A Random Time-varying Particle Swarm Optimization for Local Positioning Systems

Hui Zhu †, Syahrulanuar Ngah †, Ying Xu †, Yuji Tanabe † and Takaaki Baba †

Waseda University

Summary

The particle swarm optimizer (PSO) is a stochastic, population-based optimization technique that can be applied to a wide range of applications. This paper presents a random time variable PSO algorithm, called the PSO-RTVIWAC, introducing random time-varying inertia weight and acceleration coefficients to significantly improve the performance of the original algorithms. The PSO-RTVIWAC method originates from the random inertia weight (PSO-RANDIW) and time-varying acceleration coefficients (PSO-TVAC) methods. Through the efficient control of search and convergence to the global optimum solution, the PSO-RTVIWAC method is capable of tracking and optimizing in the highly nonlinear dynamic local positioning systems. Experimental results are compared with three previous PSO approaches from the literatures, showing that the new optimizer significantly outperforms previous approaches. Simply employing a few particles and iterations, a reasonable good positioning accuracy is obtained with the PSO-RTVIWAC method. This property makes the PSO-RTVIWAC method become more attractive since the computation efficiency is improved considerably, i.e. the computation can be completed in an extremely short time, which is crucial for the real-time local positioning systems. Our experiments on the PSO-RTVIWAC to track and optimize the tag position have demonstrated that it is especially effective in dealing with optimization functions in the nonlinear dynamic environments.

Key words:

inertia weight, acceleration coefficients, particle swarm, time of arrival, local positioning.

1. Introduction

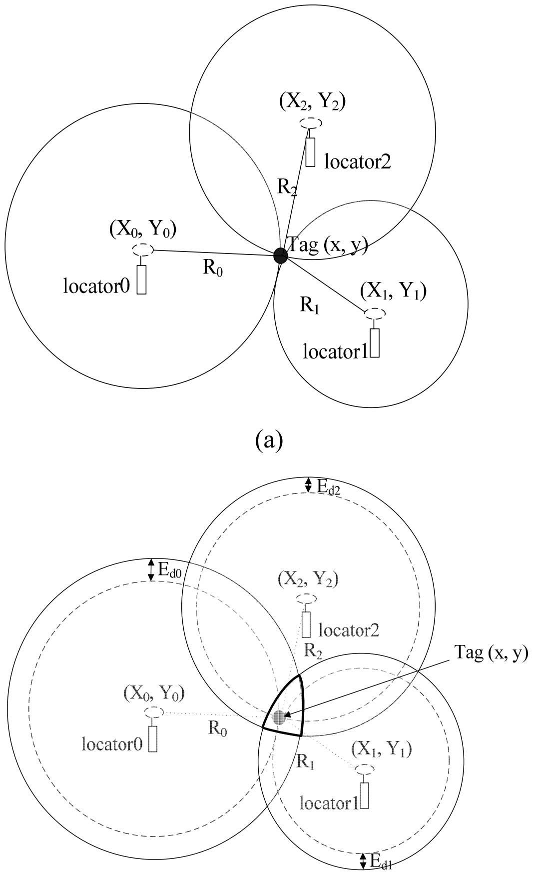
Local environments present opportunities for a rich set of position-aware applications such as navigation tools for humans and robots, interactive virtual games, resource discovery, asset tracking, position-aware sensor networking etc. Typical local positioning systems require better accuracy than what global positioning system (GPS) or other outdoor positioning systems can provide. Further, the typical local positioning systems require different types of information such as physical space, position and orientation [1-5].

Usually the local positioning systems operate in harsher environments that impede RF propagation. Besides, local positioning systems call for a smaller coverage area, often limited to a single organization compared to a typical outdoor system. Therefore, several research groups have developed a number of positioning technologies specifically tailored for local positioning systems. Some groups have developed 802.11-based positioning systems and discovered that such systems have fundamental limits resulted in a median positioning error of 3 m [6]. Bekkali [32] investigated an indoor positioning system for RFID tagged objects, which use two RFID Readers with unknown location and landmarks to achieve positioning error of 0.5 m. The UbiSense systems use a small number of UWB base stations and UWB transmitters carried by mobile users [7]. This system provides very precise position information (0.15 m accuracy) for specialized applications that operate in highly controlled environments. A configuration of local positioning systems with base stations (locators) and the device to be located (tag) is shown in Fig. 1. In Fig. 1 (a) the exact solution can be obtained for two dimensional positioning with three locators based on the time of arrival (TOA) measurements. However, in the real-world applications the measured distance error E_d is ineluctable, as depicted in Fig. 1 (b). The area highlighted by the bold line is the uncertain region, which is dominated primarily by the measured distance error E_d. Most of existing local positioning systems have measured distance errors on the order of 0.1 m to several meters. It is impossible to remove this error from the real-world applications. Usually they are considered as noises. The exact solution of tag position is difficult to be achieved under this environment. The local positioning systems are identified as highly nonlinear dynamic systems with numerous noises. There are several ways in which systems change over time [11]. First, the tag position in search space can change. Second, in the multidimensional system, the variation of position may occur on one or more dimensions, either independently or simultaneously. Third, the noises change at any time. Fourth, the locator position may change.

In local positioning systems the optimization algorithm is indispensable to convert the measured distances R to tag position, as shown in Fig. 1. If the optimization is good enough, then it is sufficient [11]. By good enough it means the accuracy requirements are satisfied reasonably well, computation finishes within an acceptable time and used

Manuscript received June 5, 2008.
Manuscript revised June 20, 2008.

available resources in efficient way [27]-[29]. Obviously, if more time is given, the optimization algorithm probably performs better. However, in the real-time local positioning systems, it is highly expected to evaluate a solution in a very short time.



(b)

Fig. 1. (a) The general positioning under the idea environment and (b) positioning under the noisy environment

The basic way to estimate the position via the measured distances may be by directly solving a set of nonlinear equations. For an over-determined system with redundant measurements, Taylor Series Expansion (TSE) [8] may be used to iteratively produce a solution to the position estimate. However, to maintain good convergence, the TSE method requires a quite accurate initial position estimate which is often difficult to be obtained in some practical applications.

The Particle Swarm Optimization (PSO) algorithm is a new sociologically inspired stochastic optimization algorithm introduced by Kennedy and Eberhart in 1995 [10]. The PSO algorithm is easy to implement, has few parameters, and has been shown to converge faster than traditional techniques like GA for a wide variety of benchmarks. The PSO algorithm can produce better results in a faster, cheaper way, compared with other methods. Years ago, PSO has been successfully applied in many research and application areas.

However, like other heuristic algorithms, PSO process is time consuming. For many real-world applications, PSO can run for days, even when it is executed on a high performance workstation. The computation time must be reduced to fulfill the hard real-time requirements of the local positioning systems. The most straightforward way is to use a small particle number and iterations. In PSO research, it is quite common to limit the number of particles to the range of 20 to 60. The maximum number of iterations may change for various applications, from 100 to several thousands. The optimization algorithm is necessary to be as fast as possible to complete the computation within a very short time. Considering above concerns, in this paper a new concept PSO algorithm is introduced to track and optimize in the local positioning systems. The main objective of this development is to reduce the computation time of PSO algorithms. Meanwhile, the tag positions are evaluated with reasonable accuracy. The concept of random time variable inertia weight and acceleration coefficients (PSO-RTVIWAC) is introduced, which is motivated by the PSO-RANDIW and PSO-TVAC concepts [15].

The rest of this paper is organized as follows. In Section 2, three significant previous developments to the original PSO methodology are summarized. These methods are used as the comparative measure of performance of our novel development in this paper. In Section 3, the proposed new extension to PSO algorithm, called the PSO-RTVIWAC, is introduced. Then the position estimate using PSO-RTVIWAC and its process are explained in Section 4 and 5, respectively. The experimental settings are explained in the Section 6. In Section 7, the PSO-RTVIWAC method is applied to investigate its ability to find the tag position in the local positioning systems. The experimental results in comparison with three previous developments are presented. Section 8 comprises the summary and conclusions of this study.

2. Previous Work on the PSO Algorithm

2.1 PSO-TVIW

Shi and Eberhart [17] introduced the concept of inertia weight to the original version of PSO, in order to balance the local and global search during the optimization process. Shi and Eberhart [18] have found a significant improvement in the performance of the PSO method with a linearly varying inertia weight over the generations. The mathematical representation of this concept is briefly illustrated as follows [19]:

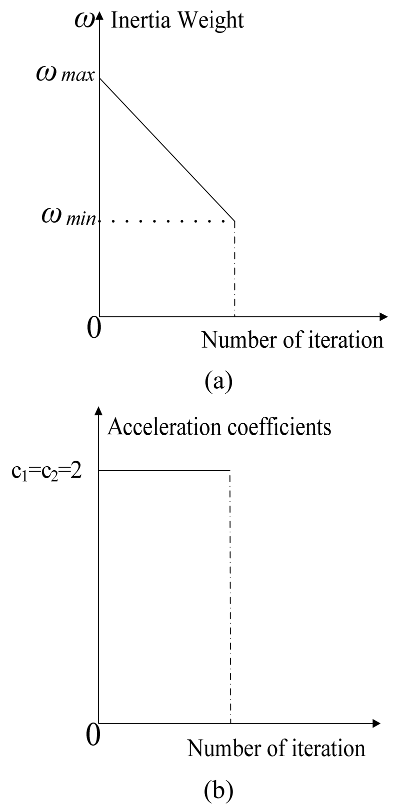


Fig. 2. (a) The inertia weight and (b) the acceleration coefficients in the PSO-TVIW

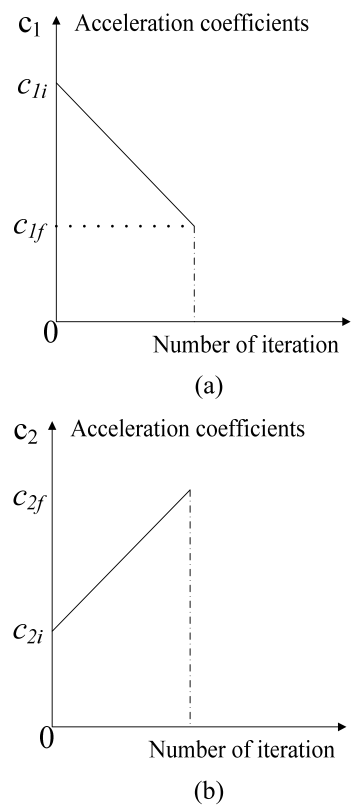


Fig. 3. (a) The inertia weight and (b) the acceleration coefficients in the PSO-TVAC

(1) A population of particles with random positions and velocities is initialized in the search space.
(2) For each particle evaluate the optimization fitness function *Fi*. Then two comparisons are executed. Firstly compare current particle's fitness evaluation with fitness evaluation of particle's local best position. If current value is better, then set fitness evaluation equal to the current value and the local best position equal to the current position in search space. Secondly compare fitness evaluation with fitness evaluation of the population's global best position. If current value is better than the previous value of fitness evaluation, then reset to the current particle's array index and value.
(3) In standard PSO, the velocity and position of the particle *i* are updated based on following equations:

$$\upsilon_{ix}^{t+1} = \omega * \upsilon_{ix}^{t} + c_1 * r_1 * (p_{lx} - x_i^t) + c_2 * r_2 * (p_{gx} - x_i^t) \quad (1)$$

$$\upsilon_{iy}^{t+1} = \omega * \upsilon_{iy}^{t} + c_1 * r_3 * (p_{ly} - y_i^t) + c_2 * r_4 * (p_{gy} - y_i^t) \quad (2)$$

$$x_i^{t+1} = \upsilon_{ix}^{t+1} + x_i^t \quad (3)$$

$$y_i^{t+1} = \upsilon_{iy}^{t+1} + y_i^t \quad (4)$$

$$\omega = \omega_{\max} - t * (\omega_{\max} - \omega_{\min}) / T \quad (5)$$

where c_1 and c_2 are the acceleration constants, r_1, r_2, r_3 and r_4 are uniformly distributed random numbers between 0 and 1, the current iteration is t, the maximum number of iterations is T and the inertia weight ω is decreased linearly from maximum value ω_{max} to minimum value ω_{min} during T.
Experience showed that when using inertia weight, the maximum velocity factor V_{max} could simply be set to the value of the dynamic range of each variable. This limitation is sometimes necessary to keep the particle from oscillating too fast around a region without adequately exploring it. In this way, there is no longer the need of a complicated strategy for setting V_{max}.
Through empirical studies, Shi and Eberhart [18] have observed that the optimal solution is improved by varying the value of inertia weight from 0.9 at the beginning to 0.4 at the end of the search for most applications. This version of PSO is referred to as time-varying inertia weight factor method (PSO-TVIW) in the following sections.
As depicted in Fig. 2 (a) and (b), in the PSO-TVIW the inertia weight is set to decrease linearly from 0.9 to 0.4 during the iterations and two acceleration coefficients are fixed at 2. However, Sugant [23] in his experiment suggested that the acceleration coefficients should not be equal to 2 all the time.

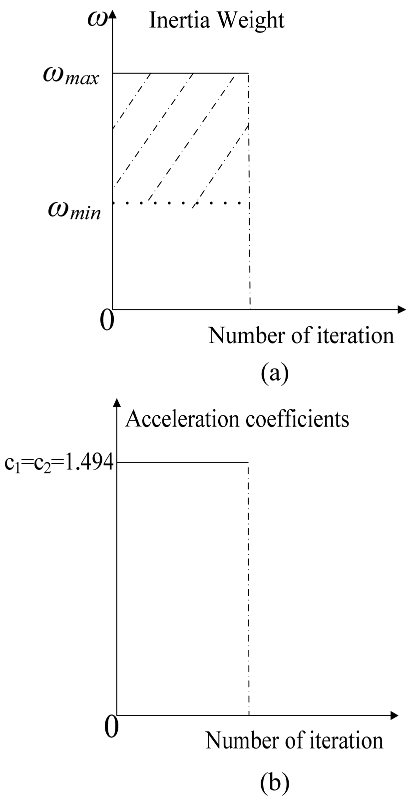


Fig. 4. (a) The inertia weight and (b) the acceleration coefficients in the PSO-RANDIW

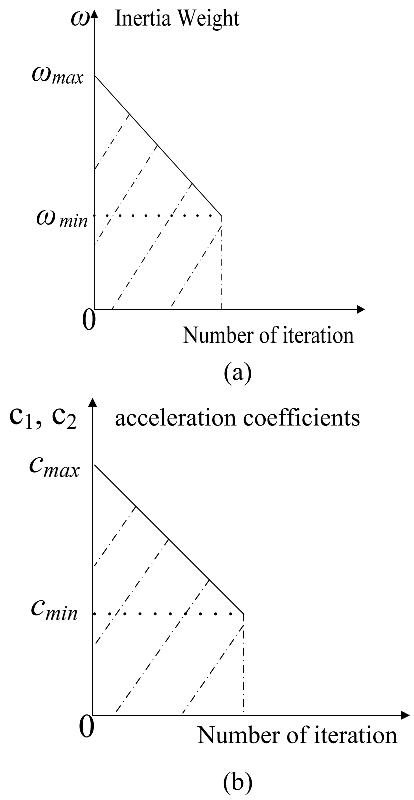


Fig. 5. (a) The inertia weight and (b) the acceleration coefficients in the PSO-RTVIWAC

2.2 PSO-TVAC

The acceleration coefficients c_1 and c_2 are usually set to the same value which is $c_1=c_2=2$. However, taking into account the adaptive step size of the algorithm, it is not suitable to set the acceleration coefficients to be constant.
According to Ratnaweera *et al.* [15], the time-varying acceleration coefficients method (PSO-TVAC) controls the local search and convergence to the global optimum solution. They suggested this method to be run with time-varying acceleration coefficients given as:

$$c_1 = (c_{1f} - c_{1i})\frac{iter}{MAXITR} + c_{1i} \quad (6)$$

$$c_2 = (c_{2f} - c_{2i})\frac{iter}{MAXITR} + c_{2i} \quad (7)$$

where c_{1i}, c_{1f}, c_{2i}, and c_{2f} are constants, *iter* is the current iteration number and *MAXITR* is the maximum number of iterations. This method runs with a TVIW factor as given in (5).
As shown in Fig. 3 (a) and (b), c_1 is recommended to change from 2.5 to 0.5 and c_2 changing from 0.5 to 2.5, over the full range of the search [15]. Under this development, the cognitive component is reduced and the social component is increased, by changing the acceleration coefficients with time. With a large cognitive component and small social component at the beginning, particles are allowed to move around the search space. Conversely, a small cognitive component and a large social component allow the particles to converge to the global optima in the latter part of the optimization.

2.3 PSO-RANDIW

Eberhart and Shi [11] have found that the PSO-TVIW concept is not very effective for tracking dynamic systems. Angeline [30] indicated that for some dynamic functions, self-adaptation is effective while for others it is detrimental. He uses three types of dynamics: linear, circular, and random. Back [31] used the same three types of dynamics as Angeline. His results indicate that self-adaptation ofvariance as utilized in a (μ,λ)-evolution strategy is an effective method for tracking dynamic environments. Instead, considering the dynamic nature of real-world applications, Eberhart and Shi [11] have proposed a random inertia weight factor for tracking dynamic systems. In this development, the inertia weight factor is set to change randomly according to the following equation:

$$\omega = 0.5 + \frac{rand}{2} \quad (8)$$

where *rand* is a uniformly distributed random number within the range of 0 and 1. When random inertia weight factor method is used the acceleration coefficients are kept constant at 1.494, as shown in Fig. 4 (a) and (b).

This version of PSO is referred as random inertia weight factor method (PSO-RANDIW) in the following sections. It has been identified in many applications that the PSO-RANDIW method achieved a reasonably good solution for most of the functions. The errors are less than what obtained by either Angeline or Back [11]. PSO-RANDIW performs better than PSO-TVAC in both unimodal and multimodal functions [15]. Therefore, this method is selected to compare the effectiveness of the novel PSO strategies introduced in this paper.

3. The Proposed PSO-RTVIWAC Algorithm

Early developments in PSO have provided many effective approaches in optimizing static applications. However, a lot of real-world applications are always recognized as nonlinear dynamic systems. Many real-world systems change state continuously. These changes result in a requirement for frequent, sometimes almost continuous, re-optimization. Clearly a PSO algorithm designed to be used in a real-world dynamic system is supposed to be tested in an environment most closely resembling the real-world situation [11], [25]. The local positioning systems are identified as nonlinear dynamic systems. First, in these applications the actual tag position in the search space changes over time. Since the position of the optimizer is moving through the search space, the optimization algorithm is expected to track it over time. Second, the objective function is inherently imprecise. The measured distance error E_d makes it difficult to approximate the optimization position in the search space.

In general, PSO results in global solutions even in high–dimensional and multimodal spaces. In contrast, there are not many experimental results about its behavior in the presence of noise, i.e. the performance of the method when noise is inserted into the fitness function and the landscape is continuously changing. In practical applications, most of methods can detect just sub–optimal solutions of the objective function. In many cases these sub–optimal solutions are acceptable but there are applications where a high-speed computation is not only desirable but also indispensable. Moreover, in many applications there are imprecise values for the input data as well as for the fitness function. Therefore, the development of robust and efficient PSO methods for dynamic environments is a subject of considerable ongoing research.

The motivation of new developed PSO concept in this paper is to achieve an acceptable accuracy in the local positioning systems, simply employing a small particle number and iterations. Benefit from the small number of particle and iterations, the computation time will be reduced to fulfill the rigid real-time conditions. Through the efficient control of search and convergence to the global optimum solution, the PSO-RTVIWAC method is capable of tracking and optimizing in the highly nonlinear dynamic local positioning systems.

Previously, a linearly decreasing inertia weight or acceleration coefficients, often decreasing from 0.9 to 0.4 during a run is used in PSO-TVIW and PSO-TVAC concepts. When tracking a nonlinear dynamic system, however, it is hard to predict whether exploration (a larger inertia weight value) or exploitation (a smaller inertia weight) will be better at any given time. Simply decrease of these parameters linearly is proved unsuccessful in the dynamic systems.

Through empirical studies with some of the well-known benchmarks, it has been identified that the PSO-RANDIW method shows rapid convergence in the early stages of the optimization process and can find a good solution with great accuracy in the non-stationary environments. A new concept of PSO, motivated by the PSO-RANDIW and PSO-TVAC, is proposed in this paper. As shown in Fig. 5 (a) and (b), as a new parameter automation strategy for the PSO concept, the time variable inertia weight and acceleration coefficients that randomly vary roughly within a linear range are introduced. The inertia weight is decreased linearly from ω_{max} to minimum value ω_{min}, multiplied by a random value r_5. It means the inertia weight randomly varies within the shadow area, as shown in the Fig. 5 (a). The acceleration coefficients c_1 and c_2 change from $c_{\max}$ to $c_{\min}$, multiplied by the other random numbers. The modifications of ω, c_1, c_2, x_{t+1} and y_{t+1} are mathematically represented as follows:

$$\omega = r_5 * \left(\omega_{\max} - t * (\omega_{\max} - \omega_{\min}) / T\right) \quad (9)$$

$$c_1 = r_6 * \left(c_{\max} - t * (c_{\max} - c_{\min}) / T\right) \quad (10)$$

$$c_2 = r_7 * \left(c_{\max} - t * (c_{\max} - c_{\min}) / T\right) \quad (11)$$

$$x_i^{t+1} = k * \upsilon_{ix}^{t+1} + x_i^t \quad (12)$$

$$y_i^{t+1} = k * \upsilon_{iy}^{t+1} + y_i^t \quad (13)$$

$$phi = 4 * (1 + r_8) \quad (14)$$

$$k = 2 \Big/ \left| 2 - phi - \sqrt{phi^2 - 4 * phi} \right| \quad (15)$$

where r_6, r_5, r_7 and r_8 are uniformly distributed random numbers between 0 and 1. This version of PSO is referred to as time-varying inertia weight factor and acceleration coefficients method (PSO-RTVIWAC) in the following sections.

The research work done by Clerc [19], [24] indicates that use of a constriction factor k may be necessary to insure convergence of the particle swarm algorithm. It is used to prevent the particles from exploring too far away into the search space since always applies a damping effect to the oscillation size of each particle over time [20]-[23].

The objective of PSO-TVAC is to enhance the global search in the early part of the optimization and to encourage the particles to converge toward the global optima at the end of the search. By changing the acceleration coefficients with time, the cognitive component is reduced and the social component is increased. However, in local positioning systems the positioning error is dominated by the measured distance error E_d, after the global search arrives the uncertain area as illustrated in Fig. 1 (b). A fast global search in the early stage is more attractive than convergence toward the global optima to find the optimum solution during the latter stage. Highly dynamic environments favor greater values for these parameters, which gives rise to faster-moving particles that are able to track a fast moving optimum better [26].

With above modification, a significant improvement of the optimum value and the rate of convergence are observed, compared with three previous PSO concepts. The PSO-RTVIWAC method shows significantly quick convergence to a good solution, employing a small number of particles and iterations. The experimental results are presented and discussed in following sections.

4. Position Estimate using PSO-RTVIWAC

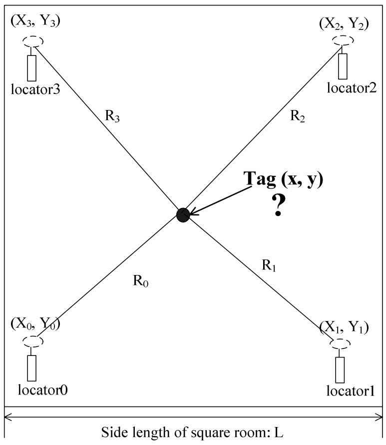
The general idea of position estimate is explained in Fig.6. In Fig.6 (a), there is a square room with side length L. As an example, four locators (*locator0*, *locator1*, *locator2* and *locator3*) are deployed in this room. Coordinates of locators (X_0, Y_0), (X_1, Y_1), (X_2, Y_2) and (X_3, Y_3) are predetermined.

The distances between locators and tag are measured in TOA technique. As an initial condition, these distances expressed as R_0, R_1, R_2 and R_3 are obtained through measurement. Tag position (x, y) is unknown at this stage. Above situation is system input conditions.

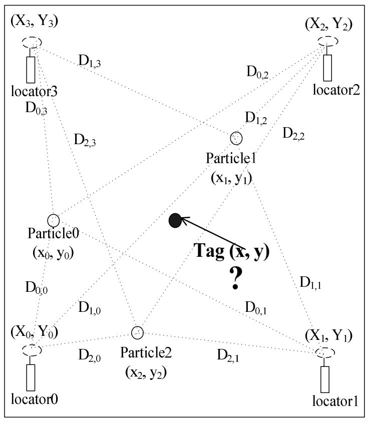
The measured distance between tag and locator m is expressed as:

$$R_m = \sqrt{(X_m - x)^2 + (Y_m - y)^2} + E_d * rd \qquad (16)$$

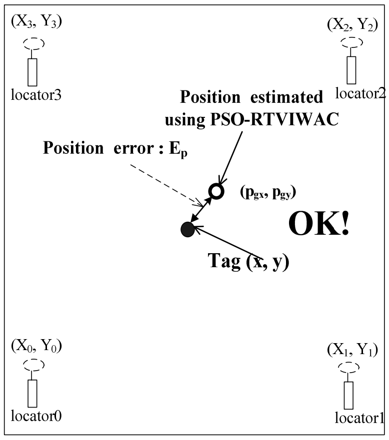
where (x, y) is an assumed coordinate of the tag, (X_m, Y_m) is the coordinates of locator m, E_d is the maximum value of measured distance error, which is shown in Fig. 1 (b), and rd is an uniformly distributed random number between 0 and 1.



(a)



(b)



(c)

Fig. 6. General idea of position estimate using PSO-RTVIWAC: (a) distance measured in TOA technique, (b) position estimated in PSO-RTVIWAC and (c) estimated results and error

Based on above assumption, system process for searching the position of tag is performed as follows. The PSO-RTVIWAC is used to estimate tag position, as illustrated in Fig.6 (b). Three particles are deployed into the room randomly: particle0 (x_0, y_0), particle1 (x_1, y_1) and particle2 (x_2, y_2). The distances between particles and locators are calculated. For instance, the distance between particle2 and locator0 is $D_{2,0}$. These particles move to the tag based on the PSO-RTVIWAC concept. The detailed process is presented in the next section.

Finally, the particles stop around the tag which is shown in Fig.6 (c). Global best position (p_{gx}, p_{gy}) is considered as the tag position. It is the system output information. The position error between tag and global best position is E_p. If a reasonable good E_p is achieved only using a few particles and iterations, the benefit of PSO-RTVIWAC is demonstrated.

5. The Process of Position Estimate

Figure 7 shows estimation process in the PSO-RTVIWAC method. More detailed process and its practical performance are discussed in this section.

5.1 System Initialization

This system is initialized as shown in Fig.7 (a). Locators are deployed in the certain position of a square room. Tag is deployed randomly. The number of locators is set to nine in this example. The coordinates of nine locators are (0, 0), (0, 5), (0, 10), (5, 0), (5, 5), (5, 10), (10, 0), (10, 5) and (10, 10) (unit: meter). Distances between tag and locators are measured as defined in equation (16).

5.2 Estimation Process in the PSO-RTVIWAC

In the second step, the program uses the PSO-RTVIWAC method to estimate tag position.

(1) In this stage as explained in Fig.7 (b), the particle swarm is initialized. These particles are deployed in the room with random positions and velocities.

(2) The program calculates the distance between particle i and locator m:

$$D_{i,m} = \sqrt{\left(X_m - x_i^t\right)^2 + \left(Y_m - y_i^t\right)^2} \quad (17)$$

(3) The optimization fitness function of particle i at t is expressed as:

$$F_i(t) = \sum_{m=1}^{9} \left(D_{i,m} - R_m\right)^2 \quad (18)$$

$F_i(t)$ is the key point of this program. If the optimization fitness function F_i is small, the distance between the particle i and tag is short. When F_i is zero, particle i reaches the position of tag.

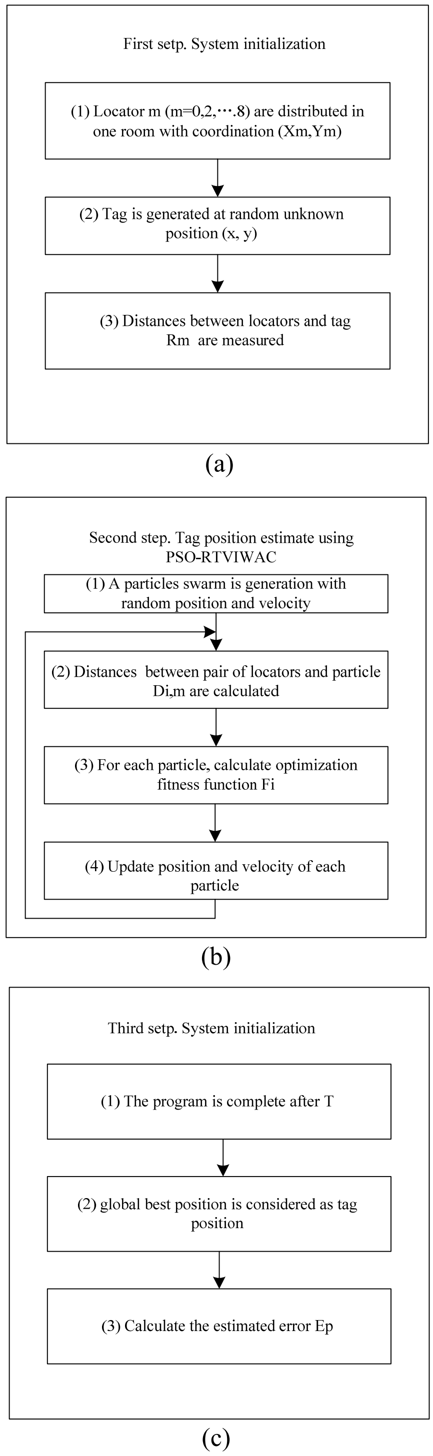


Fig. 7. Position estimate using PSO-RTVIWAC: (a) system initialization, (b) estimation process and (c) estimated results and error

From previous iteration t_p to current iteration t_c, the position of particle i is determined based on a velocity defined before. For particle i, compare current $F_i(t_c)$ with fitness evaluation $F_i(t_p)$ which is corresponding to local best position (p_{lx}, p_{ly}) at previous loop. If $F_i(t_c)$ is smaller than $F_i(t_p)$, current position for particle i is recognized as new "local best position". Otherwise, the local best position is kept as it is. The comparisons are performed over entire particle swarm. Then compare every $F_i(t_c)$ with fitness evaluation $F_i(t_p)$ for global best position (p_{gx}, p_{gy}). Find the smallest value among them. That is recognized as new "global best position".

5.3 Estimated Results and Error

In the third step, the program is completed after T. As explained in Fig.6 (c) and Fig.7 (c), all particles converge into the global best position (p_{gx}, p_{gy}). This position is the optimal solution estimated using PSO-RTVIWAC, which is considered as system output. The position error E_p is defined as:

$$E_p = \sqrt{(p_{gx} - x)^2 + (p_{gy} - y)^2} \tag{19}$$

6. Experimental Settings

6.1 Benchmark

In this section the performance of PSO-RTVIWAC is evaluated and compared with three previous methods. Simulations are executed over one thousand runs to find the global minimum in the search space. The position of tag is randomly placed in each run. The average position error $E_{p,average}$ is expressed as:

$$E_{p,average} = \sqrt{\sum_{r=1}^{1000} \left(E_p^{\ 2}\right) \Big/ 1000} \tag{20}$$

The $E_{p.average}$ is used to evaluate the performance of each PSO concept.

6.2 Population Size

It is quite common in PSO research to limit the number of particles to the range 20 to 60 [12]–[14]. Van den Bergh and Engelbrecht [20] suggested that even though there is a slight improvement of the optimal value with increasing swarm size, it increases the number of function evaluations to converge to an error limit. This is generally true in terms of performance, but not in terms of cost. More particles would search more space, and a solution would then be found sooner. However, as the population increases, each iteration represents a greater cost. First, more particles rise the initialization time. Second, a large number of particles call upon the evaluation function. Then more particles increase the computation time remarkably. In order to fulfill the hard real-time conditions, the particle number must be reduced as much as possible. Therefore, all experiments are carried out with four different population sizes: 10, 15, 20 and 25.

6.3 The Maximum Number of Iterations

A very small error may be achieved if significantly more iterations are required. However such small value is not necessary in most real-world wireless positioning systems since the measured distance error is much large than the evaluate errors. An evaluate error of 0.01 m is sufficient. Moreover, for the real-time systems, the computation time is not acceptable if more iterations are applied. The iterations should be reduced in order to finish the computations within a very short time. Then in this paper the maximum number of iterations is selected as 20 and 50.

6.4 Population Initialization

Since the optimization function used in this paper may have the global minimum at or close to the origin of the search space, the initial population is randomly distributed in the entire search space. This method is used to observe the performance of the each PSO concept introduced in this paper. A dynamic range of 50 m is used in all dimensions. Generally in PSO methods the maximum velocity of a particle is limited, in order to eliminate excessive searching outside the predefined search space. The maximum velocity is limited to the upper value of the dynamic range.

7. Experimental Results

For all simulations, the inertia weight factor is set to change from 0.9 to 0.4 over the generations. The performance of the PSO-RTVIWAC method is then observed in comparison with the PSO-TVIW, PSO-TVAC and PSORANDIW methods. Results are presented in Tables.

7.1 The Maximum number of iterations is set to 20

Table 1 summarizes the average positioning error $E_{p,average}$ as the search space is set to a dimension of 50 m × 50 m. Initially, in the Table 1, it is seen that the PSO-RTVIWAC method achieves the smallest average positioning error $E_{p,average}$ under any particle number. In four PSO concepts, the average positioning error $E_{p,average}$ evaluated by PSO-TVIW is worst. The average positioning error $E_{p,average}$ evaluated by PSO-RANDIW is smaller than what the PSO-TVIW and PSO-TVAC concepts produce. The average positioning error $E_{p,average}$ is substantially improved by the PSO-RTVIWAC method. Second, the experimental results also indicate the effect of particle number on the optimum solution. As the particle number is increased from 10 to 25, for the PSO-TVIW, PSO-TVAC and PSO-RANDIW methods, the average positioning error $E_{p,average}$ are reduced by 50 %. However, it is reduced by 75 % with the PSO-RTVIWAC method. For example, from the table above, it is clearly shown that $E_{p,average}$ for PSO-TVAC is reduced from 0.605 m to 0.303 m and 0.106 m to 0.0255 m for PSO-RTVIWAC. Obviously, if the particle number is increased, a significant improvement of convergence rate is observed with the PSO-RTVIWAC method.

Table 1. The average positioning error $E_{p,average}$ produced with 20 iterations in a dimension of 50 m × 50 m

	Particle number	Side length (m)	Max. number of iteration	$E_{p,average}$ (m)
PSO-TVIW	10	50	20	8.00E-01
PSO-TVAC	**10**	**50**	**20**	**6.05E-01**
PSO-RANDIW	10	50	20	3.02E-01
PSO-RTVIWAC	**10**	**50**	**20**	**1.06E-01**
PSO-TVIW	15	50	20	6.08E-01
PSO-TVAC	15	50	20	4.55E-01
PSO-RANDIW	15	50	20	2.17E-01
PSO-RTVIWAC	15	50	20	4.40E-02
PSO-TVIW	20	50	20	5.15E-01
PSO-TVAC	20	50	20	3.64E-01
PSO-RANDIW	20	50	20	1.68E-01
PSO-RTVIWAC	20	50	20	2.94E-02
PSO-TVIW	25	50	20	4.43E-01
PSO-TVAC	**25**	**50**	**20**	**3.03E-01**
PSO-RANDIW	25	50	20	1.44E-01
PSO-RTVIWAC	**25**	**50**	**20**	**2.55E-02**

7.2 The Maximum number of iterations is set to 50

In the Table 2, the maximum number of iterations is set to 50. Since more iterations are applied, the average positioning error $E_{p,average}$ is decreased in each PSO concept. Apparently the PSO-RTVIWAC method achieves the smallest average positioning error $E_{p,average}$, compared with other methods under any particle number. If the particle number is set to 10 and the maximum number of iterations is set to 50 (500 evaluations), an error of 0.00603 m is achieved by the PSO-RTVIWAC method, which is sufficient for the current local positioning systems. However, in other PSO concepts, more evaluations are necessary to achieve the similar positioning accuracy. For example, an error of 0.00799 m is accomplished by the PSO-RANDIW method, using 1250 evaluations (25*50). It means that to achieve the same order of the accuracy, the computation time consumed by the PSO-RANDIW method is 2.5 times of what the PSO-RTVIWAC method used.

From all above experimental results it is summarized that, for local positioning systems, the introduction of PSO-RTVIWAC has significantly improved the optimum solution, compared with other PSO strategies. If the particle number is increased, the PSO-RTVIWAC method accomplishes more improvement on the optimum solution, which is especially desirable in the local positioning systems. The computation efficiency is highly improved, i.e. the computation can be finish in an extremely short time. As a result, simply employing a few particles and iterations, the position can be obtained with reasonable good accuracy. On the other hand, an improvement of convergence rate is observed with the PSO-RTVIWAC method. Such an attractive property is significant for the real-time applications. Consistent performance has been

Table 2. The average positioning error $E_{p,average}$ produced with 50 iterations in a dimension of 50 m × 50 m

	Particle number	Side length (m)	Max. number of iteration	$E_{p,average}$ (m)
PSO-TVIW	10	50	50	2.02E-01
PSO-TVAC	10	50	50	1.03E-01
PSO-RANDIW	10	50	50	2.25E-02
PSO-RTVIWAC	**10**	**50**	**50**	**6.03E-03**
PSO-TVIW	15	50	50	1.47E-01
PSO-TVAC	15	50	50	7.42E-02
PSO-RANDIW	15	50	50	1.39E-02
PSO-RTVIWAC	15	50	50	3.26E-03
PSO-TVIW	20	50	50	1.15E-01
PSO-TVAC	20	50	50	5.83E-02
PSO-RANDIW	20	50	50	9.57E-03
PSO-RTVIWAC	20	50	50	1.55E-03
PSO-TVIW	25	50	50	9.88E-02
PSO-TVAC	25	50	50	4.73E-02
PSO-RANDIW	**25**	**50**	**50**	**7.99E-03**
PSO-RTVIWAC	25	50	50	1.25E-03

observed with the PSO-RTVIWAC method, for all experimental settings considered in this investigation.

8. Conclusions

In this paper a novel random time-varying inertia weight and acceleration coefficients (PSO-RTVIWAC) method is described aiming to improve the performance of track and optimize in the local positioning systems.

The local positioning systems are identified as nonlinear dynamic systems with numerous noises. They consist of two distinct characteristics: First, the actual position of tag in the search space changes over time so that the optimization algorithm is expected to track it over time. Second, the objective function is inherently imprecise because the measured distance error E_d makes it very difficult to approximate the precise position in the search space. An effective algorithm is essential to track and optimize the tag position. This algorithm needs to meet the accuracy requirements reasonably well and complete the computation in a very short time.

In this paper, the PSO-RTVIWAC concept is introduced to efficiently control the search and converge to the global optimum solution. Experimental results are compared with three previous PSO approaches, showing that the new optimizer significantly outperforms previous approaches. A significant improvement of the convergence rate is observed with PSO-RTVIWAC method. It means that the particle number and iterations can be reduced to a small value while accomplishing an acceptable positioning accuracy simultaneously. Therefore, through the PSO-RTVIWAC method, the computation time can be reduced to a great extent, which is crucial in the real-time local positioning systems. The experimental results indicate that PSO-RTVIWAC is a particularly efficient method for tracking and optimizing in the nonlinear dynamic local positioning systems.

Acknowledgments

This research was supported by “Ambient SoC Global COE Program of Waseda University” of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] M. Hata and T. Nagatsu, "Mobile Location using Signal Strength Measurements in a Cellular System," IEEE Trans. On Vehicular Technology, pp. 221-225, 1991.

[2] W. C. Jakes, Microwave Mobile Communications, IEEE Press, 1994.

[3] G. Turin, W. Jewell, and T. Johnston, "Simulation of urban vehicle-monitoring systems", IEEE Trans. on Vehicular technology, vol. VT-21, pp. 9-16, 1972.

[4] C. H. Knapp and G. Clioeord Carter, "The Generalized Correlation Method of Estimation of Time Delay", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 24, no. 4, pp. 320-327, 1976.

[5] L. Izzo, A. Napolitano, and L. Paura, "Modified Cyclic Methods for Signal Selective TDOA Estimation", IEEE Transactions on Signal Processing, vol. 42, no. 11, pp. 3294-3298, 1994.

[6] E. Elnahrawy, X. Li, and R. P. Martin, "The Limits of Localization Using Signal Strength: A Comparative Study", In IEEE Sensor and Ad hoc Communications and Networks Conference (SECON 2004), Santa Clara, CA, 2004.

[7] W. C. Chung and D. S. Ha, "An accurate Ultra Wideband (UWB) Ranging for Precision Asset Location", In Int. Conf. on Ultra Wideband Systems and Technologies, Reston, VA, November, pp 389–393, 2003.

[8] W.H.Foy, "Position-Location Solutions by Taylor Series Estimation", IEEE Trans on Aerospace and Electronic Systems, vol. AES-12, no. 2, pp. 187-194, 1976.

[9] 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.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, pp. 1942–1948, 1995.

[11] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in Proc. IEEE Congr. Evolutionary Computation 2001, Seoul, Korea, pp. 94–97, 2001.

[12] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in Proc. IEEE Int. Congr. Evolutionary Computation, vol. 1, pp. 84–88, 2000.

[13] Y. Shi and R. C. Eberhart, "Comparison between genetic algorithms and particle swarm optimization," in Lecture Notes in Computer Science—Evolutionary Programming VII, Proc. 7th Int. Conf. Evolutionary Programming, vol. 1447, pp. 611–616, 1998.

[14] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in Lecture Notes in Computer Science—Evolutionary Programming VII, vol. 1447, Proc. 7th Int. Conf. Evolutionary Programming, pp. 591–600, 1998.

[15] A. Ratnaweera, S K. Halgamuge and H. C. Watson, "Self Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients," IEEE Trans. on Evolutionary Computation, vol. 8, no. 3, pp. 240-255, 2004.

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

[17] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in Proc. IEEE Int. Conf. Evolutionary Computation, pp. 69–73, 1998.

[18] Y. Shi and R. C. Eberhart,, "Empirical study of particle swarm optimization," in Proc. IEEE Int. Congr. Evolutionary Computation, vol. 3, pp. 101–106, 1999.

[19] E. Ozcanand and C. K. Mohan, "Particle swarm optimization: Surfing the waves," in Proc. IEEE Congr. Evolutionary Computation 1999, vol. 3, Washington, DC, pp. 1944–1949, 1999.

[20] R.C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources", Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 81-86, 2001.

[21] M. Clerc and J. Kennedy, "The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space," IEEE Transactions on Evolutionary Computation, pp. 58-73, 2002.

[22] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," Proc. CEC 1999, Washington, DC, pp 1951-1957, 1999.

[23] P. N. Suganthan, "Particle swarm optimizer with neighborhood operator," in Proc. IEEE Int. Congr. Evolutionary Computation, vol. 3, pp. 1958–1962, 1999.

[24] R.C. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization," Congress on Evolutionary Computing, vol. 1, pp. 84-88, 2000.

[25] F. van den Bergh, "Analysis of the particle swarm optimization algorithm," Doctor's Dissertation, University of Pretoria, 2002.

[26] D. Parrott and X. Li, "A Particle Swarm Model for Tracking Multiple Peaks in a Dynamic Environment using Speciation," In Proc. of the 2004 Congress on Evolutionary Computation (CEC 2004), IEEE Service Center, Piscataway, pp. 98-103, 2004.

[27] A. Farmahini-Farahani, S. M. Fakhraie and S. Safari, "SOPC-Based Architecture for Discrete Particle Swarm Optimization," Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference pp.1003–1006, 2007.

[28] G. Kókai, T. Christ, and H. H. Frühauf, "Using hardware based particle swarm method for dynamic optimization of adaptive array antennas," in Proc. First NASA/ESA. Conference on Adaptive Hardware and Systems, K. Didier, Ed., pp. 51–58, 2006.

[29] A. Farmahini-Farahani, S. M. Fakhraie and S. Safari, "Scalable Architecture for on-Chip Neural Network Training using Swarm Intelligence," in Proc. of the Design, Automation and Test in Europe Conf. (DATE'08), Munich, Germany, pp.1340-1345, 2008.

[30] P. J. Angeline, "Tracking extreme in dynamic environments," Proc. Evolutionary Programming VI, Indianapolis, In. Berlin: Springer-Verlag, pp. 335-345, 1997.

[31] T. Back, "On the behavior of evolutionary algorithms in dynamic environments," in Proc. Int. Corf. Evolutionary Computation, Anchorage, AK. Piscataway, NJ: IEEE Press, pp. 446-451, 1998.

[32] A. Bekkali, H. Sanson and M. Matsumoto, "RFID Indoor Positioning Based on Probabilistic RFID Map and Kalman Filtering," Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, pp. 21-27, 2007.

Hui Zhu received the B.S. degree from the Department of Physics, Anhui University, Hefei, China, in 2000 and M.S. degree from the Institute of Microelectronics, Peking University, Beijing, China, in 2005. Currently, he is a doctor student in Graduate School of Information, Production and Systems of Waseda University. His research interests include hardware and software for local positioning systems.

Syahrulanuar Ngah received the B.S degree and M.S degree from Faculty of Computer Science & Information Technology, University of Putra Malaysia in 2001 and 2003 respectively. Currently, he is a Doctorate student in Graduate School of Information, Production and Systems of Waseda University. Fields of interest are local positioning systems and Wireless communication.

Ying Xu was born on May 27, 1984 in Xi'an, Shanxi province, China. She received B.S. degree from School of Telecommunications Engineering, Xidian University, China, in 2006. She is now a master student in Graduate School of Information, Production and Systems of Waseda University. Her current research interests include indoor positioning systems and relative wireless communication technology.

Yuji Tanabe was born on October 17, 1978 in Sunnyvale, CA. He received the B.S. and M.S. degrees in Electrical Engineering from Seikei University in 2001 and 2003, respectively. He is now a doctor student in Graduate School of Information, Production and Systems of Waseda University. His current research interests include design and development of UWB antennas and circuits for high accuracy positioning systems.

Takaaki Baba was born on January 10, 1949 in Aichi, Japan. He received Ms. Degree and Dr. of Engineering from Nagoya University in 1973 and 1979, respectively. He joined Matsushita Electric Industrial Co., Ltd in 1973. From 1983 to 2002 he worked for Matsushita Electric Co., of America, involving and conducting several strategic projects such as System LSI and ASIC application, wireless communication system and electronic devices. From 1980 to 1982 he was a research fellow at UC-Berkeley. From2002 to 2003 he was a research fellow at Stanford University. Since 2003 he is a professor in the system LSI application field at Graduate School of Information, Production and Systems of Waseda University. He is a member of IEEE and served as an Executive Committee member of IEEE-ISSCC from 1995 to 2003.