Quantum Algorithm in Path Optimization for Autonomous Underwater Vehicle: Hardware-in-the-Loop experiment

Kvantni algoritam za optimizaciju putanje autonomnog podvodnog vozila: eksperiment hardver-u-petlji

Viet-Dung Do
University of Transport Ho Chi Minh City
AIT Research Group
Vietnam
E-mail: dungdv@ut.edu.vn

Xuan-Kien Dang*
University of Transport Ho Chi Minh City
AIT Research Group
Vietnam
E-mail: kien.dang@ut.edu.vn

Žarko Koboević
University of Dubrovnik
Croatia
E-mail: zarko.koboevic@unidu.hr

Tien-Dat Tran
University of Transport Ho Chi Minh City
AIT Research Group
Vietnam
E-mail: tiendatck75@ut.edu.vn

Chi-Luan Le
University of Transport Technology
Hanoi, Vietnam
E-mail: luanlc@utt.edu.vn

https://doi.org/10.17818/NM/2026/3.1
UDK 629.58:656.61
510.51
Original scientific paper / *Izvorni znanstveni rad*
Paper received / *Rukopis primljen:* 10. 3. 2025.
Paper accepted / *Rukopis prihvaćen:* 21. 5. 2026.

This work is licensed under a Creative Commons Attribution 4.0 International License.

Abstract

Autonomous Underwater Vehicles (AUVs) are often used in survey, exploration, and monitoring missions of the maritime environment. Due to operating in complex environments, planning an optimal path to avoid obstacles is crucial and directly affects operational efficiency and safety. This study proposes a Quantum Algorithm (QA) for planning an AUV optimal path. The QA is based on determining the optimal path by computing the quantum nodes in a Three-Dimensional (3D) space with obstacles. The proposed algorithm computes optimal waypoints, which are assessed by safety requirements established at the stage. The results have a fast-processing time and are effective under complex operating conditions. The feasible results, compared with other pathfinding algorithms, have demonstrated the effectiveness of the proposed solution. In addition, this study develops experiments using a Hardware-in-the-Loop (HIL) model to evaluate the efficiency of the proposed solution under operating conditions that closely resemble actual conditions.

KEY WORDS

Autonomous underwater vehicles
hardware-in-the-loop
path planning
obstacle avoidance
quantum algorithm

Sažetak

Autonomna podvodna vozila (AUV) često se koriste u istraživačkim, izviđačkim i nadzornim misijama u morskom okruženju. Zbog složenosti radnog prostora, planiranje optimalne putanje koja izbjegava prepreke ključno je za operativnu učinkovitost i sigurnost. Ovo istraživanje predlaže kvantni algoritam (QA) za planiranje optimalne putanje AUV-a. Algoritam se temelji na izračunu kvantnih čvorova u trodimenzionalnom (3D) prostoru s preprekama, čime se određuje optimalna putanja. Predloženi algoritam izračunava optimalne međutočke koje se vrednuju prema sigurnosnim zahtjevima definiranima u toj fazi. Rezultati pokazuju kratko vrijeme obrade i učinkovitost u složenim radnim uvjetima. Dobiveni izvedivi rezultati, u usporedbi s drugim algoritmima za pronalaženje putanje, potvrđuju učinkovitost predloženog rješenja. Razvijeni su eksperimenti primjenom modela hardver-u-petlji (HIL) kako bi se procijenila učinkovitost rješenja u uvjetima koji uvelike sliče stvarnima.

KLJUČNE RIJEČI

autonomna podvodna vozila
hardver-u-petlji
planiranje putanje
izbjegavanje prepreka
kvantni algoritam

1. INTRODUCTION / *Uvod*

Nowadays, many applications of the AUVs are being developed in the fields of information exploitation, ocean research, and maritime environmental monitoring. The AUVs deployed for these tasks guarantee high efficiency and safety for vehicles in complex underwater environments. Therefore, the underwater work of AUVs always faces complex working environments. Dangerous obstacles, such as optical cables and reefs, significantly influence the planning of underwater routes. Consequently, to ensure reliable and collision-free navigation, research efforts have increasingly focused on upgrading and enhancing automated control algorithms for AUV operations, leading to substantial attention and rapid development in this field [1]. Recent studies on path planning focus on developing optimal algorithms that plan the optimal paths, guaranteeing the safety of AUVs [2]. In practical conditions, maintaining feasible and stable motion under actuator constraints is essential. Constraints control methods address this need by enhancing operational safety [3]. In addition, AIS data processing algorithms have been investigated to provide collision warnings in high-risk operational scenarios [4].

The AUVs' path planning has been a focus of many recent studies [5-9]. In particular, from developing basic algorithms, focusing on finding the shortest path according to the distance in a 2D space such as A, and A* [6], [10], for building the RRT solutions to find paths under limited conditions [7], [8], or combining RRT and A* [9], [11] to bring more efficiency in finding the safe paths in complex environmental conditions [12]. In addition, the studies consider the characteristics of the underwater environment, including changes in current, complex terrain, and computational limits on underwater vehicles. The main aspects and limitations of path-finding algorithms for underwater vehicles are summarized in Table 1.

* Corresponding author

Table 1 Comparison of the path planning solutions
Tablica 1. Usporedba rješenja za planiranje putanje

Solutions	Aspects	Advantages	Limitations
A* [6-10]	Finding the shortest path Collision avoidance	Theoretical basis Optimization by resolution	Not searching in 3D space
RRT [7], RRT* [8]	Exploring space Integration with AUV dynamics	Complex spaces Flexible	Not efficient for obstacles
Hybrid RRT [9-11]	Shorter path Considering kinematic constraints	Improved accuracy Collision avoidance	Affected by the original limitations of algorithms
FNN [13-14]	Combining the neural networks and the fuzzy logic	Adapting to nonlinear environments	Requiring high quality data Real-time computation
DA [15]	Discrete graph-based path search	Global optimality	Not searching in 3D space
FMA [16]	Continuous wave front propagation	Smooth distance	Not searching in 3D space

The A, A*, and hybrid algorithms with RRT provide many feasible results in finding paths of underwater vehicles. However, these studies still reveal limitations in determining the optimal path in 3D environments with many obstacles. In this study, the authors propose the QA to analyze and select the optimal path for an AUV in 3D environments with obstacles. The QA, based on quantum principles and the interaction of Q-bits, enables rapid surveying and accurate computation of the paths in complex environments. Additionally, the constraint function incorporates the path length with safety conditions, ensuring that the AUV maintains stable operation and avoids collisions [17]. In addition to classical methods such as A* or RRT, several studies have applied graph-based optimization techniques to plan the AUV's path. In particular, the Dijkstra Algorithm (DA) is capable of determining a globally optimal path. However, in a 3D environment, the number of nodes increases and remains discrete, which makes the computation more complex and limits its effectiveness when the environment contains obstacles with other shapes [15]. To address this limitation, the Fast-marching Algorithm (FMA) [16] has been developed for improving path planning in continuous space. The suggested solution is formulated by using the Eikonal equation [18], which allows smoother paths in environments with multiple obstacles. However, in 3D scenarios, both algorithms show reduced performance as the search space expands, increasing computational load and often requiring integration with other methods to ensure that the AUV follows an optimal path [19].

The main contributions are summarized as follows: *i)* The authors designed an optimal path planning framework for AUVs based on the QA, which allows for efficient handling of complex constraints in 3D maritime environments and overcomes the limitations of classical algorithms that mainly operate in 2D space; *ii)* We develop the HIL reality environment, which integrates terrain, obstacle, and current factors to visualize the path and verify the feasibility of QA in near-actual conditions; *iii)* Simulation and HIL validation experiments demonstrated that the QA provides better results than other algorithms such as DA, FMA which are reflected in shortening the path distance, reducing the steady-state error, improving computational efficiency, and avoiding collisions in real-time.

2. PROBLEM STATEMENT / *Postavljanje problema*

2.1. The AUV kinematic model / *Kinematički model AUV-a*

The AUV motion is considered in two coordinate systems: the Earth-fixed system and the AUV-body system. The body coordinate system helps determine the velocity and direction of an AUV's motion. The AUV motion in the two coordinate systems is described by Fig. 1 [20], which operates with six degrees of freedom. Therein, $v = [u, v, w, p, q, r]^T$ denotes the linear and angular velocity in the body coordinate system, and $J(\eta)$ is the rotation matrix. The AUV's position $\eta_1 = [x, y, z]^T$ and orientation $\eta_2 = [\varphi, \partial, \psi]^T$ are given by [21]-[22]

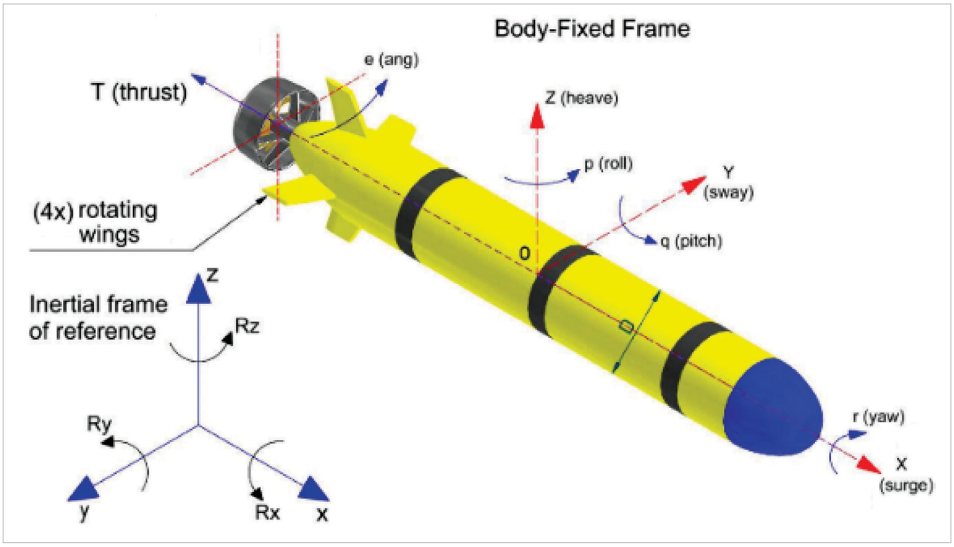


Figure 1 Inertial frame and body-fixed frame of AUV
Slika 1. Inercijski koordinatni sustav i koordinatni sustav vezan uz AUV

$$\eta = J(\eta)v \tag{1}$$

$$J(\eta) = \begin{bmatrix} \eta_1 & 0_{3\times3} \\ 0_{3\times3} & \eta_1 \end{bmatrix} \tag{2}$$

The AUV kinematic model with environmental disturbance is defined as follows [23]:

$$M\dot{v} + Cv + Dv + g(\eta) = \tau + \tau_{en} \tag{3}$$

where M denotes the inertia matrix with the Coriolis matrix C. The matrix D denotes the hydrodynamic damping terms. Additionally, the vector $g(\eta)$ indicates the gravitational and propulsive force.

The matrix $\tau = [\tau_X, \tau_Y, \tau_Z, \tau_K, \tau_M, \tau_N]$ reflects the input forces and moments that account for the various influences acting on the AUV, including contributions from the rudder, fins, and propeller. The environmental disturbance represented by τ_{en} which affects AUV motion. In deep-sea operational conditions, the primary influencing factor is the current, which is modelled as [24]

$$\tau_{en} = [u_c^E, v_c^E, w_c^E]^T \tag{4}$$

$$\begin{aligned} u_c^E &= V_c \cos(\alpha_c)\cos(\beta_c) \\ v_c^E &= V_c \sin(\beta_c) \\ w_c^E &= V_c \sin(\alpha_c)\cos(\beta_c) \end{aligned} \tag{5}$$

for V_c is the flow velocity, α_c represents the deflection angle, and β_c is the slip angle describing the direction of the flow velocity vector.

Remark 1: In the subsea environment influenced by currents and uneven terrain, the AUV's path is essential to prevent collisions with obstacles.

2.2. Obstacle model / *Model prepreka*

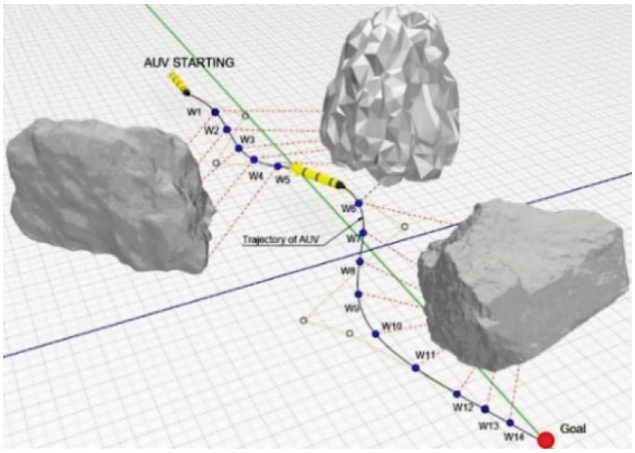


Figure 2 Desired AUV path for avoiding obstacles
Slika 2. Željena putanja AUV-a za izbjegavanje prepreka

In the scenario of an AUV navigating in the presence of obstacles, a critical task is to determine the optimal path that avoids any collisions. The set of waypoints is denoted as $W = (w_1,...,w_i,...,w_l)$, where each point $w_i \in R^3$ represents the positions along the path from the starting point to the endpoint (illustrated in Fig. 2). The depth is constrained between the surface and the maximum allowable depth, z_{max}, with the condition that $(x, y)>0$. Consequently, the obstacle space O_{obs} is given by [25]

$$O_{obs} \triangleq \{\boldsymbol{p} \in O_{space}, v(p) \cap \mathcal{O} \neq \emptyset\} \tag{6}$$

The configuration space can be expressed as [26]

$$O_{space} = \{\boldsymbol{p} \in \mathbb{R}^3 | 0 \leq z \leq z_{max}(x, y)\} \tag{7}$$

Let $v(p) \in R^3$ represent the space occupied by the AUV, whereas $\mathcal{O} \in R^3$ denotes the obstacle space. The free space, denoted as O_{free}, is defined as the complement of O_{obs}:O_{free}= $O_{space} \backslash O_{obs}$ [27]. Thus, O_{free} represents the area within O_{space} that is not occupied by obstacles, allowing the AUV to move safely. The goal is to continuously update a function h: $[0,1]\times[\![0, k-1]\!]\rightarrow X$, where X is a connected space, such that for $\forall s \in [0,1]$, and $\forall i \in [\![0, k-1]\!]$, the conditions are satisfied as

$$\begin{aligned} &h(s, i) = o_i(s) \\ &o_i: [0,1] \rightarrow O_{free} \\ &o_i(0) = \begin{cases} p, i = 0 \\ \omega_i, else \end{cases} \\ &o_i(1) = \omega_{i+1} \end{aligned} \tag{8}$$

Remark 2: The AUV motion can pass through multiple waypoints w_i while adhering to the O_{free} space constraint along various feasible paths in the 3D space. As a result, it is essential to determine the optimal path for the AUV's movement to ensure it safely avoids collisions with obstacles.

3. METHODOLOGY / *Metodologija*

3.1 Optimizing AUV path based on the Quantum-Inspired Optimization Algorithm / *Optimizacija putanje AUV-a temeljena na kvantno-inspiriranom optimizacijskom algoritmu*

The proposed QA solution aims to optimize the AUV's path in a 3D environment with obstacles. The development process is divided into two phases: establishing a simulation environment and performing path planning based on a quantum model, followed by verification using a HIL model. The overall process, illustrated in Fig. 3, demonstrates the linkage between simulation, QA algorithm, and system performance evaluation. In this study, the QA algorithm is defined as a quantum-inspired optimization approach. The methodological basis of the proposed algorithm is developed by adapting and integrating concepts from quantum-inspired optimization (such as Q-bit representation, the Quadratic Unconstrained Binary Optimization (QUBO) model, Hamiltonian mapping, cost Hamiltonian, mixing Hamiltonian, and bit-string measurement) to solve the deviation problem of the AUV path in an obstacle environment. The optimization process is performed in sequential steps, including discretizing the operating space, encoding waypoints with binary variables, constructing the QUBO function, mapping the cost function into the Hamiltonian representation, and decoding the obtained bit strings into feasible paths for the AUV.

Phase 1: Developing a 3D simulation environment for an AUV, where obstacles are strategically arranged within a defined space to compute the safe movement areas. The QA algorithm is employed to build the search space and determine an optimal waypoint list based on the cost function (8) and the QUBO function [28]. Quantum states are encoded by using Q-bits (13) and evolve according to the Hamiltonian, which incorporates two Hamiltonians, H_c and H_B. This approach aims to identify the optimal path, guaranteeing collision avoidance while defining the shortest feasible path.

Phase 2: Building the HIL model to verify the effectiveness of the QA algorithm. The HIL model's operating data and cost weights are continuously updated. In addition, the QA implements the quantum superposition $\psi(\gamma,\beta)$ (21), which is applied in bit-string decoding to compute the cost function $C(X)$ (15). The iteration process stops when the convergence condition (12) is achieved, thereby obtaining the optimal bit-string corresponding to the best waypoint for the AUV path. Furthermore, this study compares the proposed QA solution with other pathfinding algorithms (such as Dijkstra's algorithm

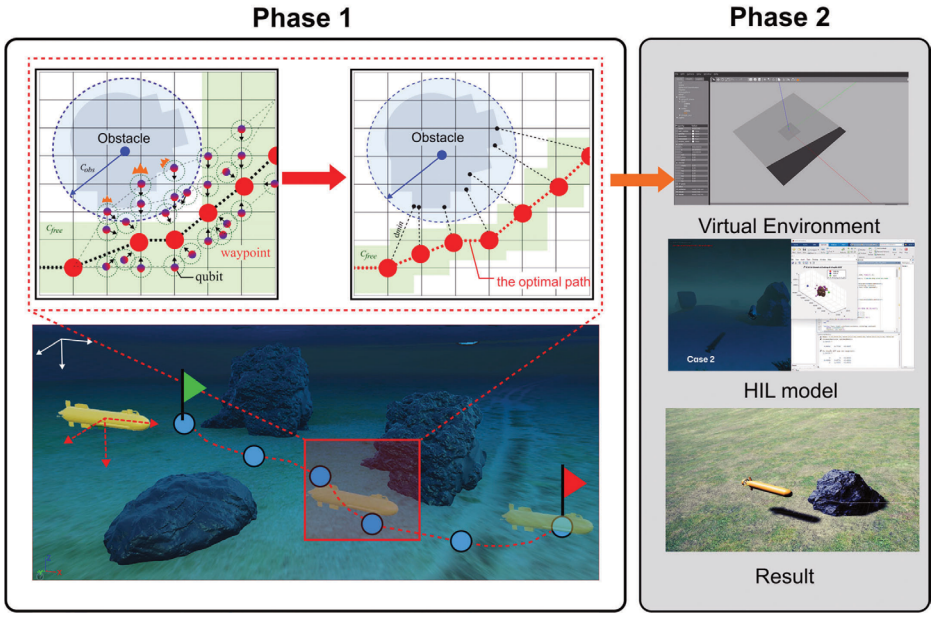


Figure 3 Overview of AUV optimal path using QA

Slika 3. Pregled optimalne putanje AUV-a korištenjem kvantnim algoritmom (QA)

and Fast Marching) to confirm the effectiveness of the proposed solution in obstacle avoidance.

In the simulation phase, the 3D operating space for the AUV is discretized into a set of candidate waypoints, and the waypoint selection process is then encoded using binary variables. Based on this representation, a path planning is formulated as a QUBO model, in which the objective function combines path length, waypoint selection constraints, start and target conditions, and obstacle-avoidance penalty terms. The QUBO model is mapped into the Hamiltonian representation, where low-energy states correspond to paths with lower cost values. In the HIL phase, the bit strings obtained after measurement are decoded into waypoint sequences, and the path with the lowest cost while satisfying the safety constraints is selected as the optimal path for the AUV. To determine the optimal path for an AUV, it is essential to minimize the total path length while adhering to safety constraints that prevent collisions with obstacles. Let's consider the AUV navigating through a series of discrete points in 3D space. The distance between two consecutive points, P_i and P_j, is computed as [29]

$$d_{(i,j)} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{9}$$

Let the set of waypoints in space be denoted by $P = [P_1, \dots, P_n]$. The order in which the AUV moves through these points is represented by a permutation $p = [p_1, \dots, p_n]$. The objective of finding the permutation p is to minimize L, while adhering to the obstacle avoidance conditions in the 3D environment, while transitioning from the starting point to the endpoint. Thus, the total path length is defined as

$$L = \sum_{i=1}^{n-1} d_{p_i,p_{i+1}} \tag{10}$$

The QA identifies the optimal solution through population evolution, where each element is represented as a quantum state. This representation enables parallel simulation of possible outcomes, assisting in computation of the best path for the AUV. The population consists of elements, with each element corresponding to a different feasible path within a d-dimensional multi-solution space. The position of the ith element at the current iteration is given by [30]

$$X_i = [x_{i1}, x_{i2}, \dots, x_{id}] \tag{11}$$

During the process of finding the optimal path, each element stores its best value, referred to as p_i^{best}. Thus, the set of optimized elements derives the global optimal value, g^{best} [31].

$$g^{best} = \frac{1}{N}\sum_{i=1}^{N} p_i^{best} \tag{12}$$

The global optimum value shows how the group is moving toward the best path for the AUV. To update the position of particle i in the next step, we look at the distance between the local best solution and the global solution, as described by

$$X_i^{t+1} = p_i^{best} + \alpha rand(0,1)(g^{best} - X_i^t) \tag{13}$$

where α represents the convergence speed adjustment factor, and $rand(0,1)$ is a random variable that is uniformly distributed in the interval (0,1). The execution of the QA will stop when either the maximum number of iterations, $iter_{max}$, is reached, or the change in the cost function between two consecutive steps is smaller than the threshold ε.

$$|L^{t+1} - L^t| < \varepsilon \tag{14}$$

This condition ensures that the convergence process is stable and stops at the solution with the smallest difference. In QAs, the solutions to the path optimization problem are encoded using Q-bit, allowing each decision variable to exist in a superposition of $|0\rangle$ and $|1\rangle$. Here, α and β represent complex probability amplitudes, and a Q-bit can be expressed as follows [32]:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$
$$|\alpha|^2 + |\beta|^2 = 1 \tag{15}$$

The solution values are shown in QUBO, which uses binary variables. This approach aims to improve the variety of the search by creating multiple binary solutions during the measurement process. It covers a wider range of options than traditional coding methods. The binary variable $x_{i,k}$ is set to 1 when waypoint i is chosen at position k, and 0 if it is not. The path of the AUV is represented by the matrix X as

$$X = [x_{i,k}]_{n\times m} \tag{16}$$

The total cost function is defined by [33]

$$C(X) = w_1 f_1(X) + w_2 f_2(X) + w_3 f_3(X) + w_4 f_4(X) \quad (17)$$

therein f_1 and f_2 represent one-hot constraints applied to rows and columns, respectively. f_3 accounts for the cost associated with relocating between waypoints, while f_4 sets the starting and ending conditions. The weights w_1, w_2, w_3, and w_4 are selected to balance the cost function with the constraints for determining the optimal path.

The objective function is designed to combine both path optimality and path feasibility. The one-hot constraints ensure a valid structure of the waypoint sequence, the relocation cost guides the algorithm toward paths with shorter total distances, and the start and end conditions ensure that the generated path is consistent with the AUV navigation task. The corresponding weights are used to balance the path-length minimization objective with satisfying the constraint requirements. From this objective function, the QUBO model is mapped into the Hamiltonian representation to support the search for the lowest-energy state. In addition, Z_i is the Pauli-Z operator acting on the ith Q-bit, while h_i and J_{ij} are linear and quadratic coefficients. The lowest energy state of H corresponds to the optimal solution of the problem. Thus, the cost function in QUBO [34] is mapped to the Hamiltonian [35]-[36] to represent it in quantum form as

$$H = \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j \quad (18)$$

Binary variables are mapped as follows:

$$x_i = 1 - Z_i / 2 \quad (19)$$

The elements within a QUBO problem are transformed into combinations of Z_i and $Z_i Z_j$. This transformation incorporates one-hot constraints and the relocation cost (17), which are represented as connections between qubits in quantum space. The cost Hamiltonian H_C guides the quantum system towards low-energy states, indicating scenarios in which the optimal path has been identified. Additionally, a mixing Hamiltonian H_B is introduced to facilitate movement between qubit states, thereby allowing the exploration of alternative paths. The dual Hamiltonian approach enables the quantum annealing algorithm to effectively balance the exploitation of the Q-bit structure with the expansion of the search space [37], ultimately enhancing its capability to identify the global optimal path.

$$H_B = \sum_i X_i \quad (20)$$

where X_i represents a Pauli-X operator, enabling bitwise inversions and facilitating the exploration of the solution space. Thus, the two values H_C and H_B are applied sequentially during the quantum evolution, resulting in an overall Q-bit state described by

$$|\psi(\gamma,\beta)\rangle = U_B(\beta)U_C(\gamma)|s\rangle \quad (21)$$

for $U_C(\gamma) = e^{-i\gamma Hc}$, $U_B(\beta) = e^{-i\beta HB}$ and $|s\rangle$ express the initial superposition state. The parameters (γ,β) are adjusted to minimize the expected value of the system energy as [38]

$$\min_{\gamma,\beta} \langle\psi(\gamma,\beta)|H|\psi(\gamma,\beta)\rangle \quad (22)$$

After the optimization process, the resulting quantum states are mapped to bit strings that correspond to feasible path. The state with the lowest energy is then selected as the optimal path for the AUV. The QA allows us to represent the AUV path planning problem as a QUBO model and to solve the optimization problem using quantum mechanisms. The structure comprises two Hamiltonians, H_C and H_B, which help satisfy the path constraint while minimizing the total path length. The proposed method is implemented in a HIL environment to verify its practical performance and evaluate the efficiency of the algorithm, which will be detailed in the next subsection.

3.2. Building the AUV virtual reality model / *Izgradnja AUV modela u virtualnoj stvarnosti*

The AUV experimental model is implemented using a HIL application to evaluate the proposed solution within a virtual reality environment. Specifically, the control algorithm is developed in the MATLAB computing environment to process sensor data and compute control values that interact with the HIL model. To enhance visualization, this study employs the Unreal Engine software to create a 3D sea environment for the AUV's real-time movement. The ARM Cortex M7 processor is used as a classical embedded processor to execute the C/C++ code of the quantum-inspired optimization algorithm under real-time conditions. This microprocessor kit was selected because it is suitable for embedded control applications, supports floating-point computation, and enables deterministic real-time execution. The ARM Cortex M7 kit is compatible with code generation in the MATLAB computing environment. Thus, the role of the processing kit is to verify the feasibility of implementing the algorithm modeled after quantum optimization mechanisms in the HIL model. The structure of the HIL model for the AUV test is illustrated in Fig. 4.

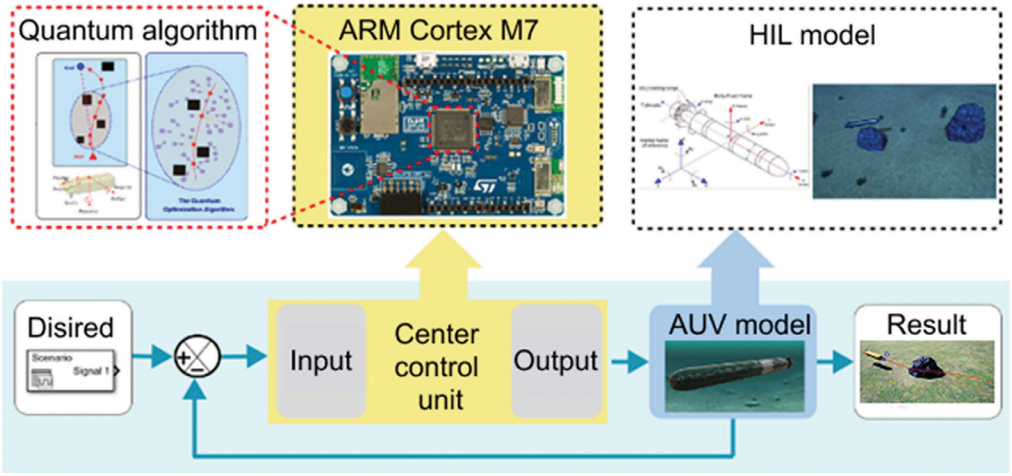


Figure 4 The HIL structure in virtual reality model

Slika 4. Struktura HIL sustava u modelu virtualne stvarnosti

Algorithm 1 Implementing the QA for HIL model	
Input: Candidate waypoints ← $P = [P_1, \ldots, P_n]$, HIL parameter ← T_S. Quantization layers ← p, Population size ← N, Convergence threshold ← ε, Iteration ← $iter_{max}$.	
Output: Optimal waypoint sequence p^*, Corresponding paths	
1	Computing the distance matrix: d_{ij} ←(9)
2	Evaluating the population evaluation : X_i: $C(X_i)$ ⟵(17)
3	Identifying p_i^{best} and g^{best} ⟵ (12)
4	Encoding configuration ← X = $[x_{ij}]$
5	Building H_C, H_B, and quantum layers: p ⟵ (16) to (18)
6	**for** $t = 1 \rightarrow iter_{max}$ **do**
7	Updating HIL data and cost weights
8	Applying Quantum superposition $\psi(\gamma,\beta)\rangle$ ← (21)
9	Measuring Quantum circuit ← Bit-string S
10	Decoding C(X_S)
11	Updating (γ,β), p_i^{best}, g^{best} ← (13)
12	Checking $\lvert L^t - L^{t-1} \rvert < \varepsilon$ → Stop if converge
13	**end for**
14	Updating the best bit-string S^*
15	Decoding the waypoint sequence p^*
16	Exporting the output optimal path (p^x, L)
17	**end**

The hardware implementation is achieved by converting the QA into C/C^{++} using Simulink Coder and Embedded Coder, and then loading the proposed algorithm onto the STM32 microcontroller to perform system control. The HIL operation process (expressed in Algorithm 1) is executed in the following steps:

- *Step 1:* Establishing the virtual reality model environment and its parameters, while developing and transforming the control algorithm. The QA algorithm is constructed on the MATLAB/Simulink platform to enhance path planning. Subsequently, the algorithm is converted into C/C^{++} code using Simulink Coder and Embedded Coder for execution on embedded hardware.
- *Step 2:* Converting the source code for the ARM Cortex M7 microcontroller to enable real-time execution of the QA algorithm. The microcontroller receives input signals from a computational model created in MATLAB, performs the necessary analysis, and then outputs control signals to the HIL model.
- *Step 3:* Transmitting the control signals from the processing kit to the AUV model within a virtual reality environment. This model is developed using Unreal Engine, featuring a 3D marine setting. The responses of the test model are visualized to enable an accurate evaluation of the proposed solution's performance.

4. RESULTS AND DISCUSSIONS / *Rezultati i rasprava*

4.1. Configuration parameters / *Konfiguracijski parametri*

The proposed solution is implemented using MATLAB R2024a on a computer equipped with an Intel Core i7-4800MQ processor and 16GB of RAM. The simulation space is designed with dimensions of 500 m × 200 m × 120 m. Obstacle configurations within the space are represented as polyspherical shapes to provide a clear visualization of the reef conditions on the seabed. The parameters utilized during the testing phase are summarized in Table 2.

Table 2 Parameters applied for QA training
Tablica 2. Parametri primijenjeni za treniranje QA (kvantnog algoritma)

Parameters		Value	Unit
AUV initial position	Initial position	[0, 0, 0]	m
	Initial angle	[0, 0, 0]	rad
	Target location	[270, 180, 0]	m
AUV Movement	Speed	2	m/s
	Length L_0	3	m
	Turning radius	2	m
QA	Number of seeds	$num_{particles}$	60-120
	Landmark parameters	W	8-65
	Maximum of iterations	$iter_{max}$	600-800
	Time step	0.1	s

During the simulation and testing of the QA algorithm for the AUV, technical parameters were established to ensure the model's accuracy and feasibility. The AUV's initial position was defined at the coordinates [0 m, 0 m, 0 m], with an initial orientation angle of [0 deg, 0 deg, 0 deg]. The moving target for the AUV was positioned at [270 m, 180 m, 0 m], maintaining a speed of 2 m/s. The AUV's specifications include a body length of 3 m and a radius of 2 m, thereby reflecting its navigation capabilities within a 3D environment. The QA algorithm is configured with a particle count ranging from 60 to 120, and the landmark parameter W is set between 8 and 65. The maximum iteration number, itermax, is established within the range of 600 to 800. To ensure high resolution in state updates and sensor data processing, the simulation time step is fixed at 0.1 s. The selection of these parameters not only aids in accurately simulating reality but also enhances the evaluation of performance in complex navigation environments, which include both moving and static objects.

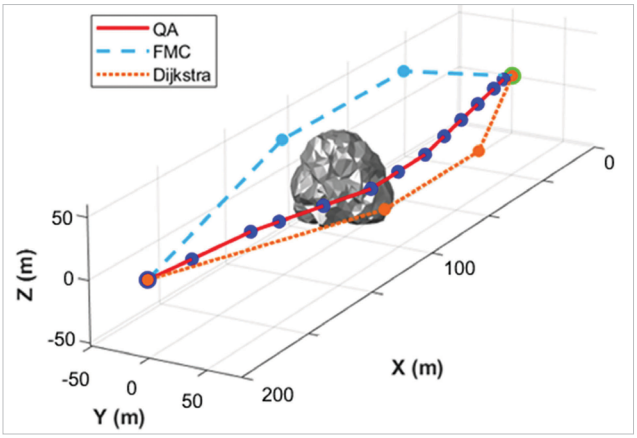
4.2. Simulation cases / *Simulacijski slučajevi*

In this study, the path-optimization algorithms are implemented on the MATLAB environment. To assess the effectiveness of the proposed solution, the results were compared with FMA and DA, which use propagation models to identify optimal paths under complex conditions. All algorithms are initialized under identical conditions in a 3D environment to assess the AUV's navigation performance in obstacle-laden settings. In the simulation cases, the randomly shaped obstacles are initialized to reflect natural seabed conditions such as coral reefs. In the first simulation case, the AUV begins at the origin O [0 m, 0 m, 0 m], then moves towards a destination point A [250 m, 100 m, 100 m]. The arrangement of obstacles forces the algorithms to optimize the navigation to avoid collisions during path planning.

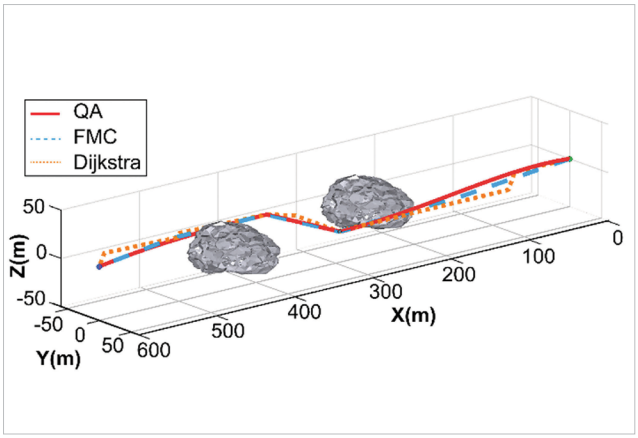
The simulation results for Case 1 are shown in Fig. 5a and Table 3. The QA algorithm provides the shortest path of 285.4 m, thus proving it is effective at optimizing routes. The FMA comes next, with a path of 298.7 m, which is still acceptable but longer than QA. The DA's results in the longest path at 342.1 m, highlighting that it is not optimal for this grid layout. In terms of speed, QA finishes the move in 241 s, which is much faster than FMA at 310 s and DA at 281 s. In the number of iterations for finding a path, the FMA only takes 10 due to its smart search feature. However, the QA requires 150 iterations to reach the best solution, and the Dijkstra takes up to 500 iterations because of its random approach. QA keeps an average safe distance of 12.3 m, ensuring high safety. FMA achieves 8.7 m, and DA only gets 5.1 m, which could bring it too close to obstacles. These results show that QA is the best choice for path length, moving time, and safety, even though it has a higher cost function value of 15, thereby satisfying the conditions stated in Remark 1. The FMA works well in static environments with known conditions, while DA is better for navigating AUVs in larger areas.

In the second case, the author sets up a 3D environment with terrain containing two obstacles. The AUV's planned path from A [0 m, 0 m, 0 m] to B [600 m, 0 m, 0 m] passes through two spherical obstacles simulating natural terrain, such as reefs or coral reefs, creating obstacles in path planning. During the movement, the AUV follows the path planned by the comparison solutions to avoid collisions in a limited space.

The comparison results are summarized in Table 4, which shows that the QA algorithm achieves the best performance among the compared solutions. In terms of path length, the result with QA is 602 m, the shortest among the three algorithms, while FMA and DA are 614 m and 625 m, respectively. The moving times clearly indicate the differences between the solutions: the QA completed the process in 268 s, which is faster than the FMA of 74 s and the DA of 37 s. In terms of iterations, the FMA only required 12 iterations thanks to its method of solving the Eikonal equation. Furthermore, the QA algorithm needs 180 iterations to converge, while DA algorithm requires 620 iterations because it involves traversing the entire graph. Despite needing a higher number of iterations, the QA still demonstrated strong performance, thus meeting the requirements specified in *Remark 2*. Regarding safety distance from obstacles, QA achieved an average of 10.9 m, which is significantly higher than FMA's 7.4 m and DA's 4.3 m. The visual results (expressed in Fig. 5b) indicate that QA is highly reliable in environments with collision risks, making it suitable for applications such as autonomous vehicles. The proposed QA solution continues to be tested in the HIL model to shorten the gap between simulation and implementation on actual models.



a. AUV path in a single-obstacle environment / *Putanja AUV-a u okruženju s jednom preprekom*



b. Path in two obstacles environment / *Putanja u okruženju s dvije prepreke*

Figure 5 Comparison of AUV path planning results in obstacle environment
Slika 5. Usporedba rezultata planiranja putanje AUV-a u okruženju s preprekama

Table 3 Comparison of testing results on simulation
Tablica 3. Usporedba rezultata testiranja u simulaciji

Aspects	QA	FMA [39]	DA [40]
Path length (m)	285.4	298.7	342.1
Total time (s)	241	310	281
Iteration	150	10	500
Safety distance (m)	12.3	8.7	5.1

Table 4 Comparison of results on second case
Tablica 4. Usporedba rezultata u drugom slučaju

Aspects	QA	FMA [39]	DA [40]
Path length (m)	602	614	625
Total time (s)	268	342	305
Iteration	180	12	620
Safety distance (m)	10.9	7.4	4.3

4.3. Experimental scenarios / *Eksperimentalni scenariji*

To demonstrate the effectiveness of the QA algorithm, the authors built a virtual reality environment using the HIL. In this approach, two experimental scenarios not only simulate actual operating conditions but also integrate realistic disturbances. The HIL model aims to verify the adaptability of the QA algorithm in environments similar to the actual operations.

In scenario 1, the AUV operates in an underwater environment with the shortest distance from the start point to the end of the path being 200 m. At the planning stage, an obstacle needs to be considered while evaluating the QA algorithm's ability to determine an optimal path. The results of Scenario 1 are illustrated in Fig. 6, where the red line represents the optimal path identified by the QA.

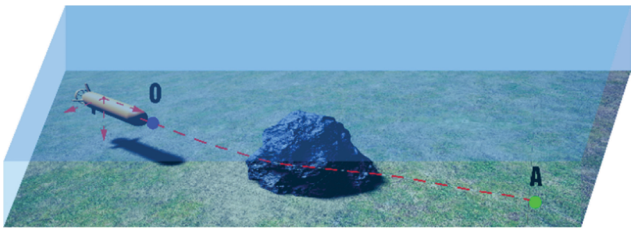


Figure 6 The AUV path for avoiding an obstacle
Slika 6. Putanja AUV-a za izbjegavanje prepreke

Table 5 Simulation results in the first scenario
Tablica 5. Rezultati simulacije u prvom scenariju

Aspects	QA	FMA [39]	DA [40]
Path length (m)	214	227	225
Total time (s)	107	113.5	112.9
Iteration	12	12	13
Safety distance (m)	5.49	6	6

The comparison results are presented in Table 5, indicating that the QA met the optimal path for a length of 214 m, which is 13 m shorter than the DA, and ensures a safe distance of 5.49 m. Although the number of QA iterations is 12, equal to FMA and DA, it still indicates a good balance between accuracy and processing speed. Meanwhile, FMA has a longer path time of 113.5 s and a higher safe distance of 6 m, reflecting its effective obstacle avoidance ability, but not optimal in terms of time. The DA provides a longer path length of 225 m, a moving time of 112.9 s, and a safe distance of 6 m, showing high accuracy but not superior in terms of response. In summary, the experimental results show that the QA algorithm has an advantage in optimizing the path to ensure AUV safety for moving through an obstacle.

In experimental scenario 2, the AUV is planned a path from point A to point B in a HIL environment with two randomly placed obstacles. The goal of this scenario is to evaluate the QA algorithm's ability to plan an optimal path in a more complex environment, where the AUV simultaneously maintains a safe path and minimizes the path length. The results of the second scenario are presented in Fig. 7, where the red line shows the optimal path searched by the QA.

The results presented in Table 6 express that the QA performs better than the comparison algorithms, with the shortest path length of 603 m, and the lowest moving time of 301 s, while maintaining an average safety distance of 5 m between the AUV and obstacles. Although the number of algorithm iterations of

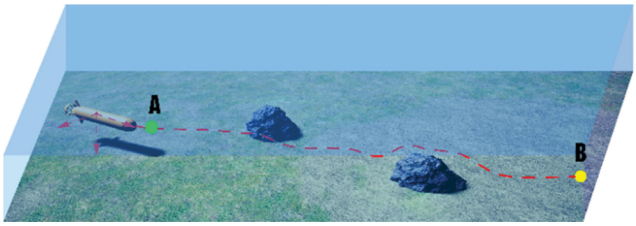


Figure 7 The AUV path for avoiding two obstacles
Slika 7. Putanja AUV-a za izbjegavanje dviju prepreka

Table 6 Simulation results in the second scenario
Tablica 6. Rezultati simulacije u drugom scenariju

Aspects	QA	FMA [39]	DA [40]
Path length (m)	603	813	645
Total time (s)	301	322	406
Iteration	13	13	12
Safety distance (m)	5.00	6.50	5.50

the QA is 13, which is equal to the FMA, it is still significantly more efficient than the DA with 12 iterations, resulting in a path longer than 645 m and a safety distance lower than 5.5 m. The FMA has a low number of iterations similar to the QA, but a longer moving time of 322 s and a safety distance higher than 6.50 m, showing good obstacle avoidance but not optimal in terms of time and path length. The Fig. 7 illustrates the AUV's path from point A to B, navigating through two obstacles, expressing the flexibility of the proposed solution. Based on the testing results of the two-obstacle scenario, the QA algorithm has the ability to balance well between safety and path optimization, making it suitable for practical applications in complex underwater environments.

In this study, the HIL plays a crucial role in evaluating the QA performance in the path optimization. The HIL application allows integration of the actual AUV control hardware with the virtual model, thereby demonstrating its effectiveness for close-to-actual operating conditions. Through two experimental scenarios with different numbers of obstacles, the QA verifies its ability to navigate effectively, guaranteeing a safe distance and optimal moving time. In particular, the QA iteration number, although higher than FMA, is significantly lower than the DA, reflecting the balance between accuracy and processing speed. The implementation of HIL not only helps validate the algorithm in real-time physical conditions but also evaluates the interaction between the virtual reality model and the control hardware, thereby improving the reliability and applicability of QA in AUV systems operating in actual environments.

5. CONCLUSION / *Zaključak*

This study develops a novel optimization method for path planning for AUVs based on the QA. The structure of QA efficiently determines optimal paths by evaluating potential paths through obstacles through the quantum superposition mechanism to find the best path. Simulation and experimental results, conducted in scenarios with one to two obstacles, have demonstrated the effectiveness of the algorithm in finding both safe and optimal paths while significantly reducing the computation time. Furthermore, the results from the HIL model validation show that the QA algorithm achieves high performance in real-time AUV path adjustment, ensuring collision avoidance and path optimization in a hardware-

integrated simulation environment. In the future, this study will integrate the real-time sensor data to handle mobile obstacles and ocean currents, and conduct field tests with physical AUV models to bridge the gap between simulation and actual deployment. These properties are further investigated using analytical solutions and stability guarantees under different obstacle configurations, ocean-current disturbances, and sensor noise. It should also be noted that the present study evaluates the proposed quantum-inspired optimization algorithm through numerical simulations and HIL experiments under the tested scenarios, in which the summarized results demonstrate the feasibility of the proposed solution.

Author Contributions: V.-D. D.: Formal analysis, numerical data calculation, algorithm implementation, writing – original draft preparation. X.-K. D.: Conceptualization, methodology, supervision, writing – review and editing. Ž. K.: Technical review, validation, writing – review and editing. T.-D. T.: Data curation, computing, simulation setup and execution. C.-L. L.: Visualization, result analysis and technical support.

Conflict of interest: The authors state that there is no conflict of interest.

Acknowledgement: The authors acknowledge the facilities, scientific and technical support from Artificial Intelligent Transportation LAB, University of Transport Ho Chi Minh City and Maritime Department, University of Dubrovnik, Croatia.

Acknowledgement of AI or AI-assisted tools use: During the preparation of this manuscript, the authors used Grammarly for the purposes of English grammar checking, spelling correction, punctuation checking, and language polishing. The authors reviewed and edited all suggested corrections and take full responsibility for the content of this publication.

REFERENCES / *Literatura*

[1] Mišković, D., Bielić, T., & Čulin, J. (2018). Impact of technology on safety as viewed by ship operators. *Transactions on Maritime Science*, *7* (1), 51-58. https://doi.org/10.7225/toms.v07.n01.005

[2] Cao, X., Chen, L., Guo, L., & Han, W. (2021). AUV global security path planning based on a potential field bio-inspired neural network in underwater environment. *Intelligent Automation and Soft Computing*, *27* (2), 391-407. https://doi.org/10.32604/iasc.2021.01002

[3] Dang, X. K., Do, V. D., & Koboević, Z. (2025). Short-term Motion Control for Semi-submersible Platform Based on Constrained Fuzzy Genetic Optimization Controller. *Advances in Electrical and Computer Engineering*, *25* (3), 79-92. https://doi.org/10.4316/AECE.2025.03009

[4] Pham, T. A., Dang, X. K., Koboević, Ž., Do, V. D., & Pham, T. D. A. (2024). Maritime Data Mining for Marine Safety Based on Deep Learning: Southern Vietnam Case Study. *Naše more*, *71* (1), 21-29. https://doi.org/10.17818/NM/2024/1.4

[5] Chen Y., Zhang, H., Zou, W., Zhang, H., Zhou, B., & Xu, D. (2025). Dynamic modelling and learning based path tracking control for ROV-based deep-sea mining vehicle. *Expert Systems with Applications*, 262, 125612. https://doi.org/10.1016/j.eswa.2024.125612

[6] Rorstad, H. Y., Krogstad, T. R., & Bitar, G. (2023). Motion Planning for Autonomous Underwater Vehicles in Near-Seafloor Operations Using State Lattices. *Proceedings of OCEANS 2023*, 1-9. https://doi.org/10.23919/OCEANS52994.2023.10337347

[7] Jiang, X., Wang, Z., & Dong, C. (2024). A Path Planning Algorithm Based on Improved RRT Sampling Region. *Computers, Materials and Continua, 80* (3), 4303-4323. https://doi.org/10.32604/cmc.2024.054640

[8] Sahoo, A., Dwivedy, S. K., & Robi, P. S. (2022). Adaptive Neuro Fuzzy PID Controller for A Compact Autonomous Underwater Vehicle. *Proceedings of OCEANS 2022*, 1-5. https://doi.org/10.1109/OCEANS47191.2022.9976983

[9] Cao, X., Sun, C., & Chen, M. (2019). Path planning for autonomous underwater vehicle in time-varying current. *IET Intelligent Transport Systems*, *13* (8), 1265-1271. https://doi.org/10.1049/iet-its.2018.5388

[10] Lai, Y., Shang, J., Liu, Z., Jiang, Z., Li, Y., & Chen, L. (2024). Deep-Sea A*+: An Advanced Path Planning Method Integrating Enhanced A* and Dynamic Window Approach for Autonomous Underwater Vehicles. *Proceedings of 2024 5th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering*, 485-491. https://doi.org/10.1109/ICBAIE63306.2024.11117128

[11] Zhang, B., Su, Y., Sun, S., Luo, W., & Huang, Q. (2025). Hybrid path planning algorithm for underactuated AUV based on RRT star and APF. *Scientific Reports*, *15* (1), 28493. https://doi.org/10.1038/s41598-025-11500-1

[12] Yao, J., Li, X., Zhang, Y., Chen, K., Zhang, D., & Ji, J. (2024). Autonomous Navigation Control of Tracked Unmanned Vehicle Formation in Communication Restricted Environment. *IEEE Transactions on Vehicular Technology*, *73* (11), 16063-16075. https://doi.org/10.1109/TVT.2024.3423430

[13] Chen, Z., Yan, J., Huang, R., Gao, Y., Peng, X., & Yuan, W. (2024). Path Planning for Autonomous Underwater Vehicles (AUVs) Considering the Influences and Constraints of Ocean Currents. *Drones*, *8* (8), 348. https://doi.org/10.3390/drones8080348

[14] Dang, X. K., Truong, H. N., & Do, V. D. (2022). A path planning control for a vessel dynamic positioning system based on robust adaptive fuzzy strategy. *Automatika*, *63* (3), 580-592. https://doi.org/10.1080/00051144.2022.2056289

[15] Singh, Y., Sharma, S., Sutton, R., Hatton, D., & Khan, A. (2018). Feasibility study of a constrained Dijkstra approach for optimal path planning of an unmanned surface vehicle in a dynamic maritime environment. *Proceedings of 2018 IEEE International Conference on Autonomous Robot Systems and Competitions*, 117-122. https://doi.org/10.1109/ICARSC.2018.8374170

[16] Tian, Z. & Chen, M. (2025). Safe and Optimal Motion Planning for Autonomous Underwater Vehicles: A Robust Model Predictive Control Frameworks Integrating Fast Marching Time Objectives and Adaptive Control Barrier Functions. *Drones*, *9* (4), 273. https://doi.org/10.3390/drones9040273

[17] Richert, D., MacNab, C. J. B., & Pieper, J. K. (2012). Adaptive haptic control for telerobotics transitioning between free, soft, and hard environments. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, *42* (3), 558-570. https://doi.org/10.1109/TSMCA.2011.2170066

[18] Dang, X. K., Corchado, J. M., Le, V. V., & Do, V. D. (2024). Non-parametric Vibration-based Structural Damage Detection for Coastal Structures: Multi-Dimension to Single Input Convolutional Neural Network Approach. *Advances in Electrical and Computer Engineering*, *24* (4), 3-18. https://doi.org/10.4316/AECE.2024.04001

[19] Krishnan, M. G., & Sankar, A. (2022). Image space trajectory tracking of 6-DOF robot manipulator in assisting visual servoing. *Automatika*, *63* (2), 199-215. https://doi.org/10.1080/00051144.2021.2022889

[20] Bell, J. J., Micaroni, V., Strano, F., Broadribb, M., Wech, A., Harris, B., & Rogers, A. (2023). Testing the impact of Remotely Operated Vehicle (ROVs) camera angle on community metrics of temperate mesophotic organisms: A 3D model-based approach. *Ecological Informatics*, 76, 102041 https://doi.org/10.1016/j.ecoinf.2023.102041

[21] Hosseinnajad, A., & Mohajer, N. (2025). Fixed-time Observer-based Saturated Non-singular Terminal Sliding Mode Controller Design for an Over-actuated ROV with Time-varying Saturation Limits. *ISA Transactions*, 165, 98-110. https://doi.org/10.1016/j.isatra.2025.06.025

[22] Mišković, N., Vukić, Z., & Barišić, M. (2009). Identification of unmanned underwater vehicles by self-oscillation method. *Automatika*, *50* (3-4), 167-183. https://hrcak.srce.hr/file/73157

[23] Dang, X. K., Koboević, Ž., Phan, N. H. L., Do, V. D., & Ly, S. (2025). Artificial Neural Network for Improving the Motion Control Quality of Underwater Remotely Operated Vehicle. *Naše more*, *72* (2), 39-48. https://doi.org/10.17818/NM/2025/2.1

[24] Gary Greene, H. (2015). Habitat characterization of a tidal energy site using an ROV: Overcoming difficulties in a harsh environment. *Continental Shelf Research*, 106, 85-96. https://doi.org/10.1016/j.csr.2015.06.011

[25] Chen, Y., Zhang, H., Zou, W., Wu, H., & Li, Q. (2025). Learning-based obstacle avoidance strategy for ROV-based deep-sea mining vehicle. *Ocean Engineering*, 339, 121996. https://doi.org/10.1016/j.oceaneng.2025.121996

[26] Azis, F. A., Aras, M. S. M., Rashid, M. Z. A., Othman, M. N., & Abdullah, S. S. (2012). Problem identification for Underwater Remotely Operated Vehicle (ROV): A case study. Procedia Engineering, 41, 554-560. https://doi.org/10.1016/j.proeng.2012.07.211

[27] Amundsen, H. B., Føre, M., Ohrem, S. J., Haugaløkken, B. O. A., & Kelasidi, E. (2024). Three-dimensional obstacle avoidance and path planning for unmanned underwater vehicles using elastic bands. *IEEE Transactions on Field Robotics*, 1, 70-92. https://doi.org/10.1109/tfr.2024.3421393

[28] Wagner, F., Nublein, J., & Liers, F. (2025). Enhancing Quantum Algorithms for Quadratic Unconstrained Binary Optimization via Integer Programming. *ACM Transactions on Quantum Computing*, *6* (2), 1-28. https://doi.org/10.1145/3711935

[29] Li, X. & Yu, S. (2023). Three-dimensional path planning for AUVs in ocean currents environment based on an improved compression factor particle swarm optimization algorithm. *Ocean Engineering*, 280, 114610. https://doi.org/10.1016/j.oceaneng.2023.114610

[30] Arnesen, B. O., Lekkas, A. M., & Schjølberg, I. (2017). 3D path following and tracking for an inspection class ROV. *Proceedings of the ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering*, 1-10. https://doi.org/10.1115/OMAE2017-61170

[31] Chen, X., Bose, N., Brito, M., Khan, F., Millar, G., Bulger, C., & Zou, T. (2022). Risk-based path planning for autonomous underwater vehicles in an oil spill environment. *Ocean Engineering*, 266, 113077. https://doi.org/10.1016/j.oceaneng.2022.113077

[32] Xie, J., Wang, Z. & Chen, Y. (2022). Joint Caching and User Association Optimization for Adaptive Bitrate Video Streaming in UAV-Assisted Cellular Networks. *IEEE Access*, 10, 106275-106285. https://doi.org/10.1109/ACCESS.2022.3211940

[33] Xia, G., Han, Z., Zhao, B., Liu, C., & Wang, X (2019). Global Path Planning for Unmanned Surface Vehicle Based on Improved Quantum Ant Colony Algorithm. *Mathematical Problems in Engineering*, *2019* (1), 1-10. https://doi.org/10.1155/2019/2902170

[34] Moussa, C., Wang, H., Bäck, T., & Dunjko, V. (2022). Unsupervised strategies for identifying optimal parameters in Quantum Approximate Optimization Algorithm. *EPJ Quantum Technology*, *9* (11), 1-19. https://doi.org/10.48550/arXiv.2202.09408; https://doi.org/10.1140/epjqt/s40507-022-00131-4

[35] Hadfield, S., Wang, Z., O'Gorman, B., Rieffel, E. G., Venturelli, D., & Biswas, R. (2019). From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, *12* (2), 34. https://doi.org/10.3390/a12020034

[36] Bravyi, S., Kliesch, A., Koenig, R., & Tang, E. (2020). Obstacles to Variational Quantum Optimization from Symmetry Protection. *Physical Review Letters*, *125* (26). https://doi.org/10.1103/PhysRevLett.125.260505

[37] Kadowaki, T., & Nishimori, H. (1998). Quantum annealing in the transverse Ising model. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, *58* (5), 5355-5363. https://doi.org/10.1103/PhysRevE.58.5355

[38] Wang, Z., Hadfield, S., Jiang, Z., & Rieffel, E. G. (2018). Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Physical Review A*, *97* (2), 022304. https://doi.org/10.1103/PhysRevA.97.022304

[39] Pêtrès, C., Pailhas, Y., Patrón, P., Petillot, Y., Evans, J., & Lane, D. (2007). Path Planning for Autonomous Underwater Vehicles. *IEEE Transactions on Robotics*, *23* (2), 331-341. https://doi.org/10.1109/TRO.2007.895057

[40] Dudi, T., Singhal, R., & Kumar, R. (2020). Shortest Path Evaluation with Enhanced Linear Graph and Dijkstra Algorithm. *Proceedings of 2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan*, 451-456. https://doi.org/10.23919/sice48898.2020.9240227