Artificial Neural Network Based Model for Local Position Systems

Syahrulanuar NGAH, Hui ZHU, Kui-Ting CHEN, Yuji TANABE and Takaaki BABA,

Graduate School of Information, Production and System, Waseda University Kitakyushu-shi, Fukuoka 808-0135, Japan

Summary
In this paper an Artificial Neural Network (ANN) based model is introduced to improve the positioning accuracy in local environment. With the learning ability to deal with unknown environment, the proposed model can be used to convert received Time of Arrive (TOA) signals into corresponding positions. Supported by the good and enough training data (related data to solve the problem), the ANN based model can provide better positioning accuracy and precision, compared with previous positioning algorithms: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Moreover, the computation time used by ANN based model is reduced to an extremely short value. Comprehensive performance comparisons including accuracy, precision and computation time are presented in this paper. Results described in this paper demonstrate that the proposed model produce a high accuracy position information. The computation time used by ANN based model is only 10% and 20% of what GA and PSO used, respectively. The proposed model can be successfully used in local positioning with high-quality solution.

Key words:
Local Positioning and Neural Network Based Model.

1. Introduction

Local positioning systems get more and more interest in recent years. Many applications can get benefits from local positioning to provide useful services, such as museum tour-guide, hospital health-care and location-based handoff. The position estimations rely on received physical information, such as time of arrival (TOA), time difference of arrival (TDOA) and received signal strength indicator (RSSI), from several nearby radio access points [1-4][15][20]. Then the positioning algorithms have been developed for converting received physical information to the coordinates of tag.

Typical local positioning systems require better accuracy than what current outdoor positioning systems provide. Outdoor positioning technologies such as GPS have poor performance because of the harsh nature of local environments. Further, local positioning systems are expected to provide different types of position information such as physical space, position and orientation. Generally, the local positioning systems operate in harsher environments that impede signals propagation. The applications in real world environments often require good positioning accuracy. Normally, local positioning systems call for a smaller coverage area, compared to a typical outdoor system, and it is often desirable to limit the coverage area to a single organization.

An artificial neural network (ANN) based model is introduced in this paper for the local positioning systems. The nonlinear problems in position estimate can be solved by the proposed model. The objective of this work is to estimate the locations of tag with small error and reduce the computation time as well.

The rest of this paper is organized as follows. In Section 2, the research scenario is briefly discussed. Then the previous work on the positioning algorithm has been introduced. The proposed new extension to ANN based model is introduced in Section 3. The position estimation using ANN based model is explained in Section 4. The simulation settings and comparison result are explained in Section 5 and 6 respectively. Section 7 comprises the summary and conclusions of this study.

2. Research Scenario

In this paper, a new local positioning is introduced. Its system framework is depicted in Fig.1. Three distinct components are included in this system.

1) The first component is distance estimation. This component is responsible for estimating information about the distances between two nodes. In this research, ultrasonic transmitters are used to measure distances. The transmitter sent ultrasonic signals to the receiver and measured the transmit time. Then the distances between transmitters and receiver will be evaluated by TOA technology. When ultrasonic signal is received, the analog signal is sampled at 240 kHz and the amplitude of the signal is converted to digital data by an analog-to-digital converter [6]. This information will be used by the next components, called position evaluation.

2) Position evaluation component is the crucial part for local positioning systems. It responsible for computing a tag position accurately based on available information. An ANN based model is proposed in this paper to convert

Manuscript received July 5, 2009
Manuscript revised July 20, 2009

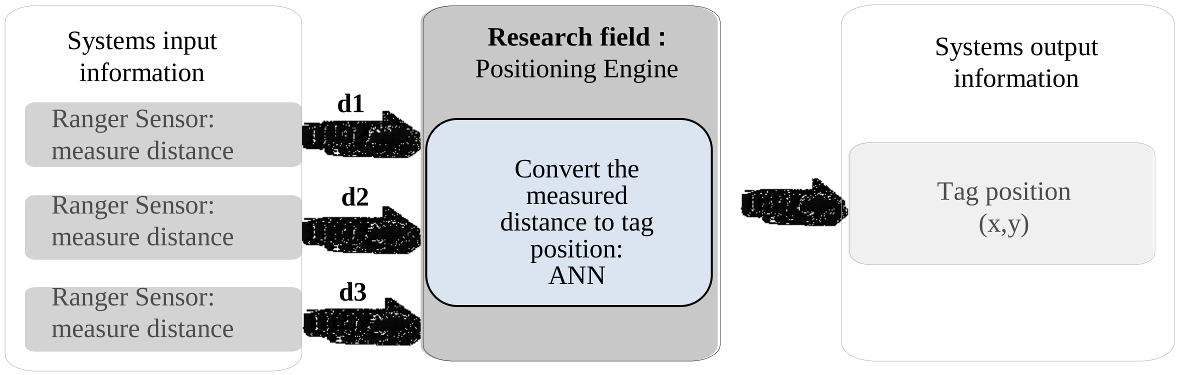


Fig.1 System framework of Local Positioning Systems.

received distance information into tag position. This information then will be passing to next component.

3) In the third component, the accurate position will be produced and used in various applications, such as museum tour-guide, hospital health-care and location-based handoff.

A configuration of local positioning with base stations (locators) and the device to be located (tag) is shown in Fig. 2(a). The exact solution can be obtained for two dimensional positioning with three locators based on the TOA measurements. However, in the real-world applications the estimation distance error E_d is ineluctable, as illustrated in Fig.2(b). The area highlighted by the dot line is the uncertain region, which is dominated primarily by the estimation distance error E_d. It is impossible to be removed and usually it considered as noised. Therefore, high positioning accuracy is more desired in this environment.

There are also many ways in which local positioning change over time [7]. First, the tag position in search space can change. Second, in the multidimensional system, the variation of position may occur on more dimensions, either independently or simultaneously. Third, the noises change at any time. Therefore, the local positioning is recognized as highly nonlinear dynamic systems with numerous noises.

In general, the positioning algorithm should be good enough and fast enough, then it is sufficient [7]. By good enough, it means the accuracy requirements are satisfied reasonably well. Moreover, by fast enough, it indicates that the computation should be finished within an extremely short time.

3. Previous Works on Position Estimate

3.1 Direct Calculation Method

The most straightforward way to estimate the position is directly solving a set of simultaneous equations based on the TOA measurements. Therefore, exact solutions can be obtained for two-dimensional positioning with three sensors [8]. It is basically a linear problem that can be finely solved with a simple Triangulation Method. The distance between sensor m and the tag is given by:

$$d_m = \sqrt{(X_m - x)^2 + (Y_m - y)^2} + E_d * rd \quad m = (1,...n) \tag{1}$$

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 estimation distance error, which is shown in Fig. 2, and rd is an uniformly distributed random number between 0 and 1. As explained in Fig.2, the exact position of tag cannot be achieved with the existence of estimation distance error E_d.

In real world environment, sometime only a few sensors can be detected, which is less than three. For two-dimension positioning, it is impossible to achieve the exact position by directly solving a set of equations in this situation. On the other hand, for an over-determined system with redundant measurements (sensors), produce a solution to the position estimation is difficult. The existence of the estimation distance error E_d makes it more complex to achieve an accurate solution. Positioning based on intersection of circles is thus inappropriate. The main difficulty is due to the fact that the equation (1) is highly nonlinear.

3.2 Optimization Based Methods

For practical applications, the position estimate algorithm should be robust and easy to implement [8]. To achieve this, the nonlinear optimization algorithms are applied for further investigation. There are many techniques in nonlinear optimization problems [8]. In this research, several practical optimization methods are applied to two-dimensional position estimation. These methods are used as the comparative measure of our development in this paper.

3.2.1 Genetic Algorithm

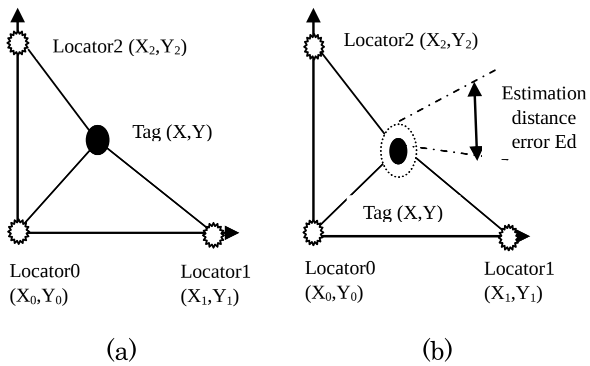


Fig.2(a) The general positioning under the idea environment and (b) positioning under the noisy environment

Genetic Algorithm (GA) is the one of the most widely known evolutionary algorithms. In GA, a problem solution is considered as the individual's chromosome and a population of such individuals strives to survive under harsh conditions. The ability to survive, represented by fitness, is what allows individuals to reproduce and generate descendants. Usually GA uses three operators: selection, crossover and mutation to obtain an optimal solution. A time-difference-of-arrival (TDOA) location algorithm based on GA has been proposed [17]-[18]. This method has higher accuracy than the fixed coverage algorithm and follows closely to the Cramer-Rao lower bound (CRLB) even at high noise level. Numerical simulations show that GA has higher accuracy.

3.2.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) algorithm is a new sociologically inspired stochastic optimization algorithm. The PSO algorithm is easy to implement, has few parameters, and has been shown to converge faster than traditional techniques for a wide variety of benchmark optimization problems. The PSO algorithm can produce good results in a faster, cheaper way, compared with other methods. PSO has been successfully applied in many research and application areas [10]-[13]. In PSO, the velocity and position of the particle *i* are updated based on following equations:

$$\upsilon_{ix}^{t+1} = \omega * \upsilon_{ix}^{t} + c_1 * r_1 * (p_{lx} - x_i^t) + c_2 * r_2 * (p_{gx} - x_i^t) \quad (2)$$

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

$$\omega = \omega_{\max} - t * (\omega_{\max} - \omega_{\min}) / T \quad (4)$$

where the acceleration constants c_1 and c_2 are constant, r_1 and r_2 are uniformly distributed random numbers between 0 and 1, 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 iterations. PSO method has been employed for source localization, using TDOA measurements. For sufficiently small noise conditions, it is shown that the accuracy of the PSO method approaches CRLB and outperforms the least squares method [16].

The GA and PSO have been proved to be effective in position estimate. Heuristic algorithms however, like GA and PSO are time consuming. For many real world applications, these algorithms can run for a long time, even when it is executed on a high performance workstation. Nonetheless, fast computation is more desirable. The positioning algorithm is necessary to be as fast as possible to complete the computation within a very short time. To decrease the processing time of GA and PSO, the most straightforward way is by using a small number of particle and iterations. The impact if the particle number and iterations are reduced, the positioning accuracy becomes intolerable.

4. ANN based model

The objective of this work is to estimate the locations of the tag with small error and reduce the computation time as well. To achieve these objectives, ANN based model has been chooses. The reason is an ANN has the learning ability. With that, ANN can achieve:

1) High accuracy and precision

For the indoor positioning systems, a good positioning accuracy and precision are expected. Usually, ANN is trained (learned) to perform a particular function by providing related training (learning) data to adjust the values of the connections (weights) between elements. That particular input will leads to a specific output. To improve the accuracy, a lot of training data can be used during the training stage. For example, in an area of 10 m * 10m, 1000 set of training data will be applied during the training stage. Once the training completed, ANN has ability to produce better positioning accuracy with high precision.

2) Less computation time

For the proposed ANN model, the acquired knowledge after the training (learning) stage, the model will be saved in the computer. When receiving physical signals, the results will be produced by ANN model. Since ANN is composed of elements operating in parallel and already has a specific knowledge for specific problems, the computation time can be reduced to an extremely short value.

From above discussion, it is found that ANN model is possible to be used to achieve good positioning accuracy and small computation time. ANN has ability to approximate very complex functions. It provides results even for inputs which were not included on training data sets [14]. With the learning ability to produce a solution for a specific problem, supported by related training data,

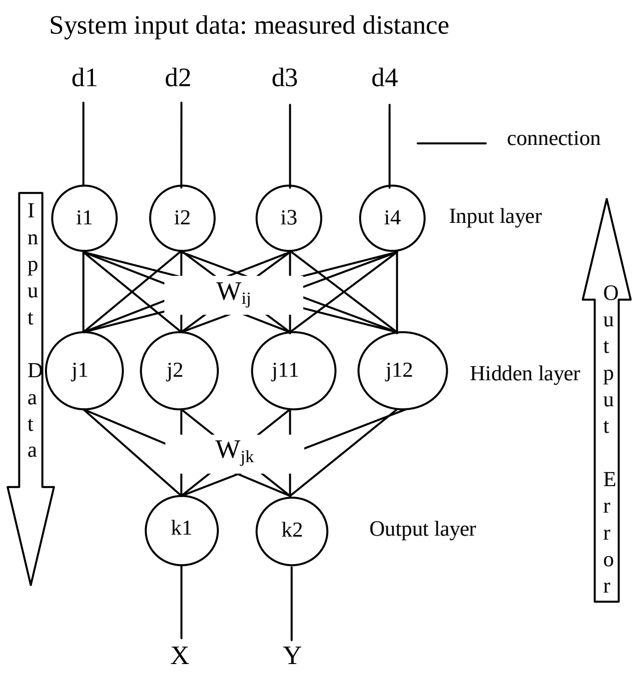


Fig.3 General Neural Network

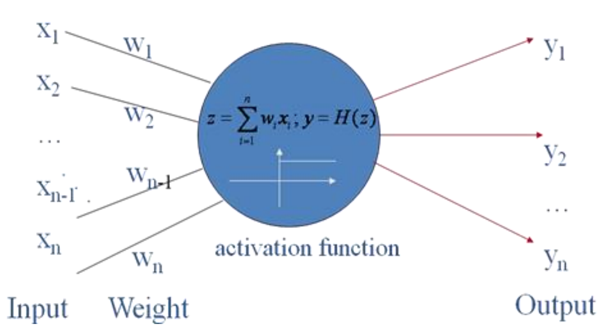


Fig.4 Abstracted model of neuron with connection

ANN can overcome the problem where normal computation cannot solve.

4.1 The structure of proposed model

Neural networks consist of a large class of different architectures. There are several choices to be made when implementing ANN to solve a problem. A good performance of ANN will be achieved only if these selections were suited to the problem to be solved [19]. The structure of proposed model is depicted in Fig.3. This model is composed of input, hidden and output layers. The number of neurons in the input and output layers are determined by system inputs (for measured distances between tag and locators) and outputs (coordinates of tag: x and y) respectively. The proposed ANN based model consisted of a mapping from a set of input variables containing information about the received distances from locators (using TOA technique) onto a set of two outputs variables representing the two dimensional tag location (x, y).

4.2 The training of proposed model

Unlike the other algorithms, ANN first needs to be train to acquire the knowledge before can be used to solve the problem. For the training phase, a set of measured distances between tag and locator has been applied to the input of the ANN. This information flows sequentially through the hidden layer to the output layer. The training is done by using two passes. The forward pass is used to evaluate the output of the ANN for the given input in the existing weights. In the reverse pass, the difference in the ANN output with the desired output is compared and feed back to the ANN as an error to change the connection weights of the network. This process will loop continuously until the ANN achieved the desired output. Figure 4 shows the abstract of the neuron model with the weight connection.

The output of ANN gives the estimated value of tag's position. The proposed artificial neural network is used to simulate the idea position environment. Therefore, the estimate distance error E_d does not exist in the training stage. Table 2 shows an example of training data set. Each data consists of four input data (measured distance d1, d2, d3 and d4) and two output data (the true tag position x and y). These data will be used in the training stage in order to ensure the ANN can perform with high accuracy. The steps for the whole training processes are shown in Fig.5. Let consider $X = [x_1(p), x_2(p) \ldots x_n(p)]$ be the input vector . The weight matrix connection between input layer and hidden layer is represented as $W = (w_{ij})$. Similarly the weight matrix connection between hidden layer and output layer represented by $w = (w_{jk})$. In step 1, the weight connection w_{ij} and w_{jk} are randomly set. Then, the input data (measured distance between locator and tag) will be feed into input layer. The data will flow from input layer to hidden layer. In step 2, each neuron in hidden layer will calculates the total input received and produces the output by applying sigmoid activation function. It can be written as:

$$y_{hidden} = \text{sigmoid}[\sum_{i=1}^{n} x_i(p) * W_{ij}(p)] \quad (5)$$

The output from hidden layer will become input to output layer. It will be used to calculate the output in output layer in step 3. It given by:

$$y_{output} = \sum_{i=1}^{n} y_{hidden}(p) * w_{jk}(p) \quad (6)$$

Comparison between the actual output (y_{output}) from the network and the desired output (y_d) is done in step 4. The output error (e_y) is the different between y_{output} and y_d.

$$e_y = y_d - y_{output} \quad (7)$$

To update the v_{ij},(Step 6) first, calculate the error gradient for neurons in the output layer.

$$\delta_k(p) = y_{output} * [1 - y_{output}] * e_y \quad (8)$$

Then calculate the weight correction with the equation:

$$\Delta W_{jk}(p) = \alpha * y_{hidden}(p) * \delta_k(p) \quad (9)$$

Update the weight at the output neuron finally can be written as:

$$W_{jk}(p+1) = W_{jk}(p) + \Delta_{jk}(p) \quad (10)$$

Step 7 is to update the w_{ij}. First, error gradient for neurons in hidden layer is calculated by equation:

$$\delta j(p) = y_{hidden}(p) * [1 - y_{hidden} \sum_{i=1}^{n} \delta k(p) * w_{jk}(p)] \quad (11)$$

Then calculate the weight correction with the equation:

$$W_{ij}(p) = \alpha * x_i(p) * \delta_j(p) \quad (12)$$

where α is a learning rate.
Update the weight at the hidden neuron is represented by:

$$w_{ij}(p+1) = w_{ij}(p) * \Delta w_{ij}(p) \quad (13)$$

The processes then get back to Step 2 and repeat until the desired output achieved. The training process is considered as an extra cost that needs to be paid by ANN to achieve higher accuracy compared with GA and PSO. Figure 6, shows the summary in term of total time taken to identify the tag location.

4.3 Training results

Table 3 shows the training result for different numbers of training data. The ANN can achieve better accuracy with the big number of training data and takes longer time to

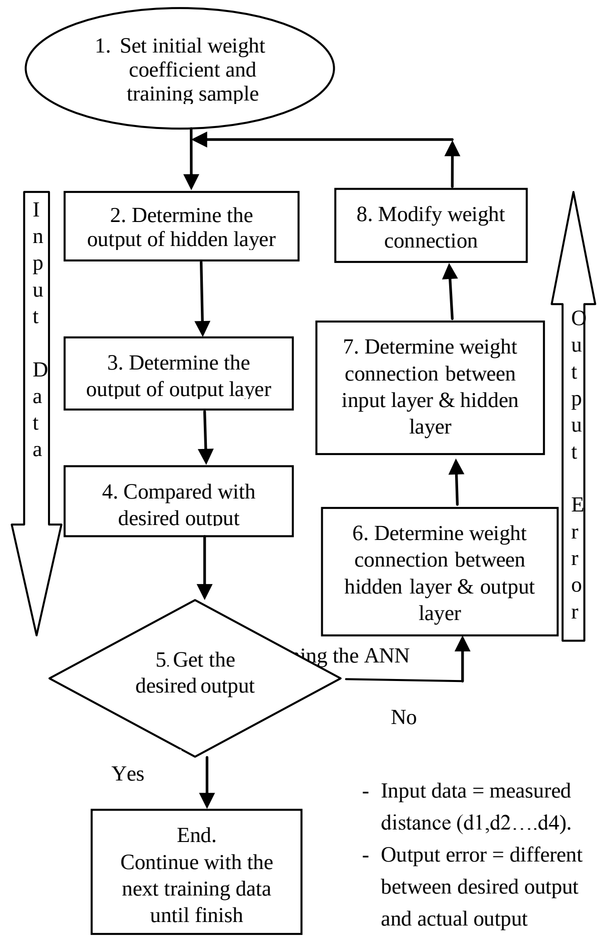


Fig.4 Training step for ANN

Table 1. ANN parameters

Parameters	Value
No. of inputs neurons	Number of locators
No. of hidden neurons	12
No. of output neurons	2
Hidden layer function	Sigmoid
Learning algorithm	Back propagation

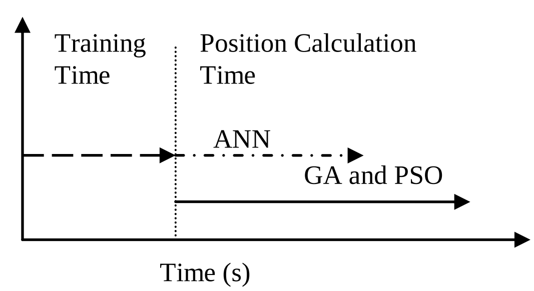


Fig 6. Total time needed to calculate tag position

complete the training. For the three locators, by increasing the number of training data from 100 to 500, ANN accuracy improved 5.5% (0.55m to 0.52). The extra time

Table 2. 1000 Training Data Set

Training data	Input1 (d1)	Input2 (d2)	Input3 (d3)	Input4 (d4)	Output1 (x)	Output2 (y)
1	9.78	2.36	12.22	7.70	9.50	2.31
2	7.77	6.25	7.95	6.47	6.06	4.86
3	11.73	7.70	9.23	2.62	8.91	7.62
………	………	………	………	………	……	……
1000	4.57	5.44	10.82	11.22	4.56	0.19

Table 3. Training Results

Total Training Data	3 Locators		4 Locators	
	Accuracy(m)	Training Times (s)	Accuracy(m)	Training Times (s)
10	3.12	2.1	1.98	2.1
100	0.55	5.3	0.52	5.3
500	0.52	9.6	0.48	9.7
1000	0.51	13.3	0.43	13.4
2000	0.51	18.2	0.42	18.2

Table 4. Final Weight Connection between Input neuron - hidden neuron – output neuron after completed 1000 training data for number of locator = 3.

Hidden Neuron	Input neuron (i1)	Input neuron (i2)	Input neuron (i3)	Output neuron (k1)	Output neuron (k2)
j 1	-0.094903	-2.0434	2.4822	2.6081	0.65288
j 2	-2.2628	-2.6416	-2.3458	4.5186	3.8595
j 3	-1.1373	-2.5998	-1.0027	3.8724	-1.2864
j 4	-0.38274	1.4934	2.442	1.768	-4.172
j 5	-1.3206	5.0755	2.8309	-1.3805	1.0122
j 6	1.1658	0.42408	-3.1492	-0.04424	-0.19662
j 7	-1.27	1.1423	2.5514	2.0639	10.442
j 8	0.60166	-0.41333	1.1144	-2.8354	7.5119
j 9	1.3066	0.2583	0.78204	-1.2491	-2.3528
j 10	2.3656	-4.7496	-4.4941	-0.82738	-4.0602
j 11	-2.5736	-0.86338	2.3991	5.2715	0.8625
j 12	0.74194	0.94055	2.7858	1.3573	4.1001

Table 5. Locator Coordinate Setting

Locator	Coordinate (m)
Locator 1	(0,0)
Locator 2	(10,0)
Locator 3	(0,10)
Locator 4	(10,10)

needed is 44.8% (4.3 seconds). By increasing again the number of training data to 1000, once again, the ANN accuracy improved by 1.9% (0.52m to 0.51m) with 27.8% (3.7 seconds) added. No further improvement in accuracy even the number of training data increased to 2000. In case of four locators; the ANN accuracy improves from 0.52 m to 0.48m (7.7%) with the data training number increased from 100 to 500. The training time is increased from 5.3 seconds to 9.7 seconds (45.4%). ANN accuracy improved 10.4% when the number of training data increased from 500 to 1000 with additional 3.7 seconds (27.6%). Meanwhile, from 1000 to 2000 training data, the accuracy improved 2.3% with additional 4.8 seconds (26.3%). By taken into account the accuracy and time taken to complete the training, the number of training data equal to 1000 is the ideal number for training the ANN in 10m*10m search space for this paper.

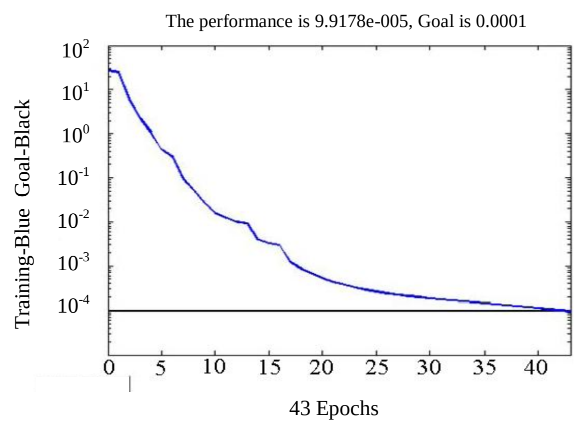


Fig.7 Training graph result for 1000 training data and number of locator = 3

Simulation result showed that 12 neurons in hidden layer are adequate to produce good solution. The same training process will be given to the ANN while using two and three locators. Once complete the training, the network do not need to be train again unless if the number of locator is increased to more than four or the search space is different than what the network has been train. From the training graph result in Fig.7, it shows that, ANN needs 43 epochs to achieve performance goal. Mean Square Error (MSE) is an indicator of network's performance. In this case, the MSE is set to 0.0001 as network performance goal. Once the ANN achieved the performance goal, the ANN is considered to have converged. Table 4 shows the final weight connection between input layer – hidden layer – output layer after complete the training with 1000 training data set. However, these weights will change if the network is being trained again.

5. Position Estimation

There are three phases involved in running this simulation. First, system initialization, then estimation process and finally estimate result and error. Numbers of locator to be used in the simulation are two, three and four.

5.1 System Initialization

This system is initialized in a square room with side length L. Locators are assumed to be deployed in the room stationary and their coordinates are predefined as shown in Table 5. Tag is deployed randomly. Distances between tag and locators are measured in TOA technique, as defined in equation (1). As an initial condition, these distances expressed as d1, d2, until dn, corresponding to the locator used as inputs. For example if the simulation using three locators (Locator 1, 2 and 3), the distances are express as d1, d2 and d3. Tag position (x, y) is unknown at this stage. Above situation is system input conditions.

5.2 Estimation Process

Based on above assumption, system process for searching the tag position is performed. In this step, the program uses the ANN that has been explained in section four to estimate tag position. The input data (measurement distance between locator and tag) will go through the input layer, hidden layer until reached the output layer.

5.3 Estimated Results and Error

Finally, the tag position is produced by the ANN based model. The position error between true and estimated values is expressed as E_p. If a smaller position error E_p is achieved, the benefit of ANN based model is proved. The position error E_p is calculated by following equation:

$$E_p = \sqrt{(X - x)^2 + (Y - y)^2} \tag{14}$$

where (X, Y) is the true position of tag and (x, y) is the optimal solution estimated using ANN, which is output data of the proposed ANN based model. In this paper, the performance of ANN method is evaluated and compared with previous methods. Simulations are carried out over 1000 runs in the problem space. The tag position is randomly placed in each run. $E_{p\cdot average}$ is used to evaluate the performance of each positioning algorithm, which is expressed as:

$$E_{p,average} = \sqrt{\sum_{r=1}^{1000} \left(E_p^{\,2}\right) \Big/ 1000} \tag{15}$$

6. Simulation Results

In this paper, the positioning results that estimated by GA and PSO are compared with the ANN based model. The position accuracy, precision and time taken for all algorithms will be discussed in this section. This system is implemented on a single 2.4 GHz Pentium 4 processor, 1.5 GB RAM, and Windows XP Pro OS. Software programs for all positioning algorithms are prepared with MATLAB language.

6.1 Parameter Setting

Since the optimization function used in this paper may have the global minimum at or close to the origin of the search space, for GA and PSO, the initial population is randomly distributed in the entire search space. This method is used to observe the performance of each concept introduced in this paper. For each algorithm, simulations are conducted over 1000 runs. The tag position is randomly placed in each run. The estimation results are investigated and compared.

The number of population is set to 20 in GA. The number of mutation children is selected as five and the number of elitist is set to five. The maximum number of iterations is 50. In the PSO method, particle number is set to 20 and the maximum number of iterations is selected as 50. The maximum velocity is limited to the upper value of the dynamic range. The acceleration constants c1 and c2 are set to two. The inertia weight ω is decreased linearly from 0.9 to 0.4.

The problem space is 10m*10m. Coordinates of locators are same as previous definition. Locator numbers n are increased from two to four. Since the exact value for estimation distance error E_d is unknown, then the values of 0.2m and 1.0m are set as estimation distance error E_d for every number of locator used in each running simulation.

6.2 Positioning Accuracy and Precision

Accuracy and precision are two main performance parameter to evaluate Indoor Positioning Systems [15][20]. Accuracy means the average distance error. Figure 8(a) and (b) shows the accuracies result for all of the algorithms when estimation distance error E_d is set to 0.2m and 1.0m respectively. From 1000 times position calculation, PSO can achieve same accuracy with ANN when the number of locator equal to four in Fig.8(a). Meanwhile GA can achieve slightly better accuracy compared with ANN when the number of locator equal to two in Fig.8(b). Other cases, ANN produces the smallest average positioning error $E_{p.average}$ compared with GA and PSO algorithms. Graph pattern also just like expected where the performance of all algorithms increased with the increasing the number of locators.

Precision is defined as the success probability of position estimations with respect to predefined accuracy [15]. Usually, Cumulative Distribution Function (CDF) is being used to compare the precision of two or more algorithms. In practice, CDF is described by the percentile format. For example, one system has a location precision of 90% within 2.3m (the CDF of distance error of 2.3m is 0.9), and 95% within 3.5m; another one has a precision of 50% within 2.3m and 95% within 3.3m. The first system has a higher precision since its distance error concentrated in small values [20]. Results from 1000 times position calculation for each algorithm are being used to produce the CDF for evaluating the precision. Figure 9(a) and (b) are the CDF for number of locator equal to three and four respectively with the estimate distance error E_d is set to 1m. From Fig.9(a), it shows that, ANN could achieve a location precision of 50% (500 times) within 0.5m error and 81%(810 times) within 0.75m error. Meanwhile GA has almost the same precision location with ANN with achieved 50% (500 times) error less than 0.5m. For the error less than 0.75m, GA only achieved 72% (720 times). PSO has the lowest location precision with 49% (490 times) within 0.5m and 62% (620 times) within 0.75m error.

In Fig.9(b), the precision for all algorithms are increased when the number of locator increased from three to four. It

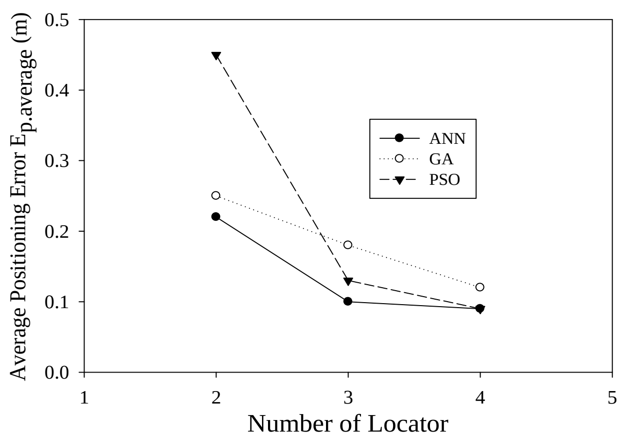


Fig.8(a) Accuracy achieved for E_d is set to 0.2m

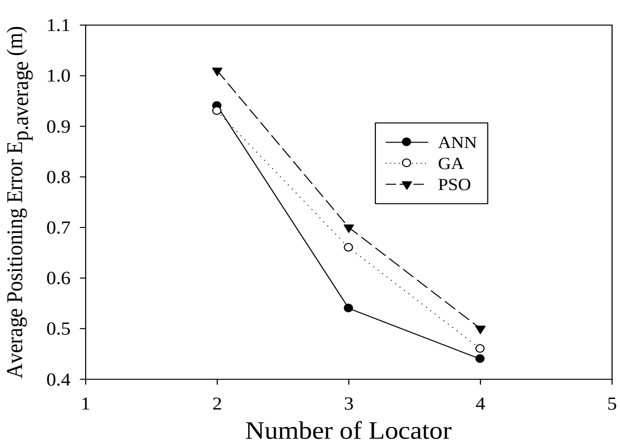


Fig.8(b) Accuracy achieved for E_d is set to 1.0m

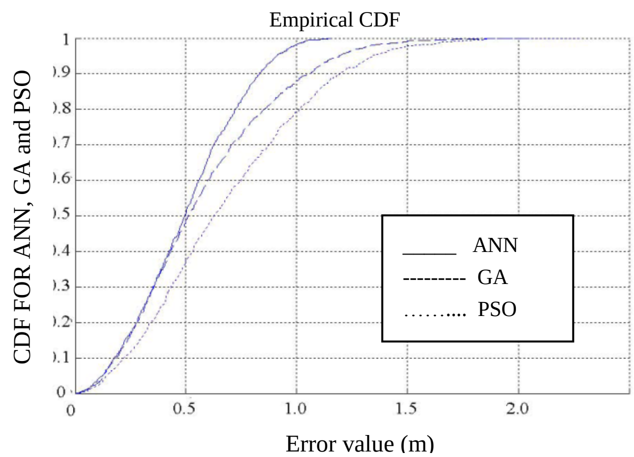


Fig.9(a) Cumulative Distribution Function (CDF) of distance error for ANN, GA and PSO with number of locator = 3 and E_d = 1.0m

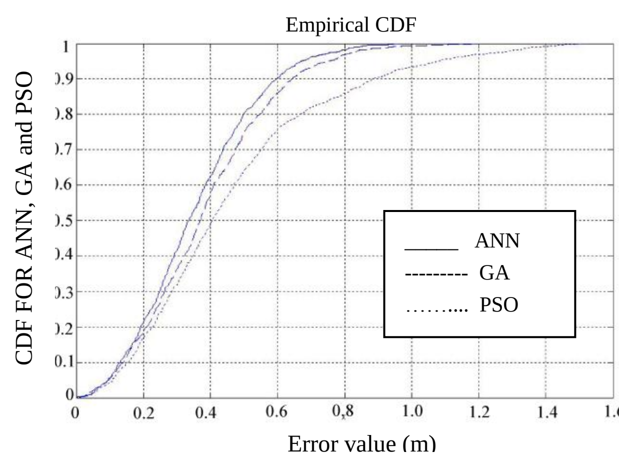
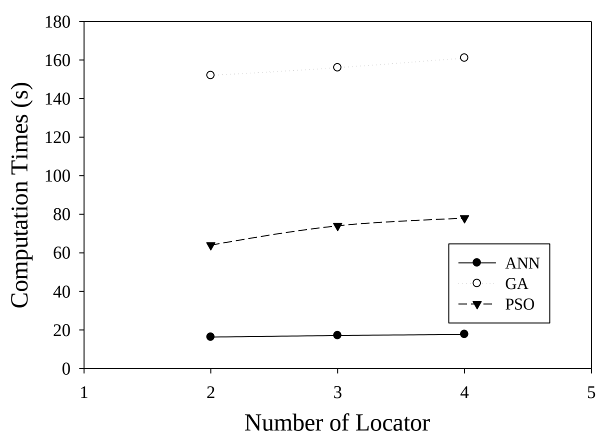


Fig.9(b) Cumulative Distribution Function (CDF) of distance error for ANN, GA and PSO with number of locator = 4 and E_d = 1.0m can be seen from the graph pattern getting more steeper compared with graph pattern in Fig.9(a). However, ANN still achieved the highest precision. ANN has a location precision of 78% (780 times) within 0.5m and 98% (980 times) within 0.75m. At the same time, GA has a location precision of 72% (720 times) within 0.5m and 93% (930 times) within 0.75m. Again, PSO has the worst location precision with only 60% (600 times) within 0.5m and 82% (820 times) within 0.75m.

From the above results, it is concluded that ANN has better precision compared with the other algorithms. The costs for preparing the network with 1000 training data become advantages when ANN based model algorithms successfully estimating the tag position in nonlinear environment.

6.3 Computation Time

Delay is another aspect of measuring the performance in Indoor Positioning Systems which contains the delay of measuring, calculating the positions of estimated target and forwarding position information [15]. The acquired knowledge during the training stage clearly gives advantages to the ANN in producing the result. From Fig.10, it is clearly showed that the calculation time used by ANN based model to estimate the target position is always lower than that being used by PSO and GA. The computation time (include training time) of ANN based model is only 10% of what GA algorithm and 20% PSO algorithm used.



F
ig.10 ANN computation time always less than GA and PSO

In general, the ANN method provides a good positioning precision throughout the search space. At the same time, the computation time is significantly less compared with the other optimization algorithms (GA and PSO). These properties make ANN based model more practical for the local positioning systems.

7. Conclusions

The use of an artificial neural network algorithm as a local positioning system is introduced. With the learning capabilities and 1000 related training data, the proposed model is particularly suitable for providing higher positioning accuracy in search space 10m*10m even in the presence of noise. Even having an extra cost for preparing (training) the network, the proposed model and TOA technique can give an advantages and improvement on local position estimation.

Performance comparison in term of accuracy, precision and time delay has been done with other two optimization algorithm (GA algorithm and PSO algorithm). All algorithms will run 1000 times position calculation. Five from six cases (number of locator = 2, 3 and 4 running with estimation distance error 0.2m and 1.0m) ANN produces better position accuracies. Turn to precision, it is obviously, ANN achieved higher precision than GA and PSO. ANN can achieve 78% or 780 times' error within 0.5m and 98% (980 times) within 0.75m. Therefore, compared with GA and PSO, the proposed model is more attractive since smaller position error could be achieved even in the presence of noise.

Moreover, the computation time is reduced to an extremely short value. By running on stated machine in section 6, ANN based model needs only 10% and 20% from total times of what GA algorithm and PSO algorithm used respectively for completing 1000 times positioning calculations including the training times.

Acknowledgment

This work was supported by funds from MEXT via Kitakyushu innovative cluster projects.

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", in IEEE Trans. on Vehicular technology, vol. VT-21, pp. 9-16, Feb 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, August 1976.

[5] "Information technology AIDC techniques - Harmonized vocabulary, Part 5 – Locating systems", pp. 1-10, 2007.

[6] Bo Huang, Peng Zhang, Syahrulanuar Ngah and Takaaki Baba, " A Method for Concealed Object Detection Using a Code Division Multiple Access-like Method with a Modulated Ultrasonic Wave", Journal of Signal Processing, Vol. 12, No. 6, pp.489-496, Nov. 2008.

[7] 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.

[8] I. Oppermann, M. Hamalainen and J. Iinatti, "UWB Theory and Applications", John Wiley & Sons Ltd, England pp.176, 2004.

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

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

[12] 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.

[13] 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.

[14] Rafal Szumny, Jozef Modelski, "Neural Network in Indoor Positioning System Based on Power Delay Profile" EUROCON 2005. The International Conference on "Computer as a Tool" vol 2, 21 – 24 Nov 2005, pp 1726 – 1729.

[15] Yanying Gu, Anthony Lo and Ignas Niemegeers "A Survey of Indoor Positioning Systems for Wireless Personal Networks" IEEE Communication Surveys & Tutorial, VOL 11, No 1, First Quarter 2009.

[16] Kenneth W. K. Lui, Jun Zheng, and H. C. So, "particle swarm optimization for time-difference-of-arrival location", in Proc. 2007 EURASIP , pp. 414-417, 2007.

[17] HOU Hui-fang, LIU Su-hua, YANG Tie-jun, "TDOA Location Technique Based on Genetic Algorithm and Simulated Annealing Algorithm", Computer Engineering, Vol. 34, No. 12, pp. 172–175, June 2008.

[18] Li, L. and Wie, F., "Position Estimation by Improved Genetic Algorithm for Hyperbolic Location," 14th IST Mobile & Wireless Communications, 2005.

[19] G. Dreyfus, "Neural Networks, methodology and applications", Berlin: Springer-Verlag, 2005.

[20] Hui Liu, Houshang Darabi, Pat Banerjee and Jing Liu, "Survey of Wireless Indoor Positioning Techniques and Systems" IEEE Transaction On System, Man and Cybernatic – Part C, Vol.37 No 6, Nov 2007.

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.

Hui ZHU received the B.S. degree from the Department of Physics, Anhui University, Hefei, China, in 2000 and M.E. degree from the Institute of Microelectronics, Peking University, Beijing, China, in 2005 and Dr. of Engineering from Graduate School of Information, Production and Systems of Waseda University, Japan in 2008. His research interests include hardware and software for real-time location systems.

Kui-Ting CHEN was born on Sep 27, 1983 in Taipei, Taiwan. He received B.S. degree from School of Electrical Engineering, I-SHOU University, Taiwan, in 2006. He is now a master student in Graduate School of Information, Production and Systems of Waseda University. His current research interests include positioning systems and relative wireless communication technology

Yuji TANABE was born on October 17, 1978 in Sunnyvale, CA. He received a B.S. and an M.S. degree in Electrical Engineering from Seikei University in 2001 and 2003, respectively. He is currently pursuing a doctoral degree at the Graduate School of Information, Production and Systems of Waseda University. His current research interests include the design and development of UWB antennas and circuits for various applications.

Takaaki BABA was born on January 10, 1949 in Aichi, Japan. He received an M.S. degree and a Dr. of Engineering degree from Nagoya University in 1973 and 1979, respectively. He joined Matsushita Electric Industrial Co., Ltd in 1973. Between 1983 and 2002 he worked for Matsushita Electric Co. in the US, where he was involved in several strategic projects, such as System LSI and ASIC applications, wireless communication systems and electronic devices. He was a research fellow at UC Berkeley between 1980 and 1982, and at Stanford University between 2002 and 2003. Since 2003, he has held the position of a professor in the field of system LSI application at the 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.