Population Density Particle Swarm Optimized Improved Multi-robot Cooperative Localization Algorithm

Mengchu TIAN[1], Yuming BO[1], Zhimin CHEN*[2], Panlong WU[1], Cong YUE[1]
(1. School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China;
2. China Satellite Maritime Tracking and Controlling Department, Jiangyin, 214431, China)
Email: 13675133070@188.com

Abstract: In light of the accuracy of particle swarm optimization-particle filter (PSO-PF) inadequate for multi-robot cooperative positioning, the paper presents population density particle swarm optimization-particle filter (PDPSO-PF), which draws cooperative co- evolutionary algorithm in ecology into particle swarm optimization. By taking full account of the competitive relationship between the environment and particle swarm, through dynamic adjustment of particle swarm densities based on Lotka-Volterra competition equations, PDPSO-PF improves particle diversities, speeds up the evolution of the algorithm and enhances the effectiveness of prediction for multi-robot positioning. Studies show that PDPSO-PF improves both the convergence speed and accuracy, thus is suitable for multi-robot cooperative positioning.

Keywords： Particle swarm optimization, particle filter, multi-robot, population density, cooperative localization

1 Introduction

To facilitate the path planning, navigation, decision-making and implementation of tasks, the robots are required to sense the environment in advancement and make accurate self-positioning [1]. Compared to a single robot [2], multi-robot cooperative positioning system enjoys high efficiency, high-precision, high fault tolerance, and high robustness [3−5], etc for multi-robot cooperative operations, thus is more suitable for a variety of complex tasks. Dieter [6], by application of Markov positioning method, proposed that for detections between one robot and another, robot motions, situational awareness information, and mutual observation models could be taken into account for adjusting positions of each robot, so as to achieve multi-robot mutual orientation. However as with no regard to the interdependence of detection information, overly optimistic estimates on positions might be resulted. Wang Ling [7], by combination of particle filter [8] and extended Kalman filter (EKF), brought up a multi-robot cooperative positioning method, which by taking account of the robustness and adaptability of particle filter as well as the high efficiency and real-time of EKF, makes the robot group members share the overall positioning information, thus effectively determining self-positions in an unknown environment in spite of no accuracies.

This paper presents an algorithm of population density particle swarm optimization-particle filter, which by taking full advantage of the accurate positioning capacity among the heterogeneous multi-robots and by adoption of population densities in to the particle swarm optimization-particle filter method, with as well as the dynamic adjustment of particle densities, can improve both the positioning accuracy and speed of robots.

The remaining part of the paper is organized as follows: Section 2 presents the information on relative observations and Section 3 provides an analysis for population density particle swarm optimization-particle filter. Comparative experiments and results are shown and discussed in Section 4. Finally, Section 5 is a conclusion of this research.

2 Access to Information on Relative Observations

In order to achieve mutual positioning of heterogeneous robots, the following assumptions are taken herewith [9]:

(1) As each robot is equipped with motion sensors, thus information can be exchanged between one robot and another.

(2) Heterogeneous multi-robots consist with high-level robots with great abilities. Such robots are equipped with external sensors that can measure the relative distances and directions of other robots, which makes it possible for accurate self-positioning; while low-level robots are configured with external sensors that can measure the relative locations.

(3) Each robot can detect the other ones with accurate identifications. Observation data can be accessed through three stages: firstly, the motions of each robot can be sensed through internal sensors based on their motion models, including moving distances and directions; next, high-level robots can directly obtain the relative positions of other nearby

accessible robots and convert these positions into their own local coordinates for calculations of relative observations; finally, the high-level robots convey the observant information on relative positions to other robots which, through processing such information, thereby calculate the distances and directions of all observant robots and robots inaccessible relative to themselves.

Assuming a queue of a number of robots that are engaged in explorations [10], and there is at least one robot can obtain accurate localizations. Use $\boldsymbol{R}=(x,y,\theta)^{\mathbf{T}}$ to represent positions of robots at some point, and positions of all robots at some point can be expressed as $(R_A,R_{B1},\cdots,R_{Bn})^{\mathbf{T}}$. The relative positions between one robot and another can be denoted as:

$$d_{R_A}^{R_{Bn}}=\sqrt{(x_{R_A}-x_{R_{Bn}})^2+(y_{R_A}-y_{R_{Bn}})^2}$$
(1)

$$\theta_{R_A}^{R_{Bn}}=\arctan\left(\frac{y_{R_{Bn}}-y_{R_A}}{x_{R_{Bn}}-x_{R_A}}\right)-\theta_{R_A}$$
(2)

Whereas, R_A and R_{Bn} represent positions of high-level and lowly-configured robots at a given moment, while $d_{R_A}^{R_{Bn}}$ indicates the relative distance from R_{Bn} to R_A; θ_{R_A} is the direction angle of R_A; $\theta_{R_A}^{R_{Bn}}$ means the direction angle of R_{Bn}.

3 PDPSO-PF

3.1 Dynamic Adjustment of Population Densities

Given a population P, the secular equation between population growth and environment as described in ecology goes as the following [11]:

$$\frac{dN}{dt}=rN\left[\frac{K-N}{N}\right]$$
(3)

Whereas, K means environmental load; r represents individual growth rate for the population; N indicates the population size; $\frac{K-N}{N}$ is the logistic coefficient.

As can be seen from the formula, logistic coefficient acts on the variation of population densities, to the effect of population densities tending to environment load. If $N>K$, logistic coefficient is negative, the population density decreases; while if $N<K$, then the logistic coefficient is positive, thus population density increases; in case $N=K$, logistic coefficient is 0, then population density remains unchanged.

3.2 Population Density Particle Swarm Optimization-particle Filter (PDPSO-PF)

Given a group composed with N particles, and the Particle i is expressed with an m-dimension vector $x_i(i=1,2,\cdots,N)$, so is the flying speeds [12–13] of the Particle i which are noted as $v_i(i=1,2,\cdots,N)$, then the updated formula of PDPSO-PF particles are as follows:

$$v_i(t+1)=wv_i(t)+c_1r_1(p_i(t)-x_i(t)) \\ +c_2r_2(p_g(t)-x_i(t))$$
(4)

$$x_i(t+1)=x_i(t)+v_i(t+1)$$
(5)

Whereas: $p_i(t)\,(i=1,2,\cdots,N)$ is the optimal location that is accessible currently for Particle i, which is expressed as:

$$p_i(t+1)=\begin{cases}p_i(t), & f(x_i(t+1))<f(p_i(t))\\ x_i(t+1), & f(x_i(t+1))\geq f(p_i(t))\end{cases}$$
(6)

$p_g(t)$ is the optimal position of the whole particle group, i.e.

$$p_g(t)\in\{p_1(t),p_2(t),\cdots,p_N(t)\mid f(p_g(t)) \\ =\max\{f(p_1(t)),f(p_2(t)),\cdots,f(p_N(t))\}\}$$
(7)

Whereas, w is inertia factor; c_1 and c_2 are non-negative constants; while r_1 and r_2 are random numbers ranging within $[0,1]$.

Given n particle swarms $P_i(i=1,2,\cdots,n)$ with $N_j(i=1,2,\cdots,n)$, PDPSO-PF undergoes evolution and cooperative processes in all iterations. Particle swarm optimization algorithm [14] is adopted for evolution process; while species population density equation shall be applied for calculations of population densities during cooperative process, and subsequently sizes of each particle swarm can be adjusted based on the calculated particle swarm densities, i.e.

$$N_i(t+1)=N_i(t)+\frac{dN_i}{dt},(i=1,2,\cdots,n)$$
(8)

In case that the growth rate $\frac{dN_i}{dt}$ of particle swarm $P_i(i=1,2,\cdots,n)$ is positive, then through random generation of $\frac{dN_i}{dt}$ particles, particle swarm $P_i(i=1,2,\cdots,n)$ is added to expand particle swarms.

In case that the growth rate $\frac{dN_i}{dt}$ of particle swarm $P_i(i=1,2,\cdots,n)$ is negative, then particle adaptabilities of the particle swarm is calculated, followed by sorting based on adaptabilities; the $\frac{dN_i}{dt}$ particles with least adaptabilities are deleted, thus shrink the particle swarms.

As seen from the foregoing, if the density of certain particle swarm increases, at least one particle will be produced randomly and added into such particle swarm, which improves the diversity of such particle swarm, thus optimizes the overall layout of particles to certain extent. In case the density of particle swarms decreases, at least one particle with the least adaptability shall be removed. Such process not only indicates the cooperative competition among swarms in the evolution process of particle swarms, but also reflects the mutual competition process among particles within the particle swarm.

3.3 Calculation steps of PDPSO-PF

PDPSO-PF algorithm goes as follows:

(a): Determine the initial parameter values and obtain the relative observations from robots to other low-configured robots. At the point when $k=0$, select N particles $\{x_0^i, i=1,...,N\}$ from importance functions at the initial moment, the importance density function goes like formula (9):

$$x_k^i \sim q(x_k^i \mid x_{k-1}^i, z_k) = p(x_k^i \mid x_{k-1}^i) \tag{9}$$

Give the objective function as:

$$f = \exp\{-sqrt[(z_t - z_{tpred})] \cdot R_k^{-1} \cdot (z_t - z_{tpred})^{\mathrm{T}} / c_3\} \tag{10}$$

Whereas, R_k represents the measured noise covariance; c_3 is a constant; z_t means the observation information at t moment; z_{tpred} indicates predictive information at t moment.

(b): calculation of significant priority

$$w_t^i = w_{t-1}^i p(z_t \mid x_{t-1}^i) = w_{t-1}^i \frac{p(z_t \mid x_t^i) p(x_t^i \mid x_{t-1}^i)}{q(x_t^i \mid x_{t-1}^i, z_t)} = w_{t-1}^i p(z_t \mid x_t^i) \tag{11}$$

(c): calculation of growth rate $\frac{dN}{dt}$ of particle swarms

$$\frac{dN}{dt} = rN \frac{K - N - \sum_{k=1}^{n} a_k N}{K} \tag{12}$$

If $\frac{dN}{dt} > 0$, then the randomly generated $\frac{dN}{dt}$ particles are added into the particle swarm;

If $\frac{dN}{dt} < 0$, then calculate the adaptabilities of particles within the swarm and sorting them accordingly; remove $\frac{dN}{dt}$ particles with the least adaptabilities: $N(t+1) = N(t) + \frac{dN}{dt}$.

(d): application of iteration by adoption of particle swarm optimization formula:

$$v_t^i(t+1) = wv_t^i(t) + c_1 r_1 (p_t^i(t) - x_t^i(t)) + c_2 r_2 (p_t^i(t) - x_t^i(t)) \tag{13}$$

$$x_t^i(t+1) = x_t^i(t) + v_t^i(t+1) \tag{14}$$

$$p_t^i(t+1) = \begin{cases} p_t^i(t), & f(x_t^i(t+1)) < f(p_t^i(t)) \\ x_t^i(t+1), & f(x_t^i(t+1)) \geq f(p_t^i(t)) \end{cases} \tag{15}$$

$$p_g(t) \in \{p_1(t), p_2(t), \cdots, p_{Nj}(t) \mid f(p_g(t)) = \max\{f(p_1(t)), f(p_2(t)), \cdots, f(p_{Nj}(t))\}\} \tag{16}$$

$$p^* \in \{p_g^1(t), p_g^2(t), \cdots, p_g^n(t) \mid f(p^*) = \max\{f(p_g^1(t)), f(p_g^2(t)), \cdots, f(p_g^n(t))\}\} \tag{17}$$

(e): In case p^* meets iteration termination conditions, then the algorithm ends; otherwise go on with step (c).

(f): based on the predicted particle concentration of robots, the relative observations between each particle and other lowly-configured robots, calculate the difference between predictive measurement $\Delta d_{R_A}^{R_{Bn}}$ and actual observation $\Delta\theta_{R_A}^{R_{Bn}}$.

(g): calculate the priority properties of optimized particles and conduct normalization.

$$w_t^i = \frac{a}{\Delta d_{R_A}^{R_{Bn}}} + \frac{b}{\Delta\theta_{R_A}^{R_{Bn}}} \tag{18}$$

$$w_t^i = w_t^i / \sum_{i=1}^{N} w_t^i \tag{19}$$

(h): status output:

$$\tilde{x} = \sum_{i=1}^{N} w_t^i x_t^i \tag{20}$$

4 Simulation Experiment

The computer to do experiments has an i5-4200U CPU and 8G memory. The sensor data and the movement measures are collected by the robot with odometer and sonar sensor. To verify the effectiveness of PDPSO-PF in multi-robot positioning by combination of relative observation measurement, PF, PSO-PF and PDPSO-PF are adopted respectively in a $5m \times 5m$ lab environment for experiment observations. Parameters applied in the experiment are: $c_1 = 1.5$, $c_2 = 1.2$, $w = 0.7$, $m = 100$. Firstly, use robot A and robot B_1 and B_2 for uniform linear motions with initial orientation angle as 0°, see Figure 1 for the actual trajectory and the reference trajectory by application of PDPSO-PF. As for the orientation error of robot B_1, see Figure 3; with robots making uniform curve motions, see the actual estimate trajectory and reference trajectory as shown in Figure 2. Figure 4 indicates the positioning error of B_1. In the figure for trajectories, the Trajectory 1, 2 and 3 represent the trajectories of B_1, A and B_2 respectively.

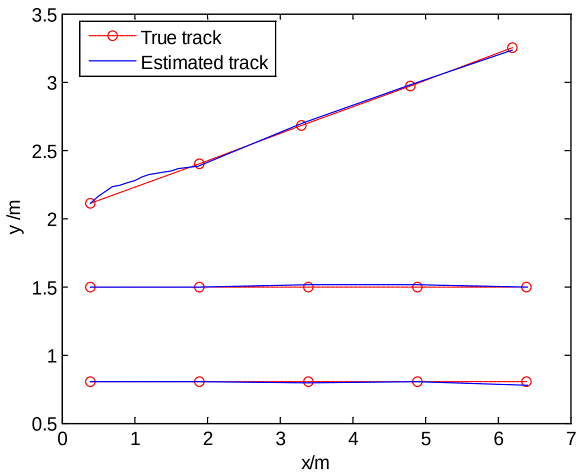

Figure 1. True track and estimated track of linear motion

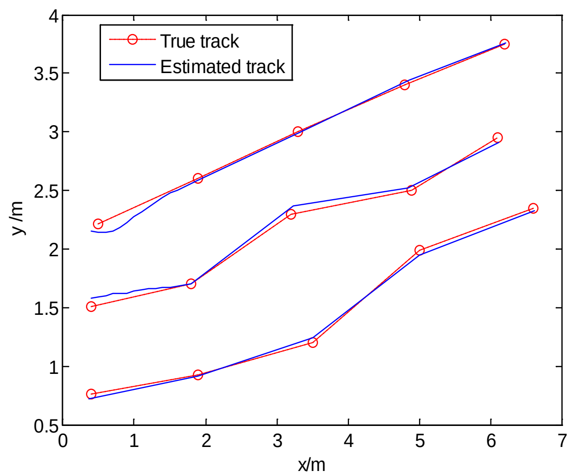

Figure 2. True track and estimated track of broken line motion

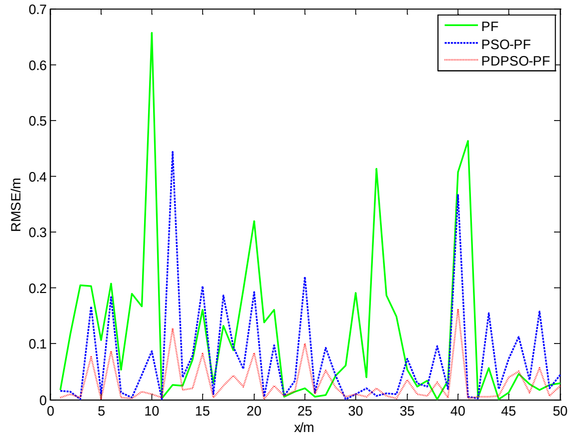

Figure 3. Localization R_{B1} -RMSE of linear motion

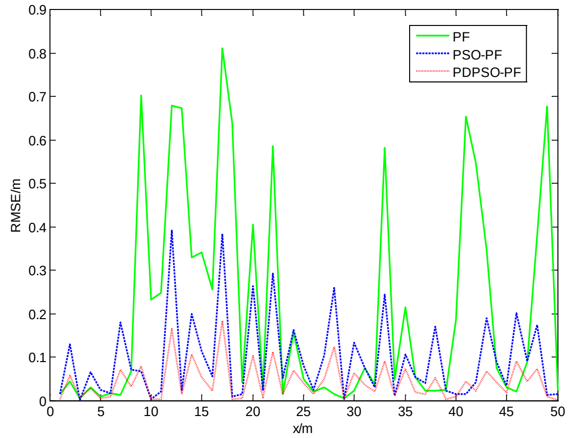

Figure 4. Localization R_{B1} -RMSE of broken line motion

Table 1. comparison of RMSE/m for different algorithm

motion mode	robot	PF	PSO-PF	PDPSO-PF
linear motion	A	0.022	0.014	0.012
	B_1	0.045	0.025	0.019
	B_2	0.051	0.040	0.031
broken line motion	A	0.023	0.022	0.018
	B_1	0.056	0.044	0.036

	B_2	0.077	0.051	0.044

The experimental results indicate that the positioning accuracy of high-level robot A is higher than that of robot B. When robot orientation angle changes, the actual estimated trajectory would deviate from the reference trajectory, thus localization estimate error would be increased; but when the orientation angle keeps unchanged, the actual positions of robots would be immediately converged, thus positioning error decreases. Compared to PSO-PF and PF, the adoption of robot corporative localization can significantly increases the positioning accuracy with less errors and better agreement with tracking trajectories, thus indicating that the algorithm presented in this paper can be well applied in robot cooperative positioning.

5 Conclusions

Through an analysis of the deficiencies of PF and PSO-PF in multi-robot cooperative localizations, the paper presents a new algorithm—population density particle swarm optimization-particle filter, which in addition to adopting individual adaptability control evolution of particle swarm, also draws population densities into particle filter, thus improves the overall optimization capabilities and speeds up the evolution of algorithm, thereby a highly-accurate positioning system comes into being. The final experimental data have verified the effectiveness of PDPSO-PF in multi-robot positioning system.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China (61501521); National Natural Science Foundation of China (U1330133); National Natural Science Foundation of China (61473153); National Natural Science Foundation of China (61403421); National Natural Science Foundation of China (61203266); Key Defense Advanced Research Project of China (40405070102). Thanks for the help.

References

[1] S. Wang, Q. Zhu, Z. Lui, et al. Study of Reinforcement Learning based on Multi-agent Robot Systems. *Journal of Computational Information Systems*, 2007,3 (5), pp: 2001- 2006.

[2] M. B. Dias, R. Zlot, N. Kalra and A. Stentz. Market-based multi-robot coordination: a survey and analysis. *Proceedings of the IEEE*, 2006, 94(7), pp:1257-1270.

[3] A. T. Rashid, M. Frasca M, A. A. Ali, et al. Multi-robot localization and orientation estimation using robotic cluster matching algorithm. *Robotics & Autonomous Systems*, 2015, 63, pp: 108–121.

[4] De Silva, Oscar, G. K. /. Mann, R. G. Gosine. An Ultrasonic and Vision-Based Relative Positioning Sensor for Multirobot Localization. *Sensors Journal IEEE*. 2015, 15(3), pp: 1716-1726.

[5] J. Pugh and A. Martinoli. Distributed adaptation in multi-robot search using particle swarm optimization. *The 10th International Conference on the Simulation of Adaptive Behavior*, 2008, pp:393-402.

[6] M. Anderson, and N. Papanikolopoulos. Implicit cooperation strategies for multi-robot search of unknown areas. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2008, 53(4), pp:381-397.

[7] Dieter F, Wolfram B, Hannes K, et al. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 2006, 8(3), pp：325−344.

[8] L. Yu, M. Wu, Z. Cai, et al. A particle filter and SVM integration framework for fault-proneness prediction in robot dead reckoning system. *Wseas Transactions on Systems*, 2011, 10(11), pp.363-375.

[9] Wang Lin, Shao Jin-xin, Wan Jian-wei, et a1. Distilbuted entropy based relative observation selection for multi-robot localization. *Acta Electronic Sinica*, 2007, 35(2), pp：333−337.

[10] M. Anderson, and N. Papanikolopoulos. Implicit cooperation strategies for multi-robot search of unknown areas. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2008,53(4), pp:381-397, 2008.

[11] P. Dasgupta. A multi agent swarming system for distributed automatic target recognition using unmanned aerial vehicles. *IEEE Transaction on Systems, Man, and Cybernetics-Part A: System and Humans*, 2008, 38(3), pp:549-563.

[12] Z. M. Chen, Y. M. Bo, P. L. Wu, et al. A new particle filter based on organizational adjustment particle swarm optimization. *Applied Mathematics & Information Science*, 2013, 7(1), pp: 179-186.

[13] Y. Ji, R. Wang, Study on PSO algorithm in solving grid task scheduling. *Journal on Communications*, 2007,10(28), pp:60-66.

[14] Y. Li, B. Bai, Y. Zhang. Improved particle swarm optimization algorithm for fuzzy multi-class SVM. *Journal of Systems Engineering and Electronics*, 2010, 21(3), pp. 509-513.

List of abbreviation

$\boldsymbol{R}$	positions of robots
R_A	positions of high-level robots
R_{Bn}	positions of lowly-configured robot
$d_{R_A}^{R_{Bn}}$	relative distance
θ_{R_A}	direction angle of R_A
$\theta_{R_A}^{R_{Bn}}$	direction angle of R_{Bn}.
$\theta_{R_A}^{R_{Bn}}$	direction angle of R_{Bn}.
K	environmental load
r	individual growth rate
N	population size
x_i	position of particle
v_i	flying speed
$p_i(t)$	optimal location of particle i
$p_g(t)$	optimal position of the whole particle group
w	inertia factor
c_1	study factor
c_2	study factor
R_k	measured noise covariance
z_t	observation information
z_{tpred}	predictive information