2016-11-15

Efficient collision-free path planning for autonomous underwater vehicles in dynamic environments with a hybrid optimization algorithm

Zhuang, Y

10.1016/j.oceaneng.2016.09.040
Ocean Engineering
Elsevier BV

"This is the author's accepted manuscript. The final published version of this work (the version of record) is published by **Ocean Engineering**, Volume 127, Pages 190-199, ISSN 0029-8018, http://dx.doi.org/10.1016/j.oceaneng.2016.09.040. (http://www.sciencedirect.com/science/article/pii/S0029801816304188). This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher."

Efficient collision-free path planning for Autonomous Underwater Vehicles in dynamic environments with a hybrid optimization algorithm

Yufei Zhuang [a,b], Sanjay Sharma[b], Bidyadhar Subudhi[c], Haibin Huang[a,b,1] and Jian Wan[b]

[a] School of Information and Electrical Engineering, Harbin Institute of Technology at Weihai, Weihai 264200, China

[b] School of Marine Science and Engineering, Plymouth University, Plymouth PL4 8AA, United Kingdom

[c] Department of Electrical Engineering, NIT Rourkela, Odisha 769008, India

Abstract: This paper presents an efficient path-planner based on a hybrid optimization algorithm for autonomous underwater vehicles (AUVs) operating in cluttered and uncertain environments. The algorithm integrates particle swarm optimization (PSO) algorithm with Legendre pseudospectral method (LPM), which is named as hybrid PSO-LPM algorithm. PSO is first employed as an initialization generator with its strong global searching ability and robustness to random initial values. Then, the searching algorithm is switched to LPM with the initialization obtained by PSO algorithm to accelerate the following searching process. The flatness property of AUV is also

[1]Corresponding author.
Email address: hhb833@gmail.com (Haibin Huang)

utilized to reduce the computational cost for planning, making the optimization algorithm valid for local re-planning to efficiently solve the collision avoidance problem. Simulation results show that the hybrid PSO-LPM algorithm is able to find a better trajectory than standard PSO algorithm and with the re-planning scheme it also succeeds in real-time collision avoidance from both static obstacles and moving obstacles with varying levels of position uncertainty. Finally, 100-run Monte Carlo simulations are carried out to check robustness of the proposed re-planner. The results demonstrate that the hybrid optimization algorithm is robust to random initialization and it is effective and efficient for collision-free path planning.

Key words: Autonomous underwater vehicle; Pseudospectral method; Particle swarm optimization; Differential flatness; Path re-planning; Collision avoidance

1. Introduction

Autonomous underwater vehicles (AUVs) are vehicles that can perform underwater tasks and missions autonomously, using onboard navigation, guidance, and control systems (Yuh, 2000). In addition to various scientific underwater exploratory missions, AUVs have also been widely utilized for military tasks and inspection of underwater structures and resources (Wang et al., 2009; Lin and Tseng, 2006; Kondo and Ura, 2004; Iwakami et al., 2002; Incze, 2011; Li et al., 2012).

AUVs usually operate in dynamic and cluttered ocean environments, and one main challenge in the development of advanced AUVs is to find a path planning scheme which can safely and effectively navigate and guide the AUVs in such environments. The path planner thus should be capable of reacting fast to changing environments and keeps the AUV away from various obstacles from its initial position to the final destination. Obviously, such planning must be completed on-line and follow some optimization strategy in order to ensure the safety and performance of the vehicles.

In recent years, a variety of solution approaches have been developed and applied to the collision-free path planning problems of underwater vehicles. These approaches can be roughly divided into two categories: global planning and local re-planning. When the environment is completely known as a *priori* with static obstacles, a global path planner can be utilized off-line via optimal control theory such as nonlinear programming (Spangelo and Egeland, 1994; Kumar et al., 2005), heuristic algorithms (Likhachev et al., 2005; Carsten et al., 2006) and artificial potential field approaches (Khatib, 1986; Daily and Bevly, 2008; Sullivan et al., 2003). Another class of algorithms to this type of optimization problems are graph search methods including A* algorithm (Carroll et al., 1992; Pereira et al., 2011, 2013) and D* algorithm (Ferguson and Stentz, 2006). On the other hand, if the vehicles operate in unknown or only partially known environments with dynamic obstacles, then subsequent local re-planning due to changing environments should be carried out on-line, which makes the path planning problem intrinsically NP hard (Non-deterministic Polynomial), and finding an optimum solution is not guaranteed. To deal with these problems, evolutionary algorithms have been used, such as genetic algorithm (GA) or particle swarm optimization algorithm (PSO) (Zeng et al., 2015; Aghababa, 2012). Evolutionary algorithms usually have better ability to converge to a global optimum or a near optimal solution than traditional optimization methods, and also not sensitive to initial guesses of solutions. However, evolutionary algorithms are prone to poor numerical accuracy and difficult constraints handling.

In this paper, a novel hybrid algorithm is proposed for time-optimal collision-free path planning of an AUV, which combines PSO algorithm and Legendre pseudospectral method (LPM). The main idea of the algorithm is that: for the first phase, PSO is used as an initial values generator due to its robustness to random initializations. It will be applied for the problem with a set of random initial values, in order to enhance the global searching capability. PSO stops iterating after a stopping criterion is achieved, and the algorithm goes to the second phase. In the second phase, the searching scheme is switched to LPM to achieve a faster and better convergence around the global optimum. The differential flatness property of AUV is also utilized to reduce the number of constraints and variables to be optimized in order to decrease the total time consumption. If the time taken for each optimization is less than the given time horizon for re-planning, then the hybrid planning algorithm can repeatedly be solved on-line. This re-planning approach introduces feedback to compensate for uncertainty, and the guidance law obtained for the AUV ensures obstacle avoidance and offers high performance.

The contributions of this paper are as follows:

- Integrating PSO and LPM as a hybrid optimization algorithm, which can improve both robustness to random initializations and convergence rate around global optimum;
- Employing flatness property of AUV to reduce the time consumption of

optimization ;

- Using re-planning scheme to deal with the collision avoidance against both static and dynamic obstacles.

The remainder of this paper is organized as follows. Section 2 introduces the mathematical models of an AUV and its flatness property; Section 3 defines the problem statement and reformulates the problem in flat outputs space by using flat transformation; Section 4 proposes the details of path re-planning scheme based on hybrid PSO-LPM algorithm; Section 5 shows the simulation results and robustness assessment of the proposed algorithm; Concluding remarks are then presented in Section 6.

2. Mathematical model of an AUV

2.1 *Nonlinear AUV equations of motions*

In general, the dynamic behaviors of an AUV are commonly described in two coordinate systems, namely earth-fixed reference frame and body-fixed reference frame as shown in Figure 1.

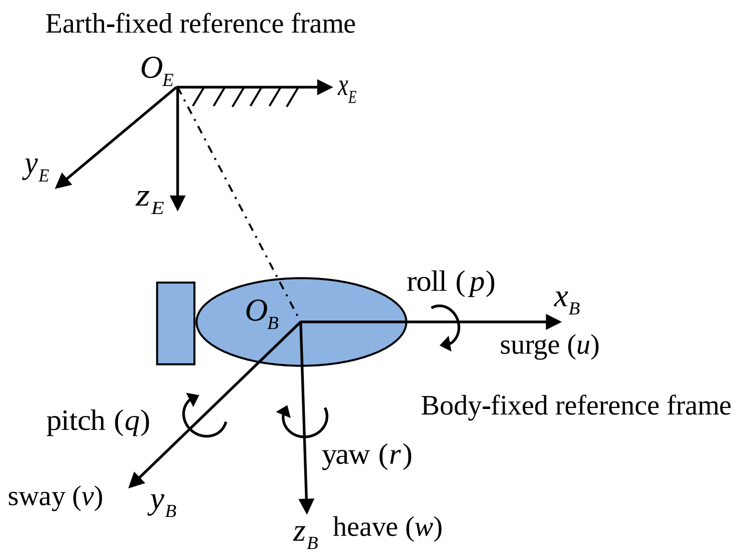


Fig. 1. Earth-fixed and body-fixed reference frames.

A general description of six-DOF nonlinear equations of AUV motions is described as follows (Fossen, 1994):

$$\begin{cases} \dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu} \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \end{cases} \tag{1}$$

where, $\boldsymbol{\nu} = [u, v, w, p, q, r]^{\mathrm{T}}$ is a velocity vector and $\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^{\mathrm{T}}$ is a displacement vector. u, v, w denote linear velocities along surge, sway and heave directions; p, q, r denote rotational velocities in roll, pitch and yaw motions; x, y, z are positions along surge, sway and heave directions, respectively and ϕ, θ, ψ show the Euler angles of the vehicle in earth-fixed frame; $\boldsymbol{J}(\boldsymbol{\eta})$ is Jacobian transformation matrix; $\boldsymbol{M}$ denotes system inertia matrix; $\boldsymbol{C}(\boldsymbol{\nu})$ is Coriolis-centripetal matrix; $\boldsymbol{D}(\boldsymbol{\nu})$ is hydrodynamic damping matrix; $\boldsymbol{g}(\boldsymbol{\eta})$ represents buoyant and gravitational forces and moments; $\boldsymbol{\tau}$ is the vector of control inputs.

Without loss of generality, it is assumed that: (i) the center of mass (CM) coincides with the center of gravity (CG) and center of buoyancy (CB); (ii) the hydrodynamic drag terms of order higher than two can be neglected; (iii) the motions in roll and pitch directions are negligible ($p = q = 0; \phi = \theta = 0$). By selecting the principal axis, the inertia matrix and Coriolis-centripetal matrix are defined as:

$$\boldsymbol{M} = \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & m & 0 \\ 0 & 0 & 0 & I_z \end{bmatrix}, \qquad \boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & -mv \\ 0 & 0 & 0 & mu \\ 0 & 0 & 0 & 0 \\ mv & -mu & 0 & 0 \end{bmatrix} \tag{2}$$

here, m is the mass of the vehicle; I_z is the moment of inertia in yaw motion. The matrix $\boldsymbol{D}(\boldsymbol{\nu})$ is assumed to be non-coupled with only uncertain linear/quadratic damping coefficients $X_u / X_{u|u|}, Y_v / Y_{v|v|}, Z_w / Z_{w|w|}$ and $N_r / N_{r|r|}$. Hydrodynamic damping matrix $\boldsymbol{D}(\boldsymbol{\nu})$ and $\boldsymbol{g}(\boldsymbol{\eta})$ can thus be described as:

$$\boldsymbol{D}(\boldsymbol{\nu}) = \begin{bmatrix} X_u + X_{u|u|}|u| & 0 & 0 & 0 \\ 0 & Y_v + Y_{v|v|}|v| & 0 & 0 \\ 0 & 0 & Z_w + Z_{w|w|}|w| & 0 \\ 0 & 0 & 0 & N_r + N_{r|r|}|r| \end{bmatrix}, \quad \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{0} \tag{3}$$

$\boldsymbol{\tau} = [T_u, T_v, T_w, 0, 0, T_r]^{\mathrm{T}}$, where T_u, T_v, T_w and T_r represent available control inputs in surge, sway, heave, and yaw directions, respectively. The kinematic and dynamic equations of AUV can be represented as:

$$\begin{cases} \dot{x} = u\cos\psi - v\sin\psi \\ \dot{y} = u\sin\psi + v\cos\psi \\ \dot{z} = w \\ \dot{\psi} = r \\ m\dot{u} - mvr + X_u u + X_{|u|u}|u|u = T_u \\ m\dot{v} + mur + Y_v v + Y_{|v|v}|v|v = T_v \\ m\dot{w} + Z_w w + Z_{|w|w}|w|w = T_w \\ I_z\dot{r} + N_r r + N_{|r|r}|r|r = T_r \end{cases} \tag{4}$$

2.2 *Flatness analysis of an AUV*

A control system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \quad \boldsymbol{x} \in \mathbf{R}^n \quad \boldsymbol{u} \in \mathbf{R}^m \tag{5}$$

is differentially flat or just flat, if there exist smooth maps $\boldsymbol{C}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ defining on open neighborhoods of $\mathbf{R}^n \times (\mathbf{R}^m)^{\rho+1}, (\mathbf{R}^m)^{\gamma+1}$ and $(\mathbf{R}^m)^{\gamma+2}$, such that

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{u}, \dot{\boldsymbol{u}}, \ddot{\boldsymbol{u}}, \ldots, \boldsymbol{u}^{(\rho)}) \\ \boldsymbol{x} &= \boldsymbol{A}(\boldsymbol{y}, \dot{\boldsymbol{y}}, \ddot{\boldsymbol{y}}, \ldots, \boldsymbol{y}^{(\gamma)}) \\ \boldsymbol{u} &= \boldsymbol{B}(\boldsymbol{y}, \dot{\boldsymbol{y}}, \ddot{\boldsymbol{y}}, \ldots, \boldsymbol{y}^{(\gamma+1)}) \end{aligned} \tag{6}$$

here ρ and γ are positive integers, $\boldsymbol{y}$ is called a set of flat outputs, and the components of $\boldsymbol{y}$ are not related by a differential relation (Fliess et al., 1995, Lévine J, 2011). The definition shows that if there exist a set of flat outputs with the same number of control inputs, then the state and control variables can both be expressed with them in flat outputs space.

By observing Eq. (4) carefully, a set of flat outputs can be easily found as $\boldsymbol{Y} = [Y_1, Y_2, Y_3, Y_4]^{\mathrm{T}} = [x, y, z, \psi]^{\mathrm{T}}$, and then the mathematical model of AUV can be transformed into

$$\begin{cases} u = \dot{x}\cos\psi + \dot{y}\sin\psi = \dot{Y}_1 \cos Y_4 + \dot{Y}_2 \sin Y_4 \\ v = \dot{y}\cos\psi - \dot{x}\sin\psi = \dot{Y}_2 \cos Y_4 - \dot{Y}_1 \sin Y_4 \\ w = \dot{z} = \dot{Y}_3 \\ r = \dot{\psi} = \dot{Y}_4 \end{cases} \tag{7}$$

$$\begin{cases} T_u = m\dot{u} - mvr + X_u u + X_{|u|u} |u| u \\ \quad = m(\ddot{Y}_1 \cos Y_4 + \ddot{Y}_2 \sin Y_4) + (X_u + X_{|u|u} |\dot{Y}_1 \cos Y_4 + \dot{Y}_2 \sin Y_4|) \cdot (\dot{Y}_1 \cos Y_4 + \dot{Y}_2 \sin Y_4) \\ T_v = m\dot{v} + mur + Y_v v + Y_{|v|v} |v| v \\ \quad = m(\ddot{Y}_2 \cos Y_4 - \ddot{Y}_1 \sin Y_4) + (Y_v + Y_{|v|v} |\dot{Y}_2 \cos Y_4 - \dot{Y}_1 \sin Y_4|) \cdot (\dot{Y}_2 \cos Y_4 - \dot{Y}_1 \sin Y_4) \\ T_w = m\dot{w} + Z_w w + Z_{|w|w} |w| w \\ \quad = m\ddot{Y}_3 + (Z_w + Z_{|w|w} |\dot{Y}_3|)\dot{Y}_3 \\ T_r = I_z \dot{r} + N_r r + N_{|r|r} |r| r \\ \quad = I_z \ddot{Y}_4 + (N_r + N_{|r|r} |\dot{Y}_4|)\dot{Y}_4 \end{cases} \tag{8}$$

3. Problem formulation and transformation

This paper aims at finding a time-optimal collision-free path planning scheme for AUV, where the optimization criterion is used to obtain the minimum travelling time whilst the collision constraints ensure that the path is collision-free from any static or moving obstacles with uncertainty.

Generally, the path planning problem can be formulated as an optimization problem: find a path $\boldsymbol{X} = [\boldsymbol{\nu}; \boldsymbol{\eta}; \boldsymbol{\tau}]^{\mathrm{T}} = [u, v, w, r, x, y, z, \psi, T_u, T_v, T_w, T_r]^{\mathrm{T}}$, which minimizes the performance index ($\bar{J}$):

$$\min_{\boldsymbol{X}} \bar{J} = t_f \tag{9}$$

subject to the vehicle dynamics described by Eq. (4), and the positional constraints from the given initial condition $\boldsymbol{X}_0$ and final destination $\boldsymbol{X}_f$ defined as:

$$\boldsymbol{X}(t_0)=\boldsymbol{X}(\boldsymbol{\nu}(t_0);\boldsymbol{\eta}(t_0);\boldsymbol{\tau}(t_0))=\boldsymbol{X}_0 \quad ; \quad \boldsymbol{X}(t_f)=\boldsymbol{X}(\boldsymbol{\nu}(t_f);\boldsymbol{\eta}(t_f);\boldsymbol{\tau}(t_0))=\boldsymbol{X}_f \tag{10}$$

where, t_f is the final time. If the initial time is assumed $t_0=0$, then t_f is the total travelling time of the AUV. The rotational velocities of the thrusters mounted on the practical AUVs will have lower and upper limitations, which results in the following control inputs constraints as

$$|\boldsymbol{\tau}|\leq|\boldsymbol{\tau}_{\max}| \tag{11}$$

where, $\boldsymbol{\tau}_{\max}$ should coincides to physical limitations of the thrusters.

In this section, to deal with the collision constraints, hybrid objective function is employed, and a weighting scheme is introduced to trade-off between the total travelling time and the risk of collision, the hybrid objective function is defined as

$$J(\boldsymbol{X})=\varepsilon_1 J_1(\boldsymbol{X})+\varepsilon_2 J_2(\boldsymbol{X}) \tag{12}$$

where, $\varepsilon_1,\varepsilon_2$ denote positive weighting values satisfying $\varepsilon_1+\varepsilon_2=1$ and $J_1(\boldsymbol{X})=t_f$ as described in Eq. (9).

The objective function for collision avoidance indicating distance information between AUV and obstacles is defined as (Liang and Lee, 2015)

$$J_2(\boldsymbol{X})=\sum_{j=1}^{S} J_2^j(\boldsymbol{X}), \qquad J_2^j(\boldsymbol{X})=\begin{cases} 0, & \left\|\boldsymbol{X}_p-Obs_j\right\|>\delta_{obsj} \\ \dfrac{1}{\left\|\boldsymbol{X}_p-Obs_j\right\|}-\dfrac{1}{\delta_{obs}}, & \left\|\boldsymbol{X}_p-Obs_j\right\|\leq\delta_{obsj} \end{cases} \tag{13}$$

where, j=1,2…S , S is the number of obstacles in the work space; Obs_j represents the center of the j^{th} obstacle; $\boldsymbol{X}_p$ is the position of AUV; δ_{obsj} denotes the given safe distance between AUV and the j^{th} obstacle, which can be obtained according to the length of the AUV and the *radii* of the obstacles.

As shown in Eqs. (9-13), the optimization process needs to determine a large number of variables, which will result in a huge time burden, especially for evolutionary algorithms. Additionally, most optimization algorithms spend majority of time on dealing with the differential equations constraints caused by the mathematical models of system.

By the definition of differential flatness above, if a dynamic system is flat, then its state and input variables can be parameterized in terms of a set of flat outputs and their derivatives. The above original optimization problem thus can be converted and reformulated in flat outputs space as: find a path $\bar{\boldsymbol{Y}} = [\boldsymbol{Y}, \dot{\boldsymbol{Y}}, \ddot{\boldsymbol{Y}} \ldots \boldsymbol{Y}^{(\gamma+1)}]^{\mathrm{T}}$ in order to minimize the objective function described as

$$\min_{\bar{\boldsymbol{Y}}(t)} J(\bar{\boldsymbol{Y}}) = \min_{\bar{\boldsymbol{Y}}(t)} [\varepsilon_1 J_1(\bar{\boldsymbol{Y}}) + \varepsilon_2 J_2(\bar{\boldsymbol{Y}})] \tag{14}$$

subject to the positional constraints as

$$\boldsymbol{A}(\boldsymbol{Y}(t_0), \dot{\boldsymbol{Y}}(t_0), \ddot{\boldsymbol{Y}}(t_0) \ldots \boldsymbol{Y}^{(\gamma)}(t_0)) = \boldsymbol{X}_0; \quad \boldsymbol{A}(\boldsymbol{Y}(t_f), \dot{\boldsymbol{Y}}(t_f), \ddot{\boldsymbol{Y}}(t_f) \ldots \boldsymbol{Y}^{(\gamma)}(t_f)) = \boldsymbol{X}_f \tag{15}$$

and the input variables constraints

$$|\boldsymbol{B}(\boldsymbol{Y}, \dot{\boldsymbol{Y}}, \ddot{\boldsymbol{Y}} \ldots \boldsymbol{Y}^{(\gamma+1)})| \leq |\boldsymbol{\tau}_{\max}| \tag{16}$$

where, flat transformation $\boldsymbol{A}$ is defined in Eq.(7), while transformation $\boldsymbol{B}$ is provided in Eq. (8).

It can be found in the reformulation in flat outputs space, the constraints caused by the nonlinear model of the AUV have been completely eliminated, and all the displacement and control input variables can be parameterized by flat outputs, thus the number of variables to be optimized has also been reduced by 60% from 12 to 4. The time taken for path planning is thus considerably faster in this case, and makes the optimization algorithm more possibly to re-plan the trajectory on-line.

4 Hybrid PSO-LPM algorithm for path planning

This paper focuses on the path planning problem of AUVs in complicated environments with static and moving obstacles. In order to seek a collision-free path, the planner should be capable of reacting fast to any new information about the environments obtained by the corresponding software and sensors mounted on the vehicles. The path planning of AUV in such environments should be a continuous and closed-loop process, and the trajectory should be locally re-planned according to the changing environments. The main idea of the re-planning scheme is illustrated in Fig. 2: where ΔT is the re-planning time horizon, t_{M_i} is the i^{th} measurement time of the sensors and t_{P_i} is the time taken for the i^{th} re-planning process. At time t_i, the AUV executes the trajectory generated by the $(i\text{-}1)^{th}$ re-planner (dotted line in Fig. 2), and this process will last until the time $t_i + \Delta t_i$, where $\Delta t_i = t_{M_i} + t_{P_i}$. An updated path will be obtained by the i^{th} re-planning process according to the environment information collected by the sensors at time $t_i + t_{M_i}$. The AUV will be guided along the new trajectory (black line in Fig. 2) until the $(i+1)^{th}$ updated trajectory is obtained. It is obvious in Fig. 2 that, if $\Delta T > \Delta t_i$, then a path update can be computed by incorporating any new information of the changing ocean environment. Moreover, if ΔT is sufficiently short, then the environment information can be fed back to the planner in real-time, which can ensure the trajectory planned more safely and efficiently. However, the shorter the planning window is, the faster the planning algorithm is required.

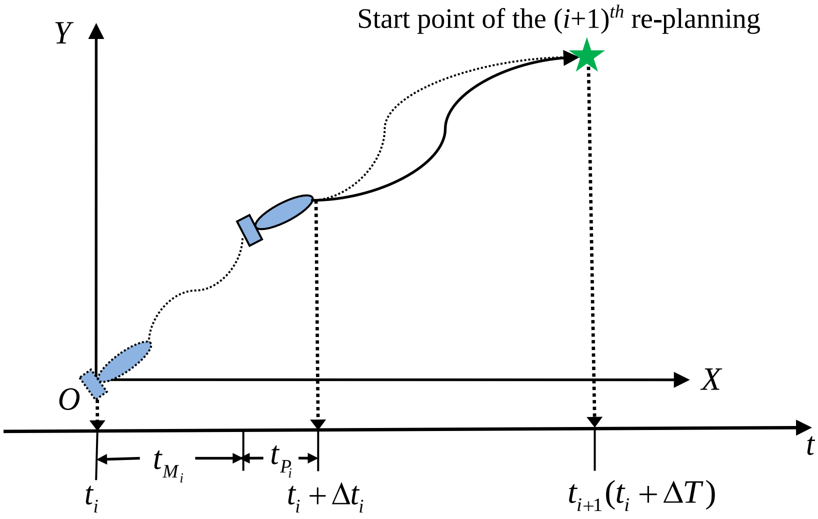


Fig. 2. Re-planning scheme.

4.1 *PSO path planning algorithm*

Particle swarm optimization (PSO) is an evolutionary computation technique, which was introduced in the mid 1990s (Kennedy and Eberhart, 1995). Every particle in the swarm represents a potential path, the parameters of each particle corresponds to the coordinates of control points generating the path. An overview of the PSO-based path planning scheme is illustrated in Table 1.

Table 1

PSO-based path planning scheme.

Initialization: Choose appropriate parameters for population size s, the maximum number of iterations $K_{\max}$. The stopping criterion is chosen as the change of the current best particle fitness values between two consecutive iterations is smaller than a predefined value κ. Input the current environmental information and initialize a set of particles positions $\boldsymbol{X}_0^i$ and velocities $\boldsymbol{V}_0^i$ randomly.

1. Evaluate each particle's fitness value subject to Eqs. (14-16), and store the current best state of each particle;
2. Evaluate the new position's fitness value; for each particle, if the fitness value of new particle is better than the original particle, swap it;
3. Compare with all the best ever positions of each particle to find the best global position, and update the velocity vector of each particle in the swarm;
4. Update the position vector of each particle, using its previous position and the updated velocity vector;
5. If the stopping criterion is satisfied or the number of iterations exceeds $K_{\max}$ then stop, otherwise, go to step2.

In Step 3, the updating scheme for the velocity vector of each particle is given by

$$\boldsymbol{V}_{k+1}^i = w_k \boldsymbol{V}_k^i + c_1 r_1 \left(\boldsymbol{P}_k^i - \boldsymbol{X}_k^i\right) + c_2 r_2 \left(\boldsymbol{P}_k^g - \boldsymbol{X}_k^i\right) \tag{17}$$

where, subscript k indicates an unit pseudo-time increment, $\boldsymbol{V}_k^i, \boldsymbol{X}_k^i$ are the velocity vector and position vector of particle *i* at iteration *k*, r_1, r_2 are two random numbers in the range $[0,1]$. The parameters c_1, c_2 are problem-dependent, where c_1 indicates the confidence level of the current particle in itself and c_2 describes the confidence level in the swarm. The parameter w_k is an inertia weighting factor which controls the global/local exploration abilities of the swarm, which is proposed as

$$w_k = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}}(k-1) \tag{18}$$

where, $w_{\min}, w_{\max}$ are the lower and upper bounds of w_k in the whole optimization.

In Step 4, the updating scheme for the position vector of each particle is described as

$$\boldsymbol{X}_{k+1}^{i} = \boldsymbol{X}_{k}^{i} + \boldsymbol{V}_{k+1}^{i} \tag{19}$$

Further, the velocity vector of a particle with violated constraints should be brought back to zero in the velocity update scheme defined as

$$\boldsymbol{V}_{k+1}^{i} = c_1 r_1 \left(\boldsymbol{P}_k^i - \boldsymbol{X}_k^i\right) + c_2 r_2 \left(\boldsymbol{P}_k^g - \boldsymbol{X}_k^i\right) \tag{20}$$

This is to ensure if a particle is infeasible, then there is a large probability that the last search direction was not feasible.

4.2 *LPM path planning algorithm*

Legendre pseudospectral method (LPM) is an efficient numerical optimization algorithm first proposed by Elnagar et al. (1995). In this paper, it is employed as a discrete optimization scheme for the NP hard problem defined by Eqs. (14-16). The main idea of LMP is to parameterize the flat outputs and their derivatives with *N*th order Lagrange polynomials L_N on $N+1$ Legendre-Gauss-Lobatto (LGL) points. Since the LGL points lie only in the interval $\sigma \in [-1,1]$, a linear transformation $\sigma = [2t - (t_f + t_0)]/(t_f - t_0) \in [-1,1]$ should be taken first to rewrite the optimization problem. The flat output functions $\boldsymbol{Y}(\sigma)$ can thus be approximated on $N+1$ LGL points as

$$\boldsymbol{Y}(\sigma) \approx \boldsymbol{Y}^N(\sigma) := \sum_{l=0}^{N} \boldsymbol{Y}(\sigma_l)\varphi_l(\sigma) = \sum_{l=0}^{N} \boldsymbol{\lambda}_l \varphi_l(\sigma) \tag{21}$$

where, LGL points $\sigma_l, l = 0,1,...,N(\sigma_0 = -1,\ \sigma_N = 1)$ are the roots of $\dot{L}_N(\sigma)$. $\varphi_l(\sigma)$ is the N^{th} degree Lagrange interpolating basis function defined as

$$\varphi_l(\sigma) = \frac{1}{N(N+1)L_N(\sigma_l)} \cdot \frac{(\sigma^2-1)\dot{L}_N(\sigma)}{\sigma-\sigma_l} \tag{22}$$

The first and the $(\gamma+1)^{th}$ derivatives of $\boldsymbol{Y}(\sigma)$ at the LGL point σ_k can be approximated respectively as

$$\begin{aligned} \dot{\boldsymbol{Y}}(\sigma_k) \approx \dot{\boldsymbol{Y}}^N(\sigma_k) &:= \sum_{l=0}^{N} \boldsymbol{D}_{1,kl}\boldsymbol{Y}(\sigma_l) = \sum_{l=0}^{N} \boldsymbol{\lambda}_l \boldsymbol{D}_{1,kl} \\ \boldsymbol{Y}^{(\gamma+1)}(\sigma_k) \approx \boldsymbol{Y}^{(\gamma+1)N}(\sigma_k) &:= \sum_{l=0}^{N} \boldsymbol{D}_{(\gamma+1),kl}\boldsymbol{Y}(\sigma_l) = \sum_{l=0}^{N} \boldsymbol{\lambda}_l \boldsymbol{D}_{(\gamma+1),kl} \end{aligned} \tag{23}$$

where, $\boldsymbol{D}_{1,kl}$ are the entries of the $(N+1)\times(N+1)$ matrix $\boldsymbol{D}_1$

$$\boldsymbol{D}_1 := [\boldsymbol{D}_{1,kl}] := \begin{cases} \dfrac{L_N(\sigma_k)}{L_N(\sigma_l)} \cdot \dfrac{1}{\sigma_k-\sigma_l} & k \neq l \\ -\dfrac{N(N+1)}{4} & k=l=0 \\ \dfrac{N(N+1)}{4} & k=l=N \\ 0 & \textit{otherwise.} \end{cases} \tag{24}$$

The matrix $\boldsymbol{D}_{(\gamma+1),kl}$ is also $(N+1)\times(N+1)$, which can be easily obtained by $\boldsymbol{D}_{(\gamma+1)} := [\boldsymbol{D}_{(\gamma+1),kl}] = \boldsymbol{D}_1^{(\gamma+1)}$.

Using LPM algorithm, the path planning problem shown in Eqs. (14-16) can be further converted into a NLP as: determine a set of coefficients $\boldsymbol{\lambda}(\sigma) = [\boldsymbol{\lambda}_0(\sigma), \boldsymbol{\lambda}_1(\sigma), ..., \boldsymbol{\lambda}_N(\sigma)]^{\mathrm{T}}$, which minimizes the cost function shown in Eq. (14), subject to all required constraints.

One of the main advantages of LPM is offering an exponential convergence rate for the approximation of analytical functions under L^2 norm, while providing Eulerian-like simplicity (Gong et al., 2006). Due to its high accuracy and competitive computational efficiency, LPM is widely used in direct optimization methods. In general, LPM has a larger radius of convergence than other numerical methods, and it may not require a set of good initial guesses for convergence. However, educated initial guesses do improve the convergence rate and robustness. In the following section, a hybrid PSO–LMP algorithm is proposed to solve the collision-free path planning problem of the AUV.

4.3 *Path re-planning with hybrid PSO-LPM algorithm*

The proposed PSO–LPM is a hybrid optimization algorithm combining PSO algorithm with LPM algorithm. The main idea of the algorithm can be divided into two phases: in phase 1, PSO algorithm serves as a start engine to generate a candidate path; in phase 2, the best solution of phase 1 is loaded as an initialization for LPM-based path planner, and then run the LPM-based path planner repeatedly on-line until the AUV reaches the final destination. Finally, the obtained optimal solutions in flat outputs space should be mapped back to the state and control input spaces. The details of PSO-LPM algorithm can be summarized as shown in Table 2:

Table 2

Hybrid PSO-LPM algorithm for re-planning process.

Initialization: Set all the parameters of PSO algorithm with appropriate values, and the number of LGL points is N+1. Select a proper value for re-planning time horizon ΔT , where ΔT could be a constant, and depends on the time consumption for each re-planning process based on LPM-based algorithm.
1. Rewrite the original problem in flat outputs space as shown in Eqs. (14-16) and approximate the flat output functions by LPM algorithm according to Eqs. (21-24);
2. Regard the undetermined variable vector $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_N]^{\mathrm{T}}$ as a single particle, and run

the PSO-based path planning algorithm in Section 4.1, until the stopping criterion is met or the number of iterations exceeds $K_{\max}$, then stop;

3. Store the best candidate solution, and regard it as a set of initial values for LPM path planner, meanwhile let i=0;
4. Update the current ocean environments information at time t_i, and run the LPM path planning algorithm;
5. Send the updated candidate path found in Step 4 to the AUV guidance system once the vehicle reaches the time $t_i + \Delta t_i$;
6. If the fitness value of the ith planning $J_{1i} > \Delta T$, store the values of $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_N]^{\mathrm{T}}$ at time $t_i + \Delta T$, and set it as an initialization for the $(i+1)^{th}$ re-planning. Then let i=i+1, and return to Step4. Otherwise, go to Step 7;
7. Store the optimal solution as $\boldsymbol{\lambda}^* = [\boldsymbol{\lambda}_0^*, \boldsymbol{\lambda}_1^*, ..., \boldsymbol{\lambda}_N^*]^{\mathrm{T}}$, and obtain the corresponding flat output variables $\bar{\boldsymbol{Y}}^*(\sigma) = [\boldsymbol{Y}^*(\sigma), \dot{\boldsymbol{Y}}^*(\sigma), \ddot{\boldsymbol{Y}}^*(\sigma), ..., \boldsymbol{Y}^{*(\gamma+1)}(\sigma)]^{\mathrm{T}}$ according to Eq. (23), then map the flat outputs space to the state and control inputs space by flat transformation;
8. Substitute the obtained optimal control input $\boldsymbol{\tau}^*$ into the system dynamic models, and obtain the actual state variables by numerical integral calculations. If the error between the actual final condition and the desired final condition does not meet the precision requirement, then increase the number of LGL points as N=N+1, and return to Step 1, else stop.

5. Results and discussion

To investigate the effectiveness and robustness of the proposed re-planning algorithm, numerical simulations have been carried out for two different cases with multi static obstacles and multi moving obstacles, respectively. The algorithm has been coded in MATLAB R2012a and simulations are run on the PC with 2.1 GHz CPU/2GB RAM. The NLP solver for re-planning process used here is KNITRO (Byrd et al., 2006).

In the cases studies, the simulation parameters for PSO algorithm are selected as: the population size *s*=30; the maximum number of iterations $K_{\max} = 1000$; $c_1 = c_2 = 2$ and the inertia weighting factor w_k scales linearly between 0.4 and 0.9. The number of LGL points is 11 with *N*=10; the re-planning time horizon is given as $\Delta T = 1s$, and the weighting values for hybrid objective function are set as $\varepsilon_1 = \varepsilon_2 = 0.5$.

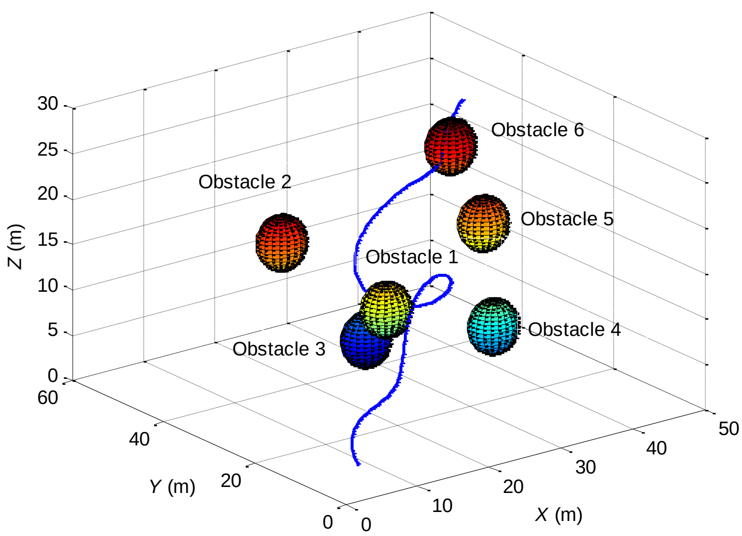
5.1 *Case1: Static obstacles avoidance*

The scenario in this case study is that an AUV is travelling in 3-D workspace, from the start point $[0,0,0,0,5,5,2,-\pi/4]^{\mathrm{T}}$ to the destination point $[0,0,0,0,45,45,22,\pi/4]^{\mathrm{T}}$. Six static obstacles are considered for evaluation of the re-planning algorithm, which are assumed to be spherical with the same radius of 3m.

Fig. 3 displays collision-free trajectories of the AUV obtained by only PSO algorithm with a random initialization and the time taken to arrive at final point is $t_f = 183\mathrm{s}$. Fig. 4 shows an optimal collision-free path obtained by hybrid PSO-LPM global planning algorithm with the final arrival time as $t_f = 111.7\mathrm{s}$. This shows that the hybrid PSO-LPM algorithm is able to find a better optimal trajectory compared to PSO algorithm alone. The PSO algorithm here is only used to find a set of initial guesses for LPM-based algorithm rather than a global optimum.

Fig. 5 shows an optimal path of AUV based on the re-planning scheme, and the total travelling time is $t_f = 130.27\mathrm{s}$. It can be found that although the globally planned trajectory is slightly different from the re-planned one; both of them can guide the AUV to the final destination successfully without collision with any obstacles. In this case all the positions of static obstacles are assumed to be exactly known as *a priori*, thus the global PSO-LPM algorithm can be utilized for the purpose of collision avoidance with sufficiently enough LGL points, in order to avoid the possible collisions between any two LGL points as shown in Fig. 4(a). It should be noted as the number of LGL points increases, the complexity and time taken for the optimization will increase, resulting in a more computational burden. The proposed algorithm deals with the obstacles by local re-planning with optimized LGL points, which not only reduces the time consumption, but also reduces the risk of collision as shown in Figs. 4(b) and 5(b) respectively. However, the re-planning scheme has to evaluate the collision risk and refine the path in each local planning process to keep the AUV a safe distance from all the obstacles. It can be seen in Figs. 4(b) and 5(b), the value of objective function for local re-planning is thus almost twenty seconds longer than that of global planning.

a



b

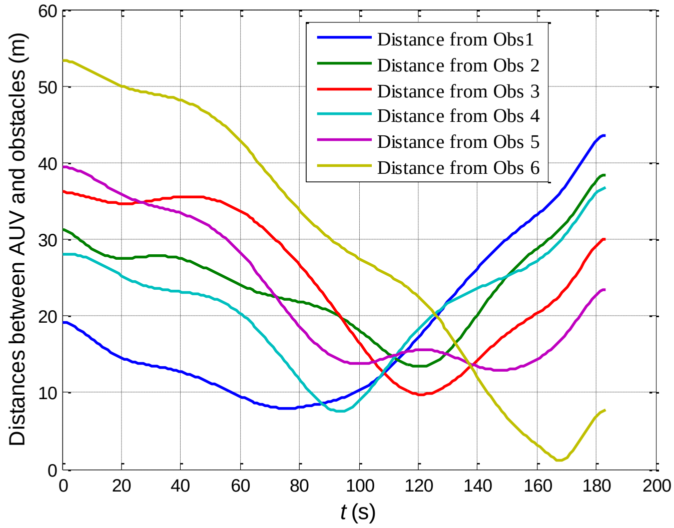
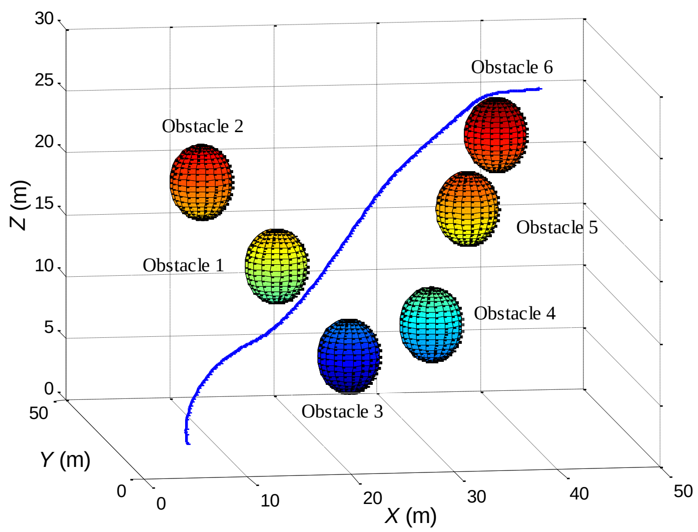


Fig. 3. Planned trajectories of AUV by PSO algorithm in Case 1. (a) Trajectory of AUV in 3-D workspace. (b) Distances between planned trajectory of AUV and each obstacle.

a



b

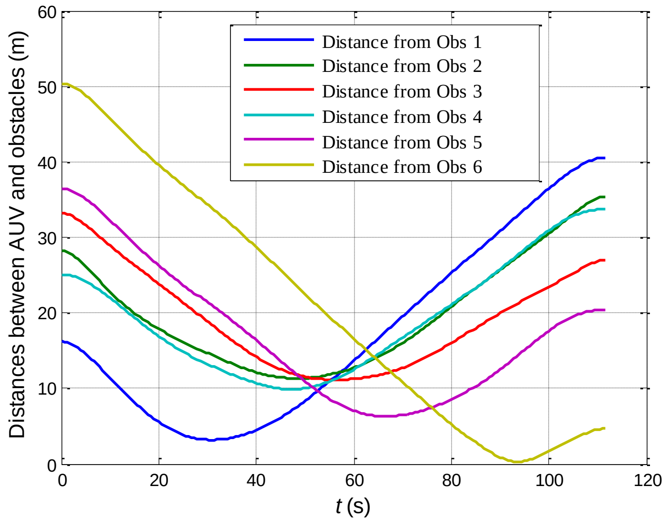
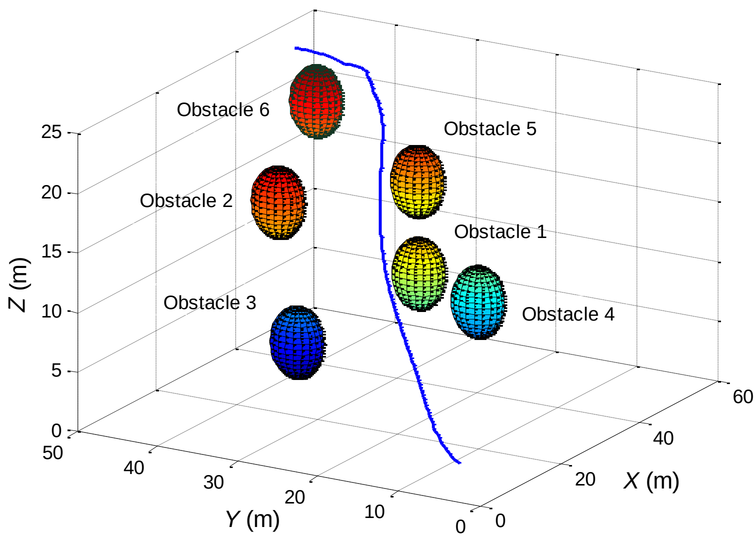


Fig. 4. Globally planned trajectories of AUV by hybrid PSO-LPM algorithm in Case 1. (a) Trajectory of AUV in 3-D workspace. (b) Distances between globally planned trajectory of AUV and each obstacle.

a



b

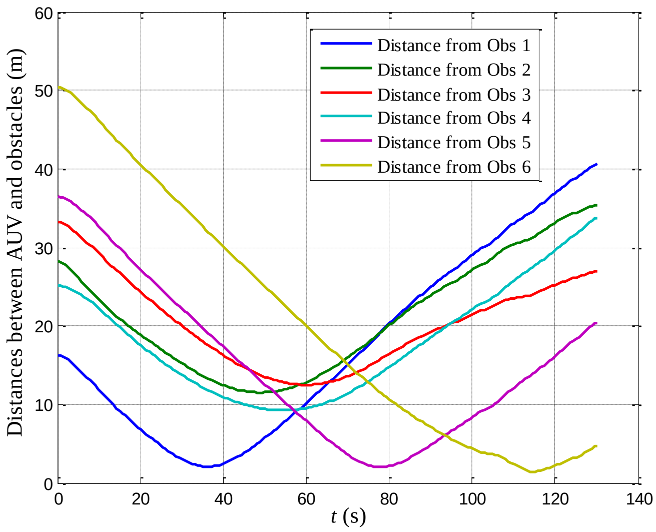
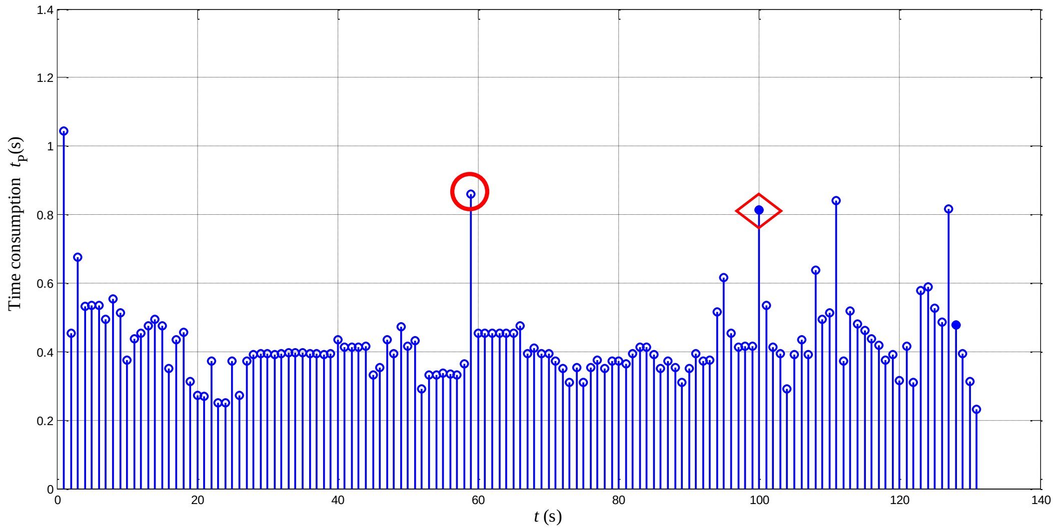


Fig. 5 Re-planned trajectories of AUV in Case 1. (a) Trajectory of AUV in 3-D workspace. (b) Distances between re-planned trajectory of AUV and each obstacle.

Fig. 6(a) shows the time taken for each planning in the whole re-planning process, where the hollow dot represents a success in finding an optimal solution while the blue dot represents a failure. It can be found the computational time for each planning except the first one is shorter than the given re-planning time horizon $\Delta T = 1\text{s}$, which ensures that the re-planning scheme can be used on-line. Fig. 6(b) displays the values of objective function obtained by each re-planning process, which gradually decrease as the AUV moves closer to the target. However, the curve is not smooth enough, i.e., it drops considerably at the time t=59s and t=101s. As shown in Fig. 6(a), the 59th re-planning process (marked with a circle) is successful to obtain an optimal solution, but the time consumption is excessive, which causes a sudden change in the value of objective function. Herein, the updated path obtained by previous successful re-planning is applied to the AUV, until the next successful re-planning is achieved. In Fig.

6(a), the 100th re-planning process (marked with a diamond) fails to find an optimum, while the 101th re-planning succeeds, which also makes the fitness values change considerably. As shown in Figs. 6(a) and 6(b), it is found that the sudden changes in the values of objective function correspond to both excessive time consumption for re-planning and the failure in finding an optimal re-planning path.

a



b

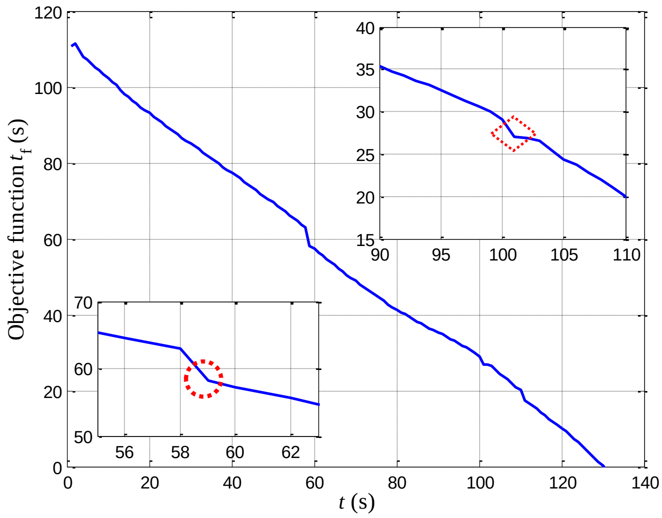


Fig. 6. Relations between computational time and objective function. (a) Computational time for each planning. (b) Values of objective function.

5.2 *Case 2: Dynamic obstacles avoidance*

In previous case, it is assumed that the positions of obstacles are precisely known, and the planned path can be executed perfectly. However, in realistic ocean fields, the locations of obstacles are not usually known precisely. In this section, the re-planning problem will tackle three moving obstacles with varying levels of position uncertainty.

The model of dynamic obstacles is assumed to be a linear and discrete-time system as defined in Zeng et. al (2015):

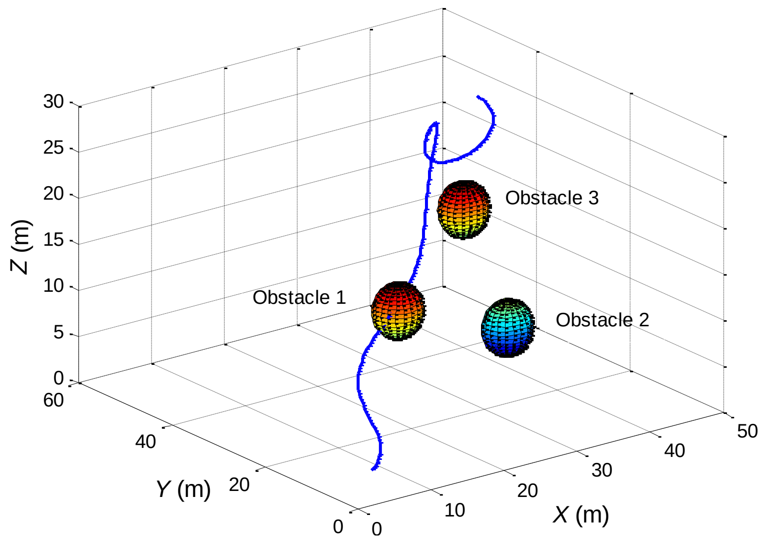
$$\boldsymbol{O}_i = \boldsymbol{H}_O \boldsymbol{O}_{i-1} + \boldsymbol{Z}_O X_{i-1} + \boldsymbol{L}_O du \tag{25}$$

where, $\boldsymbol{O}_i = [O_{Pi}, O_{Vi}, O_{Ui}]^{\mathrm{T}}$ represents the state of obstacles at time t_i (here, assuming $t_{M_i} = 0$) measured from the on-board sonar sensors, and O_{Pi}, O_{Vi}, O_{Ui} denote position, velocity and uncertainty of the obstacle at time t_i, respectively; $X_{i-1} \sim N(0, 0.005^2)$ is Gaussian disturbance acting on velocity, which is independent from the disturbances caused by $X_{0\sim i-2}$; du is the rate of uncertainty, which is set as $du = 0.005\mathrm{m/s}$. The parameter matrices are written as:

$$\boldsymbol{H}_O = \begin{bmatrix} 1 & \Delta T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{Z}_O = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \boldsymbol{L}_O = \begin{bmatrix} 0 \\ 0 \\ \Delta T \end{bmatrix} \tag{26}$$

Assuming the initial velocities for all the three moving obstacles are 0m/s, with the initial locations distributed randomly. Fig. 7 displays the optimal trajectory obtained in the first global planning, which can be regarded as the global planning problem in Case 1 with only three static obstacles. Obviously, the global planning can easily find a collision-free path as shown in Fig. 7(b) with the objective function in total time $t_f = 125.63\mathrm{s}$. In Fig. 8(a), the red line displays the re-planned optimal trajectory of AUV, while blue lines show the paths of the centers of mass of three dynamic obstacles, respectively. Further, the three spheres mark the location of each obstacle with shortest distance to the AUV in the whole re-planning process. Similarly, as illustrated in Fig. 8(b), the re-planning algorithm also succeeds in finding a time-optimal collision-free path in 3-D workspace even with uncertain moving obstacles. It can be observed the objective function obtained by re-planning is $t_f = 163.79\text{s}$, since the AUV requires more time to overcome the possible collisions caused by dynamic obstacles as well as their uncertainty in both positions and velocities.

a



b

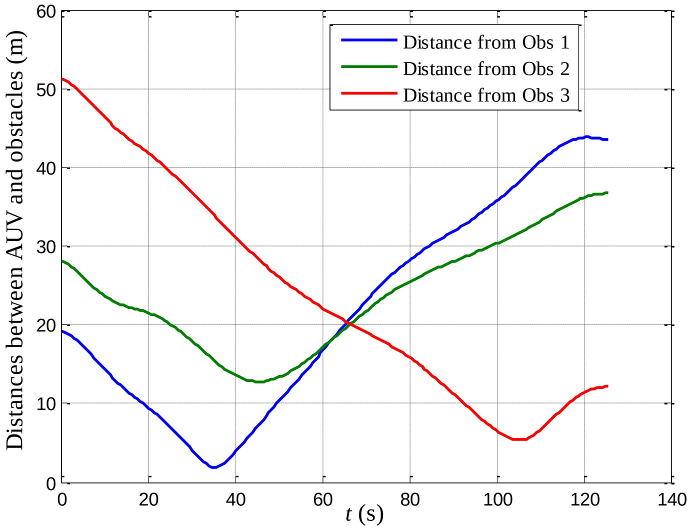
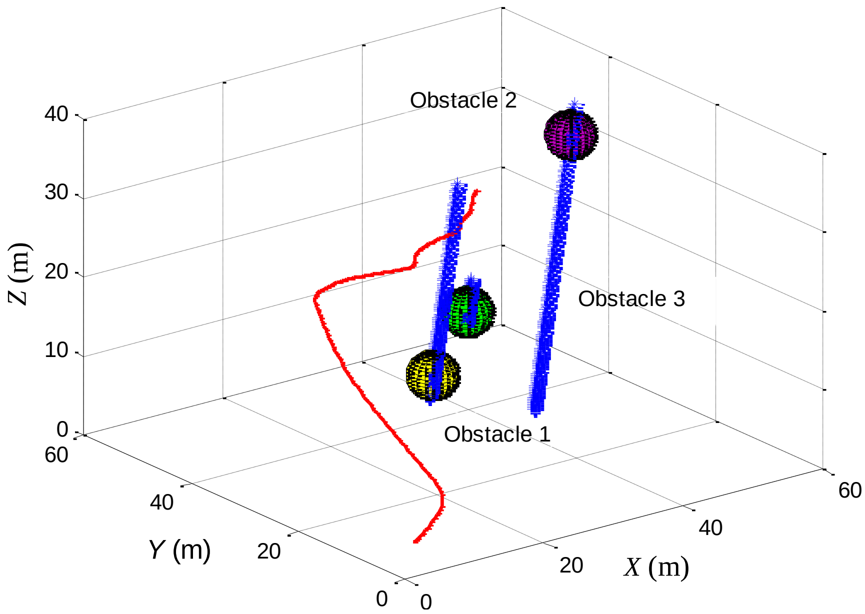


Fig. 7. Globally planned trajectories of AUV by hybrid PSO-LPM algorithm in Case 2. (a) Trajectory of AUV in 3-D workspace. (b) Distances between globally planned trajectory of AUV and each obstacle.

a



b

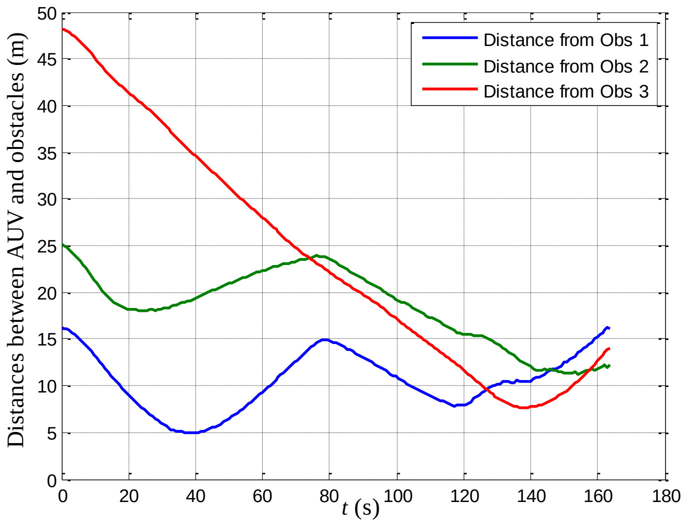
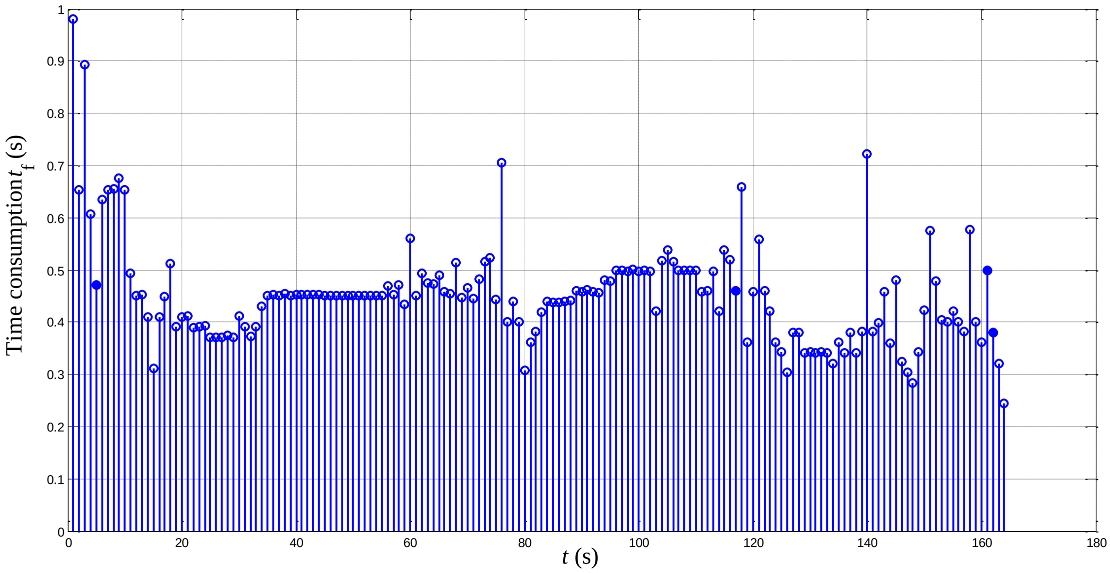


Fig. 8. Re-planned trajectories of AUV in Case 2. (a) Trajectory of AUV in 3-D workspace. (b) Distances between re-planned trajectory of AUV and each obstacle.

Fig. 9 plots the time taken for each re-planning and the relations between the values of objective function and the time taken for the whole re-planning process. In both Figs. 6(a) and 9(a), it can be found the first global planning takes the longest time than the rest re-planning process, since it is the sum of the time consumed for both PSO optimization process and LPM optimization process. And, the $(i+1)^{th}$ re-planning takes the solution obtained in the i^{th} re-planning as an initialization to decrease the total time consumption.

a



b

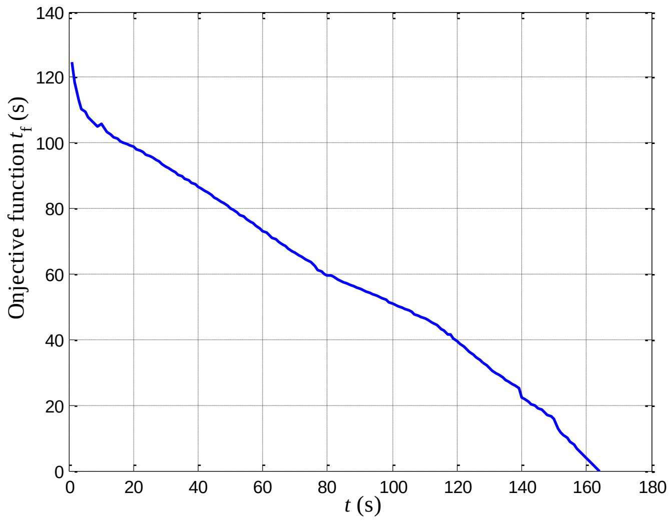


Fig. 9. Relations between computational time and objective function. (a) Computational time for each planning. (b) Values of objective function.

5.3 *Robustness assessment*

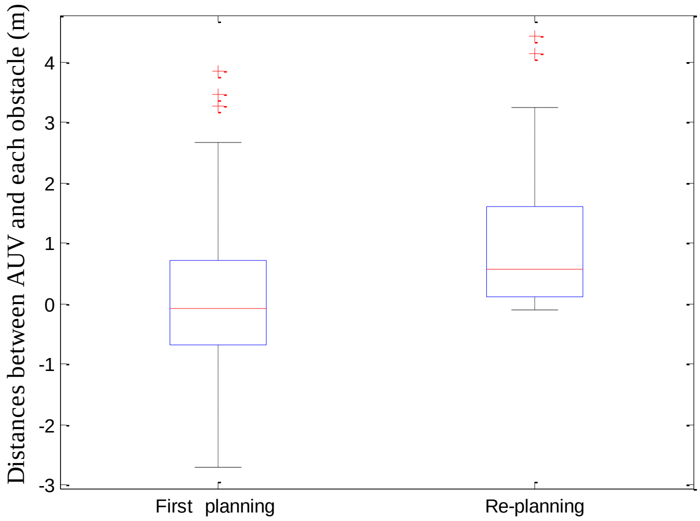
In this subsection, Monte Carlo simulations with random initial values will be carried out to demonstrate the robustness of the proposed re-planning algorithm. First, simulations are performed on a 100-run basis for Case 1 discussed in Section 5.1, and the results are illustrated in Fig. 10. Fig. 10(a) displays the shortest distances between

AUV and the obstacles in the whole re-planning process, where the positive values represent safe condition, while the negative values mean collision. It is obvious in Figs. 10(a-b), although the first global planning is superior to the re-planning scheme in objective functions, it fails to avoid collision for almost half of the 100-run Monte Carlo simulations. Fig. 10(c) plots the terminal error of AUV, which is defined as the distance between the desired final position and the actual planned destination of AUV. It is obvious that the terminal errors here are acceptable in realistic applications, and an improvement could be obtained by increasing the number of LGL points.

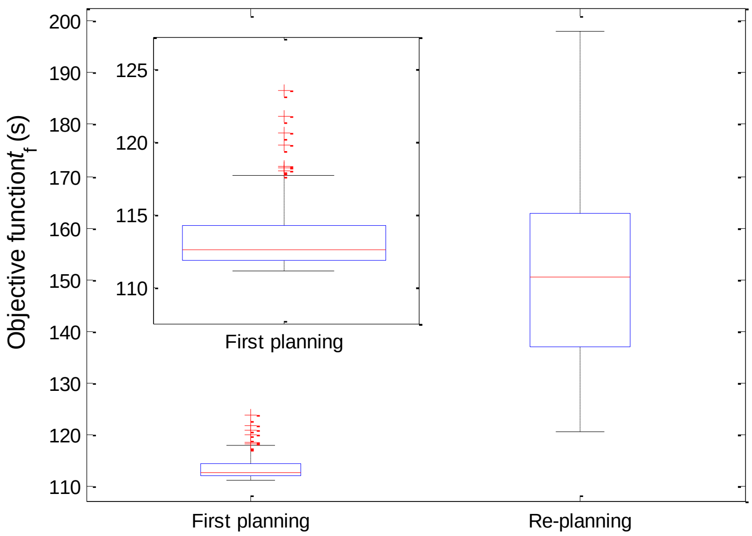
Fig. 11 shows the 100-run Monte Carlo simulation results also for Case 1 without considering the flatness property of AUV. It can be seen that the average time consumption for each re-planning is longer than the given re-planning time horizon ΔT, which causes a majority of plannings failing in the whole re-planning process, the re-planning thus cannot be executed on-line as expected. An obvious phenomenon is that the values of objective function obtained without considering flatness property are much longer than those displayed in Fig. 10(b). On the other hand, this set of Monte Carlo simulation results illustrate the flatness property of AUV is effective to reduce the time usage of planning, which sometimes is a necessary condition for the application of re-planning scheme on-line.

Fig. 12 runs 100 Monte Carlo simulations with random initial values to assess the robustness of proposed algorithm for Case 2. The results show that the PSO-LPM algorithm is not only effective for the ocean environments with static obstacles but also successful in dealing with moving obstacles with varying levels of positional uncertainty.

a



b



c

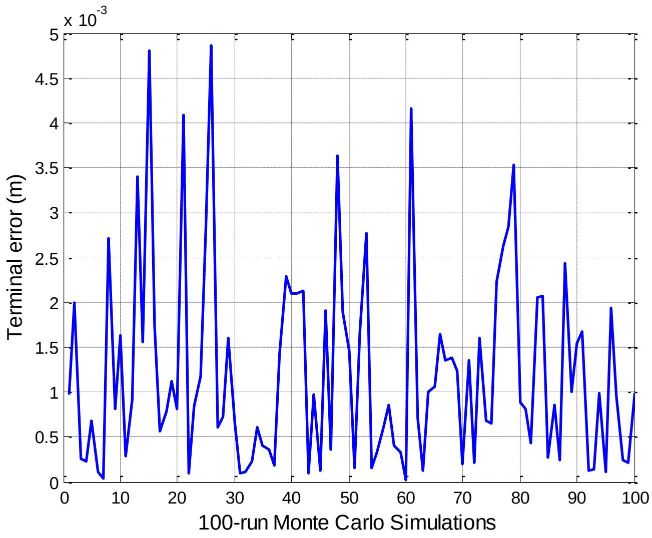


Fig. 10. Results of 100-run Monte Carlo simulations for Case 1. (a) Shortest distances between AUV and obstacles. (b) Values of objective function. (c) Terminal errors.

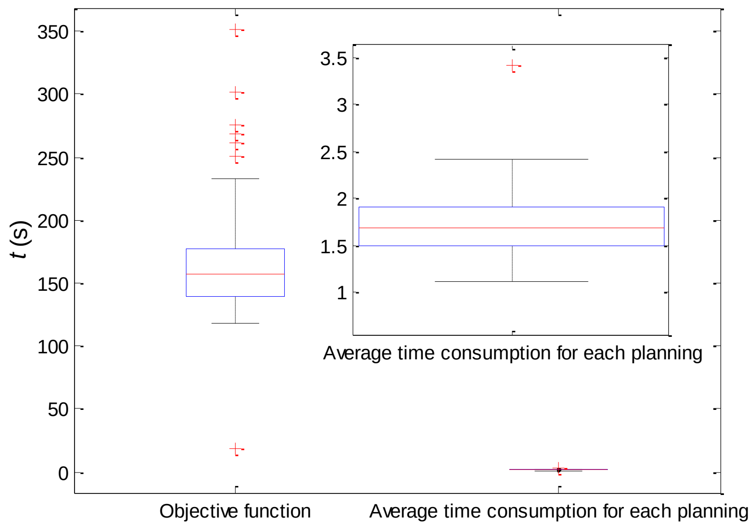
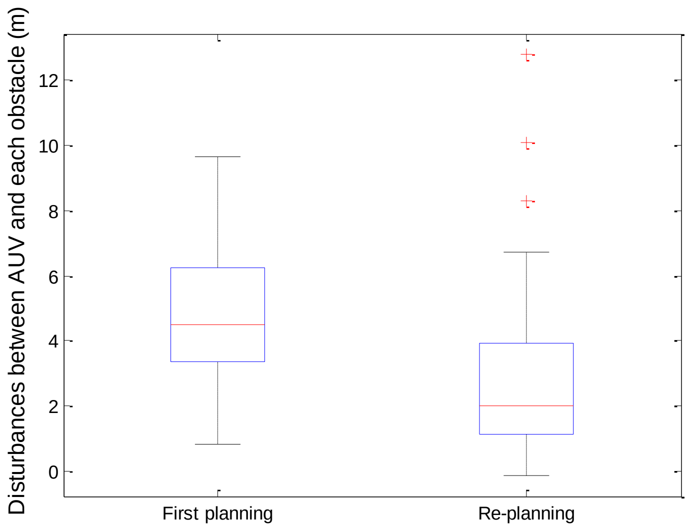
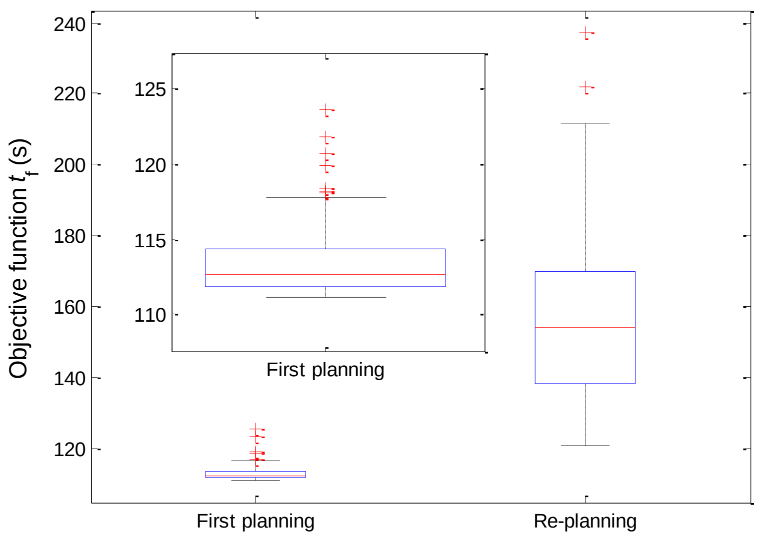


Fig. 11. Results of 100-run Monte Carlo simulations for Case 1 without flatness property of AUV.

a


Disturbances between AUV and each obstacle (m)
0
2
4
6
8
10
12
First planning
Re-planning

b


Objective function t_f (s)
120
140
160
180
200
220
240
110
115
120
125
First planning
First planning
Re-planning

c

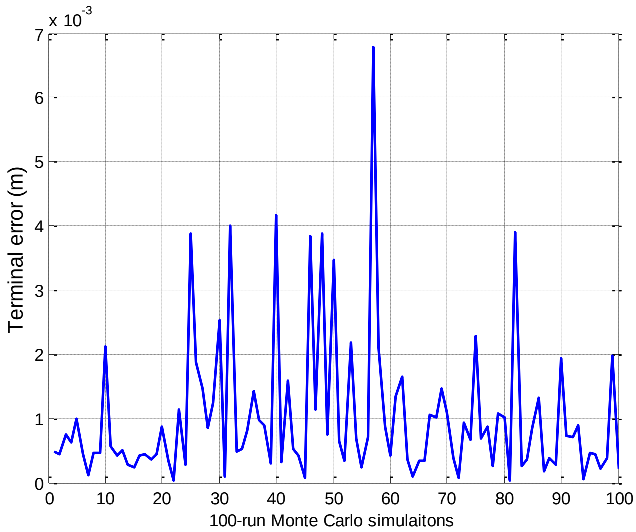


Fig. 12. Results of 100-run Monte Carlo simulations for Case 2. (a) Shortest distances between AUV and obstacles. (b) Values of objective function. (c) Terminal errors.

6. Conclusions

This paper presents an on-line collision-free path planning strategy of AUV, which incorporates PSO algorithm with LPM-based re-planning scheme to continuously refine the optimal trajectories in complex ocean environments. Simulation results illustrate that the proposed path planner succeeds in collision avoidance against both static and dynamic obstacles with uncertainty in positions and velocities, and by using PSO as an initialization generator, the hybrid PSO-LPM planner is shown to be capable of finding a more optimal solution than PSO algorithm alone. In addition, due to the differential flatness property of AUV, the time consumption for each planning process is further reduced, which ensures that the re-planning scheme can be applied on-line. Finally, Monte Carlo simulations demonstrate the robustness of the proposed scheme.

The next stage in this work is to improve the practicability of current algorithm in realistic and complex ocean environments. The ocean environments are composed of obstacles, irregularly shaped terrains and strong current fields which vary over time both in directions and strength. Thus a natural extension of the above work is to develop an efficient path planner, which can integrate current forecasts information to allow mission planning over long time duration through variable currents.

Acknowledgments:

This work is supported in part by Fundamental Research Funds for the Central Universities (No. HIT.NSRIF.2013135 and HIT.KISTP.2014029), Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (No. HIT.NSRIF.2014139) and Science and Technology Foundation for the Universities in Shandong Province (No. J14LN93) and joint works under Royal Academy of Engineering Newton Research Collaboration Programme (Reference: NRCP/1415/112). The authors thank for the financial support from China Scholarship Council (CSC).

Reference

Aghababa, M. P., 2012. 3D path planning for underwater vehicles using five evolutionary optimization algorithms avoiding static and energetic obstacles. Applied Ocean Research. 38, 48-62.

Byrd, R. H., Nocedal, J., Waltz, R. A., 2006. KNITRO: An integrated package for nonlinear optimization, in Large-scale nonlinear optimization. Springer, US.

Carsten, J., Ferguson, D., Stentz, A., 2006. 3D field D*: Improved path planning and replanning in three dimensions. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems. Beijing China. pp. 3381-3386.

Carroll, K. P., McClaran, S. R., Nelson, E. L., Barnett, D. M., Friesen D. K., William G., 1992. AUV path planning: an A* approach to path planning with consideation of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones. Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology, 1992.

Daily, R., Bevly, D. M., 2008. Harmonic potential field path planning for high speed vehicles. In: Proceedings of American Control Conference. Washington, USA. pp. 4609-4614.

Elnagar, G., Kazemi, M. A., Razzaghi, M., 1995. The pseudospectral Legendre method for discretizing optimal control problems. IEEE Transactions on Automatic Control, 40:1793-1796.

Ferguson, D., Stentz, A., 2006. Using interpolation to improve path planning the field D* algorithm. Journal of Field Robot. 23(2), 79-101.

Fliess, M., Lévine, J., Martin, P., Rouchon, P., 1995. Flatness and defect of non-linear systems: introductory theory and examples. International journal of control. 61(6), 1327-1361.

Fossen, T.I., 1994. Guidance and Control of Ocean Vehicles. John Wiley & Sons, US.

Gong, Q., Kang, W., Ross, I. M., 2006. A pseudospectral method for the optimal control of constrained feedback linearizable systems. IEEE Transaction of Automatic Control. 51(7), 1115-1129.

Iwakami, H., Ura, T., Asakawa, K., Fujii, T., Nose, Y., Kojima, J., Shirasaki, Y., Asai, T., Uchida, S., Higashi, N., Fukuchi, T., 2002. Approaching whales by autonomous underwater vehicle. Marine Technology Society Journal. 36(1), 80-85.

Incze, M. L., 2011. Light weight autonomous underwater vehicles (AUVs) performing coastal survey operations in REP10A. Ocean Dynamics. 61(11), 1955-1965.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of the IEEE

International Conference on Neural Networks. pp. 1942-1945.

Khatib, O., 1986. Real-time obstacle avoidance for manipulators and mobile robots. The international journal of robotics research. 5(1), 90-98.

Kondo, H., Ura, T., 2004. Navigation of an AUV for investigation of underwater structures. Control engineering practice. 12(12), 1551-1559.

Kumar, R. P., Dasgupta, A., Kumar, C. S., 2005. Real-time optimal motion planning for autonomous underwater vehicles. Ocean engineering. 32(11), 1431-1447.

Lévine J. On Necessary and Sufficient Conditions for Differential Flatness[J]. Applicable Algebra in Engineering Communication & Computing, 2011, 22(1):47-90.

Liang, J. H., Lee, C. H., 2015. Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm. Advances in Engineering Software. 79, 47-56.

Likhachev, M., Ferguson, D. I., Gordon, G. J., Stentz, A., Thrun, S., 2005. Anytime Dynamic A*: An Anytime, Replanning Algorithm. In: Proceedings of the 15th International Conference on Automated Planning and Scheduling. California, USA. pp. 262-271.

Lin, C. F., Tseng, C. Y., 2006. Development of a cost effective mini autonomous underwater vehicle. Journal of Marine Science and Technology. 14(2), 119-126.

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 Robot. 30(5),741–762.

Pereira, A. A., Binney, J., Jones, B. H., Ragan, M., Sukhatme, G. S., 2011. Toward risk aware mission planning for autonomous underwater vehicles. IEEE International Conference on

Intelligent Robots and Systems.

Spangelo, I., Egeland, O., 1994. Trajectory planning and collision avoidance for underwater vehicles using optimal control. IEEE Journal of Oceanic Engineering. 19(4), 502-511.

Sullivan, J., Waydo, S., Campbell, M., 2003. Using stream functions for complex behavior and path generation. In: Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit. Austin, USA.

Wang, B., Wan, L., Xu, Y. R., Qin, Z. B., 2009. Modeling and simulation of a mini AUV in spatial motion. Journal of Marine Science and Application. 8(1), 7-12.

Yuh, J., 2000. Design and control of autonomous underwater robots: A survey. Autonomous Robots. 8(1), 7-24.

Zeng, Z., Sammut, K., Lammas, A., He, F., Tang, Y., 2015. Efficient path re-planning for AUVs operating in spatiotemporal currents. Journal of Intelligent & Robotic Systems. 79, 135-153.

Li, Z., Yang, C., Ding, N., Bogdan, S., Ge, T., 2012. Robust adaptive motion control for underwater remotely operated vehicles with velocity constraints. International Journal of Control, Automation and Systems, 10 (2), 421-429.