An Autonomous Reactive Architecture for Efficient AUV Mission Time Management in Realistic Severe Ocean Environment

Somaiyeh Mahmoud.Zadeh, David M.W Powers, Karl Sammut
Centre for Maritime Engineering, Control and Imaging,
School of Computer Science, Engineering and Mathematics, Flinders University, Adelaide, SA 5042, Australia

sомaiyeh.mahmoudzadeh@flinders.edu.au
david.powers@flinders.edu.au
karl.sammut@flinders.edu.au

Abstract Today AUVs operation still remains restricted to very particular tasks with low real autonomy due to battery restrictions. Efficient motion planning and mission scheduling are principle requirement toward advance autonomy and facilitate the vehicle to handle long-range operations. A single vehicle cannot carry out all tasks in a large scale terrain; hence, it needs a certain degree of autonomy in performing robust decision making and awareness of the mission/environment to trade-off between tasks to be completed, managing the available time, and ensuring safe deployment at all stages of the mission. In this respect, this research introduces a modular control architecture including higher/lower level planners, in which the higher level module is responsible for increasing mission productivity by assigning prioritized tasks while guiding the vehicle toward its final destination in a terrain covered by several waypoints; and the lower level is responsible for vehicle’s safe deployment in a smaller scale encountering time-varying ocean current and different uncertain static/moving obstacles similar to actual ocean environment. Synchronization between higher and lower level modules is efficiently configured to manage the mission time and to guarantee on-time termination of the mission. The performance and accuracy of two higher and lower level modules are tested and validated using ant colony and firefly optimization algorithm, respectively. After all, the overall performance of the architecture is investigated in 10 different mission scenarios. The analyze of the captured results from different simulated missions confirm the efficiency and inherent robustness of the introduced architecture in efficient time management, safe deployment, and providing beneficial operation by proper prioritizing the tasks in accordance with mission time.

Keywords Autonomous vehicle, Hierarchal architecture, Mission/motion/path planning, Underwater missions, Task assignment, Time management, Decision Autonomy

1 Introduction

Most of current AUVs applications are supervised from the support vessel that provides higher-level decisions in critical situation and generally takes enormous cost during the mission [1]. A growing attention has been devoted in recent years on increasing the ranges of missions, vehicles endurance, extending vehicles applicability, promoting vehicles autonomy to handle longer missions without supervision, and reducing operation costs [2]. AUVs should operate in turbulent underwater environment with complex spatio-temporal variability, where the current variability can perturb AUVs safety conditions. Currents instabilities can influence vehicle’s motion and probably push it to an undesired direction. Robustness of AUV to this strong environmental variability is a key element to mission performance and carry out safety considerations. Restrictions of priori knowledge about later conditions of the environment reduce AUVs autonomy and robustness. Additionally, AUVs capabilities in handling mission objectives is directly influenced by performance of task allocation system. Many of the today’s AUV missions is limited to executing a list of pre-programmed instructions and completing a predefined sequences of tasks. Autonomous adaptation of AUVs in performing different tasks in dynamic and continuously changing environment has not been completely fulfilled yet and it is still necessary for the operators to remain in the loop of considering and making decisions [3].
Generally, the AUVs control architecture consists of two different execution layers based on its control requirements i.e. deliberative and reactive layers [4]. The deliberative layer manages the concurrent execution of several tasks with different priorities. The reactive layer manages real-time reactions to perform quick respond to critical events. These automatic functions are executed in the background and promote AUVs self-management characteristics. Mission timing and AUV’s time management is a fundamental requirement toward mission success. Consequently, reaching higher performance in real-time applications is a challenging issue due to limited resource availability and the dynamic changes of the environment. Thus, the vehicle should trade-off within the problem constraints and mission productivity while managing the available time and risks. Evidently, decision autonomy and vehicles routing/task assignment capability are connected in various standpoints. On the other hand, instantly adapting to the continuously changing situations of the real world where the most of the information are uncertain and unknown is strongly critical. Accordingly, in lower level also, the vehicle should be capable of coping dynamic time variant ocean current and avoiding collision to different static and dynamic obstacles to ensure safe deployment during the mission. To satisfy the addressed requirements and handling these challenges, development of more evolved embedded functions are required, which can promote the platform’s autonomy in both higher and lower levels while maintaining the trust on vehicles safety at all stages of the mission. Many efforts have been devoted in recent years for promoting AUVs’ autonomy in mission task assignment in recent years; however, there are still many challenges on achieving a satisfactory level of autonomy for AUV in this regard. Very few investigations are funded to address issues with both low and high level autonomy development and the problem remains unsolved for underwater missions. According to these fundamental concepts, various methodologies have been discussed and worked out in the literature as follows.

1.1 Previous Works on Vehicle Routing Problem (VRP) and Task Assignment Approach

Effective routing has a great impact on vehicles time management as well as mission performance by appropriate selection and arrangement of the tasks sequence. Liu and Bucknall (2015) proposed a three-layer structure to facilitate multiple unmanned surface vehicles to accomplish task management and formation route planning in a maritime environment [5]. Eichhorn (2015) implemented graph-based methods for the AUV “SLOCUM Glider” motion planning in a dynamic environment [6]. The author employed modified Dijkstra Algorithm where the applied modification and conducted time variant cost function simplifies the determination of a time-optimal route in the geometrical graph. An energy efficient fuzzy based route planning system is presented by Kladis et al., (2011) for UAVs motion planning in a graph-like terrain using priori known wind information [7]. Other methods also studied on efficient task assignment for single/multiple vehicle task assign/routing problem such as graph matching algorithm [8], partitioning method [9], Tabu search algorithm [10], branch and cut algorithm [11], A* search algorithm [12-14], Dijkstra [6,15], and evolutionary algorithms [8,16,17]. The traditional algorithms used for graph routing problem such as Dijkstra, or dynamic programming algorithm has major shortcomings such as high computational complexity for real-time applications. Another issue is that all outline literature in this scope mostly focus on routing problem for different unmanned vehicles in which task allocation is the principle direction of these papers and quality of deployment (motion) has not been addressed. The vehicle’s safe and confident deployment is a critical factor affecting vehicle’s level of autonomy and should be taken into consideration at all stages of the mission in a vast and uncertain environment. So, in the second category, existing AUVs’ trajectory/path planning approaches have been discussed, which are more concentrated on quality of vehicle’s motion encountering dynamicity of the terrain.

1.2 Path/Trajectory Planning Approaches

Path planning itself is a complicated multi-objective Non-deterministic Polynomial-time (NP) hard problem that has a great impact on vehicle’s overall autonomy. The path planning strategies proposed in most of the previous researches particularly deal with vehicle’s guidance toward the destination encountering dynamic changes of the terrain. Different methods like D*, A*, FM, FM* algorithms have been employed for AUV optimum path generation [18-21]. Particularly, the main drawback of all above-mentioned methods is that their time complexity increases exponentially with increasing the problem space, which is inappropriate for real-time applications. Evolutionary algorithms are population based optimization methods applied successfully on path planning problem that are advantaged to be implemented on a parallel machine with multiple processors, which speeds up the computation process [22,23]. The Particle Swarm Optimization (PSO) [22, 24], Evolution (DE) [24, 25], Genetic Algorithm (GA) [22, 26, 27], multi-objective genetic algorithm (NSGA II) [28], Differential, and Quantum-based PSO (QPSO) [29] are some popular types of optimization algorithms applied successfully on offline/online path planning approaches. Although various path planning techniques have been suggested for autonomous vehicles, AUV-oriented applications still face several difficulties. Path replanning in this context requires a significant computational effort for progressing data from update of entire terrain, which is problematic in large scale operations. Accurate prediction of the behavior of a dynamic large-scale terrain, far-off the vehicles sensor coverage is unreliable and impractical as only awareness of environment in vicinity of the vehicle is sufficient such that the vehicle can able to perform reaction to prompt changes. Another problem is that the path planning strategies are not designed for handling the task assignment and time management in a graph-like terrain with respect to graph routing restrictions in cases that the vehicle is required to carry out a specific sequence of ordered tasks. In these kind of problems a routing strategy is a proper approach in dealing with graph search constraints for the VRP problem and organizing the task. However, routing strategies are not accurate enough in coping dynamic variations of the terrain, but these approaches fracture the operation area to smaller beneficial sections and give overview of the pathway for vehicle’s maneuver that helps reducing the computational load. Accordingly, each approach is able to handle only a specific level of autonomy in terms of organizing
the tasks and managing the mission or dealing with variations of the environment and producing a high quality maneuver for the vehicle. This research addresses the AUV’s requirements in both high and low level autonomy toward making AUV systems more intelligent and robust in managing its availabilities and adaption to a dynamic environment.

1.3 Research Assumptions and Contributions

A novel autonomous architecture with online re-planning capability is developed to carry out the underwater missions in a large scale environment in the presence of severe environmental disturbances. The system incorporates two different execution layers, deliberative and reactive, to satisfy the AUV’s high and low level autonomy requirements in motion planning and mission management. A “Task Organize Mission Plan” (TOM-P) module is designed for the deliberative layer to provide a higher level of autonomy by ordering the execution of several tasks so that the AUV gets directed toward the targeted location. The TOM-P module in the top level simultaneously rearranges the order of tasks taking passage of time into account. The reactive layer, on the other hand, is responsible for performing real-time reactions and quick response to critical events. A “Local On-line Path-Plan” (LOP-P) module was developed in this layer and includes automatic functions that executed in the background to promote the AUV’s self-management characteristics. The LOP-P module at the lower level deals with quality and safety of deployment along the ordered tasks (produced by higher layer TOM-P) where persistent variation of the sub-area in proximity of the vehicle is considered simultaneously. The operation of each module can take longer than expectation due to unexpected dynamic changes in the environment. In order to reclaim the missed time an efficacious synchronization scheme (named “Synchron”) is added to the architecture to keep pace of modules in different layers of the system.

This research is a completion of the previous attempts [30-32] that construction of the mission-motion planners [31,32] is boosted to a consistent structure in scheme of a coherent architecture in which the performance of the proposed construction is mainly independent of the employed algorithms by modules. The main reason for remarkable performance of this system is the fashion of mixing and matching two disparate strategy from two different prospective and constructing an accurate synchronization between them. A significant benefit of such modular construction is that the modules can employ different methods or their functionality get upgraded without manipulating the system’s structure. This advantage specifically increases the reusability and versatility the control architecture and eases updating/upgrading AUV’s functionalities to be compatible with other applications, and in particular implement it for other autonomous systems such as unmanned aerial, ground or surface vehicles.

The previous study [30], which provided a primary basic idea of such a modular framework, is expanded by enhancing the synchronization process and maneuverability of the vehicle by upgrading the path planner with the added capability of reactive re-planning, generalizing the applicability of the planners by modeling more realistic underwater situations incorporating kinodynamics of the AUV, variable water currents and uncertainty of the terrain.

The paper is prepared in following sections: the mechanism of the TOM-P and LOP-P modules are demonstrated in Section 2, respectively. An overview of the adopted methods by both of modules on corresponding problems is presented in Section 3. The architectures evaluation and analyze of the simulation results is provided in Section 4 and the conclusion of this research is provided by Section 5.

2 Proposed Modular Control Architecture

Proposed architecture designed in separate modules running concurrently including TOM-P in top level with higher level of decision autonomy, and the LOP-P in lower level to autonomously carry out the collision avoidance, cope with current force and handle similar environmental challenges. The modules interact simultaneously by back feeding the situational awareness of the surrounding operating field; accordingly the system decides to re-plan the path or mission or continue the current mission. Hence, the Synchron module is designed to manage the lost time within the LOP-P/ TOM-P process and reactively adapt the system to the last update of environment and decision parameters (e.g. remaining time). This process frequently repeated until the AUV reaches to the end point. Real map data and different uncertain dynamic/ static objects are encountered to model a realistic marine environment.

![Fig.1. The information flow in proposed control architecture including TOM-P, LOP-P, and Synchron modules](image-url)
Besides, impact of the time variant water current on the AUV’s motion is incorporated. To cope dynamic variations of the environment, the LOP-P is facilitated to perform on-line re-planning from vehicles location to any predefined target point. The conceptual and mathematical model of the modular framework including TOM-P, LOP-P and Synchron modules is presented in following subsections. The schematic representation of propounded control architecture is provided by Fig. 1.

2.1 Task Organize Mission Plan (TOM-P)

The terrain modelled as waypoint cluttered operation network, in which various tasks are sparsely distributed on edges of the network and their priority presented by a weight value that initialized once in advance before starting the mission. An example of such an operation network in given by Fig. 2. In such a network, the edges that are not assigned with a task get value of one, otherwise they get value greater than one proportional to the weight of the corresponding task. Placement of the waypoints in the network performed with a uniform distribution in joint water covered sections of the map and distributed once in advance. The AUV is requested to complete its mission furnishing maximum number of tasks with higher weight while moving toward the destination. Obviously, a single AUV is not able to meet all specified tasks in one mission due to its time and energy restrictions, specifically when operating in a large-scale operation area. Hence, organizing and arranging the tasks involves a decision making procedure, in which a tradeoff exists between ordering the tasks under specified constraints and guiding the vehicle through the waypoints. Therefore, tasks should be arranged in a way that selected edges (tasks) govern the AUV to the destination and meet the time restriction, which is a joint discrete and syndetic multi-objective problem resembling both Knapsack problem and TSP. In this context, the TOM-P module at top level simultaneously tends to determine the optimum order of edges that mathematically described as follows:

∀ \( P_0^t \sim U(0,10000); P_t^t \sim U(0,100); P_{x,y,z}^t \notin \text{Map} = 0 \)

\[
\begin{align*}
q_{ij} & : t_{ij} = \frac{d_{ij}}{t} \\
q_{ij} & \notin \text{Task}_{ij} \Rightarrow w_{ij}
\end{align*}
\]

\[
T_M = \sum_{i=0}^{n} l_{ij} t_{ij} = \sum_{i=0}^{n} \frac{l_{ij} d_{ij}}{t}, \quad l \in \{0,1\}
\]

\[
\text{max} \left\{ \sum_{i=0}^{n} l_{ij} w_{ij} \right\} \quad \text{s. t. } \text{max}(T_M) < T_{\text{Total}}
\]

Definition of all parameters are given by “Nomenclature” table. Coordinates of waypoints in the geographical frame shouldn’t have any intersection with coastal area denoted by \( \text{Map} = 0 \). The \( l \) is the selection variable that shows the selected edges in the network. The total captured weight in a mission should be maximized and the mission time should approach the total available time. Safe deployment is another important task that should be taken into consideration at all stages, which is extremely critical and complicated issue in a vast and uncertain environment that is investigated in subsection 2.2.

2.2 Local On-line Path Plan (LOP-P)

2.2.1 Modelling the Ocean Terrain, Obstacles and Dynamic Current Field

The ocean environment modelled as a three dimensional volume \( \Gamma_{3D} \) covered by uncertain, static/dynamic objects and fixed waypoints. An underwater mission is commenced at a specified starting point and it is terminated when the AUV reaches to a predefined destination point (dock for example). A real map data of Adelaide-gulf st Vincent (the southern coast of Australia) [33] has been used to make the simulation process more realistic. Additional to offline map, two types of static uncertain and moving uncertain obstacles are considered in this research to cover different possibilities of the real world situations. The sonar sensors measure obstacle’s coordinates and velocity with a specific uncertainty that modelled with a Gaussian distribution. The obstacle is shown by three components of position \( \Theta_p \), radius \( \Theta_r \) and uncertainty rate \( \Theta_{ur} \). The static uncertain objects modelled with uncertain radius that is varied in a predefined bound with a normal distribution \( \Theta_r \sim (\Theta_r, \sigma) \). Uncertain moving obstacles also modelled in this research, in which the object motion is affected by current flow. The effect of current presented by uncertainty propagation proportional to current magnitude \( \Theta_c \theta \mathcal{C} \) in a circular format radiating out from the center of the object, denoted by \( (5) \). The objects position in both cases is initialized using normal distribution bounded to position of start \( (a) \) and target \( (b) \) waypoints; hence, its probability density function is defined by \( (4) \) as follows:
\[ \Theta_{f} = \left[ P_{i,3}^{a}, P_{i,3}^{b} \right] - \Theta_{f} \]  

\[ f(\Theta_{f}, \Theta_{r}, P_{i,3}^{a}, P_{i,3}^{b}) = \Theta_{f} \left( \frac{P_{i,3}^{b} - P_{i,3}^{a}}{\Theta_{f}} \right) \]

\[ \Theta_{f}(t) = \Theta_{f}(t-1) \pm \left( \sigma_{\Theta_{f}} \right) \]

\[ \Theta_{r}(t) = B_{1} \Theta_{f}(t-1) + B_{2} \Theta_{f}(t-1) + B_{1} \]

\[ B_{1} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, B_{2} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \]

where \( X_{i,0} \sim N(0, \sigma_{0}) \) is the normal distribution that assigned to each obstacle and gets updated iteratively.

On the other hand, the ocean current information can be obtained from remote devices (e.g., satellite observations) or numerical estimation models. Different predictive ocean current models have been constructed previously by [34,35]. A 3D turbulent time varying current model is constructed by this research, where the current update is modeled by multiple layered 2D current structure, in which the current circulation pattern gradually changes in each layer, given by (6):

\[ u_{c}(S) = \frac{y}{2\pi(S-\delta S)^{2}} \left[ 1 - e^{\frac{-\left| S-\delta S \right|^{2}}{\ell^{2}}} \right] \]

\[ v_{c}(S) = \frac{x}{2\pi(S-\delta S)^{2}} \left[ 1 - e^{\frac{-\left| S-\delta S \right|^{2}}{\ell^{2}}} \right] \]

\[ w_{c}(S) = \frac{1}{\delta x^{2}} \left[ \frac{\ell^{2} - \delta S^{2}}{\left( \delta_{x} \lambda_{x} \right)^{2}} \right] \]

A parameter \( \gamma \) is used to scale the vertical profile of the current \( (w_{c}) \) from the horizontal profile \((u_{c}, v_{c})\) due to weak vertical motions of the ocean current. To estimate continuous circulation patterns of each subsequent layer, a recursive application of Gaussian noise is applied to the \( S_{i,3}, \delta \) and \( f \) parameters of 2D case as follows:

\[ S_{i}^{\delta} = A_{i} S_{i}^{\delta} + A_{j} X_{i}^{\delta}, t = t \left( t-1 \right) + A_{2} X_{i}^{\delta}, t = t \left( t-1 \right) \]

\[ \delta(t) = A_{1} \delta(t-1) + A_{2} X_{i}^{\delta}, t = t \left( t-1 \right) \]

where \( X_{i,3}, X_{i,3}, X_{i,3}^{\delta} \) are Gaussian normal distributions.

### 2.2.2 Modelling Kinematics and Dynamics of the AUV

To describe vehicles motion with six degrees of freedom (6-DOF) two coordinate systems of \( \{ b \} \) frame, which is located at the centre of buoyancy of the vehicle, and “Earth-Fixed coordinate” of \( \{ n \} \) frame are utilized [36]. It is assumed that the centres of both coordinates are congruous and the Earth’s rotation is ignored. The state variables of \( \{ b \}-\{ n \} \) frames and AUV’s kinematics are described by (8) and (9), respectively:

\[ \begin{bmatrix} \dot{b} \\ v \end{bmatrix} \rightarrow \begin{bmatrix} u,v,w,p,q,r \end{bmatrix} \]

\[ \begin{bmatrix} \dot{n} \end{bmatrix} \rightarrow \begin{bmatrix} X,Y,Z,\phi,\theta,\psi \end{bmatrix} \]

\[ \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi & \cos \psi \sin \theta \\ \sin \psi \cos \theta & \cos \psi & \sin \psi \sin \theta \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \]

### 2.2.3 Mathematical Modelling of the Path Planning Problem and Path Re-planning

The path planning is a constraint optimization problem in which the main goal is to provide a safest and quickest trajectory between two points. The study takes the use of B-Spline function to generate the potential trajectories \( \phi_{i} \). The B-Spline curves captured from a set of control points like \( \delta L_{i,3} \). Placement of these control points play a substantial role in determining the optimal path. All \( \delta \) should be located in respective search space limited to predefined Upper-Lower bounds of \( \delta \) in \( \{ L_{i,3}L_{n} \} \) in Cartesian coordinates. Accordingly, path curve is generated as follows:

\[ L_{i} = P_{i,3,3}^{a} \& U_{i} = P_{i,3,3}^{b} \]

\[ \theta_{i}(t) = U_{i}^{2r} \left[ U_{i}^{2} - U_{i}^{3} \right] \]

\[ \delta_{i}(t) = U_{i}^{2r} \left[ U_{i}^{2} - U_{i}^{3} \right] \]

\[ \beta_{i}(t) = \frac{U_{i}^{2r} \left[ U_{i}^{2} - U_{i}^{3} \right]}{U_{i}^{2} - U_{i}^{3}} \]
\[
\begin{align*}
X(t) &= \sum_{i=1}^{n} \theta_{xi}(t) B_{iX}(t) \\
Y(t) &= \sum_{i=1}^{n} \theta_{yi}(t) B_{iY}(t) \\
Z(t) &= \sum_{i=1}^{n} \theta_{zi}(t) B_{iZ}(t)
\end{align*}
\]

(11)

where, the \(X(t), Y(t), Z(t)\) display the vehicle’s position at each time step \(t\) along the generated path \(\theta\) in the \(\{n\}\) frame. The vehicle’s orientation along the generated path also should be calculated as given by (12):

\[
\psi(t) = \tan^{-1} \left( \frac{Y_{i+1}(t) - Y_i(t)}{X_{i+1}(t) - X_i(t)} \right)
\]

(12)

In (12), \(\psi(t)\) and \(\theta(t)\) represent the history the vehicle’s yaw and pitch orientations at time \(t\). Water currents continually affect the vehicle’s motion, so that its position and attitude should be adjusted constantly. The vehicle is assumed to have a constant velocity \(v\). Applying the 3D water current velocity of \(V_c(u_c, v_c, w_c)\), given by (13), the vehicle’s angular velocity is calculated from the equation (14).

\[
\begin{align*}
&u_c = V_c \cos \theta_c \cos \psi_c \\
v_c = V_c \cos \theta_c \sin \psi_c \\
w_c = V_c \sin \theta_c \\
&u = \left[ \frac{\cos \theta \cos \psi + V_c \cos \theta_c \cos \psi_c}{V_c \cos \theta_c \cos \psi_c} \right] \\
v = \left[ \frac{\cos \theta \sin \psi + V_c \cos \theta_c \sin \psi_c}{V_c \cos \theta_c \sin \psi_c} \right] \\
w = \left[ \frac{\sin \theta \theta_c \sin \psi_c}{V_c \sin \theta_c \sin \psi_c} \right]
\end{align*}
\]

(13)

(14)

The path is evaluated using the following cost function, which is a combination of the time and corresponding penalty functions as described by (15).

\[
\varphi(t) = [X(t), Y(t), Z(t), \psi(t), \theta(t), u(t), v(t), w(t)]
\]

\[
T_{\varphi} = \frac{\sum_{i=1}^{n} \left| \theta_{x,y,z}(t) - \theta_{x,y,z}(t-\Delta t) \right|}{n}
\]

(15)

\[
\varphi_{Cost} = T_{\varphi} + \sum_{i=1}^{n} \gamma_{\varphi} \times \frac{1}{\left| \nabla_{\varphi} \nabla_{\Sigma M, \Theta} \right|}
\]

\[
\nabla_{\varphi} = \left[ \gamma_{\psi}, \gamma_{\theta}, \gamma_{\varphi} \right]
\]

\[
\nabla_{\Sigma M, \Theta} = \left\{ \begin{array}{ll}
\gamma_{\varphi} \varphi_{x,y,z}(t) \cap \text{Map}(x, y) = 1 & \text{if } \varphi_{x,y,z}(t) \cap \text{Map}(x, y) = 1 \\
\gamma_{\varphi} \varphi_{x,y,z}(t) \cap \bigcup \Theta(\Theta_{x}, \Theta_{y}) = 1 & \text{if } \varphi_{x,y,z}(t) \cap \bigcup \Theta(\Theta_{x}, \Theta_{y}) = 1 \\
0 & \text{Otherwise}
\end{array} \right.
\]

In (15), the \(\varphi_{Cost}\) is a weighted violation function including AUV’s kinodynamic constraints and collision boundaries. The generated path shouldn’t have any intersection with the forbidden zone \(\bigcup_{\Theta}(\Theta_{x}, \Theta_{y})\) associated with coast sections of the map and obstacles \(\Theta\). The coastal area are determined applying a k-means clustering method on offline map, which more detail is provided in section 4. The \(\gamma_{\theta}, \gamma_{\psi}, \gamma_{\varphi}, \text{ and } \gamma_{\Sigma M, \Theta}\) are impact of each violation in calculation of the \(\varphi_{Cost}\).

**On-line Path Re-planning based on Previous Solution**

This research applies on-line path planning as a continuous process for correcting the previous trajectory, so that the vehicle can be able to take a prompt reaction to cope with immediate changes of the dynamic environment. To this principle, environmental situation should be simultaneously computed and consequently the current trajectory should be refined continually during deployment toward the destination. To accomplish the re-planning, information of the vehicle’s current position and velocity along with dynamic changes of the environment are fed to the to the LOP-P module. In the proposed mechanism when the path

---

**Path Re-planning**

**Online Path Re-Planning Mechanism**

- Get the position of set start and target points from the TOM-P module
- Generate the initial optimum path using Eq.(11)
- Evaluate the generated path by defined \(\varphi_{Cost}\) in Eq.(15)
- Check Path-Planning Flag continuously

**BEGIN**

If Flag = 1

- Apply the current states of the vehicle as the initial condition
- Set the current location as the start point for LOP-P process
- Use the solution at hand (from previous path) as an initial solution or initial guess for the proposed evolutionary method
- Generate a new path
- Check the problem optimality conditions Eq.(15)

If Eq.(15) is satisfied

Follow the new path
Terminate the planning process

Otherwise

Follow the previous path
Terminate the planning process
Back to **BEGIN**

**END**

**Fig.3. On-line path re-planning mechanism**
re-planning flag is triggered the LOP-P generate a new alternative path from the vehicle’s current position to the target point by rectifying the previous trajectory. In other words, instead of regenerating the path from beginning in the problem space, it would be more appropriate if the old optimum path get refined from the existing position to the destination, which is discussed more in analysis of the simulation results. The Fig.3 clarifies this issue. In this fashion, the vehicle would be able to accurately cope with dynamic variations and uncertainty of the underlying environment. Indeed, this mechanism is resembling closed-loop guidance configuration and also reduces the computational cost [24]. This process is summarized in Fig.3.

2.3 Modelling of the Synchron Module

After the TOM-P generated a mission plan in a sequence of tasks/waypoints, the LOP-P module gets this sequence along with environmental information and starts generating safe and efficient trajectory between the waypoints while the AUV concurrently executes the corresponding tasks. In this way the AUV efficiently carries out its mission and safely guided through the waypoints. The provided trajectory is shift to the guidance controller to produce the guidance commands. During deployment between two waypoints, the LOP-P can incorporate any dynamic changes of the environment. The LOP-P repeatedly calculates the trajectory between vehicles current position and its specified target location (any time that new update of information received from the sensors). After each waypoint is visited the total path time of $T_{ij}$ gets compared to expected time $T_{ij}$ for traversing the corresponding distance, in which the $T_{ij}$ is extracted from the mission time $T_M$ given by (2). In a case that the $T_{ij}$ gets smaller than $T_{ij}$, the current order of tasks would be valid and the vehicle can continue its mission. But in the opposite case, when the $T_{ij}$ exceeds the $T_{ij}$, it means a certain amount of the available time $T_{Total}$ is spent for handling any probable challenge during the path planning process. In such a case, the previously defined mission scenario cannot be optimum anymore; therefore, mission re-planning according to mission updates would be necessary.

However, a question is raised in this situation about computational burden of the mission re-routing process. Passing a specific edge (distance) for more than ones is the time dissipation for an AUV that means repeating a task for several times. Therefore, the executable order of tasks is modified by comparing the expected times for completing all tasks. The synchronization process is shown in Fig.4.

2.4 Architecture Evaluation Criterion

Performance of the generated local path by LOP-P is evaluated by a cost function $\varphi_{Cost}$ based on travel time $T_p$, which is proportional to energy consumption and the distance travelled. The mission cost $M_{Cost}$ has direct relation to the passing distance between each pair of selected waypoints. Hence, the path cost $\varphi_{Cost}$ for any optimum local path gets used in the context of the TOM-P. The model is seeking for an optimal solution in the sense of the best combination of path and mission cost as follows:

\[
\varphi_{Cost} = T_p = \sum_{i=1}^{N_y} \left[ \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2 + (Z_{i+1} - Z_i)^2} \right] \\
\text{s.t.} \\
[X(t), Y(t), Z(t)] \cap \sum_{M, G=0} = 0 \\
u(t) \leq u_{max} & \quad \gamma_{min} \leq \gamma(t) \leq \gamma_{max} \\
\theta(t) \leq \theta_{max} & \quad \psi_{min} \leq \psi(t) \leq \psi_{max} \\
M_{Cost} \leq T_{Total} - \left[ \sum_{j=1}^{n} l_{t} \right] \left[ \psi_{Cost} + \psi_{CPU} \right] - T_{Total} + \sum_{j=1}^{n} l_{t} w_{ij} \quad t \in [0,1] \\
\text{s.t.} \quad \max(T_M) \leq T_{Total} 
\]

(16)
After visiting each waypoint, the re-planning criterion (given by Fig.4) is investigated. A computation cost encountered any time that re-routing is required. Thus, the total cost $\text{Cost}_{\text{Total}}$ for the model is defined as:

$$
\text{Cost}_{\text{Total}} = M_{\text{Cost}} \times f(\psi_{\text{Cost}}) + \sum_{i=1}^{\text{rep}} \phi_{\text{CPU}} + \sum_{i=1}^{\text{compute}} (M_{\text{CPU}})
$$

where $\text{rep}$ is the number of repeating the mission re-planning procedure.

3 Methodology Adopted by Higher/Lower Level Modules of the Architecture

There is a significant distinction between theoretical understanding of meta-heuristics and peculiarity of different applications in contrast. Specifically this gap is more highlighted when scale (size), complexity, and nature of the problem is taken to account. Application of different methods may result very diverse on a same problem due to specific nature each problem. To handle the complexity of NP-hard graph routing and task allocation problem, the TOM-P module utilizes the Ant Colony Optimization (ACO) algorithm to find an optimum order of tasks for the underwater mission. In the LOP-P module, Firefly Algorithm (FFA) is conducted to carry out path planning between each pair of the waypoints, which is efficient and fast enough in generating collision-free optimum trajectory in smaller scale. Similar to other metaheuristic algorithms, both ACO and FFA also have two inner loops through the population size $l_{\text{max}}$ and iteration $l_{\text{max}}$ so at the extreme case the algorithms complexity is $O(l_{\text{max}} \times l_{\text{max}})$, thus, the computation cost is considerably low as its complexity is linear to time.

3.1 Application of ACO Algorithm on Task-Organize Mission Planning Approach

The actual biological ants adopts swarm intelligence to discover the shortest possible route to food. The ACO algorithm has been introduced by Dorigo & Di Caro [37] inspiring the ants’ foraging process. This algorithm is proven to have a strong capability on solving vehicle routing problems and is well scaled with task assignment approaches due to its discrete nature [38-40]. In this algorithm, the ants find their path through a probabilistic decision according to level of pheromone concentration, so that the pheromone trail attract ants and guide them to grain source. Respectively, more pheromone is released on the overcrowded paths, in which produces a self-reinforcing process in finding the quickest route to target. The pheromone gets vaporized over time to helps ants to eliminate bad solutions that keep solutions from falling in to local optima. However, in evaporation process the pheromone concentration on optimum path still remains higher due to lower rate of evaporation comparing to pheromone deposit rate. The pheromone concentration on edges ($\tau$) preserve the ants’ collected information in searching process. For implementing ACO two main steps are required, first determination of the probability of targeting a node to travel among other nodes, which is calculated by (19), and the next is to update the pheromone trail after ant population completed their tour that is given by (20).

$$
p_{ij}^k = \frac{\left(\prod_{l \in N_i^k} \left(\frac{\tau_{ij}^l}{\tau_{ij}}\right)^{\alpha} \left(\frac{\Phi_{ij}^l}{\Phi_{ij}}\right)^{\beta}\right.}{\sum_{l \notin N_i^k} \left(\prod_{l \in N_i^k} \left(\frac{\tau_{ij}^l}{\tau_{ij}}\right)^{\alpha} \left(\frac{\Phi_{ij}^l}{\Phi_{ij}}\right)^{\beta}\right.} \quad j \in N_i^k
$$

$$
\begin{cases}
\tau_{ij} = (1-r)\tau_{ij} + \Delta\tau_{ij} \\
\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\tau_{ij}
\end{cases}
\quad \text{For nodes of the best ant}
\begin{cases}
\Delta\tau_{ij} = -\lambda \tau_{ij}(t) + \Omega \quad \\
-\Delta\tau_{ij}(t) \quad \text{Otherwise}
\end{cases}
$$

\[\text{Fig.5. Pseudo-code of ACO mechanism on task organizing}\]

here, $k$ is the ant index, $i$ and $j$ denote the index of current and the target nodes, $\tau_{ij}^l$ is the pheromone concentration on edge between nodes $i$ and $j$, $\alpha$ and $\beta$ are the pheromone and heuristic factors, respectively, $\Phi_{ij}^l$ denotes the heuristic information in $l^{th}$ node. $N_i^k$ indicates the set of neighbor of ant $k$ when located at node $i$. The $r \in (0, 1]$ is evaporation rate. The $\Delta\tau_{ij}^{(k)}$ denotes the pheromone level collected by the best ant $k$ based on priority of ants released pheromone ($\Omega$) on the selected nodes. $\tau_{ij}^l(t)$ and $\tau_{ij}^l(t+1)$ are the previous and current level of the pheromone between node $i$ and $j$. $Q$ is the amount of released...
pheromone on nodes taken by the best ant(s), and λ is the coefficient of pheromone evaporation. The procedure of the ACO based mission planning is described by Fig. 5. Initializing the ant population is the most critical step in implementing ACO framework. In this research, the ant population initialized with feasible waypoint sequence using a random priority vector and adjacency information of the operation graph (for more detail refer to [30]). The feasibility of the generated solutions is checked during the process of cost evaluation.

3.2 Application of FFA on the Path Planning Approach

The FFA is another swarm-intelligence-based meta-heuristic algorithm introduced by Xin-She Yang [41,42]. This algorithm inspired from the flashing patterns of fireflies, in which the fireflies attracted to each other based on brightness and regardless of their sex. As the distance of fireflies increases their brightness gets dimmed. The less bright firefly approaches to the brighter one. Attraction of each firefly is proportional to its brightness intensity received by adjacent fireflies and their distance L. The distance of L between two fireflies of \( x_i \) and \( x_j \) can be calculated as follows:

\[
L_{ij} = \sqrt{(x_j - x_i)^2} = \sqrt{\sum_{d=1}^{d} (x_{i,d} - x_{j,d})^2} \tag{21}
\]

where \( x_{i,d} \) represents the \( d \)th component of the firefly \( x_i \) in \( d \) dimensions. The attraction factor \( \beta \) and movement of a firefly \( x_i \) toward the brighter firefly \( x_j \) is calculated by

\[
\beta = \beta_0 e^{-\frac{d^2}{2}} \tag{22}
\]

\[
\alpha_t = \alpha_0 \kappa^t, \quad \kappa \in (0,1)
\]

where \( \beta_0 \) is the attraction value at \( L=0 \), \( \alpha_0 \) is the randomization parameter that control the randomness of the movement and can be tuned iteratively. The \( \alpha_0 \) is the initial randomness scaling value and \( \kappa \) is the damping factor. The \( \zeta_d \) is a random vector generated by Gaussian distribution at time \( t \). There should be a proper balance between engaged parameters, because if the \( \beta_0 \) approaches zero, the movement turns to a simple random walk, while \( \alpha_0 \) turns to a kind of particle swarm optimization [41]. The mechanism of the algorithm propels the firefly \( i \) at \( x_i \) toward the attractive firefly \( j \) at \( x_j \) using:

\[
x_i = x_i + \beta_0 e^{-\frac{d^2}{2}} (x_j - x_i) + \alpha_t \cdot \text{rand} \cdot \frac{1}{2} \tag{23}
\]

where, rand is for generating uniformly distributed random number in [0,1]. For more detail about FFA refer to [40]. The FFA is advantageous uses an automatic subdivision approach that makes it more efficient comparing to other optimization algorithms. The attraction decreases with distance and in this context entire population automatically get subdivided, where each subgroup swarm around its local optimum and the best global solution is selected among the set of the local optimums. This fact increases convergence rate of the algorithm. Such an automatic subdivision nature motivates fireflies to find all optima iteratively and simultaneously, which makes the FFA specifically suitable and flexible in dealing with continuous problems, highly nonlinear problems, and multi-objective problems [43,44]. The control parameters in FFA can be tuned iteratively that increases convergence rate of the algorithm too. To generate path by B-Spline curves, the FFA tends to accurately locate the control points of a candidate curve in the solution space. A firefly in this context corresponds to a candidate solution (path) involving a distinct number of control points. The pseudo-code of FFA process on path planning approach is provided by Fig. 6.

4 Discussion and Analysis of Simulation Results

To validate the architecture’s performance, first higher and lower level modules are evaluated separately, then their synchronous collaboration is investigated and evaluated.

4.1 Simulation Results of LOP-P Performance in Small Scale Dynamic Environment

The purpose of the LOP-P module in this study is to find the efficient trajectory adjustments making use of desirable current field, escaping undesirable current flows, and avoid intersecting uncertain borders of static/moving obstacles. Terrain situation and updates get observed and continuously tracked using on-board sensors, such as a Horizontal Acoustic Doppler Velocity Logger. A real map data of Adelaide Vincent gulf [33] is utilized to model more realistic marine environment. A k-means clustering algorithm is employed where the \( k \) (number of clusters) is set by 3 to cluster those sections of coast, water zone, and uncertain areas of the map and transform it to matrix form where the blue sector of the image is sensed as water and defined as permissible area for vehicle’s operation. Other sections depending on their color range categorized as impassable and uncertain zones. The utilized clustering algorithm is highly efficient in recognition of mentioned sectors in any alternative map. The AUV water-referenced velocity is set on 2.5 m/s. Number of control points
for generating B-spline curves is set to alter between 5 to 8 points. The common optimization configuration of FFA is set with 100 iterations and population of 80 fireflies. The attraction coefficient base value is set on $\beta_0=2$, light absorption coefficient $\varepsilon$ is assigned with 1. The damping factor of $\kappa$ is assigned with 0.95 to 0.97. The scaling variations is defined based on initial randomness scaling factor of $\alpha_0$. The parameter of randomization is set on 0.4. The simulation results for three different cases are presented by Fig.7 to Fig.11, in which the terrain gets more complex at each case. Efficiency of the proposed method is investigated at the most sever situation that closely matches the real underwater dynamics. A simple path planner and On-line path re-planner are compared at each case to declare the necessity for existence of an adaptive re-planning procedure in such a dynamic environment.

A. Path Adaptive Behavior to Dynamic Current Variations (without obstacle)

Water current may have positive or disturbing effect on vehicles deployment. An undesirable current also can push floating obstacles across the AUVs trajectory. On the other hand, the desirable current can motivate AUVs motion toward its target, which leads saving the energy. Adapting to dynamic current is even more challenging where the current flow may immediately change its magnitude or correlation. The real-time current map information is continuously captured by on-board sonar sensors with a specific uncertainty modelled by Gaussian distribution (formulated in section 2.2.1). Vehicle’s kinematic constraints also are taken into account in path planning process and investigated in the following.

![Fig.7. Path adaption to current updates in three time steps (the color map corresponds to current magnitude. The current intensity increases as the colour gets lighter from blue to yellow). Each pixel corresponds to 7x7 m$^2$.](image)

![Fig.8.(a) Path curve and its alternative generated by simple path planner encountering information of the static current map; (b) The path adaption to current updates in three levels of On-line replanning. Each pixel corresponds to 7x7 m$^2$.](image)
The behaviour of the generated path to current updates is proposed in Fig.7 and Fig.8(a,b), in which the current map gets updated in three time steps where time interval for current update is determined by division of the number of iterations to current update rate \( (U_s^f=3) \) that is set and initialized by the operator. Size of the map used in LOP-P is set on 500×500 pixels and the current field computed from a random distribution of 3 to 6 Lamb vortices in 500×500 grid; hence, each pixel in Fig.7 and Fig.8 contains a current arrow and corresponds to 7×7 m². Fig.7 presents the path curve deformation in three step current update in which the colour map presents the current magnitude and the current intensity increases as the arrows approach to the yellow areas. It is noted from Fig.7 that the generated path accurately avoids the higher intensity adverse current arrows; more specifically, it is clear in second and third time steps that the path curve is mounted on accordant current arrows to take energy efficient distance to the target point. For better presentation Fig.8 (a,b) is provided to show accuracy of the generated path by LOP-P module in which current updates and re-planned paths are presented by the same colour. In Fig.8(b) the blue arrows correspond to first current map in which the generated primary path is drawn with the same colour accordingly; afterward, the second current map arises that presented by green arrows, where the path planner tends to correct the primary path (blue) from the existing location to the target point in accordance with second current map presented with green line. In the third update, the On-line path planner again takes the accurate trajectory to the destination in accordance with third current map, where both corrected trajectory and last updated current map are presented with dark purple colour. Considering the path deformation it is noteworthy to hint the efficiency of the proposed method in adapting prompt current changes whether in avoiding undesirable turbulent or driving the desirable flows.

B. Path Adaptive Behavior in Complex Environment Including Dynamic Current and Uncertain Static-Moving Obstacles

To test the performance of the adopted method by LOP-P, the operating field is modelled to be more complex encountering static obstacles (presented by black circles) and static current (given in Fig.9(a)) and uncertain moving obstacles in presence of dynamic current (given in Fig.9(b)). In Fig.9 a set of pareto-optimum trajectories are generated in accordance with environmental situation in which the best generated path is proposed by thicker line and the sub optimum trajectories are presented by a thinner lines. In Fig.9(b) current map is updated for three times and the path re-planned adaptively at each current update. Further to current update, the uncertain obstacles are also taken into account, in which some of them are static and uncertain in position where their uncertainty grows by time in a circular format; and some others are floating with current force with uncertainty on their position and motion (as the current flow also contains uncertainty) that presented by moving circles with varying radius. Similar to previous case, each pixel in Fig.9 contains a current arrow.

![Fig.9(a)](image1)

(A) A pareto-optimum path curves encountering static current map and randomly generated static obstacles; (B) The path adaption to dynamic current and uncertain static/moving objects along with On-line replanning, where the current updates and the corresponding path deformation is presented in the same colour. Each pixel corresponds to 7×7 m².

It is derived from simulation results in Fig.9(a) the proposed simple path planner accurately handles the problem’s objectives in a static situation based on offline static current map data; however, for handling more complex situations adaptive re-planning would be necessary to correct the previous optimum trajectory according to terrain updates. As presented in Fig.9(b), the On-line re-planning accurately adapts the changes of the environment as the primary blue trajectories that generated based on first current map (blue arrows) are switched to green lines from the existing point to the target point in accordance with second current map (green arrows) and obstacles movement/deformation. The purple lines are the final correction to the trajectories based on latest update of the current map (purple arrows) and obstacle’s situation. Obviously, the employed strategy satisfies the path planning objectives at this stage; however, for gaining more confidence on efficiency of the applied method, the terrain gets even more complex to closely mimic the real-world underwater terrain that investigated in the next section.

C. On-line Path Planning in 3-D Complex Environment Encountering Real Map Data, Dynamic Current and Obstacles

Similar to case (B), both static and dynamic ocean currents are applied and different type of obstacles are considered for final evaluation of the adopted method by LOP-P module. To be more confident on accuracy of the proposed method, a part
of a real map also adopted for more realistic implementation and k-means clustering is applied to clarify water covered zone as permissive space for vehicle’s deployment and coastal areas as the forbidden zones and transform it in to a matrix format, in which the water covered sections filled by value of 1 in the generated matrix and the other sections filled by zero for coastal area or a small decimal value in range of $[0.01, 0.3]$ corresponding to recognized uncertain sections of the map.

It is clear from Fig. 10(a,b) the path planner accurately recognizes coastal areas and avoid colliding forbidden edges. Similar to results performed in previous sections the method is resistant to increasing the complexity of the terrain as it is accurately guide the vehicle toward the target point even in presence of time-varying current and uncertain static/moving obstacles. The path accurately reformed from existing point according to current update. For better presentation and easy understanding, each deformation to the path curve is presented by the same color of the updated current map. Considering Fig. 10(b) the deformed path at each replanning process accurately copes with the new current map and takes the most accordant trajectory to the desirable current flow that leads considerable save to the battery usage. On-line path re-planning uses the previous planning history to achieve more optimized path and save the computation time.

**Path Efficiency in Satisfying Vehicle’s Kinematic Constraints Incorporating Current Effects on AUV’s Motion**

The vehicular constraints of the AUV and boundary conditions on state of the vehicle also are taken into account for realistic modelling of the AUV’s operation. In fact, the optimum trajectory is obtained thru the adjustments of the path curvature considering the current effects on vehicles motion and collision boundaries. The performance of the applied method in satisfying vehicular constraints is presented by Fig. 11 using compact boxplots over 100 iterations. The displayed four parameters present vehicle’s orientation and velocity components at each single spot on the generated path curve. The red dashed line in each plot represents the violation boundary, while the blue boxes corresponds to average iterative variations of the surge, sway, yaw and theta parameters of corresponding path population. The results presented by Fig.11 reveals that this model efficiently keeps the variations of the vehicular parameters valid to the specified boundaries as all generated solutions for all four aforementioned parameters are bundled between defined borders (red horizontal lines).
The shortcomings with the LOP-P raises when vehicle is required to operate in a large scale terrain (i.e. 10 km² or even more), as it requires to consider and estimate dynamicity of the terrain adaptively. Hence, when the scale is larger the data load for adaptive re-planning progressively increase, so that the computation time extremely increases, which is a considerable defect for real-time system. Moreover, path planner is an estimate based approach and it is far from reality to give proper estimation on current or terrain behavior in hundreds of meters away from the vehicle. On the other hand this approach only deals with vehicles guidance from one point to another and does not deal with mission scenario or task assignment considerations. To cover the mission scenario and tasks priority assignment and also to handle the shortcomings of the local path planning, the TOM-P operates in a higher level to give a general overview and cut off the operating area to beneficial zones for vehicles deployment in the feature of the global route (waypoint sequence).

4.2 Simulation Results of TOM-P Performance on Task Organize and Mission Plan in a Large Scale Graph-like Terrain

The TOM-P module in this research uses prepared information about terrain, waypoints location, tasks that assigned to network’s adjacent connections and their characterization, and time/battery restrictions to compute the most appropriate order of prioritized tasks to guide the AUV toward its destination in a waypoint covered large scale terrain. There should be a compromise among the mission available time, maximizing number of highest priority tasks, and guaranteeing reaching to the predefined destination before the vehicle runs out of battery/time. The ACO algorithm is employed and tested by TOM-P module to evaluate its stability and performance in determining efficient feasible solutions and beneficiary ordering of the tasks. Several performance metrics are utilized to measure the performance of the method in a quantitative manner and the simulation results presented by Fig.12 to Fig.13. Also different criteria are embedded to keep the generated solutions concentrated to the feasible space. The ACO is configured with 100 iterations, ant population of 70. The pheromone value initialized with \( \tau = 1.5 \), the pheromone exponential and heuristic exponential weights are initialized with 1.5 and 0.9, respectively. The pheromone and heuristic factors are updated iteratively according to \( \alpha = 0.99 \times \alpha - 1 \) and \( \beta = 0.99 \times \beta - 1 \). The evaporation rate is assigned with 1.8. The performance and stability of the applied method is investigated on different network topologies, in which network complexity and size increases incrementally from 20 to 150 nodes (waypoints). The cost function used for TOM-P is a weighted function of the mission total obtained weight, mission time, distance travelled, and number of tasks.

As presented in Fig.12(a), the algorithm accurately manages the complexity growth as the cost value decrements iteratively regardless of size and topology complexity of the network. Considering Fig.12(b), the computational time increments linearly with growth of the graph nodes and the CPU time for all samples remain within the bounds of a suitable real-time solution. The cost and CPU-Time variations in Fig.12(a,b) declares that the performance of the model is almost independent of both size and complexity of the graph, whereas this is recognized as a problematic issue in many of the outlined approaches. The Fig.13 indicates the iterative variation of cost and violation of the solutions in satisfying specified objectives and constraints.

It is further inferred from Fig.13(a,b), the algorithm efficiently tends to minimise the cost over 100 iterations and the cost variation range is reduced iteratively which means the population effectively converges to the optimum solution with the minimum cost. The violation diagram shows how the generated solutions respects the defined restriction (restriction of mission time to available time and feasibility criteria). Tracking the variation of red crosses in the middle of the error bar graphs that corresponds to mean cost and mean violation, declares that the algorithm accurately enforces the solutions to approach the least cost and manage the solutions to eliminate the violation as the range of violation variations converges to zero within 100 iterations.
The 200 Monte Carlo simulation runs are carried out and performed by Fig.14 to show the stability of the applied method in a quantitative manner, in which the variation ranges of performance metrics of total obtained weight, completed tasks, total travelled time and distance, CPU time, and mission cost are presented. By analyzing the result of the Monte Carlo simulations, it is possible to be confident in the robustness and efficiency of the method in dealing with random transformation of the graph size and topology. For all Monte Carlo runs, the quantity of waypoint is set to be changed with a uniform distribution between 20 to 50 nodes and the network topology also transforms randomly with Gaussian distribution on the problem search space. The time threshold is set on 3.6×10^4(sec). According to results presented in Fig.14, the average travelled distance is approximately drawn around 80 km and within this distance nearly 8 to 16 number of tasks is completed with the average weight about 30 to 40 worth. Fig.14 shows consistency of the generated solutions by ACO in its distribution dealing with problem space deformation.

4.3 Architecture’s Performance on Scheduling and Time Management for Reliable and Efficient Operation

At the top level, the TOM-P uses offline map in the large scale to find the best fitted task sequence to the predefined available time in order to maximize the mission productivity. On the other hand, the autonomous vehicle needs to perceive the new information of the dynamic environment, comprehend the relation between perceived information and meaning of different raised situation to be able to handle raised unexpected changes of the terrain. Hence, dynamic changes of the operation field that handled by LOP-P module in smaller scale.

Quality of the generated local path also affects the optimality and productivity of the TOM-P generated mission plan, in which the mission re-planning is performed to rearrange the tasks order or cut off some of them in cases that available time is expended more than expectation for incorporating raised changes. Considering updated available time and environmental changes, the TOM-P is dynamically re-compute the mission plan until the vehicle reaches to the target point.

The particular objective of the entire architecture is to guarantee a reliable mission and accurate arrangement of the prioritised tasks so that the vehicle is guided toward the destination while takes the maximum use of available time and reach the destination before it runs out of time. The architecture capability in time management and guarantying on-time termination of the mission is investigated with 10 individual experiments as is shown in Fig.15. To establish an accurate and stringent concordance among different components of the architecture and real-time implementation of the model some other performance indexes should be highlighted additional to those which addressed in two previous sections. The first critical factor for both TOM-P and LOP-P operation is having a short computational time to provide a concurrent synchronization between modules and balancing the correlation between their operations. A fast operation for each component of the main architecture prevent any of them from dropping behind the process of the other components, since appearance of such a delay interrupts the routine flow and concurrency of the entire system. Another significant performance indicator is compatibility of the value of the local path time (T_p) and the expected time (T_c) in simultaneous operation of the LOP-P module in context of the main architecture; hence, a big difference between values of T_p and T_c is not acceptable. This issue is critical for recognizing the requisition for re-routing and re-scheduling the tasks and again influences the balance and synchronism of the whole system. To validate the performance of the introduced architecture in a quantitative feature, the concurrent and collaborative procedure of engaged modules toward satisfying addressed objectives is investigated through multiple experiments with the initial condition analogous to real underwater mission scenarios indicated by Table.2 and Fig.15. Accordingly, the operation network is initialized with 50 waypoints and 1600 edges, in which the start and final destination waypoints are assigned with nodes 1 and 50, respectively. A fixed collection of prioritized tasks is specified and randomly assigned to some of the edges. The edges that doesn’t assigned with a task are weighted with 1. The location of nodes are randomized according to \{P_i\sim U(0,10000)\} and \{P\sim U(0,100)\}. The T_Total is initialized with 10800(sec) for all experiments. The terrain is modelled as a realistic underwater environment, in which time varying ocean current and randomly generated uncertain static/moving obstacles are included.

![Statistical analysis of ACO in terms of satisfying given performance metrics in 200 Monte Carlo simulations](image)
Table 1. Notation representing the elements of Table 2 (to prevent confusion)

| \( N_{\text{TPU}} \) | \( T_{\text{TPU}} \) | \( T_{\text{MP}} \) | \( \Phi_{\text{MP}} \) | \( P_{\text{MP}} \) | \( T_{\text{Total}} \) | \( \Phi_{\text{Local}} \) | \( P_{\text{Local}} \) | \( T_{\text{Total}} \) | \( \Phi_{\text{FM}} \) | \( P_{\text{FM}} \) | \( T_{\text{Total}} \) | \( \Phi_{\text{Valid}} \) | \( P_{\text{Valid}} \) | \( T_{\text{Total}} \) | \( \Phi_{\text{Total}} \) | \( P_{\text{Total}} \) | \( T_{\text{Total}} \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1               | 2               | 3               | 4               | 5               | 6               | 7               | 8               | 9               | 10              | 11              | 12              | 13              | 14              | 15              | 16              | 17              | 18              |

Table 2. Collaborative and Concurrent process of the TOM-P and LOP-P modules in the proposed architecture.

| \( \Phi_{\text{Local}} \) | \( P_{\text{Local}} \) | \( T_{\text{Total}} \) | \( \Phi_{\text{FM}} \) | \( P_{\text{FM}} \) | \( T_{\text{Total}} \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1               | 2               | 3               | 4               | 5               | 6               |

Table 3(a). Experiment-1

| \( \Phi_{\text{Local}} \) | \( P_{\text{Local}} \) | \( T_{\text{Total}} \) | \( \Phi_{\text{FM}} \) | \( P_{\text{FM}} \) | \( T_{\text{Total}} \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1               | 2               | 3               | 4               | 5               | 6               |

Table 3(b). Experiment-2

| \( \Phi_{\text{Local}} \) | \( P_{\text{Local}} \) | \( T_{\text{Total}} \) | \( \Phi_{\text{FM}} \) | \( P_{\text{FM}} \) | \( T_{\text{Total}} \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1               | 2               | 3               | 4               | 5               | 6               |

Table 3(c). Experiment-3

| \( \Phi_{\text{Local}} \) | \( P_{\text{Local}} \) | \( T_{\text{Total}} \) | \( \Phi_{\text{FM}} \) | \( P_{\text{FM}} \) | \( T_{\text{Total}} \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1               | 2               | 3               | 4               | 5               | 6               |

Table 3(d). Experiment-4

| \( \Phi_{\text{Local}} \) | \( P_{\text{Local}} \) | \( T_{\text{Total}} \) | \( \Phi_{\text{FM}} \) | \( P_{\text{FM}} \) | \( T_{\text{Total}} \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1               | 2               | 3               | 4               | 5               | 6               |

Table 3(e). Experiment-5

| \( \Phi_{\text{Local}} \) | \( P_{\text{Local}} \) | \( T_{\text{Total}} \) | \( \Phi_{\text{FM}} \) | \( P_{\text{FM}} \) | \( T_{\text{Total}} \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1               | 2               | 3               | 4               | 5               | 6               |
Table 2 clarifies the progress of the architecture in different stages of a specific mission toward fulfilling the addressed objectives, in which the mission starts with operation of the TOM-P for generating the initial mission plan that takes maximum use of the available time. Therewith, the LOP-P starts.
its process by generating time optimum collision free path between locations of the first pair of waypoints from the sequence of nodes. When the generated local path is traversed and the targeted waypoint is visited, the expended time for traversing the generated path gets compared to the expected time, in which the $T_\epsilon$ for each edge is drawn from the estimated mission time $T_M$. Mission re-planning flag gets zero in cases that $T_P$ is smaller than $T_\epsilon$, which means the current mission plan is still valid and optimum, so the LOP-P can continue its process of next pair of waypoints; however, in the opposite case, the Mission re-planning flag gets one, the traversed edges get eliminated from the network and the TOM-P is recalled to reorganize the task sequence and generate a new route from the current waypoint based on updated available time and terrain. This synchronous process between TOM-P and LOP-P frequently repeated until the AUV reaches the destination that means mission success or available time gets minus value, which means the vehicle runs out of battery. As declared in Table.2 and Fig.15, in all of the experiments the $T_{Total}$ got a positive value which means no failure is occurred in this simulation. This issue is remarkable for having a confident and reliable mission as the failure is not acceptable for AUVs due to expensive maintenance in sever underwater environment. The mentioned process is clearly presented in each of the above experiments in Table.2.

An efficient synchronization flows between higher and lower level modules to incorporate dynamic changes of the terrain and rearrange priority of the tasks. The best possible performance for the proposed model is completion of the mission with a minimum positive available time, which means the vehicle took maximum use of the available time and terminated its mission before runs out of battery. The update process of the mission available time that presented in Fig.15, shows that maximum use of available time has been taken as the available time in the last recall of the LOP-P remarkably approaches to zero in all experiments, which confirms supreme performance of the proposed novel architecture in mission reliability and excellent time management. Moreover, the results presented in Table.2 shows a very short computational time for both TOM-P and LOP-P in all experiments that confirms real-time performance of this architecture, which makes it remarkably appropriate for real-time application.
As mentioned earlier, re-routing is required any time that $T_p$ exceeds $T_r$ and mission available time gets updated after each recall to LOP-P module. It further noted from simulation results in Fig.15 that excellent cooperation flows between TOM-P and LOP-P as the correlation between value of $T_p$ and $T_r$ is very close and accordant in multiple process of LOP-P in each experiment. Variations of $T_p$ and $T_r$ in Fig.15 reveals the stability of the architecture as the obtained results are quantitatively very similar in all experiments. This strategy guarantees on time termination of the mission.
5 Conclusion and Future Works

Generally, the AUV’s control architecture consists of two different execution layers based on its control requirements i.e. deliberative and reactive layers. The deliberative layer manages the concurrent execution of several tasks with different priorities. The reactive layer carries out the real-time reactions to perform quick respond to critical sudden events. These automatic functions are executed in the background and promote AUVs self-management characteristics. To provide a higher level of autonomy, cooperative hybrid architecture has been developed to promote vehicles capabilities in decision-making and situational awareness.

The aim of this research was to introduce a novel synchronous architecture including higher level task-organizer/mission-planner and lower level On-line path planner to provide a comprehensive control on mission time management and performing beneficial mission with the best sequence of tasks fitted to available time, while safe deployment is guaranteed at all stages of the mission. In this respect, the top level module of the architecture (TOM-P) is designed to assign the prioritized tasks in a way that the selected edges in a graph-like terrain lead the vehicle toward its final destination in predefined restricted time. Respectively, in the lower level the LOP-P module handles vehicle safe deployment along the selected edges in presence of severe environmental disturbances, in which the generated path is reactively corrected and refined to cope the sudden changes of the terrain. Precise and concurrent synchronization of the higher and lower level modules is the critical requirement to preserve the accretion and cohesion of the system that promotes stability of the architecture toward furnishing the main objectives addressed above. Developing such a structured modular architecture is advantaged to be upgraded or modified in an easier way that promotes reusability of the system to be implemented on other robotic platforms such as ground or aerial vehicles. In this way, the architecture also can be developed by adding new modules to handle vehicles inner situations, fault tolerant or other consideration toward increasing vehicle’s general autonomy. To examine the overall performance of the newly proposed architecture, first the performance of the modules are evaluated separately using ACO algorithm in context of the TOM-P module and FFA algorithm in context of the LOP-P module; afterward, the performance and robustness of the whole system is examined an evaluated through testing of 10 missions’ in 10 individual experiments. It is noteworthy to mention from analyse of the results of either TOM-P, LOP-P, and architecture, the proposed strategy enhances the vehicle’s autonomy in time management and carrying out safe, reliable and efficient mission for a single vehicle operation in a limited time interval. Besides, it is inferred from analyze of the simulation results that the variation of computational time for operation of both TOM-P and LOP-P modules is settled in a narrow bound in range of seconds for all experiments that confirm accuracy and real-time performance of the system. More importantly, referring to results of 10 experiments in Table.2, the violation value is almost zero for operation of both modules claiming that the system is robust to the variations of the environment while operating in irregularly shaped uncertain spatiotemporal environment in presence of the time-varying water current and other disturbances.

For the future work, it is targeted to integrate the inner modules of the architecture to cover internal situation awareness of the vehicle and promoted vehicles ability in risk management as its failure is not acceptable due to expensive maintenance. Also several other algorithms will be implemented and tested on inner modules of the architecture to investigate the total independency of the architecture’s performance from the adopted methods.

References

[1] Furkong, M.E., Paxton, D., Stevenson, P., Pebody, M., McPhail, S.D., Perrett, J. (2012). Autosub Long Range: A long range deep diving AUV for ocean monitoring. Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES Conf., Southampton, UK, pp. 1–7. (doi:10.1109/AUV.2012.6380737).
[2] Hobson, B., Bellingham, J.G., Kieft, B., McEwen, R., Godin, M., Zhang, Y. (2012). Tethys class long range AUVs-extending the endurance of propeller-driven cruising AUVs from days to weeks. Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES Conference, pp. 1–8.
[3] Johnson, N., Patron, P., Lane, D. (2007). The importance of trust between operator and auv: Crossing the human/computer language barrier. IEEE Oceans Europe, pp. 1159–1164.
[4] Johnson, N.A., Lane, D. (2008). An architecture for planner based control of autonomous vehicles. 27th Workshop of the UK Planning and Scheduling Special Interest Group, 2008.
[5] Liu, Y., Bucknell, R. (2015). Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. Ocean Engineering 97, pp. 126 –144.
[6] Eichhorn, M. (2015). Optimal routing algorithms for autonomous underwater vehicles in time-varying environment. Robotics and Autonomous Systems, Vol. 67, pp. 33–43.
[7] Kladis, G.P., Economou, J.T., Knowles, K., Lauber, J., Guerra, T.M. (2011). Energy conservation based fuzzy tracking for unmanned aerial vehicle missions under a priori known wind information. Engineering Applications of Artificial Intelligence, Vol. 24 (2), pp. 278–294.
[8] Kwok, K.S., Driessen, B.J., Phillips, C.A., Tovey, C.A. (2002). Analyzing the multiple-target-multiple-agent scenario using optimal assignment algorithms. Journal of Intelligent and Robotic Systems, Vol. 35(1), pp. 111–122.
[9] Liu, L., Shell, D.A. (2012). Large-scale multi-robot task allocation via dynamic partitioning and distribution. Autonomous Robots, Vol. 33(3), pp. 291–307.
[10] Higgins, A.J. (2001). A dynamic tabu search for large-scale generalised assignment problems. Computers & Operations Research, Vol. 28(10), pp. 1039–1048.
[11] Lyrgaard, J., Letchford, A.N., Eglese, R.W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming, Vol. 100(2), pp. 423–445.
[12] Al-Hasan, S., Vachtsevanos, G. (2002). Intelligent route planning for fast autonomous vehicles operating in a large natural terrain. Elsevier Science B.V., Robotics and Autonomous Systems, Vol.40, pp.1–24.
[13] Gudaitis, M.S. (1994). Multicriteria Mission Route Planning Using a Parallel A* Search. M.Sc. Thesis, Air Force Institute of Technology, Air University.
[14] Pereira, A.A., Binney, J., Hollinger, G.A., Sukhatme, G.S. (2013). Risk-aware Path Planning for Autonomous Underwater Vehicles using Predictive Ocean Models. Journal of Field Robotics, Vol.30(5), pp.741–762.
[15] Sharma, Y., Saini, S.C., Bhandhari, M. (2012). Comparison of Dijkstra’s Shortest Path Algorithm with Genetic Algorithm for Static and Dynamic Routing Network. International Journal of Electronics and Computer Science Engineering, ISSN-2277-1956/VIN2, pp.416-425.
Based on Improved Ant Algorithm for Vehicle to Variable,

Vol. 29(12), pp. 1475–1486.

IEEE Congress on Evolutionary Computation (CEC), Vancouver, Canada. July 24-29.

3D field D^3: improved path planning and replanning in three dimensions. IEEE International Conference on Intelligent Robots and Systems, pp. 3381-3386.

Energy-efficient path planning for fully propelled AUVs in congested coastal waters. Oceans 2013 MTS/IEEE Bergen: The Challenges of the Northern Dimension.

Underwater path planning using fast marching algorithms. Oceans Europe Conf., Brest, France, Vol. 2, pp. 814–819.

Path planning for autonomous underwater vehicles. IEEE Transactions on Robotics, Vol.23 (2), pp.331-341.

Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. IEEE Transactions on Industrial Informatics, Vol.9(1), pp.132–141. Doi: 10.1109/TII.2013.2198665.

Toward Efficient Task Assignment and Motion Planning for Large Scale Underwater Mission. Robotics (cs.RO). arXiv:1604.04854

AUV Rendezvous Online Path Planning in a Highly Cluttered Undersea Environment Using Evolutionary Algorithms. Robotics (cs.RO). arXiv:1604.07002

Differential Evolution for Efficient AUV Path Planning in Time Variant Uncertain Underwater Environment. Robotics (cs.RO). arXiv:1604.02523

Evolutionary Algorithm Based Offline/Online Path Planner for UAV Navigation. IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol.33(6), pp.898–912. Doi: 10.1109/tmbc.2002.804370.

Evolutionary path planning for autonomous underwater vehicles in a variable ocean. IEEE Journal of Oceanic Engineering, Vol. 29(2), pp.418–429. doi:10.1109/joe.2004.827837.

Three-dimensional optimal path planning for waypoint guidance of an autonomous underwater vehicle. Robotics and Autonomous Systems, Vol. 67, pp 23–32.

Shell space decomposition based path planning for AUVs operating in a variable environment. Ocean Engineering, Vol. 91, pp. 181-195.

A Novel Versatile Architecture for Autonomous Underwater Vehicle's Motion Planning and Task Assignment. Robotics (cs.RO). arXiv:1604.03308v2

Biogeography-Based Combinatorial Strategy for Efficient AUV Motion Planning and Task-Time Management. Robotics (cs.RO). arXiv:1604.04851

A Hierarchical Planning Framework for AUV Mission Management in a Spatio-Temporal Varying Ocean. Robotics (cs.RO). arXiv: 1604.07898

http://www.bom.gov.au/australia/satellite/

Bluelink ocean data assimilation system (BODAS). Journal of Ocean Modelling, Vol. 21(1–2), pp.46–70.

Ant algorithms for discrete optimization. Journal of Artificial Life, Vol.5(2), pp.137 – 172.

Multi-Hop Routing Optimization Method Based on Improved Ant Algorithm for Vehicle to Roadside Network. Journal of Bionic Engineering, Vol.11, pp. 490–496. Online publication date: 1-Jul 2014.

The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach. Swarm and Evolutionary Computation, Vol.21, pp.41-53.

An ant colony optimization heuristic for constrained task allocation problem. Journal of Computational Science, Vol.7, pp.37-47.

Firefly algorithms for multimodal optimization. 5th Symposium on Stochastic Algorithms, Foundations and Applications, (Eds. O. Watanabe and T. Zeugmann), Lecture Notes in Computer Science, 5792: 169-178 (2009).

Nature-Inspired Metaheuristic Algorithms. Luniver Press, UK, 2nd edition, 2010.

An Efficient Hybrid Route-Path Planning Model for Dynamic Task Allocation and Safe Maneuvering of an Underwater Vehicle in a Realistic Environment. Robotics (cs.RO). arXiv:1604.07545

Firefly Algorithm: Recent Advances and Applications. International Journal of Swarm Intelligence, Vol. 1(1), pp. 36-50.