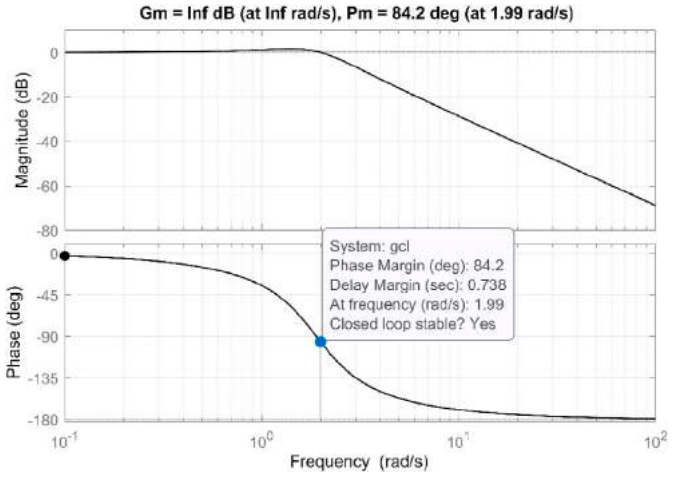

Fig. 20. Bode plot for GA optimized yaw model.

inertia of the ROV the response of the system is exactly the same. Moreover, control effort within the actuator limit for each scenario. Hence, it can be said that the designed PI controller is robust enough to overcome parameter variation.

9.1.3. Real-time implementation

At first, gains obtained from GA optimization ($k_p = 260; k_i = 70$) are used. A positive step input of 0.5 radians was provided and the output of the vehicle was recorded. The designed controller showed a good reference tracking response, as shown in Fig. 22. Also, the control effort was well within the actuator limit. The response of the system shows peak time of 1.27 s, settling time of 3.83 s, and maximum overshoot of 14%. It was found out that the experimental response is in agreement with the one predicted during simulation, and also that the GA optimized controller exhibits performance many-fold better than the root-locus tuned controller. However, as mentioned in the introduction of this paper that the PID gains obtained via root-locus cannot be regarded as optimal while GA, on the other hand, provides optimal gains. Therefore, comparing the performance of a GA tuned controller with a traditional optimal controller like LQR would be more appropriate. In this regard, the yaw motion control of the ROV under discussion is investigated using a LQR controller (Annex A). The response of the LQR yaw controller show effective reference tracking with significantly improved settling time of 1.47 s and maximum overshoot of mere 1.7 percent. However, unlike other controllers under investigation, LQR controller resulted in a slow system response with a time of 1.96 s. Not only that, the LQR controller requires nearly twice as much controller effort as its contemporary optimal algorithm. Other limitations of LQR, apart from those mentioned above, include the need for a state estimator and the cumbersome process of selecting appropriate Q and R matrices. Hence, the design of LQR controller is not pursued further in this work.

10. Marine predator algorithm problem setup

The marine predator algorithm (MPA) is a newly developed optimization method that is centred around the idea that both the predator and prey update their location according to the location of their respective food (Faramarzi et al., 2020). During the interaction amongst marine predators and prey, predators utilize the pervasive scavenging strategy known as the Brownian and Lévy random movement. If the concentration of prey in the hunting area is of high-level, predators use the Brownian method and when the prey concentration is low, they use the Lévy method. However, environmental issues such as eddy formation or fish aggregating devices (FADs) effects are among the factors that change the behaviour of marine predators. In search space, the FADs effects are considered as trapping. FADs are deemed as local optima. According to the survival of the fittest, the member of the population that has the best fitness compared to others is selected as the top member and represented through the elite matrix. Fig. 23 depicts workflow of MPA for PI controller tuning. Parameters such as upper and lower bounds are set as given in Table 5. From the optimization problem, proportional gain of 300 while integral gain of 82 is obtained (for code,

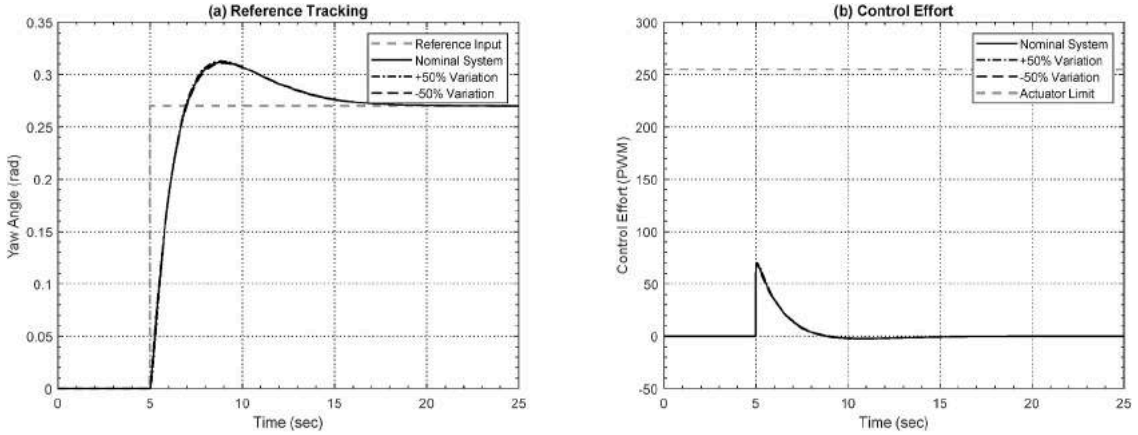

Fig. 21. Yaw step-response for nominal and varied GA optimized models.

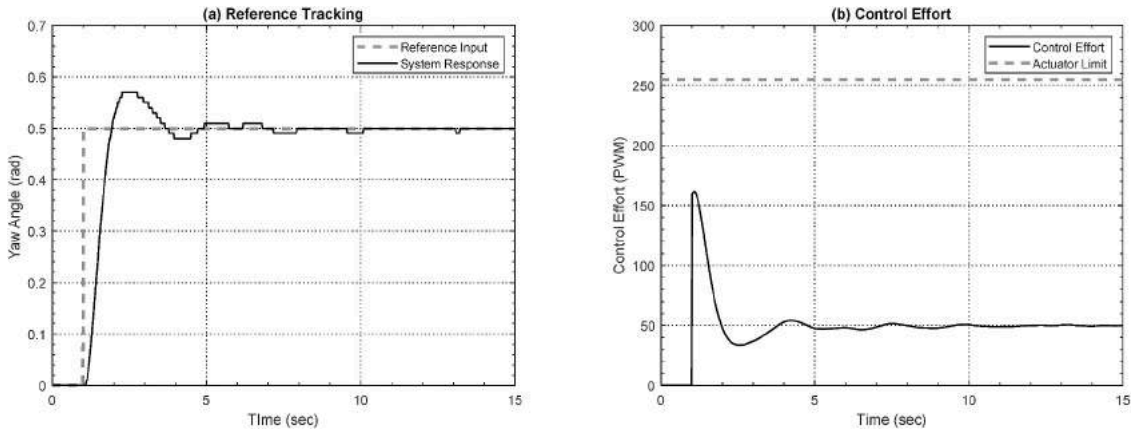

Fig. 22. GA optimized controller response to step input, (a) ref. tracking, (b) control effort.

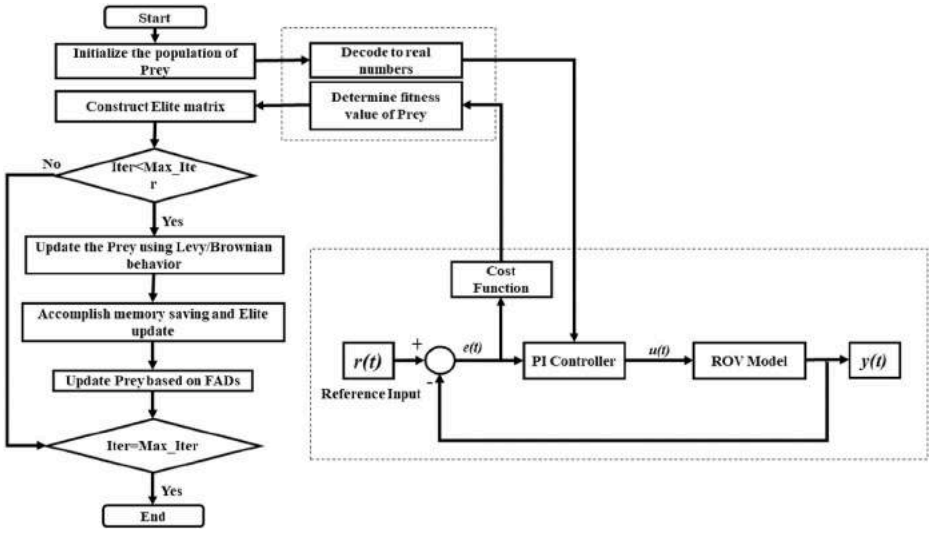

Fig. 23. A simplified workflow of Marine Predators Algorithm (MPA) for PI controller tuning.

see (Faramarzi, 2022)). The response of the MPA optimized controller show effective reference tracking with a slightly improved peak time of 1.73 s and settling time of 4 s. However, unlike other controllers under investigation, MPA results in a maximum overshoot of 22 percent. Additionally, MPA controller requires more control effort than the GA tuned controller. In short, the modest improvement in performance does not justify the high computing time and power required to execute an optimization problem using MPA. Table 6 summarizes the performance of controllers utilized in this investigation in simulation environment.

From the performance analysis in Table 6, root-locus and GA are opted for real-time implementation. A quantitative comparison between real-time performance of the GA optimized and the root-locus tuned yaw controller can be seen in Table 7. The results not only recognise the suitability of the modelling approach employed but also the premise that GA tuned controllers provide significantly superior performance with

Table 5
MPA setup parameters.

Parameter	Value
Upper Bound	300
Lower Bound	50
Number of Agents (Preys)	50
Maximum Iterations	300
Cost Function	ITAE
Number of Dimensions (Variables)	Two

Table 6
Performance comparison of controllers in simulation environment.

	Root-locus	LQR	GA	MPA
T_p	2 s	1.96 s	1.27 s	1.73 s
T_s	4.3 s	1.47 s	3.83 s	4 s
Percentage Overshoot	22%	1.7%	14%	22%

Table 7
Real-time performance comparison table for GA and r-locus tuned yaw controller.

	Root-locus	GA	Percent Improvement
GM	Inf	Inf	0
PM	73.4°	84.2°	14.7
T_p	1.34 s	1.27 s	5.2
T_s	4.52 s	3.83 s	15.3
Percentage Overshoot	18%	14%	22.2

greater parametric robustness.

11. Conclusion

Research interest in underwater vehicles, especially micro-ROVs is rapidly growing. The literature available for such systems is relatively small but growing with time. The interest is rooted in the ROVs capabilities to perform a variety of operations such as surveillance, mapping, inspection and search and rescue. These vehicles, however, present a variety of challenges. In this work, an under-actuated micro-ROV has been investigated. Although the vehicle lacks the structural integrity and robustness of counterpart industrial-grade vehicles, it can be used as a test bed to study control and modelling problems related to underwater vehicles in shallow waters and pools. The micro-ROV used in this research poses daunting challenges of modelling, control design, and real-time implementation. An attempt has been made to address these issues in this work.

Accurate linear system models are essential for controller design and evaluation. In this paper, mathematical and system identification modelling techniques were employed to achieve a reasonable system representation. A linear SISO model was obtained for yaw motion of the micro-ROV with a prior knowledge of the vehicle's physical characteristics. To instil confidence, the developed linear models were thoroughly validated using time and statistical domain tests. The modelling procedure adopted is considered suitable for the majority of ROVs whose parameters are difficult to obtain. Therefore, the presented modelling technique is an important contribution of this paper.

As far as control of the vehicle is concerned, in addition to stability, other performance criteria such as reference tracking, disturbance rejection, fast peak and settling time, and robustness were set. It was discovered from the system response that it is impossible to achieve these objectives by open-loop mechanisms alone, thus study of closed-loop strategy is inevitable. A multi-parameter root-locus as well as LQR controller served as a benchmark against which the performance of a GA optimized controller was evaluated. Also, recently developed meta-heuristic algorithm called marine predator algorithm is scrutinised against GA optimized controller. The failure of the traditional tuning procedure to achieve optimal proportional and integral gains for the controller prompted the need for meta-heuristics. The results showed that the GA optimized controller offers far superior tracking capabilities while requiring far less control effort than any of the other investigated algorithms.

Finally, real-time hardware implementation of both the controllers was carried out. Implementation of the GA optimized controller showed much improved results than its root-locus counterpart. With GA, performance requirements were met in fewer iterations. This is another advantage of the GA strategy. Robustness to parametric variations and disturbance rejection capabilities of both the controller were excellent with GA outperforming root-locus by as much as 5%–22%. Therefore, GA optimized controller has been shown to be an effective choice for the micro-ROV and similar systems.

12. Future work

In this paper SISO controller for yaw has been investigated. An obvious extension is the design and analysis of a MIMO control with inclusion of additional DoFs such as heave and surge, etc.

CRediT authorship contribution statement

Ahsan Tanveer: Conceptualization, Software, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, preparation, Visualization. **Sarvat M. Ahmad:** Conceptualization, Validation, Resources, Writing – review & editing, Supervision, Project administration, All authors have read and agreed to the published version of the manuscript.

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.

Data availability

Data will be made available on request.

Annexure A

h. LQR Controller

For the purpose of applying the control strategy, a simplified high-fidelity model of the ROV with yaw plane dynamics has been developed and is linearized (see equation (18)). The Linear Quadratic Regulator (LQR) is the control approach used to ensure that the system performs at its best under a given set of operating conditions. It is a well-established control strategy that offers useful feedback gain and effectively automates the process of selecting a suitable state feedback controller. However, for the design of an LQR controller for a state-space system $\dot{x}(t) = Ax(t) + Bu(t)$, all the states should be available for the controller.

A state-space representation for the yaw transfer function in equation (18) is given below, where

$$A = \begin{bmatrix} 0 & 1 \\ -0.4681 & -2.08 \end{bmatrix} \tag{i}$$

$$B = \begin{bmatrix} 0 \\ 0.1394 \end{bmatrix} \tag{ii}$$

$$C = [1 \quad 0] \tag{iii}$$

$$D = [0] \tag{iv}$$

The primary goal of LQR design is to obtain the state feedback gain factor K. A cost function (J) is considered where the objective is to minimize the cost function in such a manner that the system is stable. The cost function (J) is given as:

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt \tag{v}$$

Where R matrix offers the weights on the control input in the objective function and Q matrix defines the weights on the states, such that

$$Q, R \geq 0 \tag{vi}$$

The control law is given as

$$U = -Kx(t) \tag{vii}$$

The specified penalization matrices (Q,R) are used by the LQR controller to calculate the gain matrix (K). The selection of the penalization matrices may either be made methodically based on performance indices like the Integral of Absolute Error (IAE), Integral of Time and Absolute Error (ITAE), etc., or it can be done intuitively based on expertise. The design of the LQR controller for the yaw axis of the ROV is covered in the next section.

i. LQR Yaw Controller Design

The design of a full-state feedback LQR controller requires solving an algebraic Riccati equation, which is a time-consuming procedure in and of itself. Therefore, in order to design the LQR controller, MATLAB will be used. Each component of the penalization matrices Q and R is given a weight depending on their relative significance. For instance, the yaw angle is more significant than the yaw rate when it comes to yaw control. Furthermore, in order to guarantee a fast-tracking response with zero steady-state error, the steady-state error state should also be penalised more. In essence, the penalization matrices are chosen and iteratively simulated in Simulink depending on these weighting criteria. The combination of penalization matrixes that provided the desired response is

$$Q = \begin{bmatrix} 240 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{viii}$$

$$R = \begin{bmatrix} \frac{1}{1000} \end{bmatrix} \tag{ix}$$

MATLAB is used to calculate the feedback gains and integral gains of the LQR controller, which are

$$K = [458 \quad 147] \tag{x}$$

$$\overline{N} = [491] \tag{xi}$$

The designed LQR controller is then simulated in the SIMULINK simulation environment. The yaw axis of the ROV exhibits dominant 2nd order system characteristics and lacks an intrinsic integer, which means the open-loop plant will have a particular steady-state error for a step input. Therefore, the LQR based full-state feedback controller should have to augment the plant in such a way that it exhibits no steady-state error. This can be achieved by taking an integral action based on the output error. The integral action will eliminate the steady-state error from the tracking response. The signal flow and block diagram representation is shown Fig. 1.

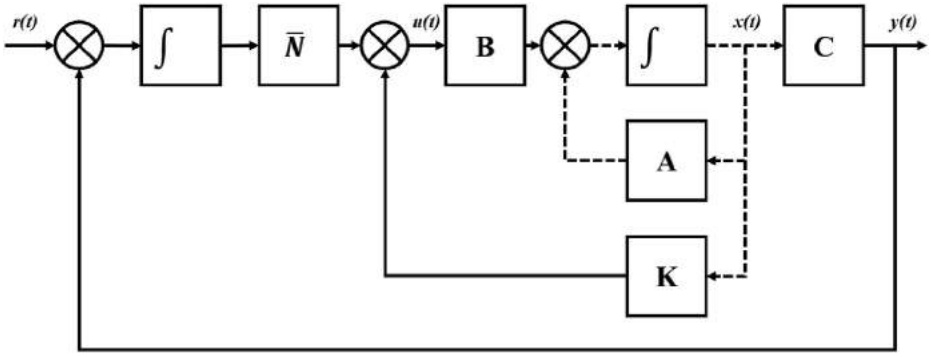

Fig. A1. Block diagram representation of an LQR servo system .

The response of the yaw LQR controller to a step input is shown in Fig. 2. The controller can track the reference input as demonstrated with no steady-state error. Additionally, the output response settles in 1.4 s with barely a 1.7% overshoot, meeting the design criteria.

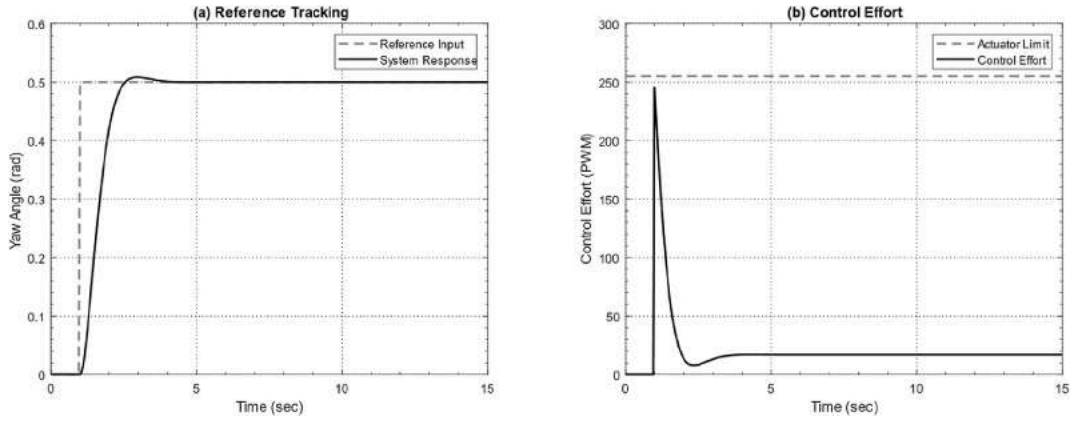

Fig. A2. LQR response to step input, (a) ref. tracking, (b) control effort.

Although, LQR yaw controller can provide a good reference tracking response with zero steady-state error along with adequate disturbance rejection, but the overall response is slow. This is due to the fact the rise time of response is dependent on the integral gain. So, to achieve fast response, the integral gain should be increased drastically. However, this may cause the system to go unstable and even the integral wind-up can lead the system to exhibit undesirable behaviour. Furthermore, LQR's requirement for control effort is fairly close to the saturation limit, which means that any external disruption to the system will cause the thrusters to become saturated. Hence, real-time implementation of LQR is not pursued.

References

Abidin, A.Z., Mardiyanto, R., Purwanto, D., 2016. Implementation of PID controller for hold altitude control in underwater remotely operated vehicle. In: International Seminar on Intelligent Technology and its Applications. ISITIA.

Ahmad, S.M, Sutton, R., 2003. Dynamic modelling of a remotely operated vehicle. In: 1st IFAC Workshop on Guidance and Control of Underwater Vehicles, GCUV'03, Newport, UK, 9-11 April. IFAC, pp. 47–52.

Antonelli, G., Fossen, T., Yoerger, D., 2008. Underwater Robotics. Springer Handbook of Robotics, Berlin.

Arif, E., Hossen, J., Murthy, G.R., Rahaman, M.A., 2018. A novel PID controller based solar panel tracking system. In: Maleka, Malaysia, Proceedings of Symposium on Electrical, Mechatronics and Applied Science.

Bäck, T., 1996. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford university press.

C, C.B., Li, X., 2008. Swarm Intelligence in Optimization. Springer.

Capocci, R., Dooly, G., Omerdic, E., Coleman, J., Newe, T., Toal, D., 2017. Inspection-class remotely operated vehicles—a review. J. Mar. Sci. Eng. 5 (1) https://doi.org/10.3390/jmse5010013.

Chen, W., Wei, Q., Zhang, Y., 2020. Research on anti-interference of ROV based on particle swarm optimization fuzzy PID. In: Chinese Automation Congress. CAC.

Christian, M., Simon, P., Leif, H., Lund, J.K., Zhenyu, Y., 2017. Modeling and control of industrial ROV's for semi-autonomous subsea maintenance. In: IFAC-PapersOnLine.

DE, D.G., Holland, J., 1988. Genetic algorithms and machine learning. Machine Learning volume 3, 95–99, 3, pages95–99 (1988).

Deng, Y., Liu, Y., Zhou, D., 2015. An improved genetic algorithm with initial population strategy for symmetric TSP. Math. Probl Eng. 1–6.

Enevoldsen, T.T., Einarsson, E.M., Pedersen, S., Yang, Z., 2018. Simplified modelling and identification of an inspection ROV. In: 3rd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production OOGP 2018. Esbjerg, Denmark.

Faramarzi, A.. Marine predators algorithm (MPA) - MATLAB central file exchange [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/74578-marine-predators-algorithm-mpa.

Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H., 2020. Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst. Appl. 152 https://doi.org/10.1016/j.eswa.2020.113377.

Folcher, J., Rendas, M., 2001. Identification and control of the Phantom 500 body motion. In: MTS/IEEE Oceans 2001. An Ocean Odyssey.

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

Fossen, T.I., 2014. Mathematical models of ships and underwater vehicles. Encyclopedia of Systems and Control. In: Baillieul, J., Samad, T. (Eds.), Encyclopedia of Systems and Control. Springer, London. https://doi.org/10.1007/978-1-4471-5102-9_121-2.

Goheen, K., Jefferys, E., 1990. The application of alternative modelling techniques to ROY dynamics. In: IEEE International Conference on Robotics and Automation, Ohio.

Hernandez-Alvarado, R., Garcia-Valdovinos, L.G., Salgado-Jimenez, T., Gómez-Espinosa, A., Navarro, F.F., 2016. Self-tuned PID control based on backpropagation Neural Networks for underwater vehicles. In: OCEANS MTS/IEEE Monterey.

Ho, M.-t., Datta, A., Keel, L., Bhattacharyya, S., 1997. Robust and optimal PID controller design. In: IFAC Proceedings Volumes, Budapest.

Jacobi, M., Karimanzira, D., 2013. Underwater Pipeline and Cable Inspection Using Autonomous Underwater Vehicles. MTS/IEEE OCEANS.

Javadi-Moghaddam, J., Bagheri, A., 2010. An adaptive neuro-fuzzy sliding mode based genetic algorithm control system for under water remotely operated vehicle. Expert Syst. Appl.: Int. J. 37 (1).

Jebelli, A., Yagoub, M., 2016. Efficient robot vision system for underwater object tracking. In: 2nd International Conference on Control Science and Systems Engineering. ICCSSE, Singapore.

S. K. Kartal, E. Ege and M. K. Leblebicioğlu, "Optimal autopilot and guidance of the ROV: SAGA," in IFAC-PapersOnLine, Turkey, 206.

Lehtomaki, N.A., Sandell, N.R., Athans, M., 1981. Robustness results in linear-quadratic Gaussian-based multivariable control designs. IEEE Trans. Automat. Control 26, 75–93.

Li, Y., Liu, R., Liu, S., 2017. The design of an autonomous underwater vehicle for water quality monitoring. In: IOP Conference Series: Materials Science and Engineering, Xiamen.

McMillan, G.K., 2012. Industrial applications of PID control. In: PID Control in the Third Millennium. Advances in Industrial Control. Springer, London, pp. 415–461.

Midtgaard, O., Hagen, P.E., Jalving, B., Storkersen, N., 2000. Autonomous underwater vehicle for mine reconnaissance. In: 5th IFAC Conference on Manoeuvring and Control of Marine Craft, Aalborg.

Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M., 2017. Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv. Eng. Software 1–29.

Nise, N.S., 2019. Control Systems Engineering, eighth ed. Wiley.

Noshahri, H., Kharrati, H., 2014. PID controller design for unmanned aerial vehicle using genetic algorithm. In: IEEE 23rd International Symposium on Industrial Electronics. ISIE).

Piotrowski, A.P., Napiorkowski, M.J., Napiorkowski, J.J., Rowinski, P.M., 2017. Swarm Intelligence and Evolutionary Algorithms: Performance versus Speed, vol. 384. Information Sciences, pp. 34–85.

Quan, Q., 2017. Introduction to Multicopter Design and Control. Springer, Singapore.

Rúa, S., Vásquez, R.E., 2016. Development of a low-level control system for the ROV Visor3. International Journal of Navigation and Observation 2016, 12.

Safonov, M.G., Athans, M., 1977. Gain and phase margin for multiloop LQG regulators. IEEE Trans. Automat. Control 22, 173–179.

Smolowitz, R., Patel, S., Haas, H., Miller, S., 2015. Using a remotely operated vehicle (ROV) to observe loggerhead sea turtle (Caretta caretta) behavior on foraging grounds off the mid-Atlantic United States. J. Exp. Mar. Biol. Ecol. 471, 84–91.

Tena, I., 2011. Automating ROV Operations in Aid of the Oil and Gas Offshore Industry. Seebyte.

Tinker, S., Bowman, A., Booth, T., 1979. Identifying submersible dynamics from free model experiments. In: RINA Annual Report and Transaction.

Ura, T., Kurimoto, Y., Kondo, H., Nose, Y., Sakamaki, T., Kuroda, Y., 2005. Observation behavior of an AUV for ship wreck investigation. In: Proceedings of OCEANS 2005. MTS/IEEE.
Vahid, S., Javanmard, K., 2016. Modeling and control of autonomous underwater vehicle (AUV) in heading and depth attitude via PPD controller with state feedback. Internation Journal of Coastal and Offshore Engineering 4, 11–18.
Venkatesan, S., 2016. AUV for Search & Rescue at sea - an innovative approach. In: IEEE/OES Autonomous Underwater Vehicles. AUV.
Walker, K.L., Stokes, A.A., Kiprakis, A., Giorgio-Serchi, F., 2020. Investigating PID control for station keeping ROVs. In: 3rd UK-RAS Conference for PhD Students & Early Career Researchers.
Wang, W., Clark, C., 2006. Modeling and simulation of the VideoRay pro III underwater vehicle. In: Oceans - Asia Pacific.
Wernli, R., 2001. Low cost UUV's for military applications: is the technology ready?. In: Pacific Congress on Marine Science and Technology, San Francisco.
Xu, J., Wang, N., 2018. Optimization of ROV control based on genetic algorithm. In: OCEANS - MTS/IEEE Kobe Techno-Oceans. OTO).
Zanoli, S.M., Conte, G., 2000. Rov depth control. In: IFAC Proceedings Volume.