Dynamic Population Adaptive Particle Swarm Optimized Particle Filter for Integrated Navigation

Zhimin CHEN*, Yuming BO, Yuanxin QU, Xiaodong LING, Xiaohong TAO, Yong LIU
(1. China Satellite Maritime Tracking and Controlling Department, Jiangyin, 214431, China)

Department (China Satellite Maritime Tracking and Controlling Departmen)
Address (Jiangyin, hongqiaobeilu#243, 214431)
China
chenzhimin@188.com

Abstract: - Particle filter based on particle swarm optimization algorithm (PSO-PF) is not precise and trapping in local optimum easily, it is not able to satisfy the requirement of advanced integrated navigation system. In order to solve these problems, a novel particle filter algorithm based on dynamic neighborhood population adaptive particle swarm optimization (DPSO-PF) is presented in this paper. This new particle filter can dynamically adjust the particle neighborhood environment, wherein each particle can adjust the number of particles in the neighborhood based on self-adaptation basis according to the neighborhood environment and their own position information, accordingly a best balance is achieved between optimal seeking and convergence rate. Finally different models are used for simulation experiment and the results indicate that this new algorithm improves the precision of GPS/INS integrated navigation system.

Key-Words: dynamic, particle filter, integrated navigation, neighborhood, population

1 Introduction

With global positioning system (GPS) position information or velocity information used for periodic calibration by inertial navigation system (INS), the performance of integrated navigation will be superior to that of any sub-system applied independently [1] . GPS/INS integrated navigation system has upgraded the overall performance of navigation system significantly [2] . Provided GPS receiver is capable of receiving the information sent by at least four satellites, GPS will be able to offer solution position information, and INS will measure the position and attitude information of the aircraft by means of angular rate sensor and linear acceleration. It is proved that integrated navigation can conquer respective deficiencies to deliver more accurate and reliable navigation information.

In the indirect estimation on the integrated navigation, navigation parameter error equation [3] constitutes the main part of system's state equation. However, considering that there is a small error, the rules of navigation parameter error can be described by the classical Kalman filtering and first-order linear approximation equation in case of the requirement for low accuracy, and its model error will not be high.

Particularly, however, the military field has made an increasingly higher demand [4] on the accuracy of integrated navigation in recent years, because of which the model error resulting from the use of low-order approximation cannot be ignored any more. Meanwhile, given that the system noise and measurement noise may be the non-Gaussian noises, particle filter can be applied to non-linear system effectively in the environment of non-Gaussian noise effectively since conventional Kalman filtering is prone to divergence, but the existence of particle filter [5] will result in sample degeneration and impoverishment and then exercise a severe influence on its estimation performance.

Particle swarm optimized particle filter (PSO-PF) is an intelligent optimized particle filter. PSO-PF introduces the latest measurement into optimization process, and meanwhile optimizes the sampling process to keep updating the particle velocity in real time so that sampling distribution will switch to the regions with a high posterior probability [6] . PSO-PF has improved not only the weight degradation, but also the accuracy of particle filter to a certain degree. Nevertheless, the neighborhood population of the particles in this algorithm is fixed, unable to make full use of the information of the particles in the neighborhood, and the excessive iterations in the optimization process have led to the excessive computing complexity. Consequently, there is a difficulty in meeting the requirement for real-time performance and accuracy of integrated navigation system[7].

The improved PSO-PF proposed herein carries out self-adaptive adjustments over particle neighborhood and number by means of neighborhood population extension and restriction factors according to the diverse changes in particle swarm. In this way, the algorithm can maintain not only high local search ability, but also the diversity of samples to heighten the locating accuracy of integrated navigation system.

2. Particle filter

Particle filter (PF) is an approximate calculation of Bayes estimation based on sampling theory. PF follows the basic thought that to gain random samples for approximation of posterior probability density[8]. Assuming the nonlinear dynamic process is expressed as equation (1) and equation (2):

$$x_k = f(x_{k-1}, v_{k-1}) \tag{1}$$

$$y_k = h(x_k, n_k) \tag{2}$$

If the initial probability density of the state is $p(x_0 \mid y_0) = p(x_0)$, then the state predictive value is:

$$p(x_k \mid y_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid y_{1:k-1}) dx_{k-1} \tag{3}$$

The state renewal equation is:

$$p(x_k \mid y_{1:k}) = \frac{p(y_k \mid x_k) p(x_k \mid y_{1:k-1})}{p(y_k \mid y_{1:k-1})} \tag{4}$$

Where

$$p(y_k \mid y_{1:k-1}) = \int p(y_k \mid x_k) p(x_k \mid y_{1:k-1}) dx_k \tag{5}$$

Importance function $q(x_{0:k} \mid y_{1:k})$ is:

$$q(x_{0:k} \mid y_{1:k}) = q(x_0) \prod_{j=1}^{k} q(x_j \mid x_{0:j-1}, y_{1:j}) \tag{6}$$

The weight of particles is

$$\begin{aligned} w_k &= \frac{p(y_{1:k} \mid x_{0:k}) p(x_{0:k})}{q(x_k \mid x_{0:k-1}, y_{1:k}) q(x_{0:k-1}, y_{1:k})} \\ &= w_{k-1} \frac{p(y_k \mid x_k) p(x_k \mid x_{k-1})}{q(x_k \mid x_{0:k-1}, y_{1:k})} \end{aligned}$$

(7)

The probability density is:

$$p(x_{k-1} \mid y_{1:k-1}) = \sum_{i=1}^{N} w_{k-1}^i \delta(x_{k-1} - x_{k-1}^i) \tag{8}$$

And the weight is:

$$w_k^i = w_{k-1}^i \frac{p(y_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, y_k)} \tag{9}$$

3. PSO-PF algorithm

The importance sampling process of conventional PF is suboptimal, whereas the incorporation of PSO algorithm will optimize the sampling process of PF, allow the weight of particle sets are more inclined to high likelihood region[9], accordingly solving the problem of particle impoverishment, and conducing to reduction of particle numbers required by PF. PSO method is fused with PF and the key lies in utilizing the optimal state value pb experienced by the particles and the state value pg of the maximum particle with the greatest objective function value, and updating the speed and position of each particle on a real-time base through equation (10) and (11), accordingly forcing the particles to be closer to the real state.

$$\begin{aligned} V_k^i &= |Rand\ n| \times (pb - X_{k-1}^i) \\ &\quad + |rand\ n| \times (pg - X_{k-1}^i) \end{aligned} \tag{10}$$

$$X_k^i = X_{k-1}^i + V_{k-1}^i \tag{11}$$

Where $|Rand\ n|$ and $|rand\ n|$ are positive Gaussian distribution random numbers.

4. Building of Integrated Navigation Model

4.1 State and Measurement Equations

The application of filter to integrated navigation is ultimately intended for a more accurate parameter [10], and the selection of filter state normally resorts to indirect process, i.e., the error $ÄX$ of the navigation parameter outputted by navigation system is taken as filter's estimated state. While indirect process is used for state estimation, the state of filter will be the combination of errors in system, without participating in the computing processes of the navigation parameters in GPS/INS integrated navigation system. Therefore, the state estimation of filter is independent of the computation, and INS can still have its superiority of high update frequency fully revealed[11].

Assuming that the combination method of GPS/INS integrated navigation system relies on the combination between velocity and attitude, GPS/INS integrated navigation system's measurement values can be divided into two values, Difference value of position measurement value and that of velocity measurement value. Difference value of velocity measurement value indicates that the difference between the information rendered by

INS and GPS receiver is worked out as another method of measurement.

The error state of GPS/INS integrated navigation system is listed as equation (12):

$$\boldsymbol{X}(t) = \boldsymbol{F}(t)\boldsymbol{X}(t) + \boldsymbol{G}(t)\boldsymbol{W}(t) \tag{12}$$

The position measurement of INS can be expressed as equation (13):

$$\begin{bmatrix} L_I \\ \lambda_I \\ h_I \end{bmatrix} = \begin{bmatrix} L_t + \delta L \\ \lambda_t + \delta\lambda \\ h_t + \delta h \end{bmatrix} \tag{13}$$

The position measurement information of global positioning system receiver can be expressed as equation (14):

$$\begin{bmatrix} L_G \\ \lambda_G \\ h_G \end{bmatrix} = \begin{bmatrix} L_t - \dfrac{N_N}{R_M} \\ \lambda_t - \dfrac{N_E}{R_{N\cos L}} \\ h_t - N_h \end{bmatrix} \tag{14}$$

Where λ_t, L_t, and h_t stand for actual location, and N_E, N_N, and N_U for the errors of global positioning system receiver in the eastward, northward and skyward directions.

The position measurement value vector is defined as equation (15):

$$Z_p(t) = \begin{bmatrix} (L_I - L_G)R_M \\ (\lambda_I - \lambda_G)R_N \cos L \\ h_I - h_g \end{bmatrix} \equiv \boldsymbol{H}_p(t)\boldsymbol{X}(t) + \boldsymbol{V}_p(t) \tag{15}$$

Where,

$$\boldsymbol{H}_p = \begin{bmatrix} 0_{3\times6} & \vdots & diag[R_M \quad R_N \cos L \quad 1] & \vdots & 0_{3\times9} \end{bmatrix}_{3\times18}$$

$$\boldsymbol{V}_p = \begin{bmatrix} N_N & N_E & N_U \end{bmatrix}^T$$

The variances of measurement noise are σ_{pN}^2, σ_{pE}^2, and σ_{pU}^2.

$$\begin{cases} \sigma_{pN} = \sigma_p \cdot HDOP_N \\ \sigma_{pE} = \sigma_p \cdot HDOP_E \\ \sigma_{pU} = \sigma_p \cdot HDOP \end{cases} \tag{16}$$

Where, σ_p is the pseudo-range measurement error of global positioning system receiver.

The velocity measurement information of INS can be expressed as equation (17):

$$\begin{bmatrix} v_{IN} \\ v_{IE} \\ v_{IU} \end{bmatrix} = \begin{bmatrix} v_N + \delta v_N \\ v_E + \delta v_E \\ v_U + \delta v_U \end{bmatrix} \tag{17}$$

Where v_E, v_N, and v_U stand for true velocities along eastward, northward and skyward axes.

The velocity measurement information of GPS can be also expressed as equation (18):

$$\begin{bmatrix} v_{GN} \\ v_{GE} \\ v_{GU} \end{bmatrix} = \begin{bmatrix} v_N - M_N \\ v_E - M_E \\ v_v - M_U \end{bmatrix} \tag{18}$$

Where M_N, M_E, and M_U constitute the components of velocity measurement errors of global positioning system receiver along northward, eastward and skyward axes.

Below is the definition of velocity measurement vector:

$$Z_p(t) = \begin{bmatrix} v_{IN} - v_{GN} \\ v_{IE} - v_{GE} \\ v_{IU} - v_{GU} \end{bmatrix} \equiv \boldsymbol{H}_v(t)\boldsymbol{X}(t) + \boldsymbol{V}_v(t) \tag{19}$$

Where, $\boldsymbol{H}_v = \begin{bmatrix} 0_{3\times3} & \vdots & diag[1 \ 1 \ 1] & \vdots & 0_{3\times12} \end{bmatrix}$, $\boldsymbol{V}_p = \begin{bmatrix} M_N & M_E & M_U \end{bmatrix}^T$.

Supposing the measurement velocity of pseudo-range rate $\dot{\rho}$ of GPS receiver is $\sigma_{\dot{\rho}}^2$, the deviations of the eastward, northward and skyward velocity errors resulting from pseudo-range rate can be expressed as equation (20):

$$\begin{cases} \sigma_{vE} = HDOP_E \cdot \sigma_{\dot{\rho}} \\ \sigma_{vN} = HDOP_N \cdot \sigma_{\dot{\rho}} \\ \sigma_{vU} = VDOP \cdot \sigma_{\dot{\rho}} \end{cases} \tag{20}$$

The combination of position measurement can be expressed as equation (21):

$$\boldsymbol{Z}(t) = \begin{bmatrix} \boldsymbol{H}_p \\ \boldsymbol{H}_v \end{bmatrix} \boldsymbol{X}(t) + \begin{bmatrix} \boldsymbol{V}_p(t) \\ \boldsymbol{V}_v(t) \end{bmatrix} = \boldsymbol{H}(t)\boldsymbol{X}(t) + \boldsymbol{V}(t) \tag{21}$$

4.2 Discretization of State and Measurement Equations

Following result can be obtained through the discretization of equation (12) and equation (21):

$$\boldsymbol{X}_k = \boldsymbol{\Phi}_{k,k-1}\boldsymbol{X}_{k-1} + \boldsymbol{\Gamma}_{k-1}\boldsymbol{W}_{k-1} \tag{22}$$

$$\boldsymbol{Z}_k = \boldsymbol{H}_k\boldsymbol{X}_k + \boldsymbol{V}_k \tag{23}$$

Where, $\boldsymbol{\Phi}_{k,k-1} = \sum_{n=0}^{\infty} [F(t_n)T]^n / n!$, $\boldsymbol{\Gamma}_{k-1} = \left\{ \sum_{n=1}^{\infty} \frac{1}{n!} [F(t_k)T]^{n-1} \right\} G(t_k)T$.

As required by filter, the integrated navigation system and measurement noises and measurement equations can be equipped as follows:

$$E\left\{W(t)W^{\mathrm{T}}(\tau)\right\} = Q(t)\delta(t-\tau) \tag{24}$$

$$E\left\{V(t)V^{\mathrm{T}}(\tau)\right\} = R(t)\delta(t-\tau) \tag{25}$$

$$E\left\{W_k W_j^{\mathrm{T}}\right\} = Q_k \delta_{kj} \tag{26}$$

$$E\left\{V_k V_j^{\mathrm{T}}\right\} = R_k \delta_{kj} \tag{27}$$

Where, $\delta_{kj} = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases}, \begin{cases} Q_k = Q(t)/T \\ R_k = R(t)/T \end{cases}$.

5. DPSO-PF Algorithm

Since the neighborhood particles are constant [12–13], particle swarm optimized particle filter cannot adjust and control the states of neighborhood population according to the specific situation of every particle. Under this situation, not only will the filter accuracy and velocity be affected, but local extremum will be caused [14–15]. The improved algorithm brings in diversity factor[16] and presents a neighborhood population extension thought. With the help of population extension thought, filter can conduct self-adaptive control over the quantity of neighborhood population in accordance with the state of the particles, then helping the filter achieve the optimal balance between optimization ability and convergence velocity.

5.1 Diversity Factor

Diversity of particles proves to be one of the significant particle filter performance indexes [17]. Higher diversity implies better population quality and performance as well as lower degradation degree. In particle swarm optimized particle filter, however, the diversity also has an impact on the local search ability and speed. Although higher diversity means stronger global search ability, the local search ability and search speed will go down accordingly, and meanwhile, particle swarm optimized particle filter performance may decline as well if the diversity is improved blindly [18–19]. For this problem, diversity factor is introduced to control the increase or decrease in the number of neighborhood population so as to optimize the neighborhood population quantity.

Let particle number be M and the set for the fitness function value of the optimal position that particles pass by be $Y_k^i = \{Y_k^1, Y_k^2, Y_k^3, \cdots Y_k^M\}$, with $Y_{\min} = \min\{Y_k^1, Y_k^2, Y_k^3, \cdots Y_k^M\}$, $Y_{\max} = \max\{Y_k^1, Y_k^2, Y_k^3, \cdots Y_k^M\}$, and $[Y_{\min}, Y_{\max}]$ set as the observation interval; Then, $[Y_{\min}, Y_{\max}]$ is divided into M equally wide subintervals to calculate the number of Y_k^i 's particles in each subinterval, with $\sum_{i=1}^{M} e_i = M$, there is

$$p_i = \frac{e_i}{M} \tag{28}$$

$$D_t = -\sum_{i=1}^{M} p_i \log(p_i), \tag{29}$$

Where D_t is diversity factor.

D_t can reflect the diversity level of the particles in the improved PSO-PF. Supposing population fall into every subinterval, i.e., e_i =1, D_t can be greatest and the diversity level will be highest; conversely, if large quantities of population fall into only one interval, D_t will be smallest and population diversity level will be lowest.

D_t can control the quantity of neighborhood population. If the diversity factor is smaller than that in previous cycle, the population diversity will begin to decline, and meanwhile, it is necessary to decrease the number of neighborhood population particles with a view to preventing the local extremum of the improved PSO-PF. There is a need to increase the number of neighborhood population particles for the sake of a higher convergence velocity of improved PSO-PF. Then, the number of iterations will be reduced and the filtering efficiency can be improved.

5.2 Neighborhood Population Extension

5.2.1 Extension Factor

The neighborhood population will be extended properly since the extension of neighborhood population needs not only to ensure the higher extension probability of the particles with high fitness function value but also to prevent the excessively rapid extension. In view of this, definition is given as follows:

The fitness fitness function value is sequenced in an ascending order to gain array h, in which let *marki* be the sequence number of particle *i*'s fitness; C_i be the number of particle i 's neighborhood particles; and *neighbor (i)* be the set of particle i's neighborhood particles; under this circumstance, if there is

$$Extend(i) = \frac{mark_i C_i}{\frac{1}{C_i}\sum_{j \in neighbor(i)} (mark_j C_j)} \tag{30}$$

Then *Extend(i)* can called the extension factor of particle *i*.

Where $\frac{1}{C_i}\sum_{j \in neighbor(i)} (mark_j C_j)$ represents the neighborhood population extension state of particle i. A greater $mark_j C_j$ means that fitness is low. If *Extend(i)* <1, the improved particle filter algorithm proposed herein will lead to a higher extension probability of particle *i*, while if *Extend(i)* >1, it will be inappropriate to extend neighborhood population. *Extend(i)* is decided by the fitness level as well as number of neighborhood population and particle state.

5.2.2 Extension Thought of Neighborhood Particles

(1) Extension factor *Extend(i)* is decided by equation (30).

(2) Assuming that random number is $r \in [0, 1]$, if r <*Extend (i)*, a particle will be selected from the non-neighborhood population of particle *i* optionally as new neighborhood particle, and *pg* will be upgraded; otherwise, the neighborhood population will not be extended.

For the population with *Extend (i)* < 1, the smaller *Extend (i)* is, the greater the extension probability will be.

5.3 Neighborhood Population Restriction

5.3.1 Restriction Factor

Local optimum is a typical problem that particle swarm optimization and particle swarm optimized particle filter face. In particle population, the particles with high fitness have a huge impact on neighboring population, and the main reason is that the population with high fitness exercises an influence over neighborhood population, which will therefore lead to local optimum and then the decline in diversity. To lessen the influence of the particles with high fitness on neighboring population, some of the particles with high fitness value in neighborhood population will be deleted, but number of deletion is supposed to satisfy the constraints on particles. The definition is as follows:

Let S_i be the number of particles in particle *i*'s neighborhood population, and then the fitness of particle *i* and the particles in its neighborhood population is sequenced to get array *g*, in which let *lmarki* be the sequence number of particle *i* 's fitness value. If *lmarki* = $S_i + 1$, the fitness value of particle *i* will outshine that of all particles in its neighborhood population, while if *lmarki* = 1, the fitness value will be low. At this time, assuming there is

$$removal(i) = \frac{S_i - lmark_i + 1}{S_i} \tag{31}$$

Where *removal(i)* is restriction factor of particle *i*.

removal(i) has explained that the fitness of particle *i*'s neighborhood particles is lower than the proportion of particle *i*'s neighborhood particles. As the algorithm evolves and iterates, *removal(i)* is decided by with neighborhood population state.

5.3.2 Restriction Thought of Neighborhood Particle

This thought mitigates the influence of the population with high fitness value by restricting the largest neighborhood particle of *removal(i)* of the particles unsuitable for neighborhood population extension. Additionally, to ensure the successful implementation of improved algorithm, at least two neighborhood particles will be included in the self-adaptive neighborhood population.

(1) Calculating *Extend(i)*, the extension factor of particle *i*; provided *Extend(i)* ≥ 1, continue with next step; otherwise, abandon neighborhood population extension thought.

(2)Calculating *removal(q)* of particle *i*'s neighborhood particle and select particle *q* with largest *removal (q)*.

(3) If S_i>2 and S_q>2, the particle *q* will be deleted from and the removal should be updated; otherwise, the neighborhood population should be constant.

5.4 Steps for DPSO-PF improvement

The improved particle filter steps are showed as follows:

(1): When *k*=0, take N particles $\{x_{0:k}^i, i = 1,..., N\}$ as samples from importance function at the initial time. The importance density function is expressed in equation (32):

$$x_k^i \sim q(x_k^i \mid x_{k-1}^i, z_k) = p(x_k^i \mid x_{k-1}^i) \tag{32}$$

Giving the fitness function:

$$Y = \exp[-\frac{1}{2R_k}(z_{New} - z_{\Pr ed})] \tag{33}$$

(2): Calculating the importance value:

$$w_k^i = w_{k-1}^i p(z_k \mid x_{k-1}^i) = w_{k-1}^i \frac{p(z_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} = w_{k-1}^i p(z_k \mid x_k^i) \quad (34)$$

(3): Calculating the $V_k^{i_m}$, the velocity of $X_k^{i_m}$, after mth iterations according to equation (35), and move $X_k^{i_m}$ to the position $X_k^{i_{m+1}}$ of next iteration under the action of velocity $V_k^{i_{m+1}}$ according to (36):

$$V_k^{i_{m+1}} = |Rand\ n| \times (pb_{k-1}^i - X_{k-1}^{i_m}) + |rand\ n| \times (pg_k - X_{k-1}^{i_m}) \quad (35)$$

$$x_{k+1}^{i_{m+1}} = x_k^{i_m} + v_k^{i_{m+1}} \quad (36)$$

(4): Calculating the fitness value and update particle's *pb* and *pg*:

$$pb_k^i = \begin{cases} pb_k^i, & Y(x_g) < Y(pb_k^i) \\ x_g, & Y(x_g) > Y(pb_k^i) \end{cases} \quad (37)$$

$$pg_k \in \{x_k^1, x_k^2, x_k^3, \cdots x_k^N \mid Y(x)\} = \max\{Y(x_k^1), Y(x_k^2), Y(x_k^3), \cdots Y(x_k^N)\} \quad (38)$$

Where *Y(pb)* is denoted by *G*.

(5): Calculating diversity factor according to equation (29);

(6): Figuring out array and mark by sequencing the fitness of particles in particle population in ascending order;

(7): Deciding whether to extend or restrict particles neighborhood population in the light of *Dt* and the optimal particle value *Gt*; if *Dt* < *Dt*-1 and *Gt*=*Gt*-1, follow step (9); otherwise, follow step (8)

(8): Expanding the particle neighborhood population of every particle in accordance with neighborhood population extension thought and turns to step (11);

(9): Sequencing the fitness value of particle *i* and its neighborhood particles to work out *lmarki*;

(10): Updating the neighborhood population structure in the light of neighborhood population restriction thought.

(11): When the optimal value of particle complies with the threshold value ε, it is indicated that the particles population have been already distributed around the high likelihood area. By now particle optimization should be stopped, and execute step (3).

(12): Calculating the importance weight of the optimized particles and perform normalization.

$$w_k^i = w_k^i / \sum_{i=1}^N w_k^i \quad (39)$$

(13): State output:

$$\tilde{x} = \sum_{i=1}^N w_k^i x_k^i \quad (40)$$

6. Experimental simulation

6.1 Simulation test of basic algorithm performance

Choosing the univariate nonstationary growth model, and the process function and measurement function of the simulated objects are llisted as follows:

$$x(t) = 0.5x(t-1) + \frac{25x(t-1)}{1+[x(t-1)]^2} + 8\cos[1.2(t-1)] + w(t) \quad (41)$$

$$z(t) = \frac{x(t)^2}{20} + v(t) \quad (42)$$

In which, $w(t)$ and $v(t)$ are zero-mean Gaussian noise. Since this system is highly non-linear and the likelihood function presents bimodal [20].

The particle upgrade process of PSO-PF turns out to be an iterative optimization process. Generally speaking, iteration will stop in two situations, i.e., particle's optimal value meets the preset requirement or the number of iteration reaches the preset highest number of iterations. It is proved that the optimization of particle filter based on high-performance particle swarm has a small number of iterations but high real-time performance.

By using PF 、PSO-PF 、DPSO-PF, state estimation and tracking of this non-linear system are performed, and the formula of root-mean-square error is

$$RMSE = [\frac{1}{T}\sum_{t=1}^T (x_t - \hat{x}_t)^2]^{1/2} \quad (43)$$

(1) Setting the particle number $N = 100$, and process noise variance Q = 10, measurement noise variance R=1, the simulation result is presented in figure 1 and figure 2, Setting $N = 100$, $Q = 20$, R=1, the simulation result is presented in figure 3 and figure 4., the result is given in Table 1.The results are given in Table 1.

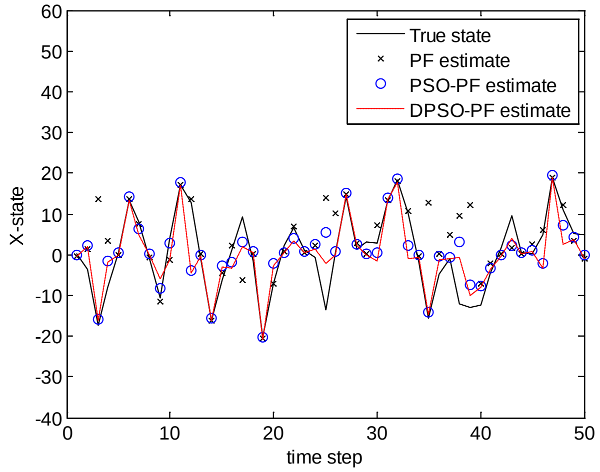

Fig.1 State estimation of different algorithm（Q=10）

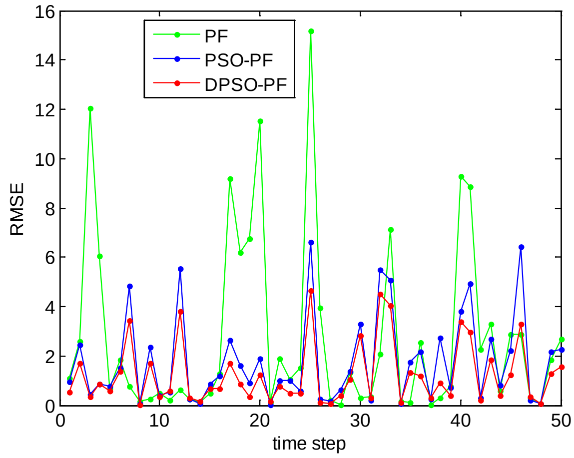

Fig.2 RMSE of different algorithm（Q=10）

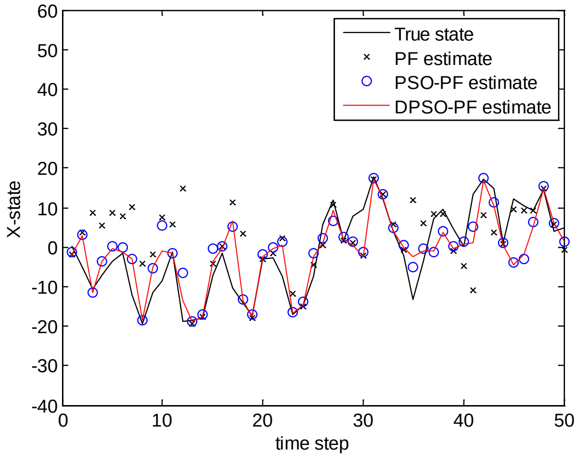

Fig.3 State estimation of different algorithm（Q=20）

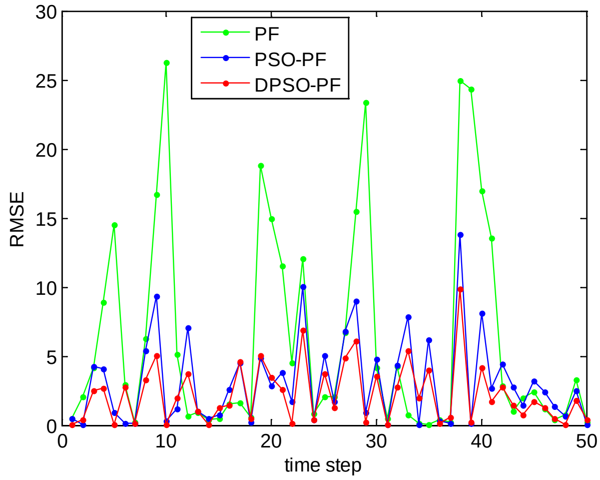

Fig.4 RMSE of different algorithm（Q=20）

Tab.1 comparison of simulation parameters by UNGM model

Parameters	Algorithms	Success rate /%	RMSE	Operation time/s
$Q=1, N=200$	PF	/	3.5901	0.6692
$Q=1, N=50$	PSO-PF	97.91	2.4524	0.5837
$Q=1, N=50$	DPSO-PF	99.33	1.4536	0.5282
$Q=20, N=200$	PF	/	5.9642	0.6803
$Q=20, N=50$	PSO-PF	97.65	4.2113	0.5924
$Q=20, N=50$	DPSO-PF	99.18	2.4300	0.5488

As shown by the emulation result, the error of the integration with particle swarm optimized particle filter is lower than that of PF, and the integration is in fact the particle optimization process of particle swarm that can improve population quality. The improved algorithm has the additional steps 5-10 compared with particle swarm optimized particle filter, but the optimization success rate of improved particle filter is higher than that of particle swarm optimized particle filter. The improved filter can optimize the number of particle's neighborhood population by dint of neighborhood extension and restriction thought, the accuracy and effective sample number of it compared with particle swarm optimized particle filter.

6.2 Simulation test of performance in integrated navigation system

Let the latitude and longitude of the initial position of system state vector be 32°and 118°, respectively; the random and constant drift errors of the gyroscope be 0.05°/*h*, respectively; the random and constant bias errors of the gyroscope be 50μg and 100μg, respectively; the update cycle of inertial navigation be 0.01*s*; the cycle of Kalman filtering be 1*s*; and the simulation time be 500s. In this paper, an analysis is implemented on the position and velocity error curves along northward, eastward and skyward directions before and after the integrated filter correction.

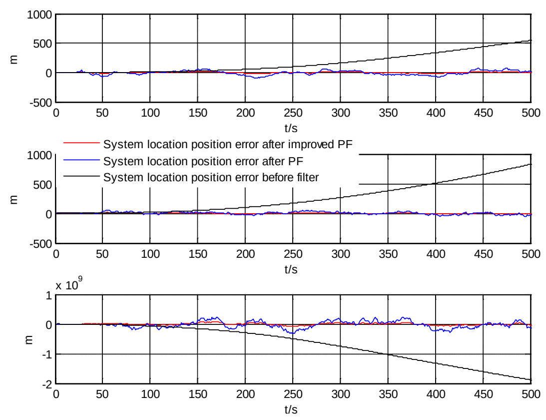

Figure 5. Position error in different directions(northward, eastward, skyward)

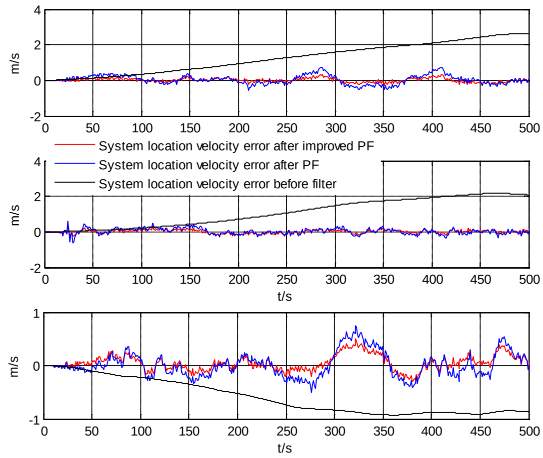

Figure 6. Velocity error in different directions(northward, eastward, skyward)

As illustrated by the figure, the system faces rapid divergence prior to the application of integrated filter, but the parameter errors of the system are correctly effectively upon the use of improved particle filter algorithm. According to the figure above, integrated filter can control the position errors within 32*m*, with the mean square deviations of position errors along the northward, eastward and skyward directions being 6.67 *m*, 7.35 m, and 4.92 *m*, respectively, and those of velocity errors being 0.22 *m/s*, 0.20 *m/s* and 0.16 *m/s*, respectively. The statistical simulation result above has verified the feasibility of the improved particle filter algorithm proposed herein, and conquered the defect that filter is prone to failure on the condition of high observation accuracy to maintain a high estimation accuracy.

7. Conclusion

This paper brings forward a novel particle filter algorithm based on neighborhood population adaptive particle swarm optimization which takes the neighborhood population information of particles into consideration. The diversity factors, neighborhood population extension factor, and neighborhood population limiting factor are used jointly to realize a self-adaption of the neighborhood particle numbers so as to control the influence of particles on the neighborhood and alleviate the local optimization phenomenon, and then a best balance will be reached between convergence speed and search ability, The experimental results show that the algorithm in this paper improves the precision and thus of high applicable value in GPS/INS integrated navigation 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] D. Bhatt, P. Aggarwal, V. Devabhaktuni. A novel hybrid fusion algorithm to bridge the period of GPS outages using low-cost INS. *Expert Systems with Applications*, vol.41, no.5, 2014, pp.2166-2173.

[2] Ben K. H. Soon, Steve Scheding, Hyung-KuenLee, An approach to aid INS using time-differenced GPS carrier phase (TDCP) measurements. *GPS Solution*, vol.12, no.4, 2008, pp.261-271.

[3] Antonio Angrisano. GNSS/INS Integration Methods. *PhD. Thesis, Department of Geomatics Engineering, University of Calgary, Canada*, 2010.

[4] S. Y. Cho. IM-filter for INS/GPS-integrated navigation system containing low-cost gyros. *IEEE Transactions on Aerospace and Electronic Systems*, vol.50, no.4, 2014, pp.2619 - 2629.

[5] 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, vol.10, no.11, pp.363-375.

[6] H. W. Li, J. Wang, H. T. Wang. A New Particle Filter Based on Differential Evolution Method. *Journal of Electronics & Information Technology*, vol.33， no.7, 2011, pp. 1639-1643.

[7] F. S. Jian, Y. M. Xu, Z. J. Yin. Enhanced multiple model particle filter for maneuvering target tracking. *Control Theory & Applications*, vol.8, pp. 1012-1016, 2010.

[8] K. Hwang, W. Sung. Load balanced resampling for real-time particle filtering on graphics processing units. *IEEE Transactions on Signal Process*, vol. 61, no. 2, 2013, pp. 411-419.

[9] Z. Fang, G. F. Tong, X. H. Xu. Particle swarm optimized particle filter. *Control and Decision*, vol.22, no.3, 2007, pp. 273-277.

[10] Christopher Hide, Terry Moore and MartinSmith. Adaptive Kalman Filtering for Low-cost INS/GPS. *Journal of Navigation*, vol.56, no.1, 2003, pp.143-152.

[11] Ding W, Wang J and Rizos C. Improving Adaptive Kalman Estimation in GPS/INS Integration. *The Journal of Navigation*, 2007, pp. 517-529.

[12] Z.M. Chen, Y.M. Bo, P.L. Wu, W.J. Zhou. A new particle filter based on organizational adjustment particle swarm optimization. *Applied Mathematics & Information Science*, vol.7, no.1, 2013, pp.179-186.

[13] M. H. Zhang, M. Xin, J. Yang. Adaptive multi-cue based particle swarm optimization guided particle filter tracking in infrared videos. *Neurocomputing*, vol. 122, 2013, pp. 163-171.

[14] A. Kong, J. Liu. sequential imputations and Bayesian missing data problems. *Journal American Statistical Association*, 1994, pp. 278-288.

[15] Z. M Chen, Y. M Bo, P. L. Wu, S. L. Yu. A particle filter algorithm based on chaos particle swarm optimization and its application to radar target tracking. *Journal of Computational Information Systems*. vol.8, no.7, 2012, pp. 3081-3090.

[16] X. J. Zhang, G. Z. Dai, N. P. Xu. Study of Diversity of Population in Genetic Algorithm. *Control Theory and Application*, vol.2, no.1, 1998, pp. 17-23.

[17] W. M. Xian, B. Long, M. Li, et al. Prognostics of Lithium-Ion Batteries Based on the Verhulst Model, Particle Swarm Optimization and Particle Filter. *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 1, 2014, pp. 2-17.

[18] Y. Li, B. Bai, Y. Zhang. Improved particle swarm optimization algorithm for fuzzy multi-class SVM.

Journal of Systems Engineering and Electronics, vol. 21, no. 3, 2010, pp. 509-513.

[19] S.A. MirHassani,N. Abolghasemi. A particle swarm optimization algorithm for open vehicle routing problem. *Expert Systems with Applications*, vol.38, no.9, 2011, pp.11547-11551.

[20] H.W. Li, J. Wang. Particle filter for manoeuvring target tracking via passive radar measurements with glint noise. *IET Radar, Sonar & Navigation*, vol. 6, no. 3, 2012, pp. 180-189.

x_k	state values
v_{k-1}	systematic noise
$p()$	probability density
$q()$	importance function
w_k^i	particle weight
pb	optimal location of particle i
pg	optimal position of the whole particle group
N_E	error of GPS receiver in eastward direction.
N_N	error of GPS receiver in northward direction
N_U	error of GPS receiver in skyward direction
σ_{pE}^2	variance of measurement noise in eastward direction
σ_{pN}^2	variance of measurement noise in northward direction.
σ_{pU}^2	variance of measurement noise in skyward direction.

v_E	true velocities along eastward, axes
v_N	true velocities along northward axes
v_U	true velocities along skyward axes
M_E	components of velocity measurement errors of GPS receiver along eastward axes.
M_N	components of velocity measurement errors of GPS receiver along northward axes.
M_U	components of velocity measurement errors of GPS receiver along skyward axes.
Y_k^i	fitness value
Q	process noise variance
R	measurement noise variance
RMSE	root-mean-square error
Extend(i)	extension factor
removal(i)	restriction factor
S_i	number of neighborhood population
$V_k^{i_m}$	velocity of particle
ε	threshold value
Dt	diversity factor