

2019-04-26

# Cooperative path planning of multiple autonomous underwater vehicles operating in dynamic ocean environment

Zhuang, Y

<http://hdl.handle.net/10026.1/13805>

---

10.1016/j.isatra.2019.04.012

ISA Transactions

Elsevier

---

*All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.*

# **Cooperative path planning of multiple autonomous underwater vehicles operating in dynamic ocean environment**

**Abstract:** This paper presents a two-stage cooperative path planner for multiple autonomous underwater vehicles operating in dynamic environment. In case of static environment, global Legendre pseudospectral method is employed for collision-free paths of vehicles for the purpose of minimum time consumption and simultaneous arrival. Moreover, in order to keep the multiple autonomous underwater vehicles safe from collisions on the path segments connecting two adjacent control nodes, an adaptive intermediate knots insertion algorithm is introduced. In the on-line planning stage, the local re-planning strategy aims at avoiding collisions with unexpected dynamic obstacles by two consecutive avoidance maneuvers, and the differential flatness property of autonomous underwater vehicle is utilized, which can help the vehicles react fast enough to avoid moving obstacles.

**Keywords:** Cooperative path planning, Autonomous underwater vehicles (AUVs), Collision avoidance, Legendre pseudospectral method, Differential flatness

## **1. Introduction**

Research on Autonomous underwater vehicles (AUVs) has been gaining attention recently due to the increasing demand in military, scientific research and commercial applications [1]. AUVs are a class of submerged marine vehicles that can perform underwater tasks and missions autonomously, using onboard navigation, guidance, and

control systems [2]. In order to enhance the level of autonomy and operational efficiency of AUVs, automatic guidance systems or proper path planning techniques for AUVs have become crucial.

In general, the path planning problem of AUVs can be regarded as an optimization process, in which the main challenge is to avoid the collisions with obstacles and allow a capable guidance for AUVs in complicated ocean environment. There have been a wide variety of algorithms employed to solve the path planning problems for AUVs, including sampling-based algorithm [3], graph search method [4], [5], artificial potential field algorithm [6], [7] and evolutionary algorithm [8]-[11], etc. In most practical applications, AUVs have to operate in unknown and potentially cluttered and dynamic environments, thus real-time path planning is of primary importance to ensure safe and efficient operations [12]-[14]. A detailed overview on path planning of AUVs can be found in [15].

Among the work reviewed, the majority only focused on the case of single AUV. In the past decade, motivated by increasingly complex and challenging missions at sea, there is widespread interest in the development of advanced techniques for multiple AUVs [16]-[19]. The core idea of multiple AUVs cooperation is to use a fleet of relatively small, simple and inexpensive AUVs to take places of specialized and expensive ones to solve underwater missions cooperatively. Simultaneous use of multiple AUVs can improve performance and robustness, as well as reducing cost; however, little attention has been paid to this problem due to the high uncertainty and complexity of the realistic ocean environment. The algorithm proposed in [20]

integrated the task assignment with path planning of multiple AUVs, which aimed to arrange a team of AUVs to reach all the appointed dynamic targets in 3-D underwater environments with obstacle avoidance without speed jump. A path planner for rendezvous of multiple AUVs and autonomous surface vessels using a distributed shell-space decomposition scheme was specified in [21], which combined with a B-spline-based quantum particle swarm optimization algorithm. Then, the rendezvous trajectories could be generated by considering both the capabilities of each vehicle and the dynamic environment. In the previous work, the dynamic models of AUVs have not been considered, and the proposed algorithms were only designed for off-line cases. Nevertheless, in practice, an on-line AUV guidance strategy is always required to regenerate paths during the course of the mission in any unstructured or unpredictable environment. Formation path planning problem of multiple unmanned surface vehicles (USVs) in realistic ocean environment using fast marching method is discussed in [22]. Here leader-follower formation control structure is adopted along with the on-line path planning scheme to largely maintain the formation shape with static and dynamic obstacles. However, the obtained paths are not optimal or even near-optimal and neither the relative moving directions of the dynamic obstacles and the affected USVs have not been considered in the process of collision avoidance.

This paper focuses on the development of a cooperative path planner for multiple AUVs to obtain near-optimal paths in rich obstacles ocean environment. In order to improve the autonomy, the proposed path planning algorithm is designed to work off-line and on-line. In the off-line stage, the paths can be pre-programmed according to

the original ocean environment information, which is completely known as *a priori* before the mission starts, and pseudospectral method will be employed to tackle the corresponding optimization problem. It is always a challenge to choose a proper number of control nodes. A large number of control nodes will be beneficial to collision avoidance for the AUVs with static obstacles as well as other members in the fleet, and furthermore the required accuracy can be easily satisfied. However, more control nodes will also cause inefficiencies and increase optimization complexity, and even make the optimization fail in finding an optimum. Therefore, in the off-line path planning process, an adaptive knots insertion pseudospectral method will be employed, which can achieve the requirements for collision avoidance and accuracy criteria by inserting proper number of knots at proper locations. Then, if unforeseen events occur during the mission, i.e. an unexpected obstacle suddenly pops up, the affected AUVs should have to begin regenerating paths on-line to avoid collisions, by using the continuously updated environment information from on-board sensors. Such re-planning process must be completed in real-time, and satisfies certain optimization criteria to ensure the safety and performance of the mission. Hence, the flatness property of AUVs will be introduced, and the corresponding optimization can then be solved more efficiently in flat outputs space. In addition, considering the relative orientation of the affected AUV and the corresponding dynamic obstacle, a practical cooperative re-planning strategy will be proposed.

The rest of this paper is organized as follows. Section 2 describes the problem formulation by means of the mathematical models. Section 3 proposes the cooperative

path planning algorithm for multiple AUVs with static and dynamic obstacles. Simulation results are shown in Section 4, followed by conclusions and future work in Section 5.

## 2. Problem formulation

### 2.1. Mathematical model

Let  $n$  be the number of AUVs in the fleet  $A = \{A_1, A_2, \dots, A_n\}$ . Assuming all the AUVs in the fleet are identical, then the mathematical model of the  $i$ th AUV moving in a horizontal plane is described as follows [23]:

$$\begin{cases} \dot{\boldsymbol{\eta}}_i = \mathbf{J}(\psi_i)\mathbf{v}_i \\ \mathbf{M}\dot{\mathbf{v}}_i = -\mathbf{C}(\mathbf{v}_i)\mathbf{v}_i - \mathbf{D}(\mathbf{v}_i)\mathbf{v}_i + \boldsymbol{\tau}_i \end{cases} \quad (1)$$

where,  $\boldsymbol{\eta}_i = [x_i, y_i, \psi_i]^T$  denotes the position  $(x_i, y_i)$  and the heading angle  $\psi_i$  of the  $i$ th AUV in earth-fixed reference frame;  $\mathbf{v}_i = [u_i, v_i, r_i]^T$  represents the velocity vector in the body-fixed reference frame,  $u_i, v_i$  denote linear velocities along surge and sway directions, and  $r_i$  is the rotational velocity in yaw motion;  $\mathbf{J}(\psi_i)$  is Jacobian transformation matrix;  $\mathbf{M}$  denotes system inertia matrix;  $\mathbf{C}(\mathbf{v}_i)$  is Coriolis-centripetal matrix;  $\mathbf{D}(\mathbf{v}_i)$  is hydrodynamic damping matrix;  $\boldsymbol{\tau}_i = [\tau_{u_i}, \tau_{v_i}, \tau_{r_i}]^T$  is the control vector including surge force  $\tau_{u_i}$ , sway force  $\tau_{v_i}$  and the yaw moment  $\tau_{r_i}$ .

Furthermore, with the assumptions in [24], the dynamic and kinematic equations of motion can be described as

$$\begin{cases} \dot{u}_i = v_i r_i - \frac{X_u + X_{|u|u}}{m} |u_i| u_i + \frac{\tau_{ui}}{m} \\ \dot{v}_i = -u_i r_i - \frac{Y_v + Y_{|v|v}}{m} |v_i| v_i + \frac{\tau_{vi}}{m} \\ \dot{r}_i = -\frac{N_r + N_{|r|r}}{I_z} |r_i| r_i + \frac{\tau_{ri}}{I_z} \\ \dot{x}_i = u_i \cos \psi_i - v_i \sin \psi_i \\ \dot{y}_i = u_i \sin \psi_i + v_i \cos \psi_i \\ \dot{\psi}_i = r_i \end{cases} \quad (2)$$

where,  $m$  is the mass of AUV;  $I_z$  is the moment of inertia about the Z-axis of the body-fixed frame;  $X_u / X_{|u|u}, Y_v / Y_{|v|v}$  and  $N_r / N_{|r|r}$  are damping coefficients.

## 2.2. Problem description

Generally, cooperative path planning of multi-AUV can be regarded as a multi-optimization problem subject to some certain criteria and constraints, which are imposed by the mission requirements, the physical characteristics of AUVs, as well as the ocean environment, then a set of optimal paths  $\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n\}$  can be generated.

This study considers four factors to determine the optimization criterion: time consumption over all participating AUVs  $F_1$ ; simultaneous arrival of AUVs at their selected destinations  $F_2$ ; collision avoidance with obstacles  $F_3$ ; collision avoidance with other AUVs  $F_4$ . The values of these criteria are calculated over the potential AUV's trajectories, which are approximated by Legendre polynomial on a set of Legendre-Gauss-Lobatto (LGL) nodes. The objective function in this study can then be defined as:

$$J(\mathbf{X}, \mathbf{O}) = \gamma_1 F_1(\mathbf{X}, \mathbf{O}) + \gamma_2 F_2(\mathbf{X}, \mathbf{O}) + \gamma_3 F_3(\mathbf{X}, \mathbf{O}) + \gamma_4 F_4(\mathbf{X}, \mathbf{O}) \quad (3)$$

where,  $\mathbf{X} = [X_1, X_2, \dots, X_n]^T$  and  $X_i = [\eta_i; v_i]^T, i = 1, 2, \dots, n$  is the state of  $A_i$ ;  $\mathbf{O} = \{O_1, \dots, O_m\}$  is the set of  $m$  obstacles;  $\gamma_1$  to  $\gamma_4$  represent positive weighting values

satisfying  $\sum_{j=1}^4 \gamma_j = 1$ .

(1) Objective criteria for time consumption and simultaneous arrivals

In this study, these two criteria should be both minimized in order to satisfy the task requirements. Set  $T_i$  as the time taken by  $A_i$  to arrive at its selected destination, then the optimization function for total time consumption can be expressed as:

$$F_1 = \sum_{i=1}^n T_i \quad (4)$$

Obviously, this criterion requires each participating AUV to reach its final position with minimum time consumption. Further, let  $\Delta T_i = \max(T_1, T_2, \dots, T_n) - T_i$ , which is the time taken by  $A_i$  to wait for the arrival of the last member, then the optimization function for simultaneous arrival can be described as:

$$F_2 = \sum_{i=1}^n \Delta T_i \quad (5)$$

(2) Objective criterion for obstacles avoidance

In this section only the static obstacles are considered, while the dynamic obstacles will be discussed later in Section 3. Assuming the safety distance between the  $i$ th AUV  $A_i$  and the  $k$ th obstacle  $O_k$  ( $k=1, 2, \dots, m$ ) is  $\delta_k$ . Then, the objective function for static obstacles avoidance can be defined as

$$F_3 = \sum_{i=1}^n \sum_{k=1}^m \sum_{u=0}^{N_i} f_{iO_k}^u \quad (6)$$

$$f_{iO_k}^u = \begin{cases} 1 & d_{iO_k}^u < \delta_k \\ 0, & \text{otherwise} \end{cases}$$

$$d_{iO_k}^u = \sqrt{(x_i^u - x_{O_k})^2 + (y_i^u - y_{O_k})^2}$$

where,  $N_i + 1$  is the number of control nodes on  $P_i$ ;  $(x_i^u, y_i^u)$  is the position of the  $u$ th control node on  $P_i$ ;  $(x_{O_k}, y_{O_k})$  is the position of the  $k$ th obstacle  $O_k$  and  $d_{iO_k}^u$  is the distance between  $O_k$  and the  $u$ th control node on  $P_i$ .



### (3) Objective criterion for AUVs avoidance

This factor is a cooperative constraint to guarantee the feasibility of the paths when all the AUVs moving simultaneously. With this purpose, set  $d_{ij}^{uv}$  as the distance between the  $u$ th control node on  $P_i$  and the  $v$ th control node on  $P_j$ , then the following model can be used to penalty this constraint:

$$\begin{aligned}
 F_4 &= \sum_{\forall i \neq j} \sum_{u=0}^{N_i} \sum_{v=0}^{N_j} f_{ij}^{uv} \\
 f_{ij}^{uv} &= \begin{cases} 1, & d_{ij}^{uv} < \delta_{\text{safe}} \quad \text{and} \quad |t_i^u - t_j^v| < t_{\text{min}} \\ 0, & \text{otherwise} \end{cases} \\
 d_{ij}^{uv} &= \sqrt{(x_i^u - x_j^v)^2 + (y_i^u - y_j^v)^2}
 \end{aligned} \tag{7}$$

where,  $\delta_{\text{safe}}$  is the safety distance between any two AUVs in the fleet;  $t_i^u$  is the time taken by  $A_i$  to arrival at its  $u$ th control node, and  $t_j^v$  is the time taken by  $A_j$  to arrival at its  $v$ th control node;  $t_{\text{min}}$  is a pre-determined value for user design.

Penalty functions are used to deal with collision avoidance as shown in Eqs. (6) and (7). Generally, to find out the optimal or even near-optimal solutions, the optimization algorithm has to avoid violations of the constrains caused by  $F_3$  and  $F_4$ . Further, large values for weights  $\gamma_3$  and  $\gamma_4$  can keep the optimization research far away from the high-risk areas where the constrains may be violated. Besides, it is always necessary to select the safety margins carefully in practical applications by considering the physical limitations of the vehicles (such as the velocities, turning rates, time consumption and so on). So that, even if the violations occur, there is still sufficient time for the AUVs to respond.

It can be found in Eqs. (6-7), each AUV should move out of the safety regions of obstacles and other AUVs due to a high risk of collisions. Moreover, in the cooperative

path planning problem, besides the optimization criteria for individual vehicle, more attention is paid to address cooperative behaviors. Thus, in order to fulfill the cooperative constraints defined in Eq. (7), not only the spatial constraints but also the temporal constraints should be taken into account.

With the optimization criteria defined above, the cooperative path planning problem for multi-AUV can be written as the following optimization problem:

$$\begin{aligned}
& \text{Minimize} && J(\mathbf{X}, \mathbf{O}) = \gamma_1 F_1(\mathbf{X}, \mathbf{O}) + \gamma_2 F_2(\mathbf{X}, \mathbf{O}) + \gamma_3 F_3(\mathbf{X}, \mathbf{O}) + \gamma_4 F_4(\mathbf{X}, \mathbf{O}) \\
& \text{Subject to} && \begin{cases} \dot{\boldsymbol{\eta}}_i = \mathbf{J}(\boldsymbol{\psi}_i) \mathbf{v}_i \\ \mathbf{M} \dot{\mathbf{v}}_i = -\mathbf{C}(\mathbf{v}_i) \mathbf{v}_i - \mathbf{D}(\mathbf{v}_i) \mathbf{v}_i + \boldsymbol{\tau}_i \\ \mathbf{X}_i(0) = [\boldsymbol{\eta}_i(0), \mathbf{v}_i(0)]^T = \mathbf{X}_{i,0}; \quad \mathbf{X}_i(T_i) = [\boldsymbol{\eta}_i(T_i), \mathbf{v}_i(T_i)]^T = \mathbf{X}_{i,T_i} \\ |\tau_{ui}| \leq \tau_{ui \max}, \quad |\tau_{vi}| \leq \tau_{vi \max}, \quad |\tau_{ri}| \leq \tau_{ri \max}, \quad \forall i \in \{1, 2, \dots, n\} \end{cases} \quad (8)
\end{aligned}$$

where,  $\mathbf{X}_{i,0}$  and  $\mathbf{X}_{i,T_i}$  are the initial and desired final states of  $A_i$ ;  $\tau_{ui \max}$ ,  $\tau_{vi \max}$  and  $\tau_{ri \max}$  are the maximum values of control inputs with respect to physical limitations of the thrusters mounted on the vehicles.

### 3. Cooperative path planner for multi-AUV

In path planning problems, safety always holds priority, especially for multi-AUV operating in complex ocean environment with high uncertainties. This section will deal with the cooperative path planning problem for multi-AUV in two phases: off-line and on-line. The main purpose of off-line process is to solve the optimization problem defined in Eq. (8) to generate a set of feasible and optimal paths, while the on-line process focuses on collision avoidance with unexpected dynamic obstacles by using a local cooperative planner. The flowchart for the cooperative path planning process is shown in Fig.1.

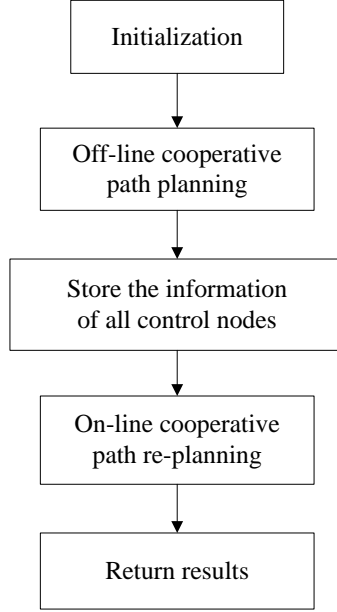


Fig. 1. Flowchart of cooperative path planning process

- (i) Initialization: load the starting and desired destination conditions of all the members in the fleet, the locations of static obstacles and the pre-determined parameters, such as  $\delta_{\text{safe}}, t_{\text{min}}$  and  $\delta_k$ ;
- (ii) Off-line cooperative path planning: according to the required waypoints distribution of the mission, a Legendre pseudospectral method based path planner integrated with an adaptive intermediate knots insertion scheme is proposed to solve the optimization problem defined in Eq. (8), which can also ensure the path segments connecting any two adjacent control nodes safe from collisions;
- (iii) Store the information of all control nodes: to reduce the bandwidth of the distributed planner, each AUV in the fleet only exchanges the information of its control nodes with other members;
- (iv) On-line cooperative path re-planning: this process employs local re-planning scheme to avoid collisions with unexpected dynamic obstacles. Once a collision is detected, i.e. an AUV moves into the safety region of a dynamic obstacle, the re-

planning process is activated to compute proper maneuver to resolve the conflict.

In this paper, the re-planning scheme can guide the affected AUV back to the original off-line path by two consecutive avoidance maneuvers. Furthermore, the differential flatness property of AUV is introduced to speed up the planning process, thus the AUV can react sufficiently fast to the changing ocean environment.

### 3.1. Legendre pseudospectral method based off-line cooperative path planner

#### (1) Legendre pseudospectral method

Legendre pseudospectral method is part of the larger theory of pseudospectral optimal control, which was originally proposed in [25]. Since then, Legendre pseudospectral method has been extended and applied for an impressive range of problems. The following section provides an overview of Legendre pseudospectral method, and more details can be found in [26].

The main idea of Legendre pseudospectral method is to parameterize the states and control inputs of the  $i$ th AUV with  $N$ th order Lagrange polynomials  $L_N$  on  $N_i + 1$  Legendre-Gauss-Lobatto (LGL) points. First, the physical domain  $t_i \in [0, T_i]$  should be mapped to a computational domain  $\sigma_i \in [-1, 1]$  by the affine transformation

$$\sigma_i = \frac{2t_i}{T_i} - 1 \quad (9)$$

The states  $\mathbf{X}_i(\sigma_i)$  can then be approximated on  $N_i + 1$  LGL points as

$$\mathbf{X}_i(\sigma_i) \approx \bar{\mathbf{X}}_i(\sigma_i) := \sum_{u=0}^{N_i} \mathbf{X}_i(\sigma_i^u) \varphi_u(\sigma_i) = \sum_{u=0}^{N_i} \lambda_i^u \varphi_u(\sigma_i) \quad (10)$$

where, LGL points  $\sigma_i^u, u = 0, 1, \dots, N_i (\sigma_i^0 = -1, \sigma_i^{N_i} = 1)$  are the roots of  $\dot{L}_{N_i}(\sigma_i)$ .  $\varphi_u(\sigma_i)$

is the  $N$ th degree Lagrange interpolating basis function defined as

$$\varphi_u(\sigma_i) = \frac{1}{N_i(N_i+1)L_N(\sigma_i^u)} \cdot \frac{(\sigma_i^2-1)\dot{L}_N(\sigma_i)}{\sigma_i - \sigma_i^u} \quad (11)$$

The first and the  $(\mu+1)$ th derivatives of  $X_i(\sigma_i)$  at the LGL point  $\sigma_i^k$  can be approximated respectively as

$$\begin{aligned} \dot{X}(\sigma_i^k) &\approx \dot{\bar{X}}(\sigma_i^k) := \sum_{u=0}^{N_i} \mathbf{D}_{1,ku} X(\sigma_i^u) = \sum_{u=0}^{N_i} \lambda_i^u \mathbf{D}_{1,ku} \\ X^{(\mu+1)}(\sigma_i^k) &\approx \bar{X}^{(\mu+1)}(\sigma_i^k) := \sum_{u=0}^{N_i} \mathbf{D}_{(\mu+1),ku} X(\sigma_i^u) = \sum_{u=0}^{N_i} \lambda_i^u \mathbf{D}_{(\mu+1),ku} \end{aligned} \quad (12)$$

where,  $\mathbf{D}_{1,ku}$  are the entries of the  $(N_i+1) \times (N_i+1)$  matrix  $\mathbf{D}_1$

$$\mathbf{D}_1 := [\mathbf{D}_{1,ku}] := \begin{cases} \frac{L_N(\sigma_i^k)}{L_N(\sigma_i^u)} \cdot \frac{1}{\sigma_i^k - \sigma_i^u} & k \neq u \\ -\frac{N_i(N_i+1)}{4} & k = u = 0 \\ \frac{N_i(N_i+1)}{4} & k = u = N_i \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

$\mathbf{D}_{(\mu+1)}$  is also an  $(N+1) \times (N+1)$  matrix, which can be easily obtained by

$$\mathbf{D}_{(\mu+1)} := [\mathbf{D}_{(\mu+1),ku}] = \mathbf{D}_1^{(\mu+1)}.$$

Using Legendre pseudospectral method based path planner, the off-line cooperative path planning shown in Eq. (8) can be further converted into a nonlinear programming (NLP), which aims at determining a set of coefficients  $\lambda = [\lambda_1^0 \dots \lambda_1^{N_1}; \lambda_2^0 \dots \lambda_2^{N_2}; \dots; \lambda_N^0 \dots \lambda_N^{N_N}]^T$ , and can be solved by a MATLAB based general commercial optimal control software package DIDO [26].

## (2) Adaptive intermediate knots insertion

In the cooperative path planner above, the collision constraints are only checked on a series of discrete-time control nodes. This collision check can be acceptable if the number of control nodes is sufficiently large; however, more nodes would lead to

inefficiencies and ill conditioning of the discrete problem. On the other hand, as shown in Fig. 2, the nodes distribution of Legendre pseudospectral method with different orthogonal polynomials have a common characteristic. They distribute densely near the two ends of the computational domain, while sparsely in center region, which would cause the gap between certain two adjacent nodes too large with a relatively small number of nodes. Hence, the path nodes may be confirmed safe from collisions, but a risk of collisions may still exist on the path segments which connect these nodes. To fix this problem, an adaptive intermediate knots insertion scheme is proposed.

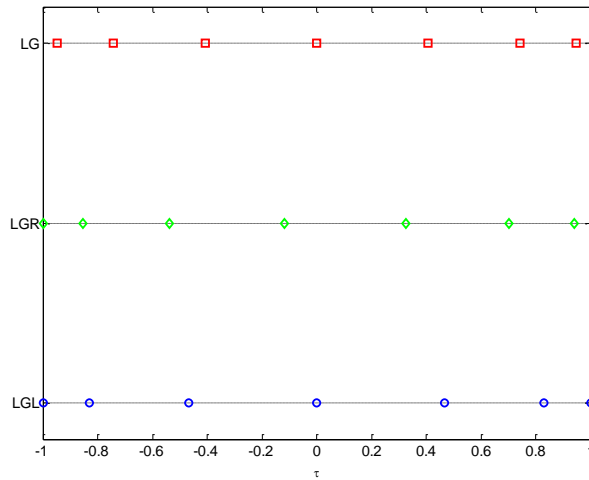


Fig.2. Distribution of nodes for Legendre pseudospectral method

It should be noted, in some practical missions AUVs may be required to visit some user-specified waypoints besides the initial and final points, which cannot be guaranteed by global orthogonal polynomials. In this paper, the pre-determined waypoints will be tackled as knots, and local pseudospectral method will be used in each segment individually. The adaptive intermediate knots insertion algorithm is shown in Table 1:

Table 1 Adaptive intermediate knots insertion algorithm

---

**Adaptive intermediate knots insertion algorithm:**

---

**Initialization:**

Set  $K_i(l)$  as the number of knots produced after the  $l$ th round of knots insertion for  $A_i$ ;  $N_{WP,i}$  is the number of waypoints for  $A_i$  to visit, and  $K_i(0) = N_{WP,i}$ ;  $S_i(l)$  is the number of segments in the  $l$ th round and  $S_{i,l}^s, s = 1, 2, \dots, S_i(l)$ ;  $N_i + 1$  is the number of LGL points on each segment and  $L$  is a parameter relatively large to ensure enough rounds of knots insertion.

**Main loop:****for**  $l = 1 : L$ **for**  $i = 1 : n$ **for**  $k = 1 : m$ **for**  $s = 1 : S_i(l)$ **do**  $c_{l-ik} \leftarrow 0$  ( $c_{l-ik} = 0$  means no collisions between  $A_i$  and  $O_k$  in the  $l$ th round, otherwise  $c_{l-ik} = 1$ ) $h_{i,s} = 0$  (number of collisions for  $A_i$  with any static obstacle on the  $s$ th segment)**for each**  $u = 1 : N_i + 1$ evaluate  $F_3(\mathbf{X}, \mathbf{O})$  on segment  $S_{i,l}^s$ **if** collision happens between the  $u$ th and the  $(u+1)$ th LGL points $c_{l-ik} \leftarrow 1$  $i(h_{i,s} + 1) = u$  $h_{i,s} = h_{i,s} + 1$ **if**  $i < n$ 

---

---

```

for  $j = (i+1):n$ 

    do  $d_{i\_ij} \leftarrow 0$  ( $d_{i\_ij} = 0$  means no collisions between  $A_i$  and  $A_j$ ,
        otherwise  $d_{i\_ij} = 1$ )

     $p_{i,s} = 0$  (number of collisions for  $A_i$  with  $A_j$ )

    for each  $v = 1:N_j + 1$ 

        if collision happens between the  $v$ th and the  $(v+1)$ th LGL points

             $d_{i\_ij} \leftarrow 1$ 

             $i(p_{i,s}+1) = v$ 

             $p_{i,s} = p_{i,s} + 1$ 

for  $i = 1:n$ 

    if any  $c_{i\_ik} == 1$ 

        Go to Algorithm 1 in Table 2 below

    else

        for  $j = (i+1):n$ 

            if any  $d_{i\_ij} == 1$ 

                if any  $c_{i\_jk} = 0$ 

                    if  $K_i(l) \leq K_j(l)$ 

                         $K_i(l) = K_i(l) + k_{i,l}^j$  ( $k_{i,l}^j$  is the number of collisions occur between  $A_i$  and
                             $A_j$  in round  $l$ )

                    else  $K_i(l) = K_i(l)$ 

                 $s = s + 1$ 

             $l = l + 1$ 

```

---



---

**return**  $K_i(l)$

---

Herein, Algorithm 1 below shows how to choose a proper set of new knots for each AUV, when they collide with static obstacles.

Table 2 Algorithm 1 for knots insertion for collisions with static obstacles

---

***Algorithm 1:***

---

Sort  $i(h_{i,s}+1)$  in ascending order as a set of possible knots, then choose the proper knots according to the following rules:

- (1) If one node appears more than once in this set, then the node must be a new knot;
- (2) If collisions are detected separately in two or more consecutive intervals (between any two LGL points) with the initial node of the first interval as  $\alpha$  and the final node of the last interval as  $\beta$ , then all the intermediate nodes will be deleted from the set, and these two end points should also be adjusted as following:
  - (i) if  $\beta \leq \lfloor N_i \rfloor$ , then delete the node  $\alpha$  in the set;
  - (ii) if  $\alpha \geq \lceil N_i \rceil$ , then delete the node  $\beta$  in the set;
  - (iii) if  $\alpha \leq \lceil N_i \rceil \leq \beta$ , then all these two end points should be chosen as new knots;

Then all  $k_{i,l}^s$  remaining nodes compose a set of new knots for  $A_i$  on segment  $S_{i,l}^s$ ,

$s=s+1$

$$K_i(l) = K_i(l) + \sum_{s=1}^{S_i(l)} k_{i,l}^s$$

---

In Algorithm 1, only some of the detected knots are taken into account, in order to improve the efficiency of the algorithm. By repeating the intermediate knots insertion process, the total number of control nodes increases, while the quantity of nodes in each

segment still keeps constant. Therefore, the risk of collision can be reduced efficiently with a relatively low computational complexity.

The flowchart for off-line cooperative path planning process is shown in Fig. 3.

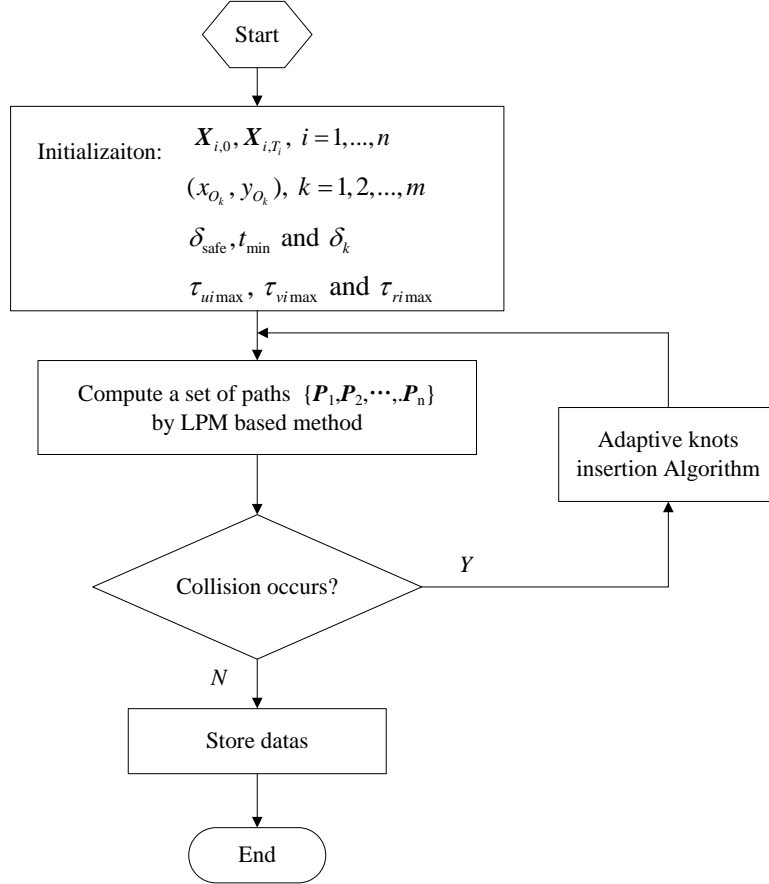


Fig. 3. Flowchart of off-line path planning process

### 3.2. On-line cooperative path planner

The previous algorithm can obtain a set of optimal paths for multiple AUVs off-line; however, the paths are not always applicable in the dynamic environment. The majority of existing algorithms focused only on the off-line implementations, since on-line path planners are always computationally expensive and their fast reaction to the changing ocean environment is very challenging. Additionally, the increased computational cost would be a heavy burden to plan multiple AUVs cooperatively.

In this paper, when multiple AUVs following the off-line optimal paths detect dynamic obstacles, an on-line path planner starts to work. Furthermore, the differential flatness property of AUV will be introduced, which can reduce the computational cost while increasing the efficiency of the on-line planner.

#### (1) Collision detection of dynamic obstacles

Assuming the number of dynamic obstacles in current ocean work space is  $m'$ , and the safety distance between the  $i$ th AUV  $A_i$  and the  $q$ th dynamic obstacle  $O'_q$ ,  $q=1,2,\dots,m'$  is  $\delta'$ , then the optimization criterion for dynamic obstacles avoidance can be written as:

$$F_5 = \sum_{q=1}^{m'} \sum_{u=0}^{N_i} f_{i,O'_q}^u \quad (7)$$

$$f_{i,O'_q}^u = \begin{cases} 1, & d_{i,O'_q}^u < \delta' \text{ and } |t_i^u - t'_q| < t'_{\min} \\ 0, & \text{otherwise} \end{cases}$$

$$d_{i,O'_q}^u = \sqrt{(x_i^u - x'_q(t'_q))^2 + (y_i^u - y'_q(t'_q))^2}$$

herein,  $(x'_q, y'_q)$  is the position of  $O'_q$  at time  $t'_q$ ;  $d_{i,O'_q}^u$  is the distance between the  $u$ th LGL point of  $A_i$  and the  $q$ th dynamic obstacle  $O'_q$ ;  $t'_{\min}$  is a user designed parameter.

#### (2) Collision resolution

In practical applications, each AUV in the fleet should have a maximum detection radius. Once the dynamic obstacle enters the detection region of the  $i$ th AUV  $A_i$  (centered at  $(x_i, y_i)$  with a radius  $\bar{R}$ ,  $\bar{R} > \delta'$ ), its motion can be detected and predicted. Fig. 4 shows the template used to construct the collision resolution algorithm, in which the obstacle  $O'_q$  moves along the  $y$ -axis of the moving reference frame, while  $A_i$  follows the off-line path (displayed by the red dashed line). Assuming  $O'_q$  enters the

detection region of  $A_i$  for the first time when  $A_i$  moves close to the  $(u-2)$ th node (marked as  $\text{Node}_i^{(u-2)}$ ) along its original path. Evaluating the optimization criterion  $F_5$  for the subsequent nodes on  $P_i$ , if there exists a risk of collision, then an on-line path planner is activated, otherwise  $A_i$  continues to move along its off-line path until the next dynamic obstacle is detected. It is noticed that, the collision should be checked with both spatial constraints and temporal constraints. Even if the instantaneous distance between the affected AUV and the dynamic obstacle is smaller than the required safety distance, the risk of collision can be ignored as long as the time span for these two to reach the corresponding points is longer than  $t'_{\min}$ .

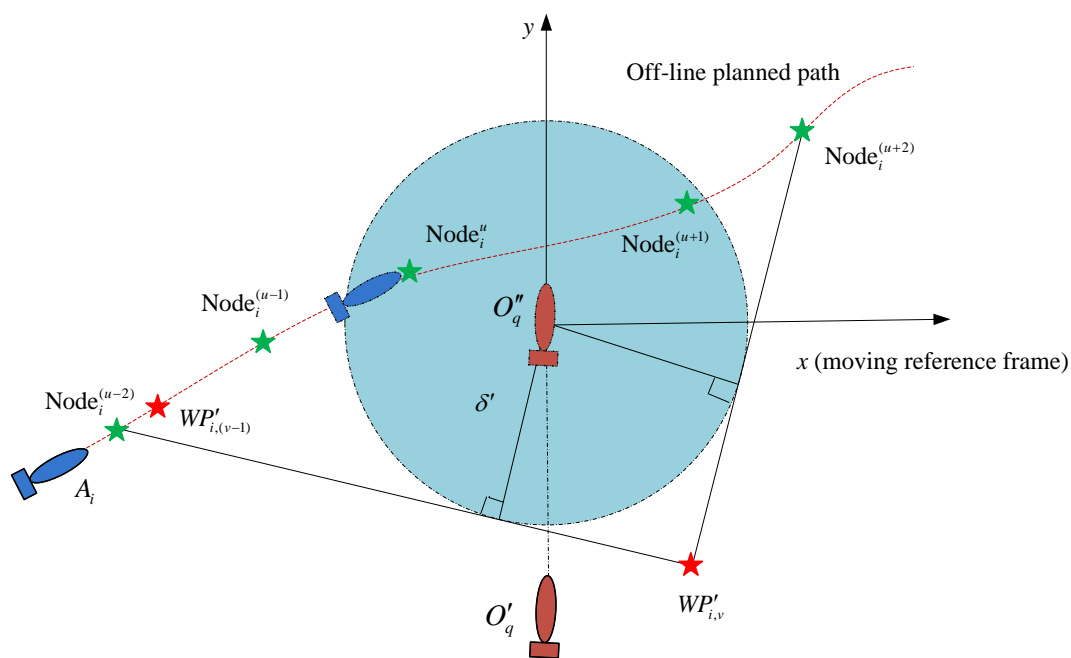


Fig. 4. Template for the collision resolution

Herein, the collision resolution algorithm includes two consecutive avoidance maneuvers, which aims at diverting the affected AUV away from the obstacle, and later guiding it back to the off-line path at the end of the re-planning process. The detailed procedure is shown in Table 3 below.

Table 3 Collision resolution algorithm

---

***Collision resolution algorithm:***

---

1. Find out the position  $O_q''$ , where the instantaneous distance between the off-line path of  $A_i$  and predicted path of  $O_q'$  is shortest, and circle the corresponding safety region with a radius  $\delta'$ ;
  2. Choose the starting point for the first re-planner as the first node detecting the dynamic obstacle such as  $\text{Node}_i^{(u-2)}$  in Fig. 4, and the ending point of the second re-planner as the first node out of the safety region of  $O_q''$  such as  $\text{Node}_i^{(u+2)}$ ;
  3. Draw the tangent lines from the starting point  $\text{Node}_i^{(u-2)}$  and ending point  $\text{Node}_i^{(u+2)}$  respectively to construct an intersection  $WP'_{i,v}$ . Take this point as a new waypoint, and also as the ending point of the first re-planner, as well as the starting point of the second re-planner;
  4. Run the Legendre pseudospectral method based algorithm twice to update the local path with these three points to fulfill all the optimization criteria.
- 

It is shown in Fig. 4, by setting the new waypoint  $WP'_{i,v}$ , the AUV can move out of the safety region of the dynamic obstacle more efficiently. Moreover, due to the time taken for the re-planning process, the actual starting point for the updated path will be some certain point on the off-line path of  $A_i$ , which is close to the starting point of the first re-planner (displayed as  $WP'_{i,(v-1)}$  in Fig. 4).

### (3) Multi-AUV on-line re-planning scheme

Obviously, the previous on-line re-planning scheme is able to clear the risk of dynamic obstacles by updating the local path of the affected AUV, but it will also make

the new path distinct from the one obtained off-line, especially the path segments around the obstacle. As discussed above, the constraints for collision avoidance with static obstacles and other AUVs can only be satisfied by the nodes on off-line paths, so the distinction may cause collisions again. Therefore, in the re-planning process, it is necessary to evaluate the optimization criteria  $F_3, F_4$  as well as  $F_5$ , which will result in a computational burden, and even make the re-planning impossible to achieve on-line when multiple AUVs are in re-planning process simultaneously.

To improve the efficiency of the distributed planner, the information exchanged by the AUVs in the fleet are only the control nodes, and the on-line path planning of multiple AUVs can be regarded as an optimization problem defined in a region centered at the starting point of the corresponding re-planner with a pre-selected radius  $R$ . In each re-planning process, only the nodes of AUVs and static obstacles locating in this region will be taken into account by the re-planner, such that the computational cost will be reduced dramatically. Furthermore, when all the AUVs are operating in the working space, the re-planning process can be carried out sequentially according to the order of detection of collisions. Thus, once the affected AUV starts to re-plan, all the other AUVs still move along their current paths until the end of the re-planning process, and then the updated control nodes will be broadcasted to all the other members in the fleet. It is noted that, the matter of prime importance in path planning problems is safety; therefore, the above scheme is proposed to simplify the path re-planning process. In order to improve the efficiency further, the differential flatness property of AUV will be discussed in the following section.

#### (4) Differential flatness of AUV

Differential flatness is an intrinsic property of nonlinear control systems, which can transform the original complex dynamical equations into a set of algebraic equations [27]. In this paper, a set of flat outputs of  $A_i$  can easily be found as  $\mathbf{Y}_i = [Y_{i,1}, Y_{i,2}, Y_{i,3}]^T = [x_i, y_i, \psi_i]^T$ , and then the mathematical model of  $A_i$  as shown in Eq.

(2) can be transformed into

$$\begin{cases} u_i = \dot{x}_i \cos \psi_i + \dot{y}_i \sin \psi_i = \dot{Y}_{i,1} \cos Y_{i,3} + \dot{Y}_{i,2} \sin Y_{i,3} \\ v_i = \dot{y}_i \cos \psi_i - \dot{x}_i \sin \psi_i = \dot{Y}_{i,2} \cos Y_{i,3} - \dot{Y}_{i,1} \sin Y_{i,3} \\ r_i = \dot{\psi}_i = \dot{Y}_{i,3} \end{cases} \quad (14)$$

$$\begin{cases} \tau_{u_i} = m\dot{u}_i - mv_i r_i + X_u u_i + X_{|u|u} |u_i| u_i \\ \quad = m(\ddot{Y}_{i,1} \cos Y_{i,3} + \ddot{Y}_{i,2} \sin Y_{i,3}) + (X_u + X_{|u|u} |\dot{Y}_{i,1} \cos Y_{i,3} + \dot{Y}_{i,2} \sin Y_{i,3}|) \cdot (\dot{Y}_{i,1} \cos Y_{i,3} + \dot{Y}_{i,2} \sin Y_{i,3}) \\ \tau_{v_i} = m\dot{v}_i + mu_i r_i + Y_v v_i + Y_{|v|v} |v_i| v_i \\ \quad = m(\ddot{Y}_{i,2} \cos Y_{i,3} - \ddot{Y}_{i,1} \sin Y_{i,3}) + (Y_v + Y_{|v|v} |\dot{Y}_{i,2} \cos Y_{i,3} - \dot{Y}_{i,1} \sin Y_{i,3}|) \cdot (\dot{Y}_{i,2} \cos Y_{i,3} - \dot{Y}_{i,1} \sin Y_{i,3}) \\ \tau_{r_i} = I_z \dot{r}_i + N_r r_i + N_{|r|r} |r_i| r_i \\ \quad = I_z \ddot{Y}_{i,3} + (N_r + N_{|r|r} |\dot{Y}_{i,3}|) \dot{Y}_{i,3} \end{cases} \quad (15)$$

It can be noted that, the number of optimization parameters to be determined has been dramatically reduced by 50% in the flat outputs space, while the constraints caused by the differential equations of the system model have also been totally eliminated. Therefore, the computational complexities as well as the time consumption will be reduced remarkably to ensure the on-line re-planning successful.

## 4. Simulation results

In this section, two case studies are tested to demonstrate the performance of the multi-AUV path planners in presence of different ocean environments: The first case aims at generating time-minimum paths for multiple AUVs travelling through a static environment, and the problem of simultaneous arrival for all vehicles at their selected destinations will also be discussed. The second case deals with dynamic ocean

environment, where unexpected moving obstacles will be considered.

The numerical simulations will be carried out in case of three identical AUVs with  $\delta_k = 10\text{m}$  ( $k = 1 \sim 4$ ) and  $\delta_{safe} = 5\text{m}$ . The initial and final states of individual AUV are given in Table 4.

Table 4 Initial and final states of individual AUV

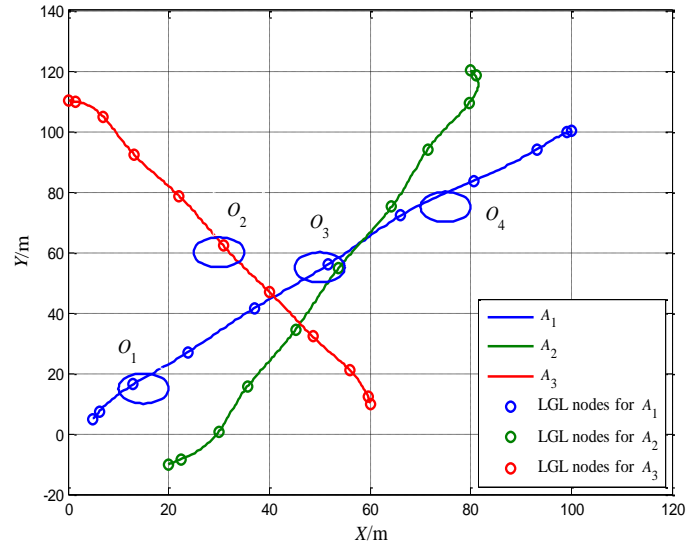
| Parameter          | Value                        | Parameter            | Value                          |
|--------------------|------------------------------|----------------------|--------------------------------|
| $\mathbf{X}_{1,0}$ | $[5, 5, -\pi/4, 0, 0, 0]^T$  | $\mathbf{X}_{1,T_1}$ | $[100, 100, \pi/4, 0, 0, 0]^T$ |
| $\mathbf{X}_{2,0}$ | $[20, -10, 0, 0, 0, 0]^T$    | $\mathbf{X}_{2,T_2}$ | $[80, 120, 2\pi/3, 0, 0, 0]^T$ |
| $\mathbf{X}_{3,0}$ | $[60, 10, \pi/3, 0, 0, 0]^T$ | $\mathbf{X}_{3,T_3}$ | $[0, 110, \pi/2, 0, 0, 0]^T$   |

The number of LGL points for the first global planning is set as  $N_i = 10$  and  $t_{\min} = 10\text{s}$ .

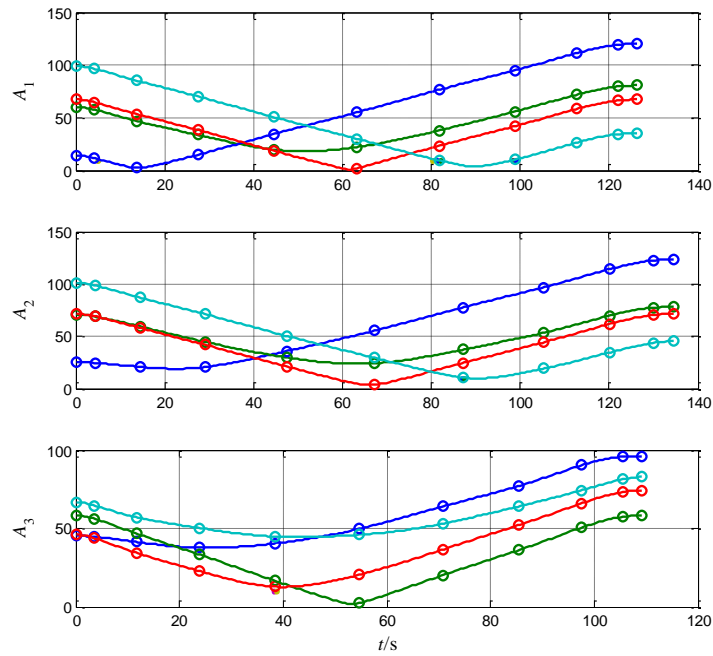
Further, the limitations of control inputs are set as  $\tau_{ui\max} = \tau_{vi\max} = 150\text{N}$  and  $\tau_{ri\max} = 50\text{N} \cdot \text{m}$ ,  $i=1\sim 3$ .

Simulation results are plotted in Figs. 5-9. Fig. 5 displays the results of time-optimal path planning for three AUVs without considering the ocean environment. The obtained paths are almost a set of straight lines connecting the initial and final points of each AUV, in order to guarantee the total time consumption minimum over all participating members. However, since the obstacle constraints have not been taken into account, the AUVs collide with the obstacles on the lines to the destinations. Furthermore, as shown in Fig. 5(c) the collisions not only occur between AUVs and obstacles but also exist between AUVs.

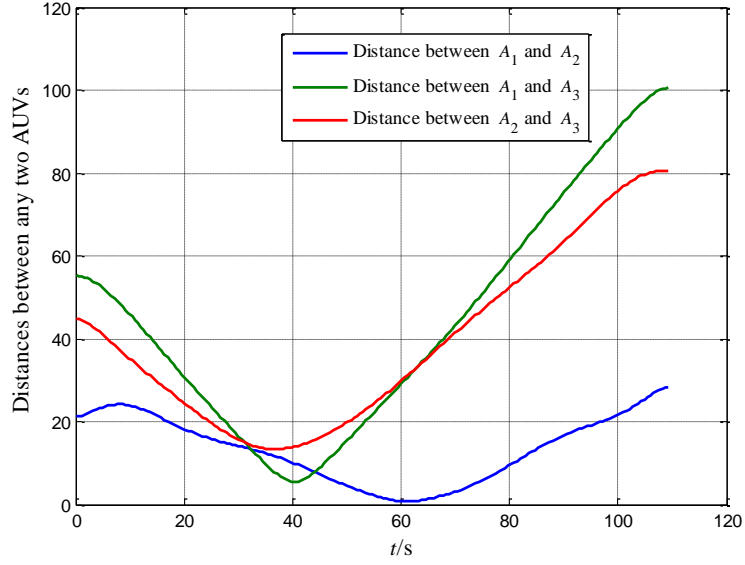




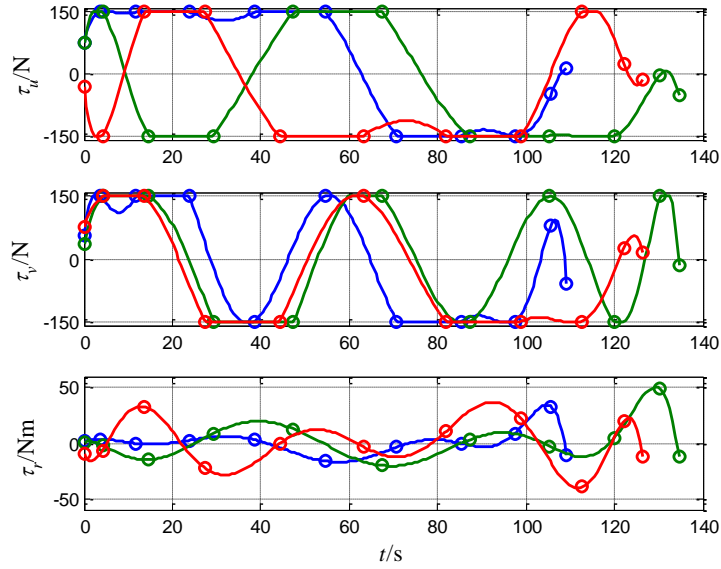
(a) Time-optimal paths for AUVs in case of static environment



(b) Distances between each AUV and static obstacles



(c) Distances between any two AUVs

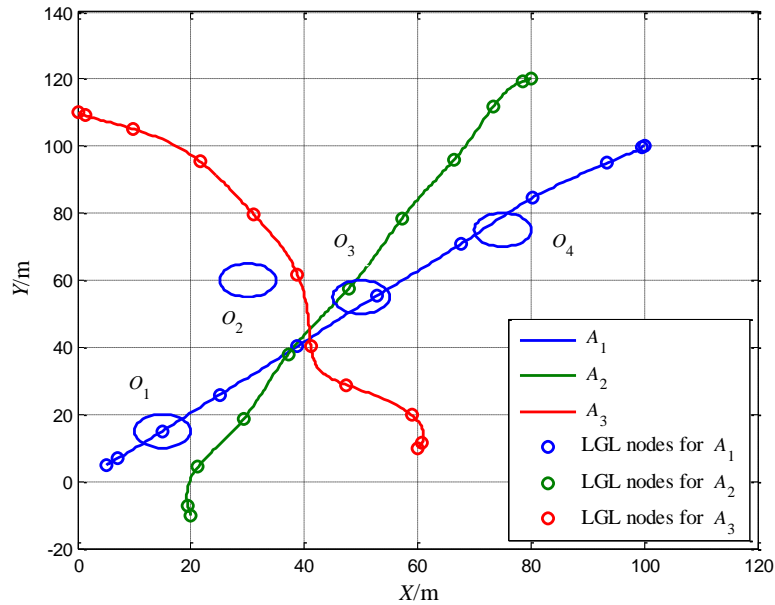


(d) Control inputs

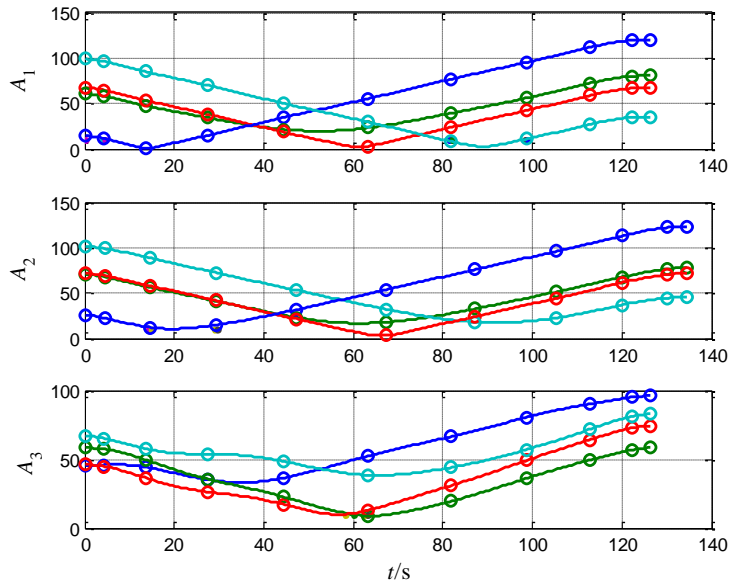
Fig. 5 Path planning of multi-AUV for minimum total time consumption in static environment

The results considering both minimum total time consumption and simultaneous arrival of multi-AUV are shown in Fig. (6). By comparing the results in Fig. 5(a) with those in Fig. 6(a), it can be found the optimal path for  $A_3$  is not a straight line any longer, since it has to detour to wait for the other two AUVs to reach their final points simultaneously. The time taken by each AUV is  $T_1 = 126.32s$ ,  $T_2 = 134.55s$  and

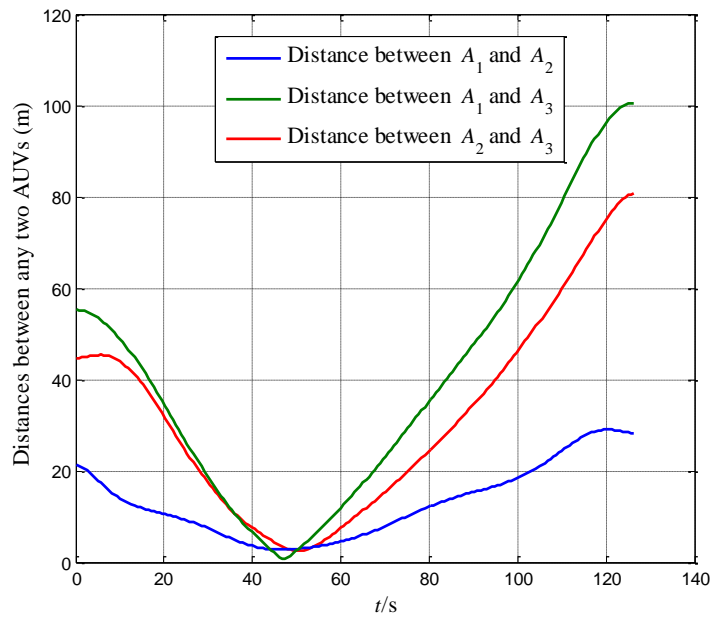
$T_3 = 126.32s$ , respectively, and the total time consumption is  $F_1 = 387.19s$ , which is a little longer than the result obtained in last case. Although more time has been spent on the purpose of simultaneous arrival, all three AUVs still are not able to arrival at their destinations simultaneously. This is due to the optimization criteria considered herein include not only  $F_2$  but also  $F_1$ , and the weighting factors are set as  $\gamma_1 = \gamma_2 = 0.5$ , which implies the total time consumption and simultaneous arrival are of equal importance in this case. Increasing the weight  $\gamma_2$  associated with  $F_2$  can indeed improve the performance of simultaneous arrival; however, much more time has to be used to achieve this goal. For example, if the weights are  $\gamma_1 = 0.3$ ,  $\gamma_2 = 0.7$ , then the time used by individual AUV can be obtained as  $T'_1 = 126.85s$ ,  $T'_2 = 139.97s$  and  $T'_3 = 139.85s$ , while the total time consumption increases by more than 30s. So, it is always necessary to make a tradeoff between these two items according to specific problems by adjusting the weights.



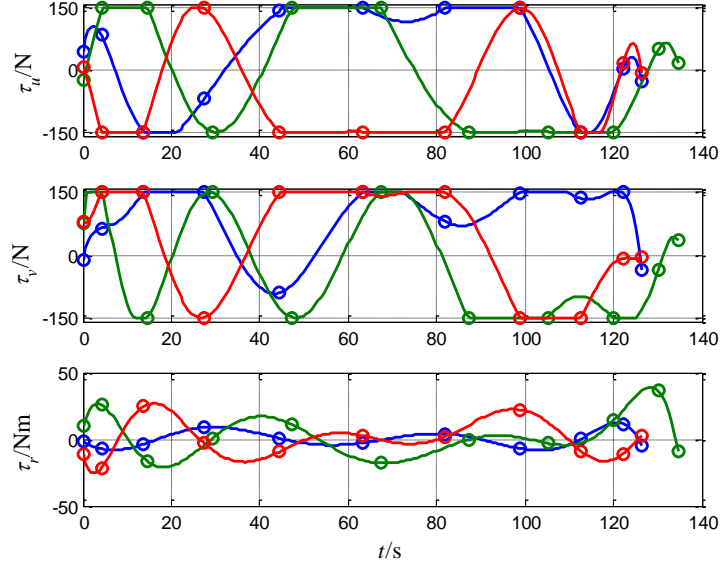
(a) Optimal paths for simultaneous arrival



(b) Distances between each AUV and static obstacles



(c) Distances between any two AUVs

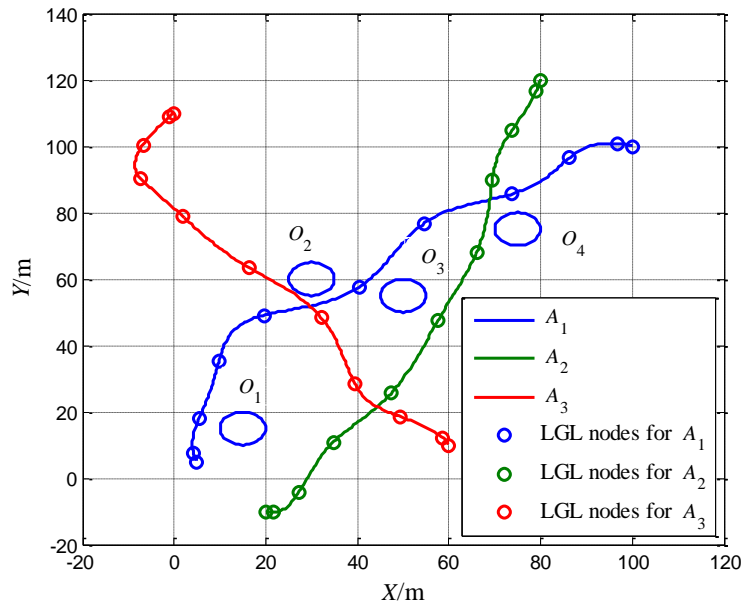


(d) Control inputs

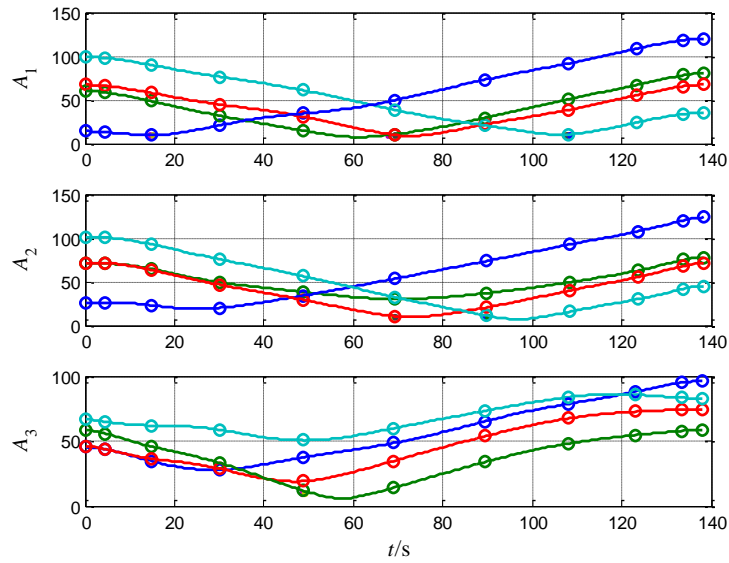
Fig. 6 Path planning of multi-AUV for simultaneous arrival and minimum total time consumption in static environment

Fig. 7 displays the results for the case in which multiple AUVs navigate through a rich obstacles environment for purpose of simultaneous arrival and minimum time consumption in 2D scenarios. It is found that the collisions on LGL nodes can be successfully avoided by considering the collision constraints. However, collisions can also be found on the segments connecting two adjacent control nodes, i.e.  $A_3$  collides with  $O_2$  on the segment connecting its 5<sup>th</sup> and 6<sup>th</sup> LGL points, and understandably the phenomenon will be more prominent when the number of LGL nodes is not sufficiently enough. Whilst, as shown in Fig. 7(c), the distance between  $A_1$  and  $A_2$  can be detected shorter than the safety distance  $\delta_{safe}$  in the time interval [99.5s, 107.2s], which can also be considered as collision. It also should be noticed as shown in Figs. 7(a) and (c), although the path of  $A_1$  crosses over the path of  $A_2$  at  $t_2 = 68.9s$ , no collision occurs around, since as described in objective criterion  $F_4$ , the collision detection

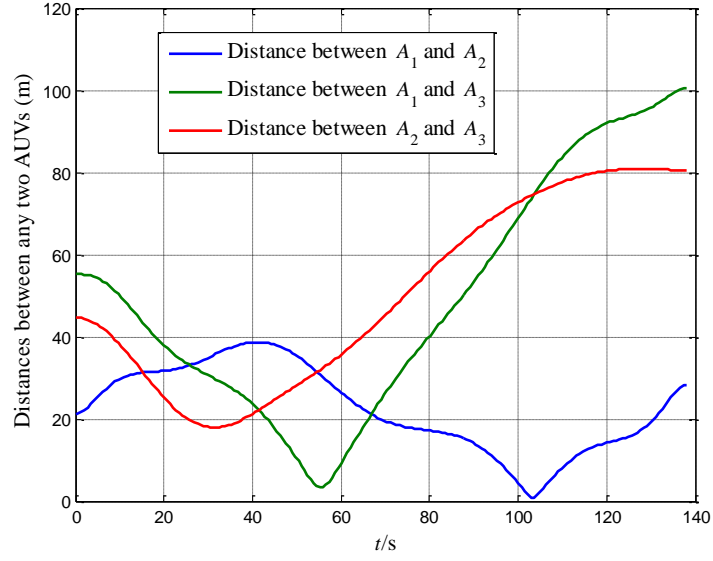
depends on both spatial constraints and temporal constraints.



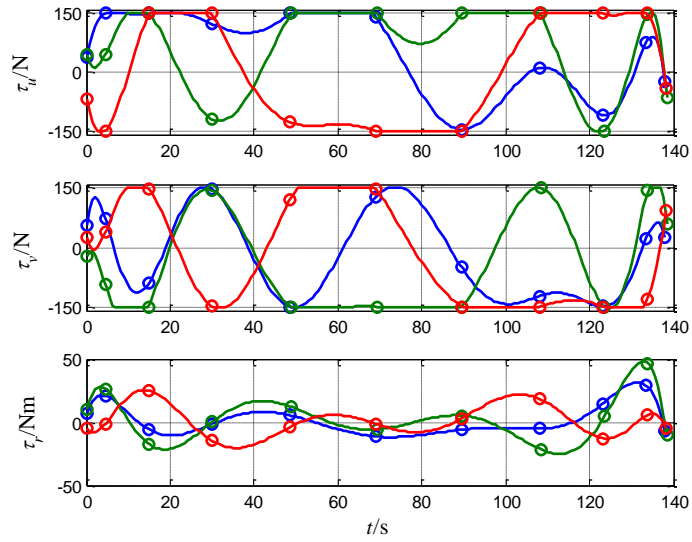
(a) Paths for AUVs in environment with static obstacles



(b) Distances between each AUV and static obstacles



(c) Distances between any two AUVs



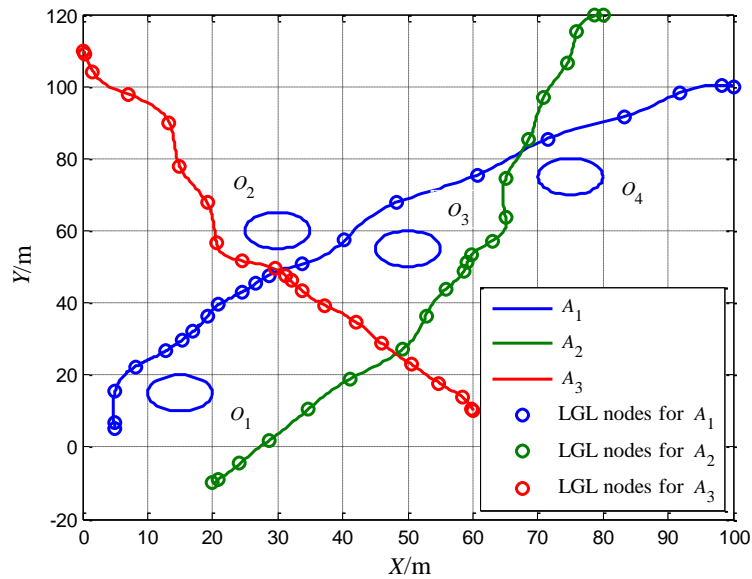
(d) Control inputs

Fig. 7 Path planning of multi-AUV for simultaneous arrival, minimum total time consumption and collision

avoidance in static environment

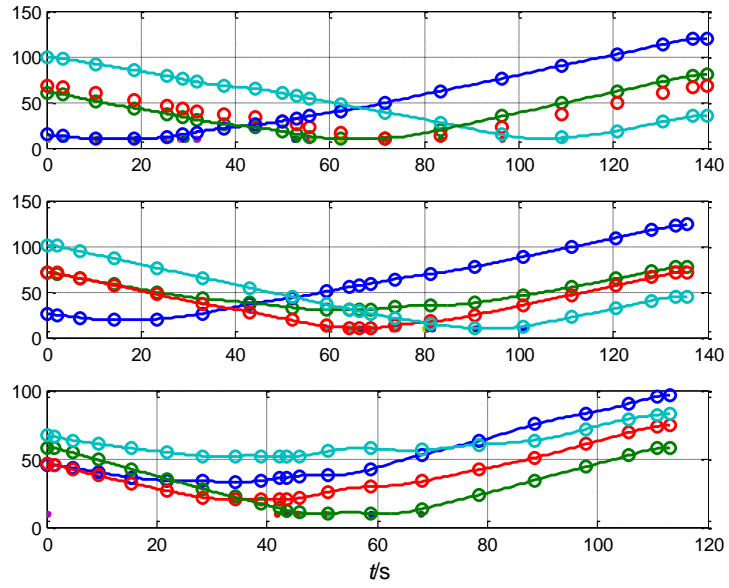
In order to reduce the computational amount and improve the efficiency, an adaptive intermediate knots insertion algorithm is introduced, and the results are displayed in Fig. 8. By comparing Fig. 7 (a) and Fig. 8 (a), it can be found the risk of collisions between  $A_3$  and  $O_2$ ,  $A_1$  and  $O_2$ , as well as  $A_2$  and  $O_4$  can all be reduced greatly by

applying knots insertion algorithm. Further, Fig.8 (c) shows inserting knots is also beneficial to avoid collisions occurring among the members in fleet. In short, all the AUVs can achieve their desired destinations successfully without colliding with any obstacles, as well as any other AUVs in the team by inserting knots at the segments where collisions are detected. Additionally, local planners sometimes can optimize the criteria further as shown in Fig. 8. The total time taken in last case is 414.97s, which is reduced to 388.35s herein. Since the segments between any two selected knots are re-planned locally, some unnecessary detours on the off-line paths can be avoided, and then the total time taken for the AUVs to reach destinations is reduced.

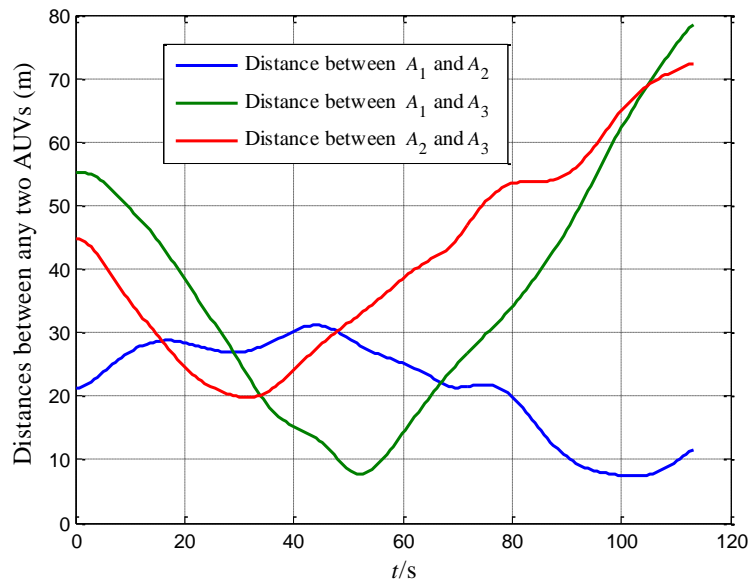


(a) Paths for AUVs in environment with static obstacles

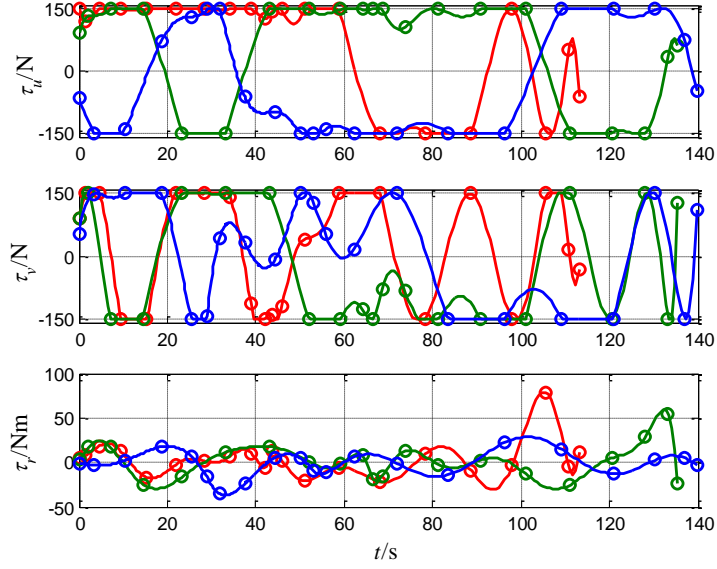




(b) Distances between each AUV and static obstacles



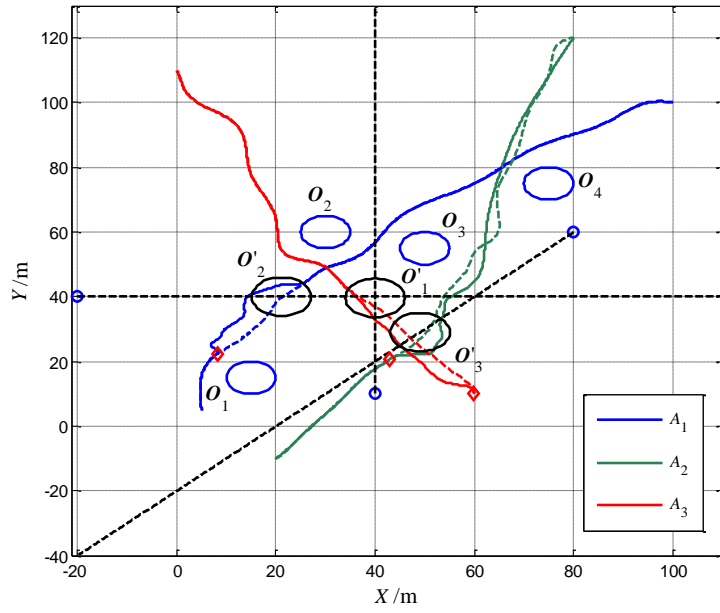
(c) Distances between any two AUVs



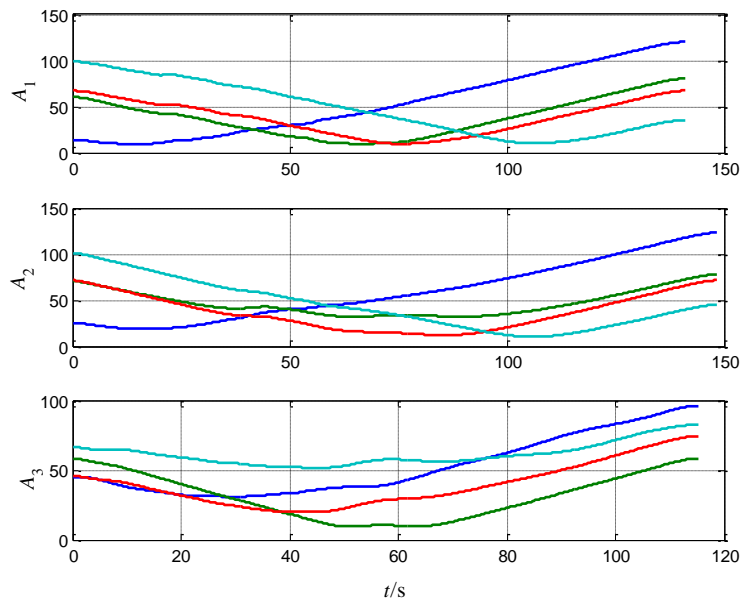
(d) Control inputs

Fig. 8 Application of knots insertion algorithm in static environment

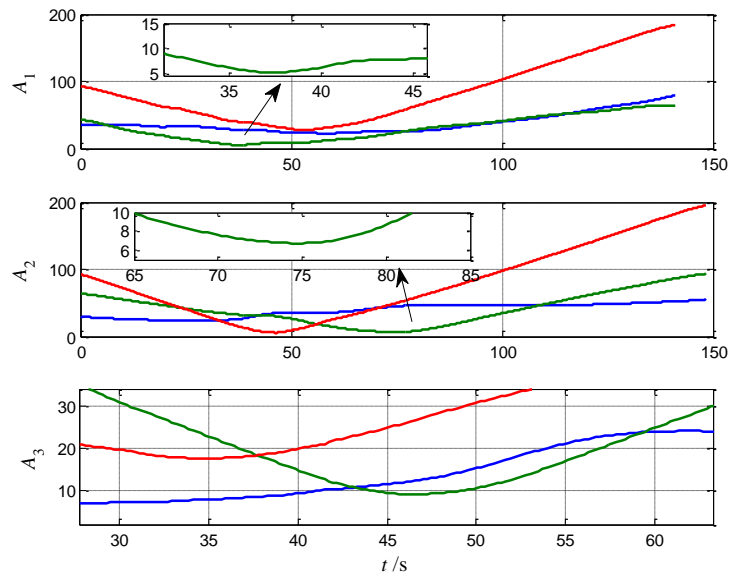
In case 2, three moving identical obstacles  $O'_1 \sim O'_3$  with the velocity 1m/s, as well as four static obstacles  $O_1 \sim O_4$  are considered operating in the work space. Herein, all the dynamic obstacles are assumed to move straight. The initial conditions of all three moving obstacles are  $[40, 10, \pi/2]^T$ ,  $[-20, 40, 0]^T$  and  $[80, 60, -3\pi/4]^T$ , and labeled by small circles respectively. The detection radius  $\bar{R}$  of AUV is set as 20m, and the safety distance  $\delta' = 5m$ . Then the results for path re-planning are displayed in Fig. 9. It can be found, by using the collision resolution listed in Table 3, all the AUVs can avoid the moving obstacles successfully, as well as the static obstacles and other AUVs.



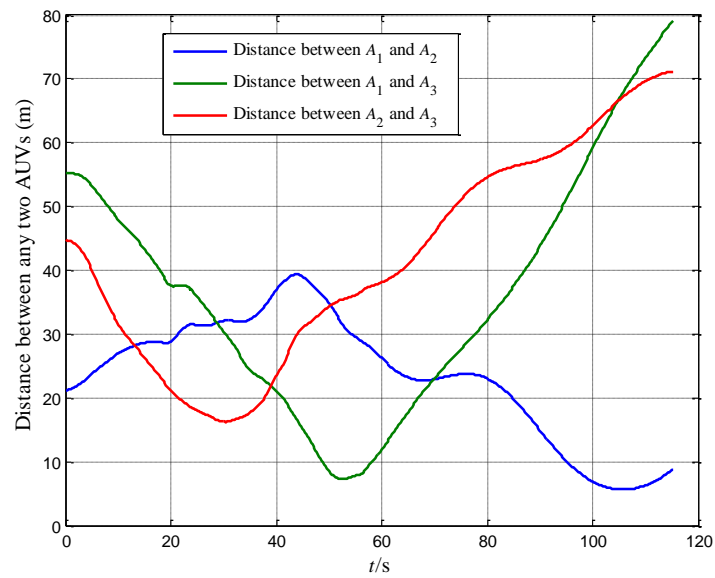
(a) Paths for AUVs in environment with static and dynamic obstacles



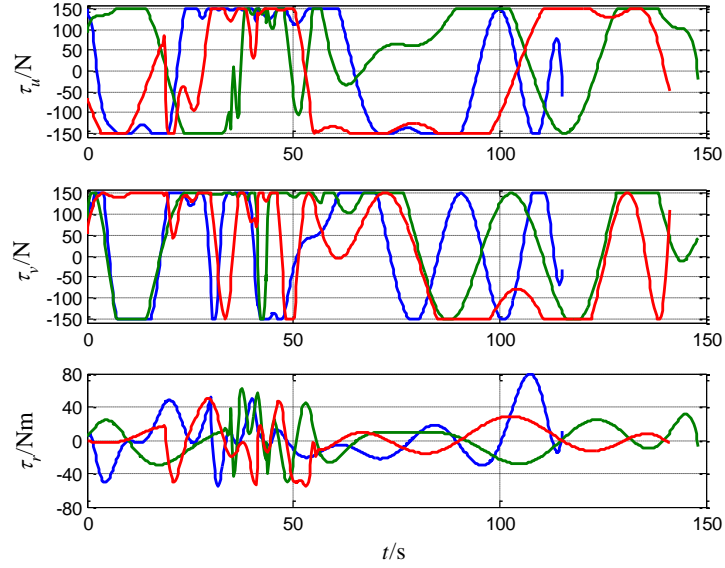
(b) Distances between each AUV and static obstacles



(c) Distances between each AUV and dynamic obstacles



(d) Distance between any two AUVS



(e) Control inputs

Fig. 9 Path re-planning of multi-AUV to deal with moving obstacles

As shown in Fig. 9(a), the risk of collision with  $O'_2$  can be detected by  $A_1$  at  $t=18.77s$ , then two consecutive maneuvers are carried out with the starting point marked by a diamond. In Fig. 9(c),  $A_1$  can be found to avoid  $O'_2$  successfully by moving along the re-planned path (the solid line) and finally return to its original path. Similarly, since the moving obstacle  $O'_1$  exists in the detection region of  $A_3$  at the initial time  $t=0$ , corresponding avoidance maneuvers are taken immediately after  $t>0$ . As discussed above, in order to make use of the results obtained in off-line planning, all the AUVs are required to move back to their original planned paths after re-planning. It can be seen in Fig. 9(a), all the AUVs in the work space can satisfy this requirement except  $A_2$ . It starts to re-plan at  $t=35.2s$ , then before it returns to the original path, the collision with  $A_1$  is detected, then one more maneuver is taken.

In the simulations, the off-line path planning can always be completed in 15 seconds, and then, the computational complexity of on-line re-planners can be reduced greatly

by taking these results as known conditions. More specifically, it is known most optimization algorithms are sensitive to initial guesses, so if a re-planner can take the results obtained in last re-planning as the initialization, the time taken can be certainly reduced and a better solution can also be obtained. In this work, the computational cost for each re-planning is always less than 2 seconds. Furthermore, in order to have the re-planner run on-line successfully, if the time consumption of iteration exceeds 3 seconds, the process will be forced to stop, and the AUVs have to move along the current paths until next re-planning begins. Meanwhile, all the AUVs are assumed to be guided by last re-planner at the first beginning of each re-planning process, until the current re-planner completes.

Additionally, in the applications of the proposed algorithm, if the dynamic obstacle is detected only a relatively short distance away with high velocity, it is difficult or even impossible for the AUV to maneuver out of the safety region always, due to its physical limitations. Then, the emergency measures should be carried out for the sake of safety, which is beyond the scope of this paper, and will be studied in our future work.

## **5. Conclusions and future work**

This paper presents a novel Legendre pseudospectral method based cooperative path planner that work off-line and on-line. In the first phase, Legendre pseudospectral method based path planner is integrated with an adaptive knots insertion algorithm to solve the collisions with static obstacles and other AUVs in the team. In the on-line phase, flatness property of AUV is employed and combined with a local re-planner to avoid the risk of collision with moving obstacles. Simulations tests have been

performed to generate a set of optimal paths with minimum time consumption for three AUVs travelling simultaneously through turbulent ocean fields in the presence of both static and moving obstacles. From the simulation results, the proposed path planner is shown to be capable of reacting fast to dynamic ocean environment, and avoid the collisions successfully and efficiently.

The ultimate aim of the development of path planning algorithms is to enhance the autonomy and operational efficiency of AUVs. Thus, the next major objective of this work is to integrate the task assignment module into the proposed algorithm, which is a self-decision making system to allocate the tasks to individual AUVs according to mission requirements. Another extension of this work is to generate the optimal paths for multi-AUV in realistic ocean environment, such as strong currents fields and irregularly shaped terrains and uncertainty dynamic obstacles. **Additionally, the performance of path planning algorithms is always closely related to the physical limitations of AUVs in practical applications, so it is also necessary to take them into account in future work.**

Although the approach is proposed only for multi-AUV, it can also deal with the cooperative path planning problem of other unmanned vehicles (eg. unmanned surface vehicles and unmanned aerial vehicles) with proper adjustments and extension, which is also an important aspect of future work.

### **Acknowledgements**

This work was supported in part by the Fundamental Research Funds for the Central Universities [grant number HIT.KISTP.2014029], Key Research and Development Plan

of Shandong Province [grant number 2018GGX105014], Shandong Province Higher Educational Science and Technology Program [grant number J18KA010], The Discipline Construction Foundation in Harbin Institute of Technology, Weihai [grant number WH20160103], Education Research Project in Harbin Institute of Technology, Weihai [grant number BKQN201619], and Shandong Province Transportation Science and Technology Program [grant number 2018B69].

## Reference

- [1] Westwood J. The AUV market place. Presentation at Oceanology International, London; 10th March.
- [2] Yuh J. Design and control of autonomous underwater robots: A survey. *Autonomous Robots* 2000; 8(1): 7-24.
- [3] Huynh VT, Dunbabin M, Smith RN. Convergence-guaranteed time-varying RRT path planning for profiling floats in 4-dimensional flow. In: *Australian Conference on Robotics and Automation*. Sydney, Australia; 2014, p. 1-9.
- [4] Eichhorn M. Optimal routing strategies for autonomous underwater vehicles in time-varying environment. *Robotics and Autonomous Systems* 2015; 67: 33-43.
- [5] Li Y, Ma T, Chen PY, Jiang YQ, Wang RP, Zhang Q. Autonomous underwater vehicle optimal path planning method for seabed terrain matching navigation. *Ocean Engineering* 2017; 133: 107-15.
- [6] Koay T, Chitre M. Energy-efficient path planning for fully propelled AUVs in congested coastal waters. In: *2013 MTS/IEEE OCEANS-Bergen*. Bergen, Norway; 2013, p. 1-9.
- [7] Cheng C, Zhu D, Sun B, Zhang S. Path planning for autonomous underwater vehicle based on



- artificial potential field and velocity synthesis. In: 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering. Halifax, NS, Canada; 2015, p. 717-21.
- [8] Fister Jr. I, Perc M, Kamal S, Fister I. A review of chaos-based firefly algorithms: Perspectives and research challenges. *Applied Mathematics and Computation* 2015; 252(1): 155-65.
- [9] Helbing D, Brockmann D, Chadeaux T, Donnay K, Blanke U, Woolley-Meza O, et al. Saving human lives: What complexity science and information systems can contribute. *Journal of Statistical Physics* 2015; 158(3): 735-81.
- [10] Zhang LY, Zhang L, Liu S, Zhou JJ, Papavassiliou C. Three-dimensional underwater path planning based on modified wolf pack algorithm. *IEEE Access* 2017; 5: 22783-95.
- [11] Ganganath N, Yuan W, Fernando T, Iu HC, Cheng CT. Energy-efficient anti-flocking control for mobile sensor networks on uneven terrains. *IEEE Transactions on Circuits and Systems-II: Express Briefs* 2018; 65(12): 2022-6.
- [12] Galceran E, Campos R, Palomeras N, Carreras M, Ridaó P. Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles. *Journal of Field Robotics* 2015; 32(7): 952-83.
- [13] Hernández JD, Vidal E, Vallicrosa G, Galceran E, Carreras M. Online path planning for autonomous underwater vehicles in unknown environments. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). Seattle, WA, USA; 2015, p. 1152-7.
- [14] Zhuang YF, Sharma S, Subudhi B, Huang HB, Wan J. Efficient collision-free path planning for autonomous underwater vehicles in dynamic environments with a hybrid optimization algorithm. *Ocean Engineering* 2016; 127: 190-9.

- [15] Zeng Z, Lian L, Sammut K, He FP, Tang YH, Lammas Andrew. A survey on path planning for persistent autonomy of autonomous underwater vehicles. *Ocean Engineering* 2015; 110: 303-13.
- [16] Das B, Subudhi B, Pati BB. Co-operative control coordination of a team of underwater vehicles with communication constraints. *Transactions of the Institute of Measurement and Control* 2015; 38(4): 1-19.
- [17] Park B. Adaptive formation control of underactuated autonomous underwater vehicles. *Ocean Engineering* 2015; 96: 1-7.
- [18] Matsuda T, Maki T, Sakamaki T, Ura T. State estimation and compression method for the navigation of multiple autonomous underwater vehicles with limited communication traffic. *IEEE Journal of Oceanic Engineering* 2015; 40(2): 337-48.
- [19] Nam H. Data-gathering protocol-based AUV path-planning for long-duration cooperation in underwater acoustic sensor networks. *IEEE Sensors Journal* 2018; 18(21): 8902-12.
- [20] Zhu, D, Cao X, Sun B, Luo C. Biologically inspired self-organizing map applied to task assignment and path planning of an AUV system. *IEEE Transactions on Cognitive and Developmental Systems* 2018; 10(2): 304-13.
- [21] Zeng Z, Sammut K, Lian L, Lammas A, He FP, Tang YH. Rendezvous path planning for multiple autonomous marine vehicles. *IEEE Journal of Oceanic Engineering* 2018; 43(3): 640-64.
- [22] Liu Y, Bucknall R. Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Engineering* 2015; 97: 126-44.
- [23] Fossen T. *Guidance and Control of Ocean Vehicles*. US: John Wiley & Sons; 1994.

- [24] Ataei M, Yousefi-Koma A. Three-dimensional optimal path planning for waypoint guidance of an autonomous underwater vehicle. *Robotics and Autonomous Systems* 2015; 67: 23-32.
- [25] Elnagar G, Kazemi M, Razzaghi M. The pseudospectral Legendre method for discretizing optimal control problems. *IEEE Transactions on Automatic Control* 1995; 40(10): 1793-6.
- [26] Ross I, Karpenko M. A review of pseudospectral optimal control: from theory to flight. *Annual Reviews in Control* 2012; 36(2): 182-97.
- [27] Fliess M, Lévine J, Martin P, Rouchon P. Flatness and defect of nonlinear systems: introductory theory and examples. *International Journal of Control* 1995; 61(6):1327-61.