Motion Planning for an Autonomous Underwater Vehicle

Tahereh Taleshian* and Sara Minagar

Dept. of Control Eng., Babol of University of Technology, Babol, Iran

*Corresponding Author's Email: t.taleshian@yahoo.com

Abstract

This paper deals with motion planning of a REMUS autonomous underwater vehicle (AUV). It is desired to move the vehicle from the surface of the water to a depth of about two meters and then to maintain it in this depth. Therefore, real-time path trajectory is first designed based on the desired goals and then an optimal PD controller following with a proportional gain are designed to force the AUV to track the desired depth trajectory with minimum tracking error. The nonlinear model of the REMUS AUV is utilized and improved particle swarm optimization algorithm in combination with SQP method (IPSO-SQP) is used to determine the optimal value of the PD and the proportional controller parameters. The cost function is hence the tracking error and the goal is to track the desired path with minimum error.

Keywords: Motion planning; AUV; Optimal control; IPSO-SQP algorithm

I. Introduction

Autonomous underwater vehicles (AUVs) are currently being utilized for scientific, commercial and military underwater applications[1]. These vehicles are playing a vital role in underwater exploration and allowing humans to explore great depths[2]. AUVs need autonomous guidance and control systems to perform underwater tasks[3]. Modeling, system identification and control of these vehicles are still major active areas of research and development[4]. AUVs are untethered, fully automated submersible platforms capable of performing underwater tasks and missions with their onboard sensor, navigation and payload equipment[5]. As there is no external communication for guidance, and the available power is limited (since the power supply is carried onboard), motion planning is extremely critical for successful operation of AUVs. There are several papers which address the problem of path planning. In [6]a new motion coordination algorithm for an AUV-manipulator system (UVMS) is proposed in which the algorithm generates the desired trajectories for both the vehicle and the manipulator in such a way that the total hydrodynamic drag on the system is minimized. In[7]an evolutionary path planning is performed for autonomous underwater vehicles in a variable ocean. The goal is to find a safe path that takes the vehicle from its starting location to a specified destination with minimum energy. In[8], path planning of autonomous underwater vehicles, in current fields with complex spatial variability, is studied. The work explored the benefits, in terms of energy cost, of path planning in marine environments showing certain spatial variability. In[9] real-time motion planning of an stable autonomous underwater is performed. In the aforementioned paper, the trajectory is designed such that the vehicle is steered from one point to another point with minimum energy. Four thrusters located symmetrically on the vehicle have been used to move AUV in different directions. In[10]planar trajectory planning and tracking control design for under-actuated AUVs is investigated. The aforementioned paper addressed the combined problem of trajectory planning and tracking control for under-actuated AUVs on the horizontal plane. Back-stepping techniques used to force the tracking error to an arbitrarily small neighborhood of zero. In[11]the problems of automatically planning autonomous AUV paths which best exploit complex current data, from computational estuarine model forecasts, while also avoiding obstacles is addressed. In [12] a dynamic neural network is proposed to solve the AUV path planning problem. In[13] a team of AUVs
are controlled to reach all appointed target locations for only one time on the premise of workload balance and energy sufficiency while guaranteeing the least total and individual consumption in the presence of the variable ocean current.

In the present paper, motion planning of a REMUS AUV is studied. Here, motion planning is divided into two steps. The first step is trajectory planning problem in which the goal is to design a path that satisfies the desired conditions and the second step is the tracking control design problem in which an optimal PID controller is designed to force the vehicle to track the desired path with minimum error. Nonlinear dynamic equations of AUV are utilized to better demonstrate the behavior of an AUV. Also, the motion is assumed to be in the depth plane. Moreover, the thruster force is constant and the fin angle is the control input.

The rest of the paper is organized as follows:
In section 2, REMUS AUV equations of motion are presented. IPSO-SQP algorithm is briefly introduced in section 3. Section 4 deals with motion planning of AUV. In this regard, the desired path trajectory is first designed based on the considered assumptions, thereafter an optimal controller is designed for tracking purposes. Finally conclusion ends the paper in section 6.

II. REMUS AUV’s Equations of Motion

In this section, the total equations governing the vehicle motion in the depth plane are presented. The equations are written in the body-fixed coordinate frame. As a result, the six degree of freedom equations of motion are as follows:

Force equation along with the x axis:

\[(m - X_u)\dot{u} + m z_c \dot{q} = -(W - B)\sin \theta + X_{z_c} \mu \theta \]

\[+ (X_{w_q} - m)\dot{w}q + (X_{w_q} + m x_c)q^2 + \text{Thrust} \quad (1)\]

Similarly, the force equation along with the z axis:

\[(m - Z_u)\dot{v} - (m x_c + Z_x)\dot{q} = +(W - B)\cos \theta \]

\[+ Z_{v_{\theta}} \dot{\theta} + Z_{\theta q} q + (Z_{w_q} + m)\dot{u}q \]

\[+ Z_{w_{\theta}} + m z_c (p^2 + q^2) + Z_{uv} \dot{u} \quad \delta_s \quad (2)\]

In parallel, the momentum equation along with the y axis:

\[m z_c \ddot{u} -(m x_c + M_y)\dot{w} + (I_{yy} - M_y)\dot{\theta} = -(z_c W - z_y B)\sin \theta \]

\[-(x_c W - x_y B)\cos \theta + M_{\theta q} \dot{\theta} + M_{\theta q} \theta \]

\[+ (M_{w_q} - m x_c)\mu \quad \delta_s \quad (3)\]

There are also three other equations which are called kinematic equations of motion. These equations are as follows:

\[\dot{\theta} = q \quad (4)\]

\[\dot{x} = u \cos \theta + w \sin \theta \quad (5)\]

\[\dot{z} = -u \sin \theta + w \cos \theta \quad (6)\]

Values of the parameters are presented in Table I- Appendix A [ref]. It is worthwhile to note that in ((1) to ((6)) the fin angle ( \delta_s ) is a bounded control input as:
State variables “u” and “w” denote the linear speed of the vehicle expressed in the x and z axis respectively. “q” indicates the angular velocity of the vehicle with respect to the y axis and “x” and “z” are the AUV translational motion along with x and z axis respectively. Finally the “Thrust” is the propulsion force which is constant in this vehicle.

III. IPSO-SQP Algorithm

A. IPSO Algorithm

Particle Swarm Optimization (PSO) is a form of evolutionary algorithm inspired initially by flocking birds and is stochastic in nature[14]. In this algorithm, there are a set number of particles that fly through the hyperspace of the problem. A minimization (or maximization) of the problem topology is found both by a particle remembering its own past best position and the entire flock’s best overall position. The PSO algorithm is based on the concept that complex behavior follows from a few simple rules[15]. The Basic PSO algorithm consists of the velocity and position equations as follows[16]:

\[
\begin{align*}
    v_{i}^{k+1} &= \omega v_{i}^{k} + c_{1} r_{1} (p_{\text{best}}^{k} - x_{i}^{k}) + c_{2} r_{2} (g_{\text{best}}^{k} - x_{i}^{k}) \\
    x_{i}^{k+1} &= x_{i}^{k} + v_{i}^{k+1}
\end{align*}
\]

(8) (9)

Where i is the particle index, k is discrete time index, v is the velocity of ith particle, x is the position of ith particle, pbest is the best position found by ith particle (personal best), gbest is the best position found by swarm (global best, best of personal bests) and c1 and c2 are random numbers on the interval [0,1] applied to ith particle[15].

The most used form of PSO is to include a constant inertia term and acceleration; however, the success of PSO during a search is highly dependent on a good balance between exploration and exploitation[16]. Exploration is to search the entire search space by ensuring the redirection of the search towards new regions, while exploitation favors a quick convergence towards the optimum[16]. To do a good balance between exploration and exploitation, one can use an appropriate adaptation mechanism for inertia weight factor. As a result, IPSO algorithm is used in this paper in which the inertia weight is determined according to the following equation:

\[
w_{i}^{k} = \frac{1}{1 + \exp(-\alpha F(p_{\text{best}}^{k}))}
\]

(10)

This adaptation rule enables the algorithm to evaluate the inertia weight for every particle in kth iteration. According to (10), during the search process of IPSO algorithm, whenever the fitness of a particle is far away from the global optimal, the value of the inertia weight will be large resulting in strong global search abilities[16]. Meanwhile, when the fitness of a particle is achieved near the real global optimal, the inertia weight will be set small, depending on the nearness of its best fitness to the optimal value, to facilitate a finer local explorations and hence accelerate convergence[16].

B. SQP Algorithm

The sequential quadratic programming (SQP) is one of the most recently developed and perhaps one of the best methods of optimization[17]. The method has a theoretical basis that is related to the solution of a set of nonlinear equations using Newton’s method, and the derivation of simultaneous nonlinear equations using Kuhn–Tucker conditions to the Lagrangian of the constrained optimization problem. The solution procedure of the sequential quadratic programming approach is as follows[17]:
minimize: \( J(x) \)
subjected to: \( \psi_i(x) \leq 0, \quad i = 1, 2, \ldots, m \) \hspace{1cm} (11)

Where \( J(x) \) is the cost function and \( \psi_i(x) \) indicates the constraint. The Lagrangian function \( L(x, \lambda) \) corresponding to problem ((11) is given in terms of the Lagrangian multiplier \( \lambda \) and the cost function together with the constraint which is as follows:

\[
L(x, \lambda) = J(x) + \sum_{i=1}^{m} \lambda \psi_i(x)
\]

In fact the SQP consists of three main parts:

1. Update the Hessian of the Lagrangian function based on the following equation:

\[
H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k^T H_k}{s_k^T H_k s_k}
\]

\[
H_0 = I
\]

\[
s_k = X_{k+1} - X_k
\]

\[
q_k = \nabla f(x_{k+1}) + \sum_{i=1}^{n} \lambda_i \nabla g_i(x_{k+1})
\]

\[
- \left( \nabla f(x_k) + \sum_{i=1}^{n} \lambda_i \nabla g_i(x_k) \right)
\]

Solve the quadratic programming sub-problem according to:

\[
\min \quad \frac{1}{2} d_k^T H_k d_k + \nabla f(x_k)^T d_k
\]

\[
\nabla \psi_i(x_k)^T d_k + \psi_i(x_k) = 0 \quad i = 1, \ldots, m_c
\]

\[
\nabla \psi_i(x_k)^T d_k + \psi_i(x_k) \geq 0 \quad i = m_c, \ldots, m
\]

Do a linear search to find a solution for the next iteration:

\[
X_{k+1} = X_k + \alpha d_k
\]

The algorithm is repeated until the stopping criterion (maximum iteration or convergence criterion) is satisfied. The step length parameter \( \alpha \) can be determined by a linear search procedure.

C. IPSO-SQP algorithm

To achieve faster convergence speed around the global optimum and also higher convergence accuracy, the proposed IPSO is combined with successive quadratic programming (SQP) algorithm[16]. Although SQP has not enough ability to escape local optimum, it can achieve faster convergence speed around global optimum and the accuracy can be higher[18]. The main idea in the IPSO-SQP method is that at the beginning stage of the searching process for the optimum, IPSO algorithm is employed to find a near optimum solution. When the change in fitness value is smaller than a predefined value, namely the particles being close to the global optimum, the best solution found by IPSO algorithm will be taken as the initial starting point for the SQP and the searching
process is switched to SQP searching to accelerate the search process and find an accurate solution. In this way, this hybrid algorithm may find an optimum solution more quickly and accurately[16]. The procedure of IPSO-SQP algorithm can be briefly defined as follows:

**Step1:** Initiate the number of variables. Determine the maximum iteration and the switching criteria.

**Step2:** Initialize the position and velocity of each particle.

**Step3:** Evaluate the fitness function for each particle.

**Step4:** If the maximum iteration criterion is met, go to step8, otherwise; go to step5.

**Step5:** Evaluate and store the global and personal best. If the change between the current global best and its previous one is smaller than a pre-defined value, go to step8, otherwise continue.

**Step6:** Evaluate the inertia weight for each particle according to ((10)).

**Step7:** Update the particles velocity and position according to ((8) and ((9) respectively and go to step3.

**Step8:** Use the global best achieved by IPSO algorithm as the initial starting point of the SQP algorithm. Switch to the SQP algorithm to search around the found global best.

### IV. Motion Planning

In this section, motion planning of AUV is investigated. First the depth trajectory is designed in the trajectory planning process based on the desired goals. Then, in the tracking control design process, an optimal PID controller is designed for the nonlinear dynamic of REMUS AUV to track the desired depth trajectory with minimum error.

#### A. Trajectory Generation

The depth trajectory is approximated by a number of polynomials. The number of coefficients in each polynomial is chosen in accordance with the number of boundary conditions to be satisfied. The goal is to steer the vehicle from the surface of the water to a depth of two meters, and then maintain the vehicle in this depth. Therefore, the desired depth trajectory is composed of two parts. The first part is the path from the depth of zero (surface of the water) to the depth of two meters and the second is the track of two meters depth. The polynomial for the first part is considered as follows:

\[
z_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5
\]

The boundary conditions are given by:

\[
\begin{align*}
z_d(0) & = 0, & \dot{z}_d(0) & = 0, & \ddot{z}_d(0) & = 0, \\
z_d(T) & = z_f, & \dot{z}_d(T) & = 0, & \ddot{z}_d(T) & = 0
\end{align*}
\]

Where \( z_f \) is the desired depth and \( T \) is the time that the vehicle reaches to the desired depth. Here, the value of \( z_f \) and \( T \) is considered 2 meters and 6 seconds respectively.

According to the aforementioned boundary conditions, the coefficients are achieved as follows:

\[
a_0 = a_1 = a_2 = 0, \quad a_3 = 10 \frac{z_f}{T^3}, \quad a_4 = -15 \frac{z_f}{T^4}, \quad a_5 = 6 \frac{z_f}{T^5}
\]

Substituting the values of \( z_f \) and \( T \) yields:

\[
a_0 = 0, \quad a_1 = 0, \quad a_2 = 0, \quad a_3 = 0.0926, \quad a_4 = -0.0231, \quad a_5 = 0.0015
\]

Therefore, the first part of depth trajectory is defined as follows:
Finally, the total desired trajectory is given by:

\[
z_d(t) = \begin{cases} 
0.0926t^3 - 0.0231t^4 + 0.0015t^5 & t \leq 6 \\
2 & t > 6 
\end{cases}
\] (21)

The depth trajectory is depicted in figure 1.

\[z_d\] is multiplied by -1 to have a better view.

In the next section a proper controller is designed that enables the vehicle to track the desired trajectory while minimizing the tracking error.

**B. Tracking problem**

In the tracking control design an optimal PD controller following with a proportional controller is designed for the nonlinear equations of the vehicle such that the desired depth trajectory achieved earlier is tracked with minimum error. The block diagram of the proposed optimal control design problem is shown in figure 2.

![Control design block diagram](image)

In the above block diagram, it is desired to both control the depth and the pitch angle simultaneously. In other words, when the vehicle is reached the final depth, it is desired to have a zero pitch angle in order to maintain the vehicle at this depth. According to figure 2, when the tracking error is zero or the vehicle is reached to the final depth, the final pitch angle is also zero. As can be seen in figure 2, two actuators are used due to the restrictions of fin and pitch angles, which is indicated by

\[-9\pi/180 \leq \delta \leq 9\pi/180\] and \[-30\pi/180 \leq \theta \leq 30\pi/180\].

Now, the goal is to determine optimal value of the controller parameters to have minimum tracking error. To do this, IPSO-SQP algorithm is employed. The parameters \(K_p\), \(K_D\) and \(\gamma\) are considered as particles in IPSO-SQP algorithm and the cost function is considered as follow:

\[
J = \int \left( k_z(z_d(t) - z(t))^2 + k_{\theta}(\theta_d(t) - \theta(t))^2 \right) dt
\] (22)
Where $z_d$ and $z$ are the desired and actual depth and $\theta_d$ and $\theta$ are the desired and actual pitch angle respectively. In the IPSO-SQP algorithm the number of swarm is considered as $s=30$, $c_1=c_2=0.9$ and the upper and lower bound of $K_p$, $K_D$ and $\gamma$ are set as $-200 \leq K_p, K_D, \gamma \leq 200$.

The negative value of the parameters represent the difference in sign conventions between the stern plane angle ($\delta_s$) and vehicle pitch angle ($\theta$). Positive stern plane angle will generate a negative moment about the y-axis, forcing the vehicle to pitch down (negative pitch rate). The algorithm is run for twenty times and the best achieved results are presented in the following Table.

<table>
<thead>
<tr>
<th>$K_p$</th>
<th>$K_D$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-53.6758</td>
<td>-66.7665</td>
<td>-3.6780</td>
</tr>
</tbody>
</table>

For the best results presented in Table I, the figures corresponding to the depth, depth tracking error, pitch angle and the control input (fin angle) are illustrated in the following. First, the desired and actual depth trajectories are depicted in figure 3 and 4.

![Fig. 3. Desired and actual depth](image)

![Fig. 4. Depth tracking error](image)

It can be seen from figure 3 and 4 that the proposed controller is able to force the vehicle to track the desired path with minimum error. The tracking error is very close to zero and this verifies the proper performance of the proposed controller.

In the following the desired and actual pitch angle trajectories are illustrated:

![Fig. 5. Desired and actual pitch angle](image)
As is obvious from figure 5 and 6 the vehicle tracks the desired pitch angle with minimum error. Moreover, the control input (fin angle) is presented in figure 7.

**Conclusion**

In this paper, motion planning of a REMUS AUV is studied. The procedure involves two steps. In the first step or the trajectory planning step, the desired depth is designed according to some assumptions. In the second step or the tracking control design step, a PD controller following with a proportional controller is designed to track the desired depth trajectory. In this regard a special control design structure is proposed in which both the depth and the pitch angle are controlled simultaneously. The parameters of the controller are determined by means of IPSO-SQP algorithm. The applied algorithm is commonly used to solve nonlinear optimal control problems. The simulation results verify that the proposed controller is able to track the desired path with minimum error.
Appendix A

The value of parameters in ((1)) to ((6)) is presented in the following Table.

### TABLE II. VALUE OF PARAMETERS IN AUV EQUATIONS OF MOTION

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_G$</td>
<td>0</td>
<td>m</td>
</tr>
<tr>
<td>$z_G$</td>
<td>0.0196</td>
<td>m</td>
</tr>
<tr>
<td>$X_e$</td>
<td>-9.30e-001</td>
<td>kg</td>
</tr>
<tr>
<td>$X_{e,q}$</td>
<td>-3.55e+001</td>
<td>kg/rad</td>
</tr>
<tr>
<td>$Z_e$</td>
<td>-1.93e+000</td>
<td>kg/m/rad</td>
</tr>
<tr>
<td>$Z_{e,q}$</td>
<td>-3.55e+001</td>
<td>kg/rad</td>
</tr>
<tr>
<td>$Z_{w,q}$</td>
<td>-5.22e+000</td>
<td>kg/rad</td>
</tr>
<tr>
<td>$Z_{w,e}$</td>
<td>-2.86e+001</td>
<td>kg/rad</td>
</tr>
<tr>
<td>$Z_{H}+1$</td>
<td>-1.31e+002</td>
<td>kg/rad</td>
</tr>
<tr>
<td>$Z_{H}$</td>
<td>-6.32e-001</td>
<td>kg/m/rad²</td>
</tr>
<tr>
<td>$Z_{w,e}$</td>
<td>-9.64e+000</td>
<td>kg/(m/rad)</td>
</tr>
<tr>
<td>$M_{w,q}$</td>
<td>-2.00e+000</td>
<td>kg/m/rad</td>
</tr>
<tr>
<td>$M_{H}$</td>
<td>3.18e+000</td>
<td>kg</td>
</tr>
<tr>
<td>$M_{H}+1$</td>
<td>-1.88e+002</td>
<td>kg.m²/rad²</td>
</tr>
<tr>
<td>$M_{w}$</td>
<td>34.6</td>
<td>kg.m²/s²</td>
</tr>
<tr>
<td>$I_{w}$</td>
<td>3.45</td>
<td>kg.m²</td>
</tr>
<tr>
<td>Thrust</td>
<td>9.25</td>
<td>N</td>
</tr>
<tr>
<td>U</td>
<td>1.54</td>
<td>m/s</td>
</tr>
</tbody>
</table>

### References


