Convex Optimization Algorithms for Cooperative Localization in Autonomous Underwater Vehicles

LIU Ming-Yong[1] LI Wen-Bai[1] PEI Xuan[1]

Abstract In this paper, a cooperative localization algorithm for autonomous underwater vehicles (AUVs) is proposed. A "parallel" model is adopted to describe the cooperative localization problem instead of the traditional "leader-follower" model, and a linear programming associated with convex optimization method is used to deal with the problem. After an unknown-but-bounded model for sensor noise is assumed, bearing and range measurements can be modeled as linear constraints on the configuration space of the AUVs. Merging these constraints induces a convex polyhedron representing the set of all configurations consistent with the sensor measurements. Estimates for the uncertainty in the position of a single AUV or the relative positions of two or more nodes can then be obtained by projecting this polyhedron onto appropriate subspaces of the configuration space. Two different optimization algorithms are given to recover the uncertainty region according to the number of the AUVs. Simulation results are presented for a typical localization example of the AUV formation. The results show that our positioning method offers a good localization accuracy, although a small number of low-cost sensors are needed for each vehicle, and this validates that it is an economical and practical positioning approach compared with the traditional approach.

Key words Autonomous underwater vehicle (AUV), convex optimization, cooperative localization, uncertainty region, screening algorithm, approximation algorithm

DOI 10.3724/SP.J.1004.2010.00704

A fundamental capability for autonomous underwater vehicles (AUVs) operations is localization, which is the ability of an AUV to estimate its position in the environment. Within this field of research, there is a more recent and narrower focus on cooperative localization for AUVs. However, the absence of GPS underwater makes navigation and localization for AUVs a challenge. Without an external reference in the form of acoustic beacons at known positions, the vehicles have to rely on proprioceptive information obtained through a compass, a Doppler velocity logger (DVL) or an inertial navigation system (INS). Independent of the quality of the sensors used, the error in the position estimate based on dead-reckoning information grows without bound. Moreover, by surfacing the AUV, we can obtain a position update through its GPS, but this is impossible or undesirable for many applications. The use of static beacons in the form of a long baseline (LBL) array limits the operation area to a few square kilometers and requires a substantial deployment effort before operations, especially in deep water[1].

A traditional method to realize cooperative localization for AUVs is the "leader-follower" model[2−4]. That is, the leader vehicle is equipped with various navigation equipment (DVL, INS, etc.) with high precision and reliability, while the follower′s equipment has a relatively low accuracy. Then, by using integrated navigation algorithms, real-time localization results for each follower vehicle can be obtained with high accuracy. However, these sophisticated equipment is expensive, which cannot always be afforded in many practical applications. Under such a motivation, we adopt a "parallel" model to deal with the cooperative localization problem. That is, each of the AUV is carried with the same relatively low-accuracy navigation equipment, via employing sensor-fusion techniques, the exact location for each AUV can be calculated through the acoustic sensor measurements. Compared with the "leader-follower" method, the advantages of our method are the small number of low-cost sensors (only an acoustic modem and bearing and range sensors are considered) and the relatively high localization accuracy of the AUV swarm. Therefore, it is more economical and practical.

In recent years, linear and nonlinear programming and convex optimization methods have been widely applied to multirobot field to realize formation control, path tracking, and cooperative positioning for robot teams. Through the graph-theoretic and convex optimization methods, Doherty et al.[5] proposed a method for estimating unknown node positions in a robot team. Spletzer et al.[6−10] studied the task of repositioning a formation of robots to a new shape while minimizing either the maximum distance that any robot travels or the total distance traveled by the formation. The optimal solutions here can be obtained via exploring second-order cone programming techniques and convex optimization methods. Furthermore, even in the case of communication constraints, some multivehicle path coordination problems can still be well solved via using mixed integer linear programming and nonlinear programming methods[11−14]. This has great significance for multi-AUV underwater navigation.

In this paper, linear programming and convex optimization methods are used to solve the multi-AUV localization problem. Consider the cooperative localization problem of AUVs in $\mathbf{R}^3$ under an unknown-but-bounded model assumption for sensor noise. As depth can be accurately measured with a pressure sensor, the AUV can use its depth and the depth received from the communication and navigation aid-AUV (CNA) to project the CNA′s position into a $\mathbf{R}^2$ plane, and thereby reducing the cooperative localization from a 3-D to a 2-D problem[2−3]. Hence, the bearing and range measurements can be modeled as linear constraints on the configuration space of the AUVs in $\mathbf{R}^2$. Merging these constraints induces a convex polyhedron representing the set of all configurations consistent with the sensor measurements. Estimates for the uncertainty in the absolute position of a single AUV or the relative positions of two or more nodes can then be obtained by projecting this polyhedron onto appropriate subspaces of the configuration space. However, recovering the exact projection, which plays a core role in our approach, is rather hard. A novel method is proposed to recover the exact projection here. It is simpler and more efficient than the traditional

Manuscript received March 9, 2009; accepted July 28, 2009
Supported by National High Technology Research and Development Program of China (863 Program) (2007AA809502C), National Natural Science Foundation of China (50979093), and Program for New Century Excellent Talents in University (NCET-06-0877)

1. College of Marine, Northwestern Polytechnical University, Xi′an 710072, P. R. China

methods, while the latter requires exponential time[6−7, 10].

The principal contributions of this paper are as follows: 1) The multi-AUV cooperative localization problem is transformed into a linear programming problem with constraints, and the mathematical model has a universal applicability; 2) Two different optimization algorithms are proposed to recover the uncertainty region of each AUV via using convex optimization method. For different numbers of AUVs, these algorithms have good convergence property and positioning accuracy.

The paper is organized as follows. Section 1 generates linear constraints of the AUV formation. Section 2 describes some preparatory work about projection and optimization. Section 3 describes two different localization algorithms via using convex optimization methods and compares the computational complexities of them. Section 4 presents simulation results.

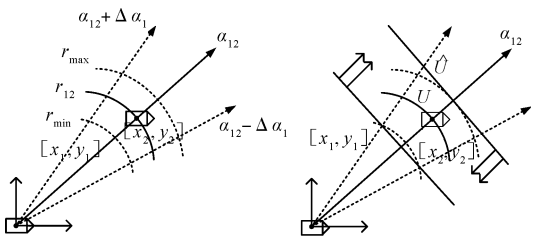
1 Generating linear constraints

The approach here relies upon modeling the bearing and range measurements obtained by the AUVs as linear constraints on the configuration of the AUV team. This section focuses on the transformation required to realize this. Let $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbf{R}^2$ represent the positions of AUV, i, j relative to some common reference frame, where $\boldsymbol{x}_i = (x_i, y_i)$ is the component expression of the position vector. Again in a slight abuse of notation, $\boldsymbol{x}_i$ is also used to denote the i-th AUV itself. Let each of the n AUVs be equipped with sensors allowing it to measure bearing and possibly range to another AUV with bounded error. Denote by α_{ij} and r_{ij} the bearing and range measurements taken by $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$, respectively. Furthermore, assume that each AUV is able to infer its orientation θ with respect to a specified reference direction with bounded error.

Note that bearing measurements are easy to translate into linear constraints. Considering Fig. 1 (a), by assuming bounded error in bearing measurements, α_{12} defines an uncertainty sector formed by the rays $[\boldsymbol{x}_1, \alpha_{12} + \Delta\alpha_1]$ and $[\boldsymbol{x}_1, \alpha_{12} - \Delta\alpha_1]$, where $\Delta\alpha_1$ reflects the uncertainty in both bearing (α_{12}) and orientation (θ) measurements. By denoting $\alpha_{\max} = \alpha_{12} + \Delta\alpha_1$ and $\alpha_{\min} = \alpha_{12} - \Delta\alpha_1$, then the uncertainty region is constrained by the linear inequalities

$$\begin{aligned}(x_2 - x_1)\sin\alpha_{\min} + (y_1 - y_2)\cos\alpha_{\min} \le 0\\ (x_1 - x_2)\sin\alpha_{\max} + (y_2 - y_1)\cos\alpha_{\max} \le 0\end{aligned} \quad (1)$$

For range measurements, the procedure seems a little more complicated. Again, suppose that the true range measurement can be bounded by $r_{\min} \le r_{12} \le r_{\max}$. Combining this with the bearing constraints results in an annular sector of uncertainty U. Then, by choosing two linear constraints orthogonal to α_{12} and supporting the critical points of U generates a new uncertainty region $\hat{U}|U \subset \hat{U}$ ($\hat{U}$ is a trapezoidal area), as illustrated in Fig. 1 (b). According to the geometric relationship in Fig. 1 (b), we have



(a) Bearing measurement (b) Range measurement

Fig. 1 Bearing and range measurements in $\mathbf{R}^2$

$$\begin{aligned}(x_2 - x_1)\cos\alpha_{12} + (y_2 - y_1)\sin\alpha_{12} - r_{\max} \le 0\\ (x_1 - x_2)\cos\alpha_{12} + (y_1 - y_2)\sin\alpha_{12} - r_{\min} \le 0\end{aligned} \quad (2)$$

2 Projection and optimization

In this section, we will do some preparatory work. These are theoretical basis to the localization approach here.

2.1 Generating projection

Let $\mathcal{C} \subset \mathbf{R}^{m\times n}$ denote the configuration space of the AUVs and $\bar{\boldsymbol{x}} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{C}$ denote the current configuration of the ensemble. Here, m and n denote the dimension of these position vectors and the number of AUVs, respectively. With these representations, the bearing and range sensor measurements can then be modeled as linear constraints on $\bar{\boldsymbol{x}}$, as described in the previous section. By combining these constraints, a system of linear inequalities is obtained

$$A\bar{\boldsymbol{x}} \le \boldsymbol{b} \quad (3)$$

where $A \in \mathbf{R}^{l\times(m\times n)}$ represents the sensor constraints. Equation (3) induces a convex polyhedron $P \subset \mathcal{C}$ on the configuration space of the formation, which corresponds to the set of all configurations consistent with the sensor measurements.

The notion of the projection of the convex polyhedron P is defined as

$$\mathcal{P} = \Pi(P) = \{\pi(\bar{\boldsymbol{x}}) | \bar{\boldsymbol{x}} \in P\} \quad (4)$$

where the projective mapping has the form

$$\begin{aligned}\pi_i(\bar{\boldsymbol{x}}) &= \boldsymbol{x}_i\\ \pi_{ij}(\bar{\boldsymbol{x}}) &= \boldsymbol{x}_i - \boldsymbol{x}_j\end{aligned} \quad (5)$$

which corresponds to the position of $\boldsymbol{x}_i$, and the relative positions of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively. Clearly, $\mathcal{P} \subset \mathbf{R}^2$ is the convex polygon related to P.

Generally, the notion of projection can be extended to describe the process of projecting a region P onto a k-dimensional subspace

$$\pi(\bar{\boldsymbol{x}}) = D\bar{\boldsymbol{x}} \quad (6)$$

where the projection matrix D maps the configuration vector $\bar{\boldsymbol{x}}$ from $\mathbf{R}^{m\times n}$ to $\mathbf{R}^k$.

2.2 Geometric properties of P

The convex polyhedron P induced by (3) is called the feasible region[15]. By adding a set of artificial variables, (3) can be rewritten as

$$\tilde{A}\bar{\bar{\boldsymbol{x}}} = \boldsymbol{b} \quad (7)$$

where $\bar{\bar{\boldsymbol{x}}} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{x}_{n+1}, \boldsymbol{x}_{\tilde{n}})$, $\boldsymbol{x}_{n+1}, \cdots, \boldsymbol{x}_{\tilde{n}} \ge 0$, and $\tilde{A} = [A, I_{\tilde{n}-n}]$. Assume $\mathrm{rank}\tilde{A} = \tilde{m}$ and $\tilde{A} = [\tilde{B}, \tilde{N}]$, where $\tilde{B}$ is a nonsingular matrix of order $\tilde{m}$. Without loss of generality, we can always assume that the front $\tilde{m}$ columns of $\tilde{A}$ are linearly independent, or one only needs to use column permutation to achieve it. Furthermore, denote

$$\bar{\bar{\boldsymbol{x}}} = \begin{bmatrix}\bar{\bar{\boldsymbol{x}}}_{\tilde{B}}\\ \bar{\bar{\boldsymbol{x}}}_{\tilde{N}}\end{bmatrix}$$

where the components in $\bar{\bar{\boldsymbol{x}}}_{\tilde{B}}$ and $\bar{\bar{\boldsymbol{x}}}_{\tilde{N}}$ correspond to the column vectors in $\tilde{B}$ and $\tilde{N}$, respectively. Hence, (7) is rewritten as

$$(\tilde{B}, \tilde{N})\begin{bmatrix}\bar{\bar{\boldsymbol{x}}}_{\tilde{B}}\\ \bar{\bar{\boldsymbol{x}}}_{\tilde{N}}\end{bmatrix} = \boldsymbol{b}$$

and then

$$\bar{\tilde{\boldsymbol{x}}}_{\tilde{N}} = \tilde{B}^{-1}\boldsymbol{b} - \tilde{B}^{-1}\tilde{N}\bar{\tilde{\boldsymbol{x}}}_{\tilde{N}} \tag{8}$$

The column vectors in $\bar{\tilde{\boldsymbol{x}}}_{\tilde{N}}$ are equal to the free variables in linear algebra. Particularly, let $\bar{\tilde{\boldsymbol{x}}}_{\tilde{N}} = \boldsymbol{0}$; then

$$\bar{\tilde{\boldsymbol{x}}} = \begin{bmatrix} \bar{\tilde{\boldsymbol{x}}}_{\tilde{B}} \\ \bar{\tilde{\boldsymbol{x}}}_{\tilde{N}} \end{bmatrix} = \begin{bmatrix} \tilde{B}^{-1}\boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \tag{9}$$

According to linear programming theory, if $\tilde{B}^{-1}\boldsymbol{b} \geq 0$, then (9) is a basic feasible solution of problem (7) subject to $\bar{\tilde{\boldsymbol{x}}} \geq \boldsymbol{0}$, and $\tilde{B}$ is a feasible basis matrix with its components $\bar{\tilde{\boldsymbol{x}}}_{\tilde{B}_1}, \cdots, \bar{\tilde{\boldsymbol{x}}}_{\tilde{B}_{\tilde{m}}}$ constituting a feasible basis set.

The following lemma which reflects the relationships between the extreme points set of P and the basic feasible solutions is especially significant to our approach.

Lemma 1[15]**.** Let $K = \{\boldsymbol{x} | A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}, A \in \mathbf{R}^{m\times n}$, and $\mathrm{rank}A = m$. Then the set of the extreme points of the polyhedron (feasible region) induced by $A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$ is equal to the basic feasible solutions of K.

With this lemma, the polygon $\mathcal{P}$ can be recovered via the following two steps.

Step 1. Obtain all of the basic feasible solutions of problem (7) subject to $\bar{\tilde{\boldsymbol{x}}} \geq \boldsymbol{0}$;

Step 2. Apply projective mappings (5) to get the extreme points of $\mathcal{P}$.

Step 2 is obvious. Let us turn to consider Step 1. In fact, to obtain the basic feasible solutions of problem (7), we only need to use the elementary transformation techniques to matrix $\tilde{A}$ to get different feasible basis matrices. Since $\mathrm{rank}\tilde{A} = \tilde{m}$, the number of the basic feasible solutions do not exceed

$$\binom{\tilde{n}}{\tilde{m}} = \frac{\tilde{n}!}{\tilde{m}!(\tilde{m}-\tilde{n})!} \tag{10}$$

3 Localization algorithm

According to the number of AUVs in the team, two approaches will be used to recover the vertices (extreme points) of the projection $\mathcal{P}$. Once obtaining the uncertainty region $\mathcal{P}$, we realize the positioning of the AUVs.

3.1 Screening algorithm

This algorithm is used to recover the uncertainty region $\mathcal{P}$ by screening all the critical points and removing the inner points of the projection of P. The following example combined with illustrations is used to explain this procedure.

Without loss of generality, suppose that the vertices set of the polyhedron P is $V = \{V_1, \cdots, V_6\}$ and the corresponding vertices set of the polygon $\mathcal{P}$ is $\tilde{\mathcal{V}} = \{\tilde{\mathcal{V}}_1, \cdots, \tilde{\mathcal{V}}_k\}$. Note that $k \leq 6$, since not all of the vertices obtained via projecting V to $\mathbf{R}^2$ are the true vertices of $\mathcal{P}$ (some projective points may just be the inner points). Next, the screening algorithm is applied to get the true vertices set $\tilde{\mathcal{V}}$ in two steps.

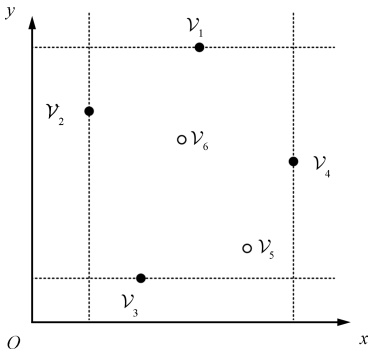
Step 1. Determine the four critical vertices of $\mathcal{P}$. As shown in Fig. 2 (a), denote the whole vertices obtained via projecting V to $\mathbf{R}^2$ as $\mathcal{V}_i(x_i, y_i), i = 1, \cdots, 6$. With $\tilde{\mathcal{V}}$ initialized to $\emptyset$, the screening algorithm is given as follows:

Begin
Because $\max_{1\leq i\leq 6}\{x_i | x_i \in \mathcal{V}_i(x_i, y_i)\} = x_4$
therefore add $\mathcal{V}_4$ to $\tilde{\mathcal{V}}$;
Because $\min_{1\leq i\leq 6}\{x_i | x_i \in \mathcal{V}_i(x_i, y_i)\} = x_2$
therefore add $\mathcal{V}_2$ to $\tilde{\mathcal{V}}$;
Because $\max_{1\leq i\leq 6}\{y_i | y_i \in \mathcal{V}_i(x_i, y_i)\} = y_1$
therefore add $\mathcal{V}_1$ to $\tilde{\mathcal{V}}$;
Because $\min_{1\leq i\leq 6}\{y_i | y_i \in \mathcal{V}_i(x_i, y_i)\} = y_3$
therefore add $\mathcal{V}_3$ to $\tilde{\mathcal{V}}$.
End
Thus, $\tilde{\mathcal{V}} = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4\}$.

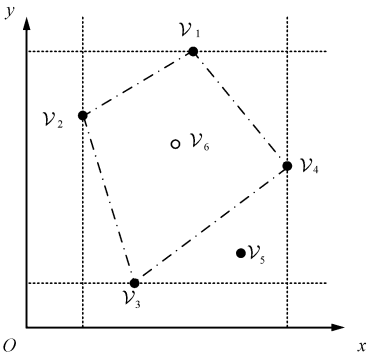
Step 2. Judge the remaining vertices of $\mathcal{V}$. As shown in Fig. 2 (b), let $y = \kappa_i x + b_i$, $i = 1, \cdots, 4$ be the equations of four lines generated in turn by $(\mathcal{V}_1, \mathcal{V}_2), (\mathcal{V}_3, \mathcal{V}_4), (\mathcal{V}_1, \mathcal{V}_4), (\mathcal{V}_2, \mathcal{V}_3)$, respectively. Consider $\mathcal{V}_j(x_j, y_j) \in \mathcal{V} \setminus \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4\}, j = 5, 6$.

Begin
If $y_j = \kappa_i x_j + b_j \leq 0, \quad i = 1, 2; j = 5, 6$
or $y_j = \kappa_i x_j + b_j \geq 0, \quad i = 1, 2; j = 5, 6$
then add $\mathcal{V}_j$ to $\tilde{\mathcal{V}}$;
Else if $y_j = \kappa_i x_j + b_j \leq 0, \quad i = 3, 4; j = 5, 6$
or $y_j = \kappa_i x_j + b_j \geq 0, \quad i = 3, 4; j = 5, 6$
then add $\mathcal{V}_j$ to $\tilde{\mathcal{V}}$;
Else delete $\mathcal{V}_j$.
End
Hence, the true vertices set is $\tilde{\mathcal{V}} = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4, \mathcal{V}_5\}$.

There is a degenerate case when the values of some $\kappa_i, i = 1, \cdots, 4$ are infinity. However, this case is easier for us to deal with, since one only needs to consider the component x_i of $\mathcal{V}_i$. Anyway, it shows that the above algorithm holds for the degenerate case as well.



(a) Corresponding to Step 1



(b) Corresponding to Step 2

Fig. 2 A screening algorithm to recover the vertices of the projection $\mathcal{P}$

3.2 Approximation algorithm

For our purposes, when the number of the AUVs is large (for example, $n > 10$), recovering the true projection $\mathcal{P}$ by the screening algorithm is computationally too expensive. Instead, the approximation approach chooses a search direction and finds the maximum extent of $\mathcal{P}$ in this direction.

This amounts to solving the linear programming problem

$$\max_{\bar{\boldsymbol{x}}} \boldsymbol{c}_i^{\mathrm{T}} \bar{\boldsymbol{x}} \qquad \text{s.t. } A\bar{\boldsymbol{x}} \leq \boldsymbol{b} \tag{11}$$

where $\boldsymbol{c}_i$ corresponds to the search direction and is chosen parallel to the projection subspace ($\boldsymbol{c}_i \in \mathrm{span}(D^{\mathrm{T}})$).

The solution to this linear programming problem yields a vertex $v_i^* \in \mathcal{V}$ in direction $\boldsymbol{c}_i$ and its supporting hyperplane H_i, where $\mathcal{V} \subset P$ is the set of extreme points of $\mathcal{P}$. The projection of H_i induces a half-space constraint on the subspace under consideration. By repeating this process for several search directions, an approximation $\mathcal{P}^+$ for the true projection is obtained by intersecting these half-spaces. Note that $\mathcal{P}^+$ is guaranteed to bound the true projection $\mathcal{P}$.

Then, let $\mathcal{V}^-$ denote the set of projected extreme points corresponding to different solutions of the linear programming problem in (11). Similarly, let $\mathcal{V}$ and $\mathcal{V}^+$ represent the vertices of the true projection $\mathcal{P}$ and the bounding approximation $\mathcal{P}^+$, respectively. These are represented in Fig. 3.

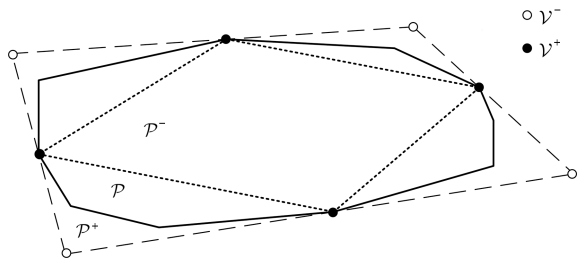


Fig. 3 Set hierarchy generated through projection estimates ($\mathcal{P}^+$ and $\mathcal{P}^-$ correspond to upper and lower bounds to the true set projection $\mathcal{P}$, respectively.)

Note that $\mathcal{V}^- \subset \mathcal{V}$, i.e., the set of the projected vertices $\mathcal{V}^-$ is the subset of the true projection′s vertex set $\mathcal{V}$. This means a lower bound estimate $\mathcal{P}^-$ for $\mathcal{P}$ from the convex hull of $\mathcal{V}^-$ can also be constructed. This leads to a set hierarchy $\mathcal{P}^- \subset \mathcal{P} \subset \mathcal{P}^+$, from which the performance indicator of the approximation algorithm can be defined as

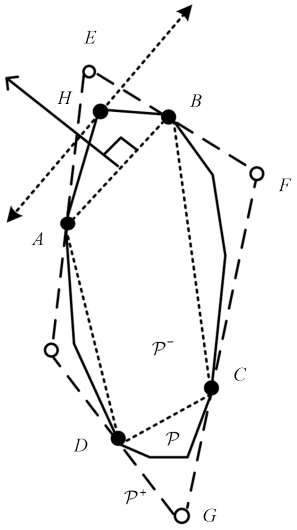
$$PAA = \frac{\mathrm{Area}(\mathcal{P}^-)}{\mathrm{Area}(\mathcal{P}^+)} \tag{12}$$

Obviously, $0 < PAA \leq 1$.

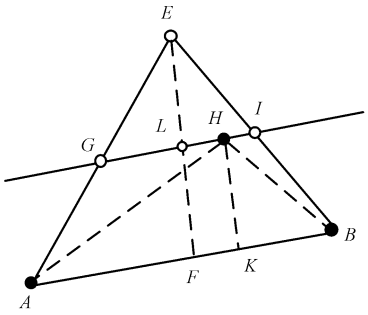
Recall that the initial search direction $\boldsymbol{c}_1 \in \mathrm{span}(D^{\mathrm{T}})$ can be chosen randomly without affecting the convergence analysis. Hence, the search algorithm is given as follows:

Begin
Choose $\boldsymbol{c}_1 \in \mathrm{span}(D^{\mathrm{T}})$
$\boldsymbol{c}_2 \Longleftarrow -\boldsymbol{c}_1$
$v_1 \Longleftarrow \pi(\max_{\bar{\boldsymbol{x}}} \boldsymbol{c}_1^{\mathrm{T}}\bar{\boldsymbol{x}}$ s.t. $A\bar{\boldsymbol{x}} \leq \boldsymbol{b})$
$v_2 \Longleftarrow \pi(\max_{\bar{\boldsymbol{x}}} \boldsymbol{c}_2^{\mathrm{T}}\bar{\boldsymbol{x}}$ s.t. $A\bar{\boldsymbol{x}} \leq \boldsymbol{b})$
$\boldsymbol{c}_3 \Longleftarrow \overline{v_1v_2}^{\perp}$ {Choose $\boldsymbol{c}_3$ orthogonal to segment v_1v_2 }
$\boldsymbol{c}_4 \Longleftarrow -\boldsymbol{c}_3$
$v_3 \Longleftarrow \pi(\max_{\bar{\boldsymbol{x}}} \boldsymbol{c}_3^{\mathrm{T}}\bar{\boldsymbol{x}}$ s.t. $A\bar{\boldsymbol{x}} \leq \boldsymbol{b})$
$v_4 \Longleftarrow \pi(\max_{\bar{\boldsymbol{x}}} \boldsymbol{c}_4^{\mathrm{T}}\bar{\boldsymbol{x}}$ s.t. $A\bar{\boldsymbol{x}} \leq \boldsymbol{b})$
$\mathcal{V}_4^- \Longleftarrow \{v_1, v_2, v_3, v_4\}$
$\mathcal{P}_4^- \Longleftarrow \mathrm{Polygon}(\mathcal{V}^-)$
$v_1^+ = (v_1, \boldsymbol{c}_1) \bigcap (v_3, \boldsymbol{c}_3)$
$v_2^+ = (v_3, \boldsymbol{c}_3) \bigcap (v_2, \boldsymbol{c}_2)$
$v_3^+ = (v_2, \boldsymbol{c}_2) \bigcap (v_4, \boldsymbol{c}_4)$
$v_4^+ = (v_4, \boldsymbol{c}_4) \bigcap (v_1, \boldsymbol{c}_1)$ {Vertices recovered from the intersection of projected hyperplanes}
$\mathcal{V}_4^+ \Longleftarrow \{v_1^+, v_2^+, v_3^+, v_4^+\}$
$\mathcal{P}_4^+ \Longleftarrow \mathrm{Polygon}(\mathcal{V}^+)$
End

After four times of searches, we obtain $\mathcal{P}^+ = 2\mathcal{P}^-$, and the set $\mathcal{P}^+ \setminus P^-$ corresponds to disjoint regions, as shown in Fig. 4 (a).



(a) After four times of searches, subsequent search directions explore the largest disjoint region



(b) By projecting normal to the edge $\overline{AB}$ in $\mathcal{P}^-$, we create two disjoint regions ABH and EGI. These will be added to and subtracted from $\mathcal{P}^-$ and $\mathcal{P}^+$, respectively

Fig. 4 Search procedure of the approximation algorithm

The subsequent search strategy then proceeds as follows:

1) Determine the prospective search region of the greatest area;

2) Choose a search direction $\boldsymbol{c}^{\perp}$ normal to the corresponding edge of $\mathcal{P}^-$;

3) Solve (11) with $\boldsymbol{c} = \boldsymbol{c}^{\perp}$;

4) Refine estimates for $\mathcal{P}^+$ and $\mathcal{P}^-$ accordingly.

This is illustrated in Fig. 4 (a). Using such a strategy allows us to bound the number of searches necessary to recover $\mathcal{P}$ as a function of the number of its vertices $\mathcal{V}$.

Next, consider the performance of the approximation algorithm (PAA). As shown in Fig. 4 (b), let $\mathcal{A}^+$ and $\mathcal{A}^-$ correspond to Area($\mathcal{P}^+$) and Area($\mathcal{P}^-$), respectively, and S_{s+1} denote the area of the region being searched. Then, PAA_{s+1} is defined as

$$PAA_{s+1} = \frac{\mathcal{A}_s^- + hS_{i+1}}{\mathcal{A}_s^+ - (1-h)^2 S_{i+1}} = \frac{\hat{\mathcal{A}}_s^- + h}{\hat{\mathcal{A}}_s^+ - (1-h)^2} \tag{13}$$

where $\hat{\mathcal{A}}_s^- = \mathcal{A}_s^-/S_{i+1}, \hat{\mathcal{A}}_s^+ = \mathcal{A}_s^+/S_{i+1}$, and h corresponds to the height of the vertex found normalized to the height of the search region ($h = \overline{HK}/\overline{EF}$ in Fig. 4 (b)). Minimize (13) with respect to h, i.e.,

$$\frac{\mathrm{d}(PAA_{s+1})}{\mathrm{d}h}|_{h=h^*} = 0$$

Then,

$$h^* = -\hat{\mathcal{A}}_s^- + [(\hat{\mathcal{A}}_s^- + 1)^2 - \hat{\mathcal{A}}_s^+]^{\frac{1}{2}}$$

Substituting this into (13) yields

$$PAA_{s+1} = \frac{\hat{\mathcal{A}}_s^- - \hat{\mathcal{A}}_s^- + [(\hat{\mathcal{A}}_s^- + 1)^2 - \hat{\mathcal{A}}_s^+]^{\frac{1}{2}}}{\hat{\mathcal{A}}_s^+ - [1 + \hat{\mathcal{A}}_s^- - ((\hat{\mathcal{A}}_s^- + 1)^2 - \hat{\mathcal{A}}_s^+)^{\frac{1}{2}}]^2} =$$

$$\frac{[(\hat{\mathcal{A}}_s^- + 1)^2 - \hat{\mathcal{A}}_s^+]^{\frac{1}{2}}}{-2[((\hat{\mathcal{A}}_s^- + 1)^2 - \hat{\mathcal{A}}_s^+) - ((\hat{\mathcal{A}}_s^- + 1)^2 - \hat{\mathcal{A}}_s^+)^{\frac{1}{2}}\hat{\mathcal{A}}_s^+]} =$$

$$\frac{[(\hat{\mathcal{A}}_s^- + 1)^2 - \hat{\mathcal{A}}_s^+]^{\frac{1}{2}} + \hat{\mathcal{A}}_s^- + 1}{2\hat{\mathcal{A}}_s^+} =$$

$$\frac{S_{s+1} + \mathcal{A}_s^- + [(\mathcal{A}_s^- + S_{s+1})^2 - \mathcal{A}_s^+ S_{s+1}]^{\frac{1}{2}}}{2\mathcal{A}_s^+} =$$

$$PAA_s + \frac{S_{s+1} - \mathcal{A}_s^- + [(\mathcal{A}_s^- + S_{s+1})^2 - \mathcal{A}_s^+ S_{s+1}]^{\frac{1}{2}}}{2\mathcal{A}_s^+}$$

This corresponds to the convergence in the worst case for our approximation algorithm.

After examining Fig. 5, we will achieve PAA levels of 0.99 after 30 iterations even if in the worst case of convergence.

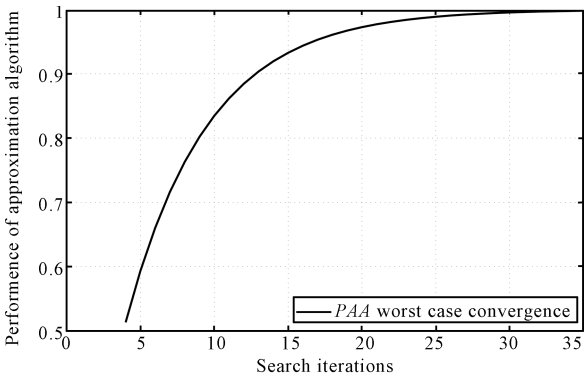


Fig. 5 Worst case convergence of PAA vs. the number of search iterations

3.3 Complexity

It is clear from Fig. 6 that with the increase in the number of AUVs, the computational complexities of the two algorithms have a significant difference. When the number of AUVs is small (for example, $n \leq 10$), both the complexities of the two algorithms are at the same level. However, when it is large, the computational complexity of the screening algorithm increases dramatically and cannot be ignored. In contrast to this, the approximation algorithm has a proper computational complexity when the number of AUVs becomes large.

Here, we need to explain two questions:

1) When should we select the screening algorithm? When should we use the approximation algorithm? Is there a principle?

2) Since the complexities of the two algorithms are at the same level when the number of AUVs is small, can we use the approximation algorithm instead of the former one, in other words, are there any essential differences between them?

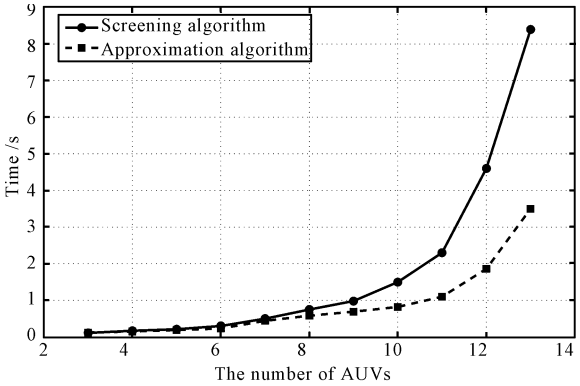


Fig. 6 The computational complexities of the screening algorithm and the approximation algorithm

To answer these questions, we should analyze the two algorithms from a theoretical level. Considering the screening algorithm, there is no approximation throughout the whole process of recovering the uncertainty region, therefore it does not have any theoretical errors. On the contrary, the approximation algorithm does have theoretical errors since the approximation is used to deal with the recovering process. Thus, selecting a correct recovering algorithm should make the computational complexity balanced with the theoretical error, and this is why we should use the screening algorithm but not the approximation algorithm when the number of AUVs is small.

4 Simulation results

Suppose that each AUV is capable of measuring the bearing to its neighbor in a common reference frame with a tolerance of $\pm 5^\circ$. For range and bearing sensors, we assume a tolerance of $\pm 10^\circ$ and an accuracy of $\pm 10\,\%$ of the measured range.

Firstly, assume that there are three AUVs in our AUV team. Let AUV 1 be the origin of common reference frame, i.e., the coordinate of AUV 1 is $O(0, 0)$. We further assume that AUV 2 is at $A\,(5\,\mathrm{m}, 5\,\mathrm{m})$ and AUV 3 is at $A_1\,(10\,\mathrm{m}, 0)$. Moreover, let $r_{\max} = \overline{OC} = \overline{A_1C_1} = (1{+}10\,\%)r = (1{+}10\,\%)\overline{OA} = (1{+}10\,\%)\overline{AA_1}$, $r_{\min} = \overline{OB} = \overline{A_1B_1} = (1 - 10\,\%)r = (1 - 10\,\%)\overline{OA} = (1 - 10\,\%)\overline{AA_1}$, $\alpha_{12} = 45^\circ, \alpha_{32} = 135^\circ$, and $\Delta\alpha_1 = \Delta\alpha_3 = 5^\circ$. Then, via using the screening algorithm, it is shown that the uncertainty region (Region 1) of AUV 2 is bounded by the intersection of two annular regions BB_1CC_1 generated by AUV 1 and AUV 3 from Fig. 7.

To compare with the screening algorithm, this example is simulated again by using the approximation algorithm under the same conditions, which is also illustrated in Fig. 7. We find that there is a vertex of the uncertainty region (Region 2) of AUV 2 outside the region BB_1CC_1 (By calculating, the "bad" vertex is exactly the $D\,(5.978\,\mathrm{m}, 4.958\,\mathrm{m})$ in Fig. 7). This phenomenon does not happen by chance. Since there are only three AUVs in this experiment and few measurements can be used to generate linear constraints. This leads to the approximation error accumulation and slowly convergent rate when we use the approximation algorithm to recover the uncertainty region. However, since the screening algorithm does not have any theoretical errors

during the recovering process, it converges sharply when the number of the AUVs is small.

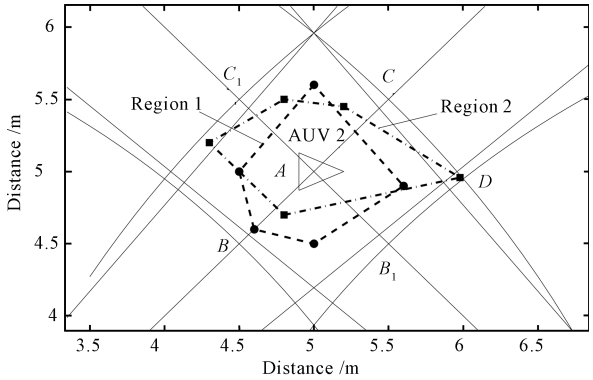


Fig. 7 Uncertainty regions generated by using the two different algorithms when the number of AUVs is three

Then, we add 10 other AUVs to the experiment under the previous condition. As clearly shown in Fig. 8, the two uncertainty regions of AUV 2 via using the two different algorithms reflect the performance of the algorithms. This example shows that some vertices of Region 1 (generated by the screening algorithm) are outside the bounded region, which is generated by the other 12 AUVs, while Region 2 is completely in the bounded region. These results reveal the fact that when the number of the AUVs is large, the performance of the approximation algorithm improves dramatically, while that of the screening algorithm is declining.

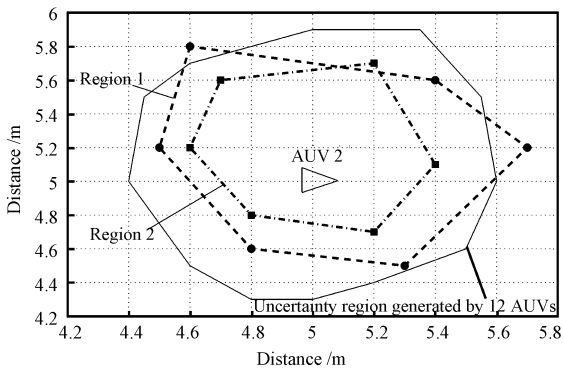


Fig. 8 Uncertainty regions generated by using two different algorithms when the number of AUVs is 13

In conclusion, the above two examples point out that the more accurate uncertainty region can be obtained by using the screening algorithm when the number of AUVs is not large. However, when it is large (for example $n > 10$), the error accumulation of the screening algorithm caused by computational complexity cannot be ignored, since the increment of the basic feasible solutions is combinatorial complexity from (10). In this case, the approximation algorithm should be selected, and this also validates our analysis at the end of Section 3.

Thus, the position and localization error for AUV 2 can be estimated from the centroid of the uncertainty region of AUV 2. In the two examples, localization errors are 2.3 % and 2.1 % with respect to the range measurements, respectively. These error results have almost met the accuracy level of the traditional "leader-follower" methods (about 1 % $\sim$ 2 % with respect to the range measurements[16−18]), while there are smaller number of low-cost sensors needed to be equipped for vehicles here. Therefore, it is an economical and practical positioning approach.

5 Conclusion

In this paper, a cooperative localization algorithm for AUVs is proposed. Instead of the traditional "leader-follower" model, a "parallel" model is adopted to describe the cooperative localization problem, and it is accomplished by using convex optimization algorithms. Although the accuracy of the sensors is not high enough for each AUV, satisfactory localization results of the AUV swarm can still be obtained through these algorithms. The key to our approach is how to recover the uncertainty region generated by the linear constraints. And two different algorithms are given to achieve this process. How to select a suitable algorithm to solve practical problems depends on the number of the AUVs. That is, when the number of the AUVs is not large, we select the screening algorithm; otherwise, the approximation algorithm should be used. Simulation results show that our positioning method offers a good localization accuracy, and it is an economical and practical positioning approach compared with the traditional approach.

References

1 Bahr A, Leonard J. Cooperative localization for autonomous underwater vehicles. In: Proceedings of the 10th International Symposium on Experimental Robotics. Berlin, Germany: Springer, 2006. 387−395

2 Santos N, Matos A, Kruppa H, Cruz N. Navigation of an autonomous underwater vehicle in a mobile network [Online], available: http://www.oceansys.fe.up.pt/publications/2008 SantosMa-tosCruz.pdf, March 10, 2009

3 Matos A, Santos N. AUV navigation and guidance in a moving acoustic network. In: Proceedings of the 2005 Europe IEEE Conference and Exhibition. Breast, France: IEEE, 2005. 1−6

4 Santos N. Navigation of Autonomous Vehicle in a Moving Acoustic Network [Master dissertation], Faculty of Engineering, Porto University, Portugal, 2008

5 Doherty L, Pister K, Ghaoui L. Convex position estimation in wireless sensor networks. In: Proceedings of the 12th Annual Joint Conference of the IEEE Computer and Communications Societies. Alaska, USA: IEEE, 2001. 1655−1663

6 Spletzer J, Fierro R. Optimal positioning strategies for shape changes in robot teams. In: Proceedings of IEEE International Conference on Robotics and Automation. Washington D. C., USA: IEEE, 2005. 742−747

7 Spletzer J, Taylor C. A bounded uncertainty approach to multi-robot localization. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. Las Vegas, USA: IEEE, 2003. 1−9

8 Das A, Fierro R, Kumar V, Ostrowski J, Spletzer J, Taylor C. A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 2002, **18**(1): 813−825

9 Derenick J, Spletzer J. Convex optimization strategies for coordinating large-scale robot formations. *IEEE Transactions on Robotics*, 2007, **23**(6): 1252−1259

10 Taylor C, Spletzer J. A bounded uncertainty approach to cooperative localization using relative bearing constraints. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Washington D. C., USA: IEEE, 2007. 2500−2506

11 Abichandani P, Benson H, Kam M. Multi-vehicle path coordination under communication constraints. In: Proceedings of the American Control Conference. Seattle, USA: IEEE, 2008. 650−656

12 Benson H, Sen A, Shanno D, Vanderbei R. Interior-point algorithms, penalty methods and equilibrium problems. *Computational Optimization and Applications*, 2006, **34**(1): 155−182

13 Vanderbei R, Shanno D. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 1999, **13**(1-3): 231−252

14 Lepetic M, Klancar G, Skrjanc I, Matko D, Potocnik B. Time optimal path planning considering acceleration limits. *Robotics and Autonomous Systems*, 2003, **45**(3): 199−210

15 Boyd S, Vandenberghe L. Convexo optimization [Online], available: http://www.stanford.edu/boyd/cvbook/, December 20, 2008

16 Stutters L, Liu H, Tiltman C, Brown D J. Navigation technologies for autonomous underwater vehicles. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2008, **38**(4): 581−589

17 Chandrasekhar V, Seah W K, Choo Y S, Ee H V. Localization in underwater sensor networks: survey and challenges. In: Proceedings of the 1st ACM International Workshop on Underwater Networks. Los Angeles, USA: ACM, 2006. 33−40

18 Tian Tan. *Technologies of Underwater Localization and Navigation*. Beijing: National Defense Industry Press, 2007 (in Chinese)

LIU Ming-Yong Professor at the College of Marine, Northwestern Polytechnical University. He received his bachelor degree from Northwestern Polytechnical University in 1992. His research interest covers navigation and control of autonomous underwater vehicles.
E-mail: lijun601@nwpu.edu.cn

LI Wen-Bai Ph. D. candidate at the College of Marine, Northwestern Polytechnical University. He received his bachelor degree from Northwestern Polytechnical University in 2006. His research interest covers navigation and control of autonomous underwater vehicles. Corresponding author of this paper.
E-mail: liwenbai@163.com

PEI Xuan Ph. D. candidate at the College of Marine, Northwestern Polytechnical University. He received his bachelor degree from Northwestern Polytechnical University in 2006. His research interest covers formation control and navigation of autonomous underwater vehicles.
E-mail: peixuan-001@163.com