PARTICLE SWARM OPTIMIZATION FOR TIME-DIFFERENCE-OF-ARRIVAL BASED LOCALIZATION

Kenneth W. K. Lui, Jun Zheng, and H. C. So

Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
Tel : (852) 2788 7780 Fax: (852) 2788 7791 email: hcso@ee.cityu.edu.hk

ABSTRACT

Time-difference-of-arrival (TDOA) based source localization has been intensively studied and broadly applied in many fields. In this paper, particle swarm optimization (PSO) is employed for positioning with TDOA measurements in the circumstances of known and unknown propagation speed. The optimization criterion is first developed and the PSO technique is then employed to search the global minimum of the cost function. For sufficiently small noise conditions, simulation results show that the PSO approach provides accurate source location estimation for both known and unknown propagation speed, and also gives an efficient speed estimate in the later case.

Index terms : time-difference-of-arrival, source localization, particle swarm optimization

1. INTRODUCTION

Time-difference-of-arrival (TDOA) based source localization is an important issue in many applications such as sonar, radar, navigation and surveillance. In the TDOA method, the differences in arrival times of the emitted source signal at multiple pairs of sensors are measured. Multiplying the TDOAs by the propagation speed will then yield the corresponding range differences. In the noise-free case and assuming two-dimensional (2-D) positioning, each range difference defines a hyperbolic locus on which the source must lie and its position is given by the intersection of two or more hyperbolas. Let $\mathbf{x} = [x, y]$ and $\mathbf{x}_k = [x_k, y_k]$, $k = 1, 2, \cdots, L$, be the unknown source location and known position of the kth sensor, respectively. Denote $D_{k,1}$ as the TDOA with respect to the first sensor, then we have the following relationship:

$$cD_{k,1} = \|\mathbf{x}_k - \mathbf{x}\|_2 - \|\mathbf{x}_1 - \mathbf{x}\|_2, \qquad k = 2, 3, ..., L \tag{1}$$

where c is the propagation speed which is either known or unknown, $\|\cdot\|_2$ denotes the Euclidean norm and L is the number of receivers.

In practice, the TDOA measurements are noisy and L is usually larger than 3. Positioning based on intersection of hyperbolas is thus inappropriate and generally the source position is solved using (1). In fact, many techniques have been proposed for TDOA based source localization in the literature [1]-[6] but they either assume known propagation speed and/or provide suboptimal estimates. The difficulty is due to the fact that the equations of (1) are highly nonlinear.

In this paper, we propose to employ the particle swarm optimization (PSO), an evolutionary search algorithm, to provide a robust and accurate solution for both known and unknown speed cases. It is noteworthy to mention that pioneer works of source localization with PSO have been presented in the applications of wireless sensor networks [7] and acoustics [8] where the former uses signal intensity and the latter integrates with particle filtering in scenario of reverberant environments.

The rest of the paper is organized as follows. An introduction of the PSO algorithm is given in Section II. The PSO algorithm is utilized for source localization in Section III for the case of known propagation speed, and for joint source position and propagation speed estimation in Section IV when the latter is not available. Simulation results are included in Section V to evaluate the estimation performance of the PSO approach. Finally, conclusions are drawn in Section VI.

2. PARTICLE SWARM OPTIMIZATION

PSO is an evolutionary technique invented by Eberhart and Kennedy in 1995 [9]-[10] from their study of bird flocking and fish schooling. Their idea is based on the observation on the food searching of a swarm of birds: When the only information for each bird (particle) is its distance from the food, a simple yet efficient way is to search the neighborhood of the bird which is the closest to the food (target). Assuming that the particles have memories, at each search (generation), there is a best result obtained by the ith particle from its searching experience, denoted by $\mathbf{p}_i^{best}$, and a best result obtained by the whole group of particles, denoted by $\mathbf{G}^{best}$. For a dimension N search with M particles, $\mathbf{p}_i^{best}$ and $\mathbf{G}^{best}$ are vectors of length N, and both are updated at each generation. The solution of the problem is then achieved from $\mathbf{G}^{best}$ at the last generation.

The PSO algorithm is thus formulated as [9]-[11]

$$\begin{aligned}\mathbf{v}_i(t+1) &= w\mathbf{v}_i(t) + c_1\mathbf{R}_{i1}(\mathbf{p}_i^{best}(t) - \mathbf{p}_i(t)) \\ &\quad + c_2\mathbf{R}_{i2}(\mathbf{G}^{best}(t) - \mathbf{p}_i(t)) \\ \mathbf{p}_i(t+1) &= \mathbf{p}_i(t) + \mathbf{v}_i(t+1), \qquad i = 1, 2, \cdots, M\end{aligned} \tag{2}$$

where

$$\mathbf{v}_i(t) = [v_{i,1}(t), v_{i,2}(t), \cdots, v_{i,N}(t)]^T$$
$$\mathbf{p}_i(t) = [p_{i,1}(t), p_{i,2}(t), \cdots, p_{i,N}(t)]^T$$

are the velocity and current location of the ith particle at time t, respectively, w is the inertia weight, $\mathbf{R}_{i1}$ and $\mathbf{R}_{i2}$ are $N \times N$ diagonal matrices with elements uniformly drawn from $[0, 1]$, which limit the search range for each particle in each dimension for a single iteration. The c_1 and c_2 are learning factors, which normally range from $[1, 4]$, and increasing their values will increase the search steps but some optimal points may be missed.

When the velocity grows to unreasonably large, algorithm divergence may occur. To avoid divergence and control the resolution, each particle's velocity is bounded by a maximum velocity, denoted by v_j^{max}, $j = 1,2,\cdots,N$. As a result, we have [10], [12]:

$$v_{i,j}(t) = \begin{cases} v_{i,j}(t) & \text{if}|v_{i,j}| < v_j^{max} \\ \text{sign}(v_{i,j}(t))v_j^{max} & \text{if}|v_{i,j}| \geq v_j^{max} \end{cases}$$
$$i = 1,2,\cdots,M \quad j = 1,2,\cdots,N \tag{3}$$

where $\text{sign}(u) = 1$ if $u > 0$, otherwise $\text{sign}(u) = -1$. After the initialization of particle velocity and position, a fitness value is computed for each particle. The $\mathbf{p}_i^{best}$ and $\mathbf{G}^{best}$ are determined based on the fitness values, and then the velocity and position of each particle are updated by (2) with the use of (3), respectively.

Note that the PSO is more suitable than simulated annealing (SA) [13] in our non-linear positioning cost function with numerous local optima. It is because SA escapes from local optima by occasional jumps, but PSO which contains many particles scratched within the search space, will jump at every iteration and thus the latter provides a better anti-locking ability. Compared to the genetic algorithm (GA) approach, PSO is similar to it in that the populations are both initialized by random solutions. On the other hand, PSO differs from GA in that it does not involve crossover and mutation operations. Instead, each particle updates itself with the information of itself and the particle group. Since PSO is simple in concept, easy to implement, has fewer user-defined parameters, and more computationally efficient than GA [14], it has been widely used in many applications, for example, neural networks training, power control, target positioning, and adaptive filtering [14]-[15].

3. SOURCE LOCALIZATION WITH KNOWN PROPAGATION SPEED

In this Section, the PSO algorithm is employed for the TDOA-based source localization in the case of known propagation speed. Generation of initial estimates for faster convergence is also discussed. More general case of the unknown propagation speed is discussed in the next Section.

First, we propose a fitness function

$$f(\mathbf{x}) = \sum_{k=2}^{L} \left(cD_{k,1} - \|\mathbf{x}_k - \mathbf{x}\|_2 + \|\mathbf{x}_1 - \mathbf{x}\|_2\right)^2 \tag{4}$$

In fact, this nonlinear least squares cost function corresponds to the maximum likelihood estimator when the disturbance is zero-mean white Gaussian. The estimate of the source location is derived by employing the PSO algorithm to find the global minimum of (4).

When implementing the PSO, particle initialization with random solutions will lead to slow convergence. To allow faster convergence, we propose to use the least squares (LS) method to provide initial parameter estimates, which are given by [6]

$$\hat{\theta}_1 = (\mathbf{A}_1^T\mathbf{A}_1)^{-1}\mathbf{A}_1^T\mathbf{b}_1 \tag{5}$$

where

$$\hat{\theta}_1 = \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{d} \end{bmatrix}; \qquad \mathbf{A}_1 = \begin{bmatrix} (\mathbf{x}_2 - \mathbf{x}_1)^T & cD_{2,1} \\ \vdots & \vdots \\ (\mathbf{x}_L - \mathbf{x}_1)^T & cD_{L,1} \end{bmatrix}$$

and

$$\mathbf{b}_1 = \frac{1}{2}\begin{bmatrix} x_1^2 + y_1^2 - x_2^2 - y_2^2 + c^2D_{2,1}^2 \\ \vdots \\ x_1^2 + y_1^2 - x_L^2 - y_L^2 + c^2D_{L,1}^2 \end{bmatrix}$$

with $\hat{\mathbf{x}}$ the estimate of $\mathbf{x}$, $d = \|\mathbf{x}_1 - \mathbf{x}\|_2$, $\hat{d}$ the estimate of d, $(\cdot)^T$ and $(\cdot)^{-1}$ denote the matrix transpose and inverse, respectively.

Then, $\hat{\mathbf{x}}$ is used in the initialization step for the PSO algorithm:

$$\mathbf{p}_i(0) = \mathbf{Q}_i\hat{\mathbf{x}} \tag{6}$$

$$\mathbf{Q}_i = \text{diag}(1 + c_3\sigma_1^2 q_{i,1}, 1 + c_3\sigma_2^2 q_{i,2}, \cdots, 1 + c_3\sigma_N^2 q_{i,N})$$
$$i = 1,2,\cdots,M$$

where σ_j^2, $j = 1,2,\cdots,N$, is the noise power of the jth dimension, $c_3 > 0$ is the proportional factor which controls the particle spreading in each dimension, $\mathbf{q}_i = [q_{i,1}, q_{i,2}, \cdots, q_{i,N}]$ is a random vector with $q_{i,j}$ uniformly drawn from $[0,1]$.

The implementation of the proposed method is illustrated by the following pseudo code:

```
For each p_i
    Initialize p_i(0) using (6)
    Initialize v_i(0) with zeros
    Initialize p_i^best(0) with p_i(0)
End
t = 1
Do
   For each p_i(t)
        Calculate the fitness value f(p_i(t)) with (4)
        If f(p_i(t)) < f(p_i^best(t))
            p_i^best(t) = p_i(t)
        End
   End
   Choose the p_i with smallest f(p_i(t)) as G^best(t)
    For each p_i(t)
        Update v_i and p_i with (2)
    End
   t = t + 1
while t < Iter
```

where the number of iterations, denoted by $Iter$, is chosen sufficiently large so that convergence is reached.

4. SOURCE LOCALIZATION WITH UNKNOWN PROPAGATION SPEED

This Section discusses about a more general case in which the propagation speed is also unknown, in addition to the source location. Based on (4), the corresponding cost function to be minimized is then

$$f(c,\mathbf{x}) = \sum_{k=2}^{L} \left(cD_{k,1} - \|\mathbf{x}_k - \mathbf{x}\|_2 + \|\mathbf{x}_1 - \mathbf{x}\|_2\right)^2 \tag{7}$$

Similar to Section III, the initial estimates are given by the LS solution [6]:

$$\hat{\theta}_2 = (\mathbf{A}_2^T\mathbf{A}_2)^{-1}\mathbf{A}_2^T\mathbf{b}_2 \tag{8}$$

where

$$\hat{\theta}_2 = \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{u} \\ \hat{v} \end{bmatrix}; \qquad \mathbf{A}_2 = \begin{bmatrix} 2(\mathbf{x}_2-\mathbf{x}_1)^T & 2D_{2,1} & D_{2,1}^2 \\ \vdots & \vdots & \vdots \\ 2(\mathbf{x}_L-\mathbf{x}_1)^T & 2D_{L,1} & D_{L,1}^2 \end{bmatrix}$$

and

$$\mathbf{b_2} = \begin{bmatrix} x_1^2+y_1^2-x_2^2-y_2^2 \\ \vdots \\ x_1^2+y_1^2-x_L^2-y_L^2 \end{bmatrix}$$

with $u = c\|\mathbf{x}_k-\mathbf{x}\|_2, v = c^2$, and $\hat{u}$ and $\hat{v}$ the estimates of u and v, respectively.

Similar to (6), the initial estimate of the source location $\hat{\mathbf{x}}$ and the initial estimate of the propagation speed $\hat{c} = \sqrt{\hat{v}}$ are employed as the initial guesses for PSO:

$$\mathbf{p}_i(0) = \mathbf{Q}_i[\hat{\mathbf{x}}^T \hat{c}]^T, \qquad i = 1,2,\cdots,M \tag{9}$$

where $\mathbf{Q}_i$ is the same as that in (6) except it has one more dimension.

The implementation flow for the unknown propagation speed case is the same as the the known propagation case, except that $\mathbf{p}_i^{best}$ and $\mathbf{G}^{best}$ have one more dimension and the speed c in the fitness function is replaced by its estimate, namely, the last element of $\mathbf{p}_i$. It is also worth noting that the cost function in (7) is set to infinity when the estimated propagation speed is negative, as the propagation speed cannot be less than zero.

5. SIMULATION RESULTS

Computer simulations have been conducted to evaluate the performance of the proposed method for source localization and speed estimation by comparing it with the LS solution of [6] as well as Cramér-Rao lower bound (CRLB) [1], [5]. The learning parameters c_1 and c_2 in (2), and proportional parameter c_3 in (6) are all set to 2. The inertia weight is set to 0.8 and all elements of $\mathbf{v}^{max} = [v_1^{max}, v_2^{max}, \cdots, v_N^{max}]$ are set to 5 [16] for fast convergence. Both the number of particles and iteration number $Iter$ are chosen to be 100. For simplicity, $\sigma_j^2,\ j = 1,2,\cdots,N$ in (6) are assumed to be identical.

The sensor-source geometry is shown in figure 1. Five sensors are placed at (200, 200)m, (100, 300)m, (300, 300)m, (100, 100)m, and (300, 100)m while the unknown source is located at (110, 130)m. The propagation speed is set to 360ms^{-1} . The noise-free TDOA's are added by correlated Gaussian noises with covariance matrix given by $\mathbf{Q}$, and equal noise power is assigned for simplicity. Thus, the diagonal elements of $\mathbf{Q}$ are two times of the noise power, and all other elements equal to the noise power. All simulation results are averages of 1000 independent runs.

Mean square error (MSE) is used as the performance measure of the source location and speed estimation. The MSE of the location estimate is defined as $E\left\{(x-\hat{x})^2+(y-\hat{y})^2\right\}$, where $\hat{x}$ and $\hat{y}$ denote the estimates of x and y, respectively. The MSE of the speed estimate is defined as $E\left\{(c-\hat{c})^2\right\}$ where $\hat{c}$ is the estimate of c.

Figures 2 compares the positioning accuracy of the PSO method with the LS scheme and CRLB for the known propagation speed case. It can be seen that the MSE performance of the proposed method outperforms the LS method by approximately 5 dB and approaches the CRLB. For the unknown propagation speed case, Figures 3 and 4 show that the MSEs of the proposed method are very close to the CRLB and is superior to the LS algorithm for both location and speed estimation.

6. CONCLUSION

Particle swarm optimization (PSO) method is employed for source localization in the cases of known and unknown propagation speed, using time-difference-of-arrival (TDOA) measurements. To guarantee fast convergence, least squares (LS) method is utilized to provide initial estimates of the parameters. For sufficiently small noise conditions, it is shown that the accuracy of the proposed method approaches Cramér-Rao lower bound and outperforms the LS method. This paper aims to provide a simple and efficient approach for dealing with highly non-linear problems. Our future work includes further exploration of solutions for such problems. For example, ant colony optimization, another type of swarming technique, will be studied.

Acknowledgement

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 119606].

REFERENCES

[1] Y.T. Chan and K.C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Trans. Signal Processing*, vol.42, pp.1905-1915, Aug. 1994

[2] K.W. Cheung, H.C. So, Y.T. Chan and W.-K. Ma, "A constrained least squares approach to mobile positioning: algorithms and optimality," *EURASIP Journal on Applied Signal Processing*, vol.2006, Article ID 20858, pp.1-23, 2006

[3] J.O. Smith and J.S. Abel, "Closed-form least-squares source location estimation from range-difference measurements," *IEEE Trans. Acoust., Speech, Signal Processing*, vol.35, pp.1661-1669, Dec. 1987

[4] B. Friedlander, "A passive localization algorithm and its accuracy analysis," *IEEE J. Ocean. Eng.*, vol.12, pp.234-245, Jan. 1987

[5] C.W. Reed, R. Hudson, and K. Yao, "Direct joint source localization and propagation speed estimation," *in Proc. ICASSP'99*, vol.3, pp.1169-1172, Mar. 1999

[6] A. Mahajan and M. Walworth, "3-D position sensing using the differences in the time-of-flights from a wave source to various receivers," *IEEE Trans. Robotics Automation*, vol.17, pp.91-94, Feb. 2001

[7] M.N. Neol, P.P. Joshi and T.C. Jannet, "Improved maximum likelihood estimation of target position in wireless sensor networks using particle swarm optimization," *in Proceedings of the Third International Conference on Information Technology: New Generations*, pp.274-279, April 2006

[8] R. Parisi, P. Croene and A. Uncini, "Particle swarm localization of acoustic sources in the presence of reverberation," *IEEE International Symposium on Circuits and Systems*, pp.4739-4742, May. 2006

[9] J. Kennedy and R.C. Eberhan, "Particle swarm optimization," *IEEE the lntemational Conference on Neural Networks*, vol.4, 1942-1948, 1995

[10] R.C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proceedings of the Sixth International Symposium on Micro Machines and Human Science*, pp.39-43, Oct. 1995

[11] Y. Shi and R.C. Eberhart, "A modified particle swarm optimizer," *IEEE International Conference on Evolutionary Computation*, pp.69-73, 1998

[12] R.C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," *Congress on Evolutionary Computation*, pp.81-86, 2001, Hawaii

[13] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, "Optimization by simulated annealing," *Science*, vol.220, pp.671-680, 1983

[14] R.C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," *The 7th Annual Conference on Evolutionary Programming*, pp.611-615, 1998

[15] D.J. Krusienski and W.K. Jenkins, "Particle swarm optimization for adaptive IIR filter structures," *Congress on Evolutionary Computation*, vol.1, pp.965-970, June 2004

[16] Y. Shi and R.C. Eberhart, "Parameter selection in particle swarm optimization," *Evolutionary Programming VI*, New York: Springer-Verlag, pp.591-600, 1998

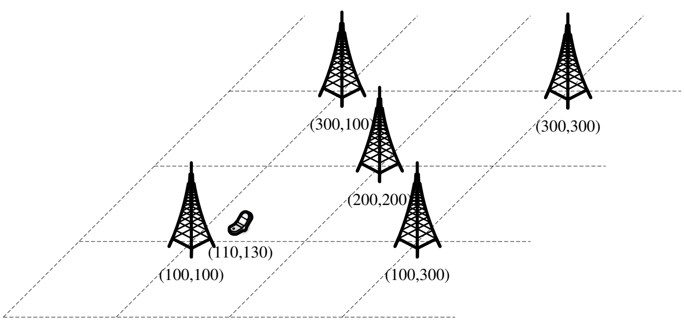

Figure 1: Sensor-source geometry

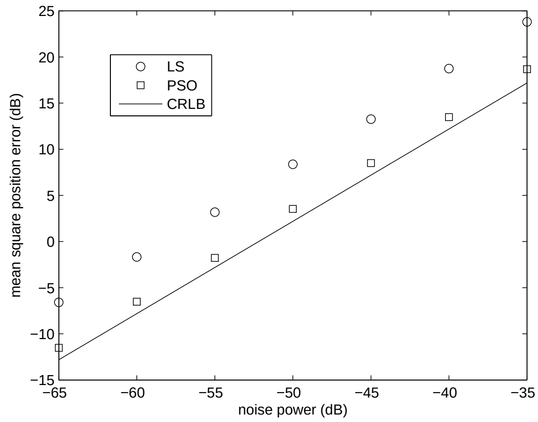

Figure 2: Mean square position error with known propagation speed

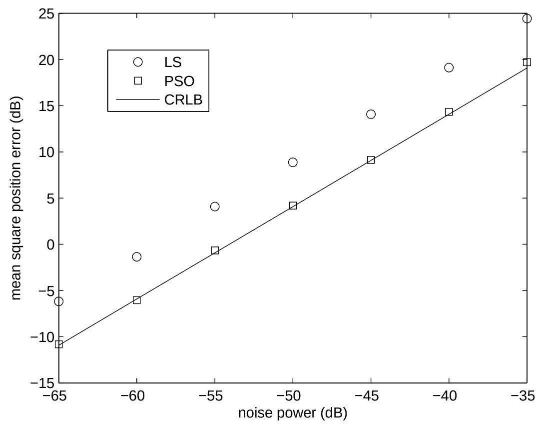

Figure 3: Mean square position error with unknown propagation speed

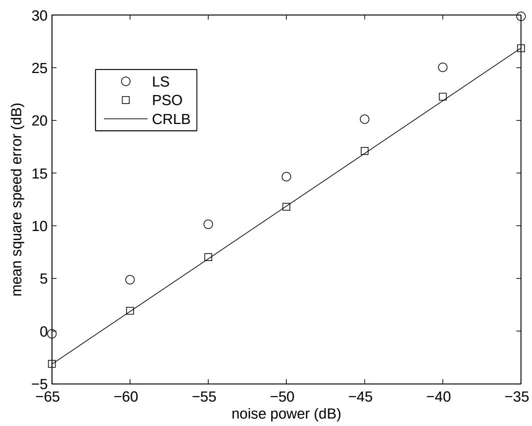

Figure 4: Mean square speed error