2016-11-15

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

Zhuang, Y

http://hdl.handle.net/10026.1/6737

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

46 keeps the AUV away from various obstacles from its initial position to the final

47 destination. Obviously, such planning must be completed on-line and follow some

48 optimization strategy in order to ensure the safety and performance of the vehicles.

49   In recent years, a variety of solution approaches have been developed and applied to

50 the collision-free path planning problems of underwater vehicles. These approaches can

51 be roughly divided into two categories: global planning and local re-planning. When

52 the environment is completely known as a *priori* with static obstacles, a global path

53 planner can be utilized off-line via optimal control theory such as nonlinear

54 programming (Spangelo and Egeland, 1994; Kumar et al., 2005), heuristic algorithms

55 (Likhachev et al., 2005; Carsten et al., 2006) and artificial potential field approaches

56 (Khatib, 1986; Daily and Bevly, 2008; Sullivan et al., 2003). Another class of

57 algorithms to this type of optimization problems are graph search methods including

58 A* algorithm (Carroll et al., 1992; Pereira et al., 2011, 2013) and D* algorithm

59 (Ferguson and Stentz, 2006). On the other hand, if the vehicles operate in unknown or

60 only partially known environments with dynamic obstacles, then subsequent local re-

61 planning due to changing environments should be carried out on-line, which makes the

62 path planning problem intrinsically NP hard (Non-deterministic Polynomial), and

63 finding an optimum solution is not guaranteed. To deal with these problems,

64 evolutionary algorithms have been used, such as genetic algorithm (GA) or particle

65 swarm optimization algorithm (PSO) (Zeng et al., 2015; Aghababa, 2012).

66 Evolutionary algorithms usually have better ability to converge to a global optimum or

67 a near optimal solution than traditional optimization methods, and also not sensitive to

68 initial guesses of solutions. However, evolutionary algorithms are prone to poor

69 numerical accuracy and difficult constraints handling.

70   In this paper, a novel hybrid algorithm is proposed for time-optimal collision-free

71 path planning of an AUV, which combines PSO algorithm and Legendre

72 pseudospectral method (LPM). The main idea of the algorithm is that: for the first phase,

73 PSO is used as an initial values generator due to its robustness to random initializations.

74 It will be applied for the problem with a set of random initial values, in order to enhance

75 the global searching capability. PSO stops iterating after a stopping criterion is achieved,

76 and the algorithm goes to the second phase. In the second phase, the searching scheme

77 is switched to LPM to achieve a faster and better convergence around the global

78 optimum. The differential flatness property of AUV is also utilized to reduce the

79 number of constraints and variables to be optimized in order to decrease the total time

80 consumption. If the time taken for each optimization is less than the given time horizon

81 for re-planning, then the hybrid planning algorithm can repeatedly be solved on-line.

82 This re-planning approach introduces feedback to compensate for uncertainty, and the

83 guidance law obtained for the AUV ensures obstacle avoidance and offers high

84 performance.

85 The contributions of this paper are as follows:

86 • Integrating PSO and LPM as a hybrid optimization algorithm, which can improve

87   both robustness to random initializations and convergence rate around global

88   optimum;

89 • Employing flatness property of AUV to reduce the time consumption of

90   optimization ;

91   • Using re-planning scheme to deal with the collision avoidance against both static

92   and dynamic obstacles.

93   The remainder of this paper is organized as follows. Section 2 introduces the

94   mathematical models of an AUV and its flatness property; Section 3 defines the

95   problem statement and reformulates the problem in flat outputs space by using flat

96   transformation; Section 4 proposes the details of path re-planning scheme based on

97   hybrid PSO-LPM algorithm; Section 5 shows the simulation results and robustness

98   assessment of the proposed algorithm; Concluding remarks are then presented in

99   Section 6.

100   **2. Mathematical model of an AUV**

101   2.1 *Nonlinear AUV equations of motions*

102   In general, the dynamic behaviors of an AUV are commonly described in two

103   coordinate systems, namely earth-fixed reference frame and body-fixed reference frame

104   as shown in Figure 1.

105

106

107

108

109

110

111

112

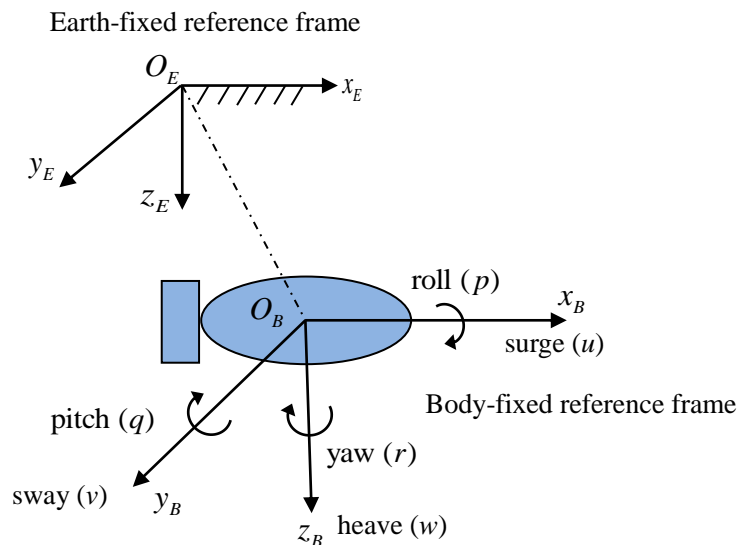113

114

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

116   A general description of six-DOF nonlinear equations of AUV motions is described

117   as follows (Fossen, 1994):

118

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

119   where, $\boldsymbol{v} = [u, v, w, p, q, r]^{\mathrm{T}}$ is a velocity vector and $\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^{\mathrm{T}}$ is a

120   displacement vector. u, v, w denote linear velocities along surge, sway and heave

121   directions; p, q, r denote rotational velocities in roll, pitch and yaw motions; x, y, z

122   are positions along surge, sway and heave directions, respectively and ϕ, θ, ψ show the

123   Euler angles of the vehicle in earth-fixed frame; $\boldsymbol{J}(\boldsymbol{\eta})$ is Jacobian transformation

124   matrix; $\boldsymbol{M}$ denotes system inertia matrix; $\boldsymbol{C}(\boldsymbol{v})$ is Coriolis-centripetal matrix; $\boldsymbol{D}(\boldsymbol{v})$ is

125   hydrodynamic damping matrix; $\boldsymbol{g}(\boldsymbol{\eta})$ represents buoyant and gravitational forces and

126   moments; $\boldsymbol{\tau}$ is the vector of control inputs.

127   Without loss of generality, it is assumed that: (i) the center of mass (CM) coincides

128   with the center of gravity (CG) and center of buoyancy (CB); (ii) the hydrodynamic

129   drag terms of order higher than two can be neglected; (iii) the motions in roll and pitch

130   directions are negligible ($p = q = 0; \phi = \theta = 0$). By selecting the principal axis, the

131   inertia matrix and Coriolis-centripetal matrix are defined as:

132

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

133   here, m is the mass of the vehicle; I_z is the moment of inertia in yaw motion. The matrix

134      $\boldsymbol{D}(\boldsymbol{v})$ is assumed to be non-coupled with only uncertain linear/quadratic damping

135   coefficients $X_u / X_{u|u|}, Y_v / Y_{v|v|}, Z_w / Z_{w|w|}$ and $N_r / N_{r|r|}$. Hydrodynamic damping

136   matrix $D(v)$ and $g(\eta)$ can thus be described as:

137
$$D(v) = \begin{bmatrix} X_u + X_{u|u|} \mid u \mid & 0 & 0 & 0 \\ 0 & Y_v + Y_{v|v|} \mid v \mid & 0 & 0 \\ 0 & 0 & Z_w + Z_{w|w|} \mid w \mid & 0 \\ 0 & 0 & 0 & N_r + N_{r|r|} \mid r \mid \end{bmatrix}, \quad g(\eta) = 0 \quad (3)$$

138    $\tau = [T_u, T_v, T_w, 0, 0, T_r]^T$, where T_u, T_v, T_w and T_r represent available control inputs in

139   surge, sway, heave, and yaw directions, respectively. The kinematic and dynamic

140   equations of AUV can be represented as:

141
$$\begin{cases} \dot{x} = u\cos\psi - v\sin\psi \\ \dot{y} = u\sin\psi + v\cos\psi \\ \dot{z} = w \\ \dot{\psi} = r \\ m\dot{u} - mvr + X_u u + X_{|u|u} \mid u \mid u = T_u \\ m\dot{v} + mur + Y_v v + Y_{|v|v} \mid v \mid v = T_v \\ m\dot{w} + Z_w w + Z_{|w|w} \mid w \mid w = T_w \\ I_z \dot{r} + N_r r + N_{|r|r} \mid r \mid r = T_r \end{cases} \quad (4)$$

142   2.2 *Flatness analysis of an AUV*

143   A control system

144
$$\dot{x} = f(x, u) \quad x \in \mathbf{R}^n \quad u \in \mathbf{R}^m \quad (5)$$

145   is differentially flat or just flat, if there exist smooth maps C, A and B defining on open

146   neighborhoods of $\mathbf{R}^n \times (\mathbf{R}^m)^{\rho+1}, (\mathbf{R}^m)^{\gamma+1}$ and $(\mathbf{R}^m)^{\gamma+2}$, such that

147
$$\begin{aligned} y &= C(x, u, \dot{u}, \ddot{u}, ..., u^{(\rho)}) \\ x &= A(y, \dot{y}, \ddot{y}, ..., y^{(\gamma)}) \\ u &= B(y, \dot{y}, \ddot{y}, ..., y^{(\gamma+1)}) \end{aligned} \quad (6)$$

148   here ρ and γ are positive integers, y is called a set of flat outputs, and the components of

149   y are not related by a differential relation (Fliess et al., 1995, Lévine J, 2011). The

150   definition shows that if there exist a set of flat outputs with the same number of control

151　inputs, then the state and control variables can both be expressed with them in flat

152　outputs space.

153　　By observing Eq. (4) carefully, a set of flat outputs can be easily found as

154　$Y = [Y_1, Y_2, Y_3, Y_4]^T = [x, y, z, \psi]^T$, and then the mathematical model of AUV can be

155　transformed into

156
$$
\begin{cases}
u = \dot{x}\cos\psi + \dot{y}\sin\psi = \dot{Y}_1\cos Y_4 + \dot{Y}_2\sin Y_4 \\
v = \dot{y}\cos\psi - \dot{x}\sin\psi = \dot{Y}_2\cos Y_4 - \dot{Y}_1\sin Y_4 \\
w = \dot{z} = \dot{Y}_3 \\
r = \dot{\psi} = \dot{Y}_4
\end{cases}
\tag{7}
$$

157
$$
\begin{cases}
T_u = m\dot{u} - mvr + X_u u + X_{|u|u}|u|u \\
\quad = m(\ddot{Y}_1\cos Y_4 + \ddot{Y}_2\sin Y_4) + (X_u + X_{|u|u}|\dot{Y}_1\cos Y_4 + \dot{Y}_2\sin Y_4|)\cdot(\dot{Y}_1\cos Y_4 + \dot{Y}_2\sin Y_4) \\
T_v = m\dot{v} + mur + Y_v v + Y_{|v|v}|v|v \\
\quad = m(\ddot{Y}_2\cos Y_4 - \ddot{Y}_1\sin Y_4) + (Y_v + Y_{|v|v}|\dot{Y}_2\cos Y_4 - \dot{Y}_1\sin Y_4|)\cdot(\dot{Y}_2\cos Y_4 - \dot{Y}_1\sin Y_4) \\
T_w = m\dot{w} + Z_w w + Z_{|w|w}|w|w \\
\quad = m\ddot{Y}_3 + (Z_w + Z_{|w|w}|\dot{Y}_3|)\dot{Y}_3 \\
T_r = I_z\dot{r} + N_r r + N_{|r|r}|r|r \\
\quad = I_z\ddot{Y}_4 + (N_r + N_{|r|r}|\dot{Y}_4|)\dot{Y}_4
\end{cases}
\tag{8}
$$

158　**3. Problem formulation and transformation**

159　　This paper aims at finding a time-optimal collision-free path planning scheme for

160　AUV, where the optimization criterion is used to obtain the minimum travelling time

161　whilst the collision constraints ensure that the path is collision-free from any static or

162　moving obstacles with uncertainty.

163　　Generally, the path planning problem can be formulated as an optimization problem:

164　find a path $X = [\nu; \eta; \tau]^T = [u, v, w, r, x, y, z, \psi, T_u, T_v, T_w, T_r]^T$, which minimizes the

165　performance index ($\bar{J}$):

166
$$
\min_{X} \bar{J} = t_f
\tag{9}
$$

167 subject to the vehicle dynamics described by Eq. (4), and the positional constraints from

168 the given initial condition X_0 and final destination X_f defined as:

169 $$X(t_0) = X(\nu(t_0); \eta(t_0); \tau(t_0)) = X_0 \quad ; \quad X(t_f) = X(\nu(t_f); \eta(t_f); \tau(t_0)) = X_f \qquad (10)$$

170 where, t_f is the final time. If the initial time is assumed $t_0 = 0$, then t_f is the total

171 travelling time of the AUV. The rotational velocities of the thrusters mounted on the

172 practical AUVs will have lower and upper limitations, which results in the following

173 control inputs constraints as

174 $$|\tau| \leqq \tau_{max}| \qquad (11)$$

175 where, τ_{max} should coincides to physical limitations of the thrusters.

176   In this section, to deal with the collision constraints, hybrid objective function is

177 employed, and a weighting scheme is introduced to trade-off between the total

178 travelling time and the risk of collision, the hybrid objective function is defined as

179 $$J(X) = \varepsilon_1 J_1(X) + \varepsilon_2 J_2(X) \qquad (12)$$

180 where, $\varepsilon_1, \varepsilon_2$ denote positive weighting values satisfying $\varepsilon_1 + \varepsilon_2 = 1$ and $J_1(X) = t_f$ as

181 described in Eq. (9).

182   The objective function for collision avoidance indicating distance information

183 between AUV and obstacles is defined as (Liang and Lee, 2015)

184 $$J_2(X) = \sum_{j=1}^{S} J_2^j(X), \qquad J_2^j(X) = \begin{cases} 0, & \left\| X_p - Obs_j \right\| > \delta_{obsj} \\ \dfrac{1}{\left\| X_p - Obs_j \right\|} - \dfrac{1}{\delta_{obs}}, & \left\| X_p - Obs_j \right\| \leq \delta_{obsj} \end{cases} \qquad (13)$$

185 where, $j=1,2 \ldots S$, S is the number of obstacles in the work space; Obs_j represents the

186 center of the jth obstacle; X_p is the position of AUV; δ_{obsj} denotes the given safe

187 distance between AUV and the jth obstacle, which can be obtained according to the

188 length of the AUV and the *radii* of the obstacles.

189 As shown in Eqs. (9-13), the optimization process needs to determine a large number

190 of variables, which will result in a huge time burden, especially for evolutionary

191 algorithms. Additionally, most optimization algorithms spend majority of time on

192 dealing with the differential equations constraints caused by the mathematical models

193 of system.

194 By the definition of differential flatness above, if a dynamic system is flat, then its

195 state and input variables can be parameterized in terms of a set of flat outputs and their

196 derivatives. The above original optimization problem thus can be converted and

197 reformulated in flat outputs space as: find a path $\bar{Y} = [Y, \dot{Y}, \ddot{Y}...Y^{(\gamma+1)}]^{\mathrm{T}}$ in order to

198 minimize the objective function described as

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

199

200 subject to the positional constraints as

201 $A(Y(t_0), \dot{Y}(t_0), \ddot{Y}(t_0)...Y^{(\gamma)}(t_0)) = X_0; \quad A(Y(t_f), \dot{Y}(t_f), \ddot{Y}(t_f)...Y^{(\gamma)}(t_f)) = X_f$ (15)

202 and the input variables constraints

$$| B(Y, \dot{Y}, \ddot{Y}...Y^{(\gamma+1)}) | \leq | \tau_{\max} | \tag{16}$$

203

204 where, flat transformation A is defined in Eq.(7), while transformation B is provided in

205 Eq. (8).

206 It can be found in the reformulation in flat outputs space, the constraints caused by

207 the nonlinear model of the AUV have been completely eliminated, and all the

208 displacement and control input variables can be parameterized by flat outputs, thus the

209 number of variables to be optimized has also been reduced by 60% from 12 to 4. The

210   time taken for path planning is thus considerably faster in this case, and makes the

211   optimization algorithm more possibly to re-plan the trajectory on-line.

212   **4   Hybrid PSO-LPM algorithm for path planning**

213   This paper focuses on the path planning problem of AUVs in complicated

214   environments with static and moving obstacles. In order to seek a collision-free path,

215   the planner should be capable of reacting fast to any new information about the

216   environments obtained by the corresponding software and sensors mounted on the

217   vehicles. The path planning of AUV in such environments should be a continuous and

218   closed-loop process, and the trajectory should be locally re-planned according to the

219   changing environments. The main idea of the re-planning scheme is illustrated in Fig.

220   2: where ΔT is the re-planning time horizon, t_{M_i} is the ith measurement time of the

221   sensors and t_{P_i} is the time taken for the ith re-planning process. At time t_i, the AUV

222   executes the trajectory generated by the $(i\text{-}1)^{th}$ re-planner (dotted line in Fig. 2), and this

223   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

224   obtained by the ith re-planning process according to the environment information

225   collected by the sensors at time $t_i + t_{M_i}$. The AUV will be guided along the new

226   trajectory (black line in Fig. 2) until the $(i+1)^{th}$ updated trajectory is obtained. It is

227   obvious in Fig. 2 that, if $\Delta T > \Delta t_i$, then a path update can be computed by incorporating

228   any new information of the changing ocean environment. Moreover, if ΔT is

229   sufficiently short, then the environment information can be fed back to the planner in

230   real-time, which can ensure the trajectory planned more safely and efficiently. However,

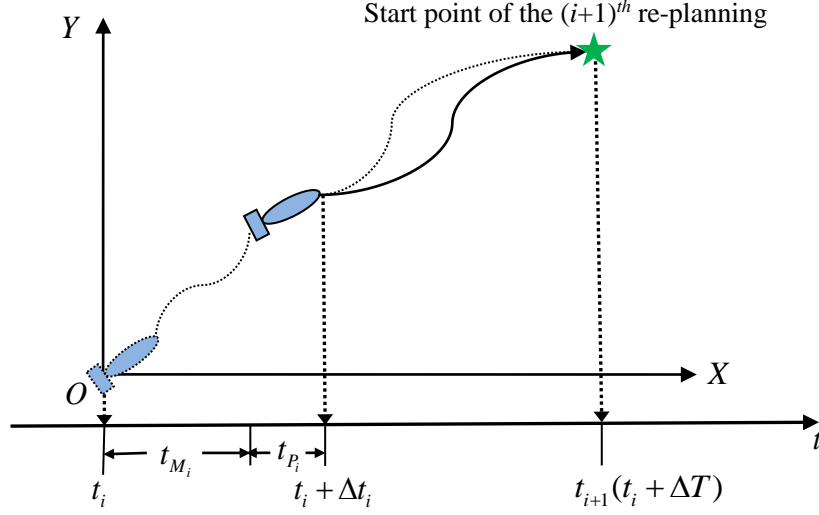231   the shorter the planning window is, the faster the planning algorithm is required.

232

233

234

235

236 Start point of the $(i+1)^{th}$ re-planning

237

238

Y

239

240

241 X

O

242 t

t_{M_i} t_{P_i}

243 t_i $t_i + \Delta t_i$ $t_{i+1}(t_i + \Delta T)$

244 **Fig. 2.** Re-planning scheme.

245 4.1 *PSO path planning algorithm*

246   Particle swarm optimization (PSO) is an evolutionary computation technique, which

247 was introduced in the mid 1990s (Kennedy and Eberhart, 1995). Every particle in the

248 swarm represents a potential path, the parameters of each particle corresponds to the

249 coordinates of control points generating the path. An overview of the PSO-based path

250 planning scheme is illustrated in Table 1.

251 **Table 1**

252 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 X_0^i and velocities 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.

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

254
$$V_{k+1}^i = w_k V_k^i + c_1 r_1 \left(P_k^i - X_k^i \right) + c_2 r_2 \left(P_k^g - X_k^i \right) \tag{17}$$

255   where, subscript k indicates an unit pseudo-time increment, V_k^i, X_k^i are the velocity

256   vector and position vector of particle i at iteration k, r_1, r_2 are two random numbers in

257   the range $[0,1]$. The parameters c_1, c_2 are problem-dependent, where c_1 indicates the

258   confidence level of the current particle in itself and c_2 describes the confidence level

259   in the swarm. The parameter w_k is an inertia weighting factor which controls the

260   global/local exploration abilities of the swarm, which is proposed as

261
$$w_k = w_{max} - \frac{w_{max} - w_{min}}{k_{max}} (k-1) \tag{18}$$

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

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

264   described as

265
$$\boldsymbol{X}_{k+1}^{i} = \boldsymbol{X}_{k}^{i} + \boldsymbol{V}_{k+1}^{i} \tag{19}$$

266   Further, the velocity vector of a particle with violated constraints should be brought

267   back to zero in the velocity update scheme defined as

268
$$\boldsymbol{V}_{k+1}^{i} = c_1 r_1 \left(\boldsymbol{P}_{k}^{i} - \boldsymbol{X}_{k}^{i} \right) + c_2 r_2 \left(\boldsymbol{P}_{k}^{g} - \boldsymbol{X}_{k}^{i} \right) \tag{20}$$

269   This is to ensure if a particle is infeasible, then there is a large probability that the last

270   search direction was not feasible.

271   4.2 *LPM path planning algorithm*

272   Legendre pseudospectral method (LPM) is an efficient numerical optimization

273   algorithm first proposed by Elnagar et al. (1995). In this paper, it is employed as a

274   discrete optimization scheme for the NP hard problem defined by Eqs. (14-16). The

275   main idea of LMP is to parameterize the flat outputs and their derivatives with Nth order

276   Lagrange polynomials L_N on $N+1$ Legendre-Gauss-Lobatto (LGL) points. Since the

277   LGL points lie only in the interval $\sigma \in [-1,1]$, a linear transformation

278     $\sigma = [2t - (t_f + t_0)]/(t_f - t_0) \in [-1,1]$ should be taken first to rewrite the optimization

279   problem. The flat output functions $\boldsymbol{Y}(\sigma)$ can thus be approximated on $N+1$ LGL

280   points as

281
$$\boldsymbol{Y}(\sigma) \approx \boldsymbol{Y}^{N}(\sigma) := \sum_{l=0}^{N} \boldsymbol{Y}(\sigma_l) \varphi_l(\sigma) = \sum_{l=0}^{N} \lambda_l \varphi_l(\sigma) \tag{21}$$

282   where, LGL points $\sigma_l, l = 0,1,...,N(\sigma_0 = -1, \ \sigma_N = 1)$ are the roots of $\dot{L}_N(\sigma)$. $\varphi_l(\sigma)$

283   is the Nth degree Lagrange interpolating basis function defined as

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

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

286 respectively as

$$\dot{\boldsymbol{Y}}(\sigma_k) \approx \dot{\boldsymbol{Y}}^N(\sigma_k) := \sum_{l=0}^{N} \boldsymbol{D}_{1,kl}\boldsymbol{Y}(\sigma_l) = \sum_{l=0}^{N} \lambda_l \boldsymbol{D}_{1,kl}$$

$$\boldsymbol{Y}^{(\gamma+1)}(\sigma_k) \approx \boldsymbol{Y}^{(\gamma+1)N}(\sigma_k) := \sum_{l=0}^{N} \boldsymbol{D}_{(\gamma+1),kl}\boldsymbol{Y}(\sigma_l) = \sum_{l=0}^{N} \lambda_l \boldsymbol{D}_{(\gamma+1),kl} \tag{23}$$

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

$$\boldsymbol{D}_1 := [\boldsymbol{D}_{1,kl}] := \begin{cases} \dfrac{L_N(\sigma_k)}{L_N(\sigma_l)} \cdot \dfrac{1}{\sigma_k-\sigma_l} & k \neq l \\[2mm] -\dfrac{N(N+1)}{4} & k = l = 0 \\[2mm] \dfrac{N(N+1)}{4} & k = l = N \\[2mm] 0 & otherwise. \end{cases} \tag{24}$$

290 The matrix $\boldsymbol{D}_{(\gamma+1),kl}$ is also $(N+1)\times(N+1)$, which can be easily obtained by

291 $\boldsymbol{D}_{(\gamma+1)} := [\boldsymbol{D}_{(\gamma+1),kl}] = \boldsymbol{D}_1^{(\gamma+1)}$.

292   Using LPM algorithm, the path planning problem shown in Eqs. (14-16) can be

293 further converted into a NLP as: determine a set of coefficients

294 $\lambda(\sigma) = [\lambda_0(\sigma), \lambda_1(\sigma), ..., \lambda_N(\sigma)]^T$, which minimizes the cost function shown in Eq.

295 (14), subject to all required constraints.

296   One of the main advantages of LPM is offering an exponential convergence rate for

297 the approximation of analytical functions under L^2 norm, while providing Eulerian-like

298 simplicity (Gong et al., 2006). Due to its high accuracy and competitive computational

299 efficiency, LPM is widely used in direct optimization methods. In general, LPM has a

300 larger radius of convergence than other numerical methods, and it may not require a

301   set of good initial guesses for convergence. However, educated initial guesses do

302   improve the convergence rate and robustness. In the following section, a hybrid PSO–

303   LMP algorithm is proposed to solve the collision-free path planning problem of the

304   AUV.

305   4.3 *Path re-planning with hybrid PSO-LPM algorithm*

306   The proposed PSO–LPM is a hybrid optimization algorithm combining PSO

307   algorithm with LPM algorithm. The main idea of the algorithm can be divided into two

308   phases: in phase 1, PSO algorithm serves as a start engine to generate a candidate path;

309   in phase 2, the best solution of phase 1 is loaded as an initialization for LPM-based path

310   planner, and then run the LPM-based path planner repeatedly on-line until the AUV

311   reaches the final destination. Finally, the obtained optimal solutions in flat outputs space

312   should be mapped back to the state and control input spaces. The details of PSO-LPM

313   algorithm can be summarized as shown in Table 2:

314   **Table 2**

315   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 $\lambda = [\lambda_0, \lambda_1, ..., \lambda_N]^{\mathrm{T}}$ 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 $\lambda = [\lambda_0, \lambda_1, ..., \lambda_N]^T$ 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 $\lambda^* = [\lambda_0^*, \lambda_1^*, ..., \lambda_N^*]^T$, and obtain the corresponding flat output variables $\bar{Y}^*(\sigma) = [Y^*(\sigma), \dot{Y}^*(\sigma), \ddot{Y}^*(\sigma), ..., Y^{*(\gamma+1)}(\sigma)]^T$ 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 τ^* 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

319 obstacles and multi moving obstacles, respectively. The algorithm has been coded in

320 MATLAB R2012a and simulations are run on the PC with 2.1 GHz CPU/2GB RAM.

321 The NLP solver for re-planning process used here is KNITRO (Byrd et al., 2006).

322   In the cases studies, the simulation parameters for PSO algorithm are selected as: the

323 population size s=30; the maximum number of iterations $K_{max} = 1000$; $c_1 = c_2 = 2$ and

324 the inertia weighting factor w_k scales linearly between 0.4 and 0.9. The number of LGL

325 points is 11 with N=10; the re-planning time horizon is given as $\Delta T = 1s$, and the

326 weighting values for hybrid objective function are set as $\varepsilon_1 = \varepsilon_2 = 0.5$.

327 5.1 *Case1: Static obstacles avoidance*

328   The scenario in this case study is that an AUV is travelling in 3-D workspace, from

329 the start point $[0,0,0,0,5,5,2,-\pi/4]^{\mathrm{T}}$ to the destination point $[0,0,0,0,45,45,22,\pi/4]^{\mathrm{T}}$.

330 Six static obstacles are considered for evaluation of the re-planning algorithm, which

331 are assumed to be spherical with the same radius of 3m.

332   Fig. 3 displays collision-free trajectories of the AUV obtained by only PSO algorithm

333 with a random initialization and the time taken to arrive at final point is $t_f = 183s$. Fig.

334 4 shows an optimal collision-free path obtained by hybrid PSO-LPM global planning

335 algorithm with the final arrival time as $t_f = 111.7s$. This shows that the hybrid PSO-

336 LPM algorithm is able to find a better optimal trajectory compared to PSO algorithm

337 alone. The PSO algorithm here is only used to find a set of initial guesses for LPM-

338 based algorithm rather than a global optimum.

339   Fig. 5 shows an optimal path of AUV based on the re-planning scheme, and the total

340 travelling time is $t_f = 130.27s$. It can be found that although the globally planned

341   trajectory is slightly different from the re-planned one; both of them can guide the AUV

342   to the final destination successfully without collision with any obstacles. In this case all

343   the positions of static obstacles are assumed to be exactly known as *a priori*, thus the

344   global PSO-LPM algorithm can be utilized for the purpose of collision avoidance with

345   sufficiently enough LGL points, in order to avoid the possible collisions between any

346   two LGL points as shown in Fig. 4(a). It should be noted as the number of LGL points

347   increases, the complexity and time taken for the optimization will increase, resulting in

348   a more computational burden. The proposed algorithm deals with the obstacles by local

349   re-planning with optimized LGL points, which not only reduces the time consumption,

350   but also reduces the risk of collision as shown in Figs. 4(b) and 5(b) respectively.

351   However, the re-planning scheme has to evaluate the collision risk and refine the path

352   in each local planning process to keep the AUV a safe distance from all the obstacles.

353   It can be seen in Figs. 4(b) and 5(b), the value of objective function for local re-planning

354   is thus almost twenty seconds longer than that of global planning.

355   **a**

356

357

358

359 **b**

360
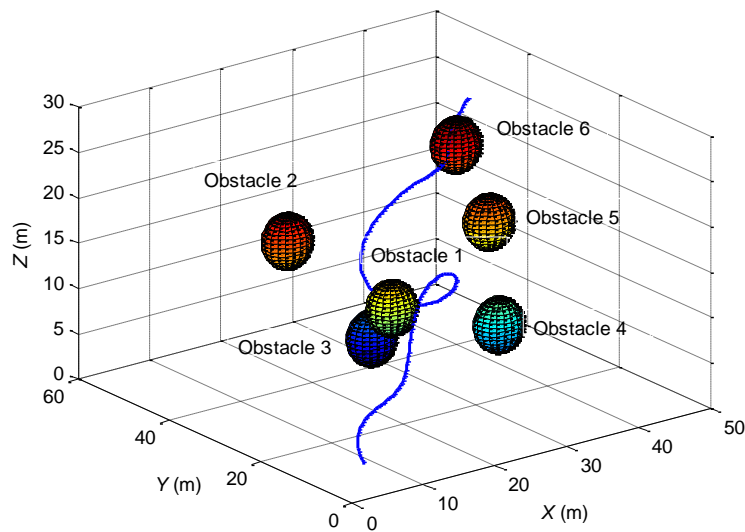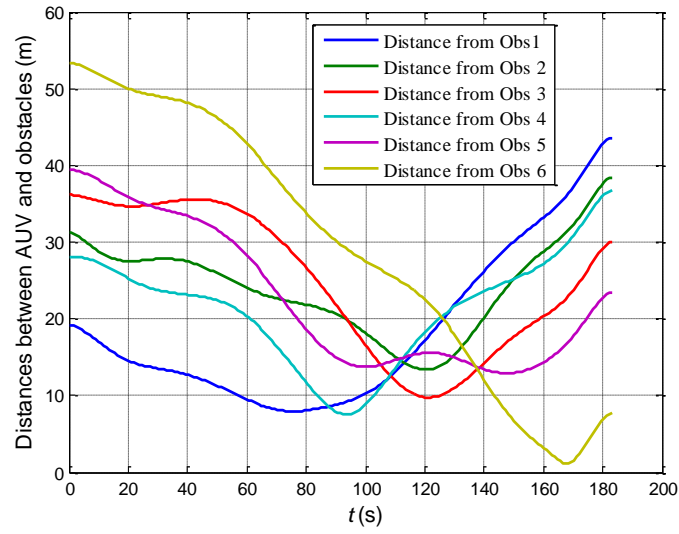
361   **Fig. 3.** Planned trajectories of AUV by PSO algorithm in Case 1. (a) Trajectory of AUV in 3-

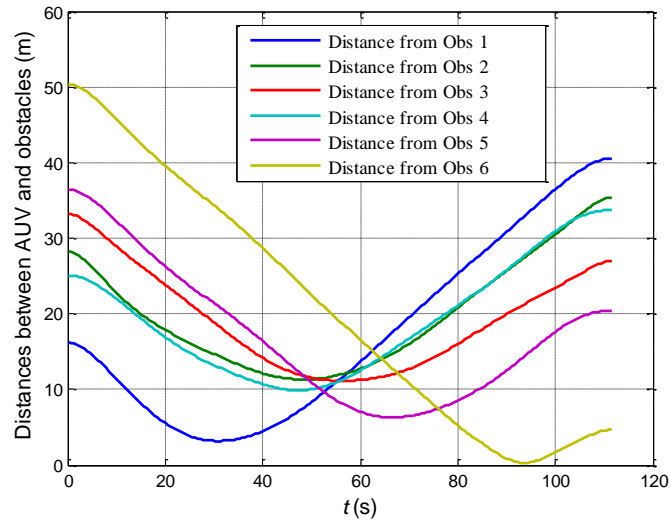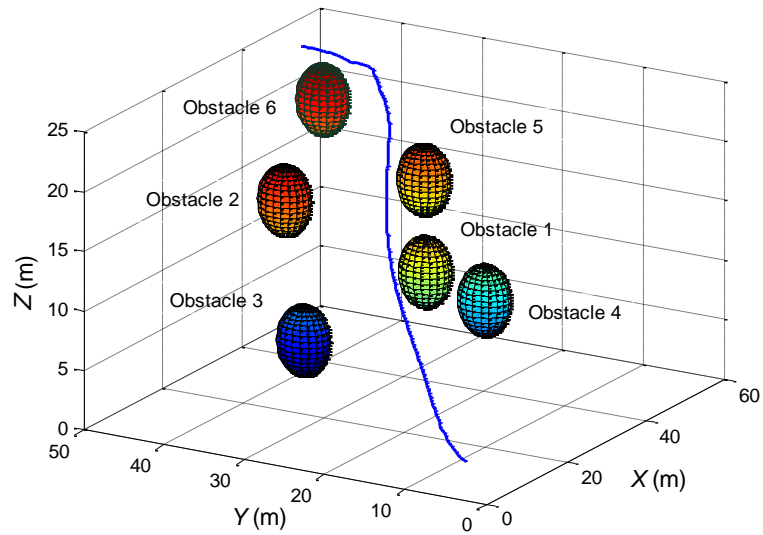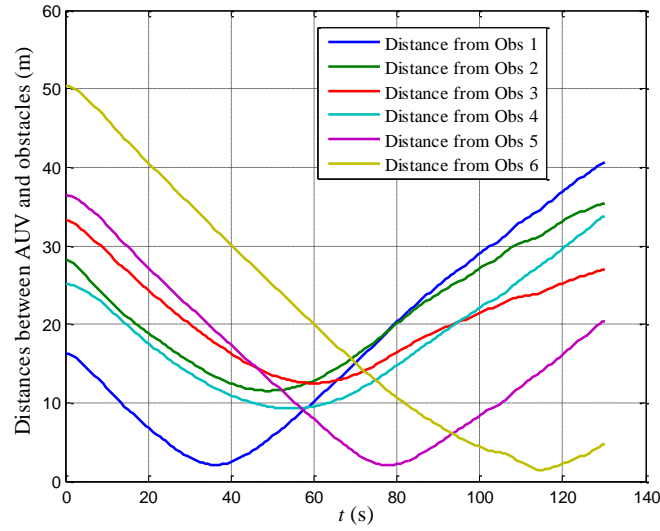362   D workspace. (b) Distances between planned trajectory of AUV and each obstacle.

363   **a**

364

365

366

367

368

369

370   **b**

371

372   **Fig. 4.** Globally planned trajectories of AUV by hybrid PSO-LPM algorithm in Case 1. (a)

373   Trajectory of AUV in 3-D workspace. (b) Distances between globally planned trajectory of

374   AUV and each obstacle.

375   **a**

376
377

378

379

380

381

b

383

384   Fig. 5 Re-planned trajectories of AUV in Case 1. (a) Trajectory of AUV in 3-D workspace. (b)

385   Distances between re-planned trajectory of AUV and each obstacle.

386   Fig. 6(a) shows the time taken for each planning in the whole re-planning process,

387   where the hollow dot represents a success in finding an optimal solution while the blue

388   dot represents a failure. It can be found the computational time for each planning except

389   the first one is shorter than the given re-planning time horizon $\Delta T = 1$s , which ensures

390   that the re-planning scheme can be used on-line. Fig. 6(b) displays the values of

391   objective function obtained by each re-planning process, which gradually decrease as

392   the AUV moves closer to the target. However, the curve is not smooth enough, i.e., it

393   drops considerably at the time t=59s and t=101s. As shown in Fig. 6(a), the 59th re-

394   planning process (marked with a circle) is successful to obtain an optimal solution, but

395   the time consumption is excessive, which causes a sudden change in the value of

396   objective function. Herein, the updated path obtained by previous successful re-

397   planning is applied to the AUV, until the next successful re-planning is achieved. In Fig.

398   6(a), the 100th re-planning process (marked with a diamond) fails to find an optimum,

399   while the 101th re-planning succeeds, which also makes the fitness values change

400   considerably. As shown in Figs. 6(a) and 6(b), it is found that the sudden changes in the

401   values of objective function correspond to both excessive time consumption for re-

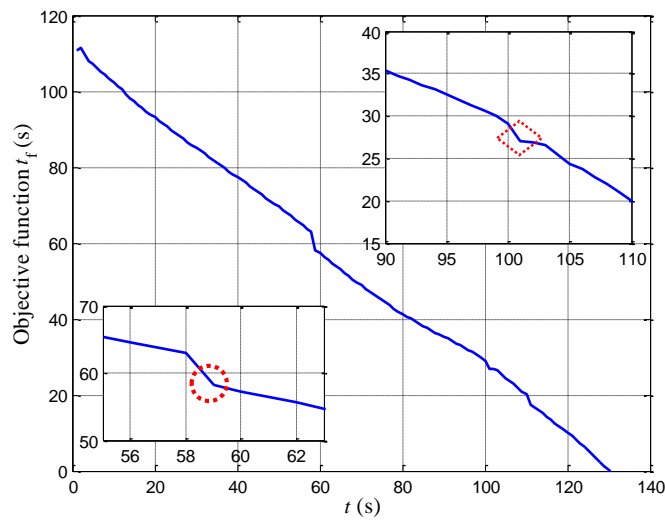402   planning and the failure in finding an optimal re-planning path.

403   **a**

404

405   **b**

406
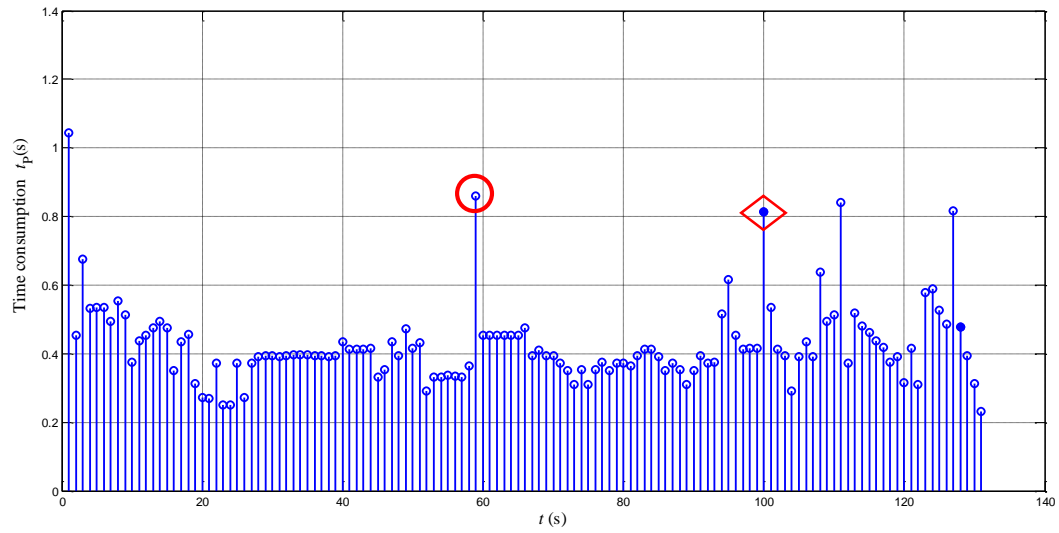
407   **Fig. 6.** Relations between computational time and objective function. (a) Computational
408   time for each planning. (b) Values of objective function.

409   5.2 *Case 2: Dynamic obstacles avoidance*

410   In previous case, it is assumed that the positions of obstacles are precisely known,

411   and the planned path can be executed perfectly. However, in realistic ocean fields, the

412   locations of obstacles are not usually known precisely. In this section, the re-planning

413   problem will tackle three moving obstacles with varying levels of position uncertainty.

414   The model of dynamic obstacles is assumed to be a linear and discrete-time system

415   as defined in Zeng et. al (2015):

416 $$O_i = H_O O_{i-1} + Z_O X_{i-1} + L_O du \qquad (25)$$

417   where, $O_i = [O_{Pi}, O_{Vi}, O_{Ui}]^{\mathrm{T}}$ represents the state of obstacles at time t_i (here, assuming

418    $t_{M_i} = 0$) measured from the on-board sonar sensors, and O_{Pi}, O_{Vi}, O_{Ui} denote position,

419   velocity and uncertainty of the obstacle at time t_i, respectively; $X_{i-1} \sim N(0, 0.005^2)$ is

420   Gaussian disturbance acting on velocity, which is independent from the disturbances

421   caused by $X_{0 \sim i-2}$; du is the rate of uncertainty, which is set as $du = 0.005 \mathrm{m/s}$. The

422   parameter matrices are written as:

423 $$\boldsymbol{H}_O = \begin{bmatrix} 1 & \Delta T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{Z}_O = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \boldsymbol{L}_O = \begin{bmatrix} 0 \\ 0 \\ \Delta T \end{bmatrix} \qquad (26)$$

424   Assuming the initial velocities for all the three moving obstacles are 0m/s, with the

425   initial locations distributed randomly. Fig. 7 displays the optimal trajectory obtained in

426   the first global planning, which can be regarded as the global planning problem in Case

427   1 with only three static obstacles. Obviously, the global planning can easily find a

428   collision-free path as shown in Fig. 7(b) with the objective function in total time

429    $t_f = 125.63 \mathrm{s}$. In Fig. 8(a), the red line displays the re-planned optimal trajectory of

430   AUV, while blue lines show the paths of the centers of mass of three dynamic obstacles,

431 respectively. Further, the three spheres mark the location of each obstacle with shortest

432 distance to the AUV in the whole re-planning process. Similarly, as illustrated in Fig.

433 8(b), the re-planning algorithm also succeeds in finding a time-optimal collision-free

434 path in 3-D workspace even with uncertain moving obstacles. It can be observed the

435 objective function obtained by re-planning is $t_f = 163.79$s , since the AUV requires

436 more time to overcome the possible collisions caused by dynamic obstacles as well as

437 their uncertainty in both positions and velocities.

438   **a**

439

440

441

442

443

444

445

446   **b**

447
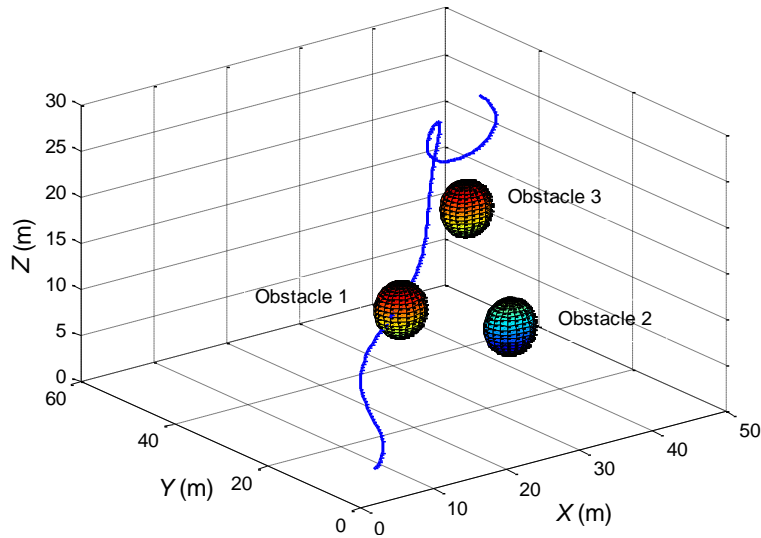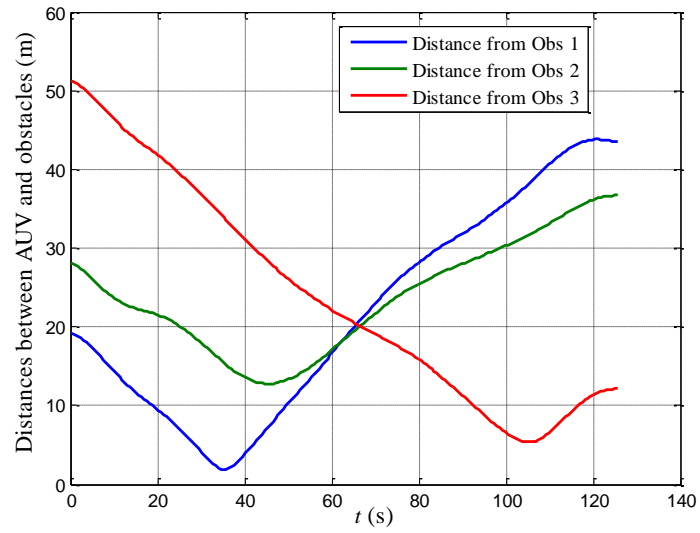
448   **Fig. 7.** Globally planned trajectories of AUV by hybrid PSO-LPM algorithm in Case 2. (a)

449   Trajectory of AUV in 3-D workspace. (b) Distances between globally planned trajectory of

450   AUV and each obstacle.

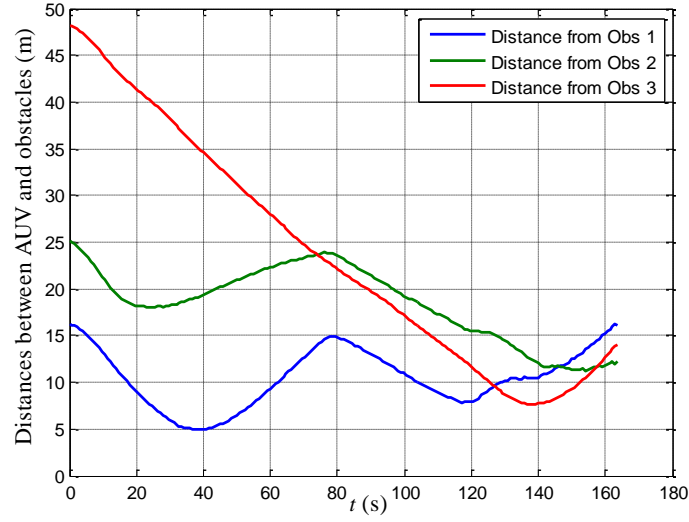451   **a**

452

453

454

455

456

457   **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 ith re-planning as an initialization to decrease the total time
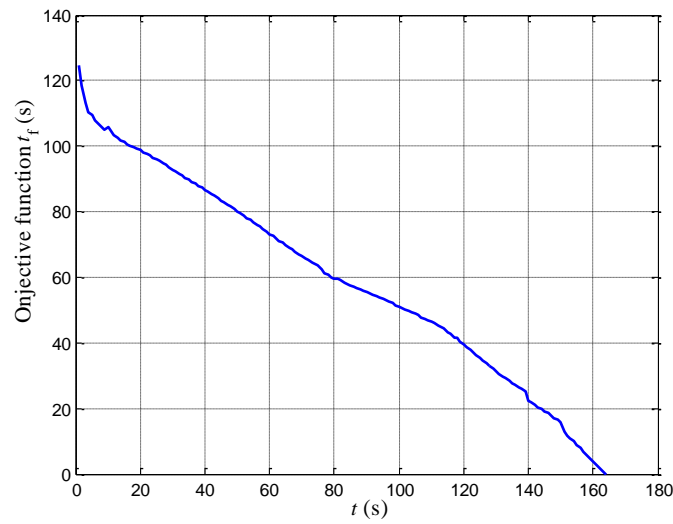
consumption.

a

474

475   **b**

476
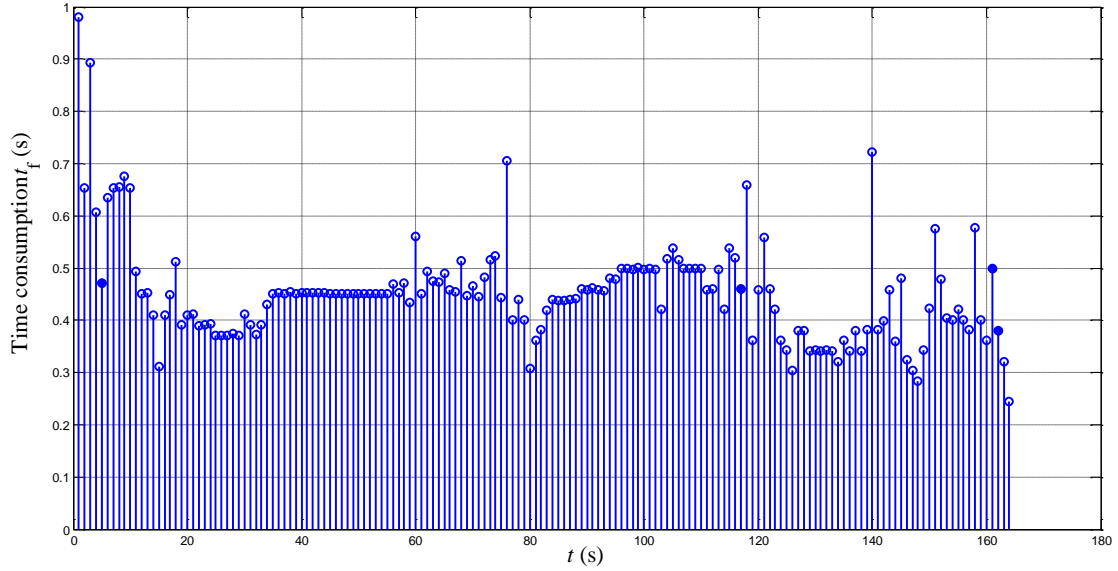
477   **Fig. 9.** Relations between computational time and objective function. (a) Computational

478   time for each planning. (b) Values of objective function.

479   5.3 *Robustness assessment*

480   In this subsection, Monte Carlo simulations with random initial values will be

481   carried out to demonstrate the robustness of the proposed re-planning algorithm. First,

482   simulations are performed on a 100-run basis for Case 1 discussed in Section 5.1, and

483   the results are illustrated in Fig. 10. Fig. 10(a) displays the shortest distances between

484 AUV and the obstacles in the whole re-planning process, where the positive values

485 represent safe condition, while the negative values mean collision. It is obvious in Figs.

486 10(a-b), although the first global planning is superior to the re-planning scheme in

487 objective functions, it fails to avoid collision for almost half of the 100-run Monte Carlo

488 simulations. Fig. 10(c) plots the terminal error of AUV, which is defined as the distance

489 between the desired final position and the actual planned destination of AUV. It is

490 obvious that the terminal errors here are acceptable in realistic applications, and an

491 improvement could be obtained by increasing the number of LGL points.

492 Fig. 11 shows the 100-run Monte Carlo simulation results also for Case 1 without

493 considering the flatness property of AUV. It can be seen that the average time

494 consumption for each re-planning is longer than the given re-planning time horizon ΔT,

495 which causes a majority of plannings failing in the whole re-planning process, the re-

496 planning thus cannot be executed on-line as expected. An obvious phenomenon is that

497 the values of objective function obtained without considering flatness property are

498 much longer than those displayed in Fig. 10(b). On the other hand, this set of Monte

499 Carlo simulation results illustrate the flatness property of AUV is effective to reduce

500 the time usage of planning, which sometimes is a necessary condition for the

501 application of re-planning scheme on-line.

502 Fig. 12 runs 100 Monte Carlo simulations with random initial values to assess the

503 robustness of proposed algorithm for Case 2. The results show that the PSO-LPM

504 algorithm is not only effective for the ocean environments with static obstacles but also

505 successful in dealing with moving obstacles with varying levels of positional
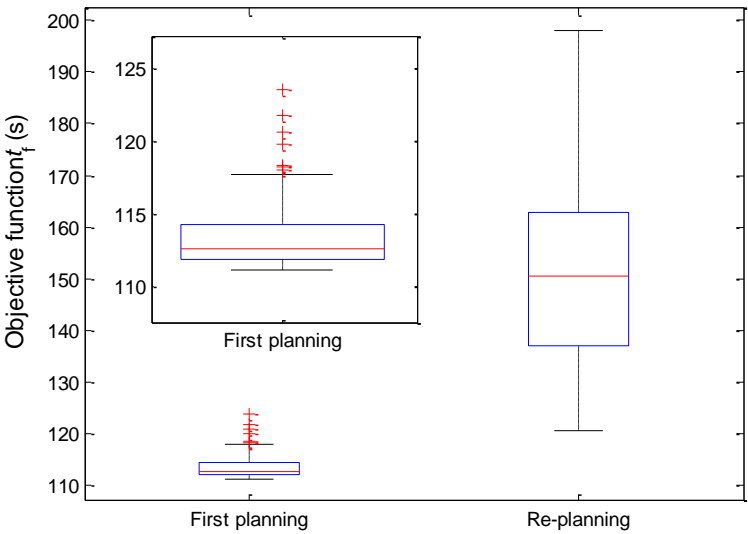
506   uncertainty.

507   **a**
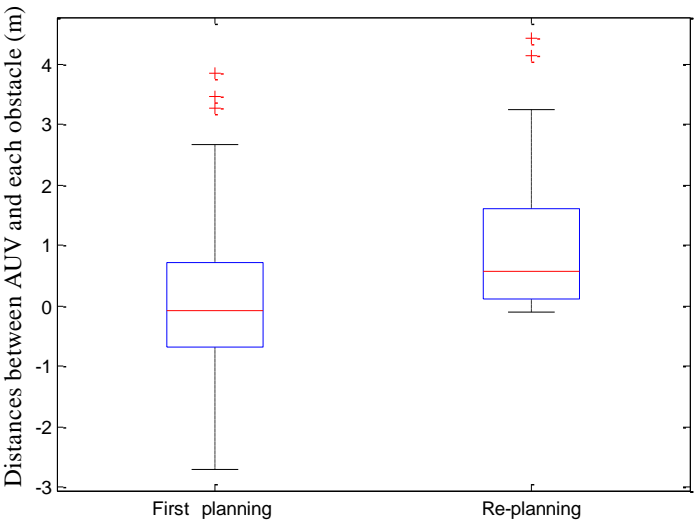
508

509   **b**

510

511

512

513
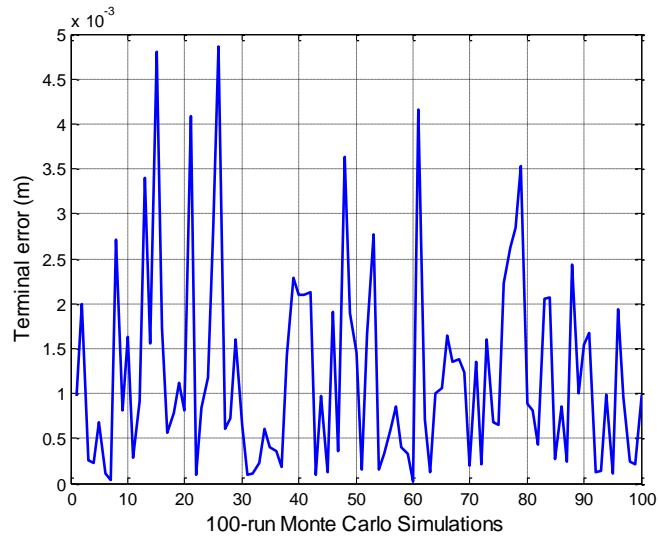
514

515

516

517   **c**

518

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.
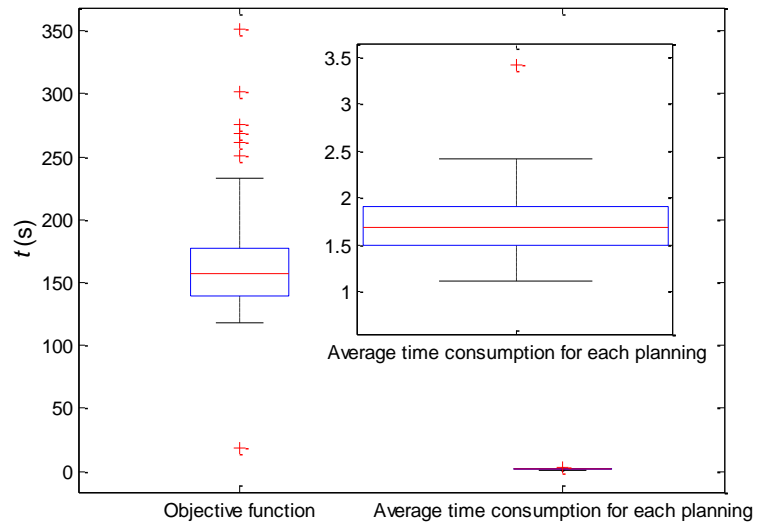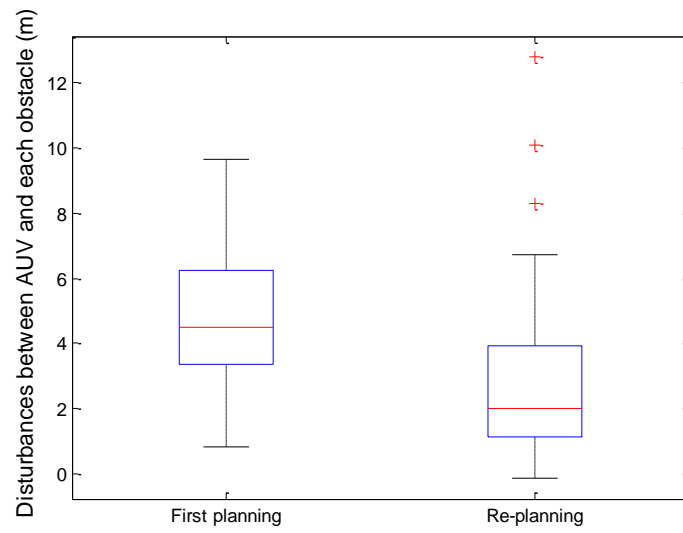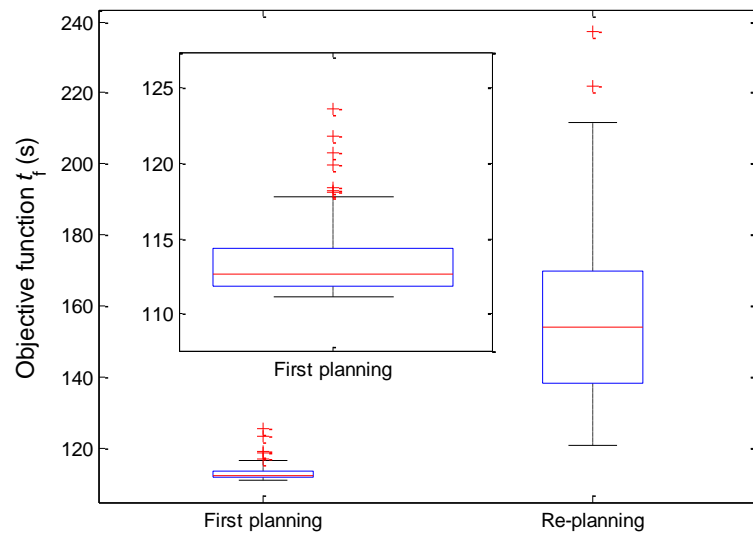

521

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

523
524
525
526
527
528
529
530
531
532

533   **a**

534

535   **b**

536

537

538

539

540

541

542

543

544
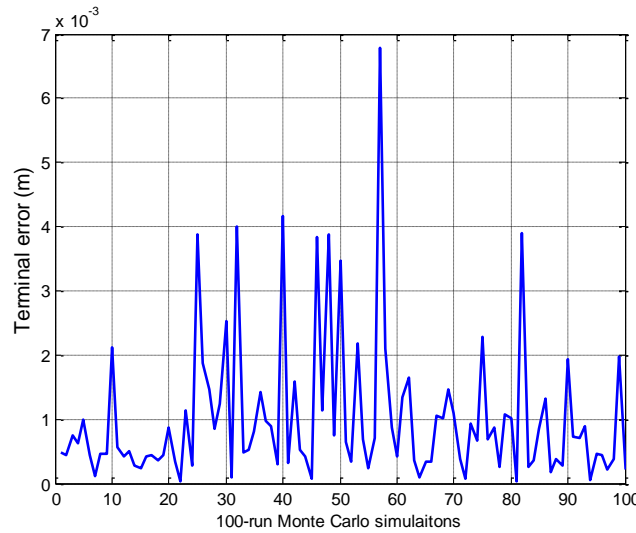
545

546   **c**

547

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.

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.

586   Carroll, K. P., McClaran, S. R., Nelson, E. L., Barnett, D. M., Friesen D. K., William G., 1992. AUV

587   path planning: an A* approach to path planning with consideation of variable vehicle speeds

588   and multiple, overlapping, time-dependent exclusion zones. Proceedings of the 1992

589   Symposium on Autonomous Underwater Vehicle Technology, 1992.

590   Daily, R., Bevly, D. M., 2008. Harmonic potential field path planning for high speed vehicles. In:

591   Proceedings of American Control Conference. Washington, USA. pp. 4609-4614.

592   Elnagar, G., Kazemi, M. A., Razzaghi, M., 1995. The pseudospectral Legendre method for

593   discretizing optimal control problems. IEEE Transactions on Automatic Control, 40:1793-1796.

594   Ferguson, D., Stentz, A., 2006. Using interpolation to improve path planning the field D* algorithm.

595   Journal of Field Robot. 23(2), 79-101.

596   Fliess, M., Lévine, J., Martin, P., Rouchon, P., 1995. Flatness and defect of non-linear systems:

597   introductory theory and examples. International journal of control. 61(6), 1327-1361.

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

599   Gong, Q., Kang, W., Ross, I. M., 2006. A pseudospectral method for the optimal control of

600   constrained feedback linearizable systems. IEEE Transaction of Automatic Control. 51(7),

601   1115-1129.

602   Iwakami, H., Ura, T., Asakawa, K., Fujii, T., Nose, Y., Kojima, J., Shirasaki, Y., Asai, T., Uchida,

603   S., Higashi, N., Fukuchi, T., 2002. Approaching whales by autonomous underwater vehicle.

604   Marine Technology Society Journal. 36(1), 80-85.

605   Incze, M. L., 2011. Light weight autonomous underwater vehicles (AUVs) performing coastal

606   survey operations in REP10A. Ocean Dynamics. 61(11), 1955-1965.

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

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

609   Khatib, O., 1986. Real-time obstacle avoidance for manipulators and mobile robots. The

610   international journal of robotics research. 5(1), 90-98.

611   Kondo, H., Ura, T., 2004. Navigation of an AUV for investigation of underwater structures. Control

612   engineering practice. 12(12), 1551-1559.

613   Kumar, R. P., Dasgupta, A., Kumar, C. S., 2005. Real-time optimal motion planning for autonomous

614   underwater vehicles. Ocean engineering. 32(11), 1431-1447.

615   Lévine J. On Necessary and Sufficient Conditions for Differential Flatness[J]. Applicable Algebra

616   in Engineering Communication & Computing, 2011, 22(1):47-90.

617   Liang, J. H., Lee, C. H., 2015. Efficient collision-free path-planning of multiple mobile robots

618   system using efficient artificial bee colony algorithm. Advances in Engineering Software. 79,

619   47-56.

620   Likhachev, M., Ferguson, D. I., Gordon, G. J., Stentz, A., Thrun, S., 2005. Anytime Dynamic A*:

621   An Anytime, Replanning Algorithm. In: Proceedings of the 15th International Conference on

622   Automated Planning and Scheduling. California, USA. pp. 262-271.

623   Lin, C. F., Tseng, C. Y., 2006. Development of a cost effective mini autonomous underwater

624   vehicle. Journal of Marine Science and Technology. 14(2), 119-126.

625   Pereira, A. A., Binney, J., Hollinger, G. A., Sukhatme, G. S., 2013. Risk-aware path planning for

626   autonomous underwater vehicles using predictive ocean models. Journal of Field Robot.

627   30(5),741–762.

628   Pereira, A. A., Binney, J., Jones, B. H., Ragan, M., Sukhatme, G. S., 2011. Toward risk aware

629   mission planning for autonomous underwater vehicles. IEEE International Conference on

630   Intelligent Robots and Systems.

631   Spangelo, I., Egeland, O., 1994. Trajectory planning and collision avoidance for underwater

632   vehicles using optimal control. IEEE Journal of Oceanic Engineering. 19(4), 502-511.

633   Sullivan, J., Waydo, S., Campbell, M., 2003. Using stream functions for complex behavior and path

634   generation. In: Proceedings of AIAA Guidance, Navigation, and Control Conference and

635   Exhibit. Austin, USA.

636   Wang, B., Wan, L., Xu, Y. R., Qin, Z. B., 2009. Modeling and simulation of a mini AUV in spatial

637   motion. Journal of Marine Science and Application. 8(1), 7-12.

638   Yuh, J., 2000. Design and control of autonomous underwater robots: A survey. Autonomous

639   Robots. 8(1), 7-24.

640   Zeng, Z., Sammut, K., Lammas, A., He, F., Tang, Y., 2015. Efficient path re-planning for AUVs

641   operating in spatiotemporal currents. Journal of Intelligent & Robotic Systems. 79, 135-153.

642   Li, Z., Yang, C., Ding, N., Bogdan, S., Ge, T., 2012. Robust adaptive motion control for

643   underwater remotely operated vehicles with velocity constraints. International Journal of

644   Control, Automation and Systems, 10 (2), 421-429.