International Journal of Artificial Intelligence,
ISSN 0974-0635; Int. J. Artif. Intell.
Autumn (October) 2011, Volume 7, Number A11

An Application of Ant Colony Optimization, Kalman Filter and Artificial Neural Network for Multiple Target Tracking Problems

Endra Joelianto[1] and **Ifan Wiranto**[2]

[1]Instrumentation and Control Research Group
Faculty of Industrial Technology
Bandung Institute of Technology, Bandung 40132, Indonesia
E-mail: ejoel@tf.itb.ac.id

[2]Department of Electrical Engineering
Gorontalo State University
Gorontalo, Indonesia

The main problem in multi-sensor-multi-target tracking is data association problem that leads to multidimensional assignment problems. In this paper, a multidimensional assignment problem is solved by merging ant colony optimization, Kalman Filter and artificial neural network. The ant colony optimization algorithm is first applied to acquire the number of targets, next the states of the targets are estimated using the Kalman filter, and finally, the estimation results are improved by using the artificial neural network. In ant colony optimization algorithm, it has been investigated the process of choosing initial pheromone value of ants, ants placement in the second cycle and afterward. It is found that the ant colony optimization algorithm will produce better and faster search results if the initial pheromone value equals to the visibility and new ants are placed at the initial position in the next cycle on the point of the last visit of the previous cycle of ants. The proposed traget tracking system leads to improved performances in the simulation cases.

Key Words: *Multiple Target Tracking; Ant Colony Optimization; Artificial Neural Network; Kalman Filter*

Mathematics Subject Classification (Msc): *74p99, 90c99, 92b20, 92b99, 93e99*
Computing Classification System (Ccs): *G.1.6, I.5.4*
Journal Of Economic Literature (Jel): *C32, C63*

1. INTRODUCTION

Target tracking has wide applications either military or civilian, such as tracking the ground targets, air traffic control, surveillance, and radar tracking. Target tracking is a combination of tracking techniques employed for a large number of purposes. The objectives of multiple target tracking are to identify and to estimate states of multiple targets while distinguishing them. The states of the targets consist of position, velocity and acceleration. It is known that some of the results of measurements by the sensors are not correct as often corrupted by measurement noise. This causes the number of targets is not known a priori. The problem becomes how to determine the number of targets, where the measurement results mix between the correct measurements (derived from the targets) and incorrect (false alarm) which is called data association problem. In the case of multi-sensors, for the number of sensors $N \geq 3$, the problem becomes a multidimensional assignment problem. The multidimensional assignment problem is NP (Nondeterministic Polynomial)-Hard (Kammerdiner et al., 2007), and to solve this type of problem, computational intellegence offers heuristic algorithm to deal with difficulties in developing computational algorithm (Sumathi and Paneerselvam, 2010).

Some studies for multiple target tracking based on the search methodology to solve the multidimensional assignment problem have been done by Chen and Hong, 1997; Turkmen and Guney, 2006; Turkmen and Guney, 2008. In these studies, the problem is solved by using genetic algorithms (Chen and Hong, 1997; Turkmen and Guney, 2008), tabu search (Turkmen and Guney, 2006), genetic algorithm and particle swarm optimization (Badamchizadeh et al., 2010). In several studies that apply the search methodology to traveling salesman problem, namely genetic algorithms, simulated annealing, and ant colony optimization algorithm (ACO), it was found that the ant colony optimization algorithm produces better solutions compared with other algorithms in terms of shorther tour length and less iterations. Ant colony optimization has also shown comparable results with those acquired by other all-purpose heuristic algorithms in implementation to other combinatorial optimization problems such as the quadratic assignment problem, graph coloring, job-shop scheduling, sequential ordering, and vehicle routing (Bonabeau et al., 1999). In this paper, the ant colony optimization algorithm is proposed to solve the multidimensional assignment problem in multiple target tracking.

Ant colony optimization is a biomimetic model of ant behavior known to be able to find the shortest distance between nests and food sources (Dorigo, 1992; Dorigo et al., 1996; Bonabeau et al., 1999; Di Caro and Dorigo, 1998; Dorigo and Stützle, 2004; Duan et al., 2006; Sumathi and Paneerselvam, 2010). The ACO algorithm uses a set of artificial ants that work together in order to obtain solution of problems by means of pheromone deposition. In a colony of ants, when the ants walk, they place chemical substance called pheromone in their path. With the intercession of this pheromone occurs indirect communication and exchanges information among ants while building a solution. The ants create a positive feedback choice based on preference visit of the path that increases the deposition of the pheromone. Very soon, all ants will use the path with more pheromone. The probability that an ant takes a path increases with the number of ants that previously chose the same path.

The Kalman filter, developed by R.E. Kalman, 1960; Kalman and Bucy, 1961, is extensively used in diverse fields such as navigational and guidance systems, radar tracking, sonar ranging, and satellite orbit determination, seismic data processing, nuclear power plant instrumentation, econometrics, etc. The application of Kalman Filter in trajectory estimation was started during the satellite orbit determination in 1960's (to illustrate the Ranger, Apollo, and Mariner missions) (Naidu, 2003). It was the first full implementation of the Kalman filter to give precise estimates of a complex real time process with high measurement noise. The merging of the Kalman filter algorithm with target tracking has satisfied the requirement for data accuracy in real time target tracking and has produced a vital target tracking systems to the present day. The computational efficiency of the Kalman filter and its ease of realization in computer programming have made it as an ideal filter for target tracking applications combining image processing and filtering/estimation. The target tracking problem is frequently obscured by the measurement noise. The data measured by tracking devices needs to be separated out from the noise in order to predict the actual path of a moving target (Bar-Shalom et al., 2001).

Artificial neural networks are modern term of neural networks that are composed of interconnected artificial neurons or nodes that are directly connected to each other and exchange information. The artificial neural network is a program which is created to imitate the functions of biological neurons circuit. A real biological system model is not necessarily needed by the artificial neural networks in solving artificial intelligence problems (Sumathi and Paneerselvam, 2010). The artificial neural network is a method which proves to be very good for building a complex relationship and nonlinear between input and output data (Kosko, 1992; Krose and van der Smagt, 1998). The artificial neural networks are built by layer networks consist of neurons which calculate a function of their inputs and send the result to the neurons in the next layer. The input signal is fed forward from one layer to the subsequent layer through the networks. The output of a neuron is typically a nonlinear activation function of a weighted combination of the incoming signals and a threshold value. During learning process, the weight values are increased or decreased to alter the overall function of the network

In this paper, the ACO, the Kalman filter and the artificial neural networks are merged into a target tracking system as shown in Figure 1. The proposed system is divided into two mains parts, the ACO algorithm-the Kalman filter and the artificial neural network (ANN). The multiple traget tracking problem is first solved by using the ACO algorithm to obtain the the actual number of the targets by separating the true targets and the false targets. After that, the Kalman filter is used to estimate the states of each target. The Kalman filter is an optimal filter which is extensively used to estimate the state of linear dynamic systems (Grewal and Andrews, 2008). Finally, inaccuracies of the estimation by the Kalman filter are then revised by using the artificial neural network. The paper extends the introduction by adding more overview and references that show the computational intelligence has received considerable attention in multiple traget tracking problems than the previous version of the paper appeared in Wiranto and Joelianto, 2010. In addition, analysis and discussion have been improved by giving the complete parameters in the simulation and by displaying the related algorithms and results in figures.

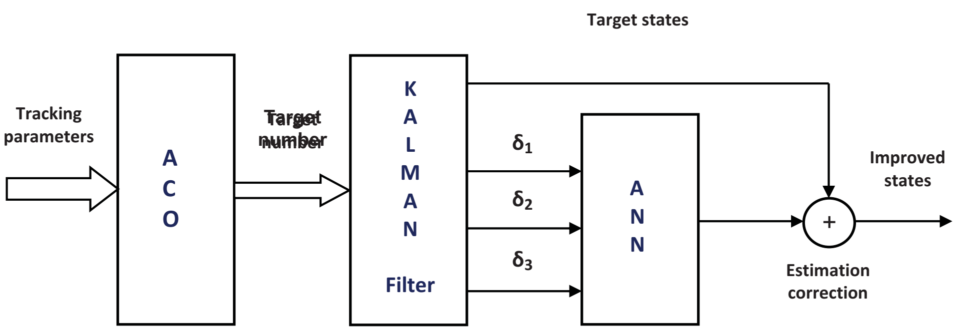

Figure 1 Diagram block of a proposed target tracking system

2. MULTI SENSOR FOR MULTIPLE TARGET TRACKING

Suppose there are N sensors used to find targets in a space. There is a number of targets q (q is unknown) in the surveillance view. The position of the target q is defined by $\theta_q = (x_q, y_q, z_q)^T$. The detection probability of sensor s is denoted by P_{ds}. Suppose the number of measurements from sensor s is n_s, s = 1,2,3, ..., N. The i_s-th measurement of sensor s is denoted as z_{si_s}, i_s= 1,2,3,…,n_s and assumed that these measurements are derived from actual targets plus Gaussian noise $N(0,\sigma_s^2)$. A target may not be detected or appears as a false alarm on each scan (Stimson, 1998). To simplify notation, a dummy measurement $z_{s0}(i_s=0)$ is added to the measurement set of each sensor to describe a false alarm and also for the association of an incomplete measurement of the target caused by miss detection. The false alarm of a data set $Z(k)$ is represented by Z_{γ_i} with $\gamma_i=(0,0,\cdots,0,i_k,0,\cdots,0,0)$ and the miss detection is represented by Z_{γ_i} with $\gamma_i=(i_1,i_2,\cdots,i_{k-1},0,i_{k+1},\cdots,i_{N-1},i_N)$.

The likelihood function that an N-tuple of measurement $Z_{i_1,i_2,\cdots,i_N}=\{z_{1i_1},z_{2i_2},\cdots,z_{Ni_N}\}$ derived from the same target with the known state θ_q is described in (Popp et al., 2001) as folows

$$\Lambda(Z_{i_1,i_2,\cdots,i_N}\mid q)=\prod_{s=1}^{N}[P_{ds}\quad p(z_{si_s}\mid\theta_q]^{u(i_s)}[1-P_{ds}]^{1-u(i_s)} \tag{1}$$

where $p(z_{si_s}\mid\theta_q)$ is the probability density function of the target q, and $u(i_s)$ is an indicator function, which indicates that if $i_s\neq 0$. The sensor s detects the target q, then $u(i_s)=1$, otherwise if $i_s=0$, the sensor s does not detect the target q, then $u(i_s)=0$. The likelihood function that the measurements are all spurious or unrelated to target q, i.e. $q=\emptyset$, is given by

$$\Lambda(Z_{i_1,i_2,\cdots,i_N}\mid\theta_q=\emptyset)=\prod_{s=1}^{N}[\frac{1}{\psi_s}]^{u(i_s)} \tag{2}$$

where ψ_s is the volume of the surveillance view (field of view) of the sensor s. Cost function to associate a measurement of N-tuple with a target q is given by

$$c_{i_1,i_2,\cdots,i_N}=-\ln\frac{\Lambda(Z_{i_1,i_2,\cdots,i_N}\mid q)}{\Lambda(Z_{i_1,i_2,\cdots,i_N}\mid q=\emptyset)} \tag{3}$$

Because θ_q is unknown, it is usually replaced by its maximum likelihood estimation, i.e.

$$\hat{\theta}_q=\arg\max_{\theta_q}\quad\Lambda(Z_{i_1,i_2,\cdots,i_N}\mid q) \tag{4}$$

The cost function is then defined by

$$c_{i_1,i_2,\cdots,i_N}=\sum_{s=1}^{N}\{u(i_s)[\frac{1}{2}[z_{si_s}-\hat{\theta}_q]^T\sigma_s^{-2}[z_{si_s}-\hat{\theta}_q]+\ln\frac{\sqrt{2\pi\sigma}}{P_{ds}.\psi_s}]-(1-u(i_s))\ln[1-P_{ds}]\} \tag{5}$$

The goal is to find the most likely set of N-tuple of measurement so that each measurement is assigned to a target or declared false, and each measurement is assigned to one target at most. This problem can be formulated as an N-multidimensional assignment problem (Chen and Hong, 1997) which is described by the following expression.

Minimize:

$$\min_{\chi_{i_1,i_2,\cdots,i_N}} \sum_{i_1=0}^{n_1}\sum_{i_2=0}^{n_2}\cdots\sum_{i_N=0}^{n_s} c_{i_1,i_2,\cdots,i_N}\,\chi_{i_1,i_2,\cdots,i_N} \tag{6}$$

subject to:

$$\sum_{i_2=0}^{n_2}\cdots\sum_{i_N=0}^{n_N}\chi_{i_1,i_2,\cdots,i_N}=1, \qquad i_1=1,\cdots,n_1$$

$$\sum_{i_1=0}^{n_1}\cdots\sum_{i_{k-1}=0}^{n_{k-1}}\sum_{i_{k+1}=0}^{n_{k+1}}\cdots\sum_{i_N=0}^{n_N}\chi_{i_1,i_2,\cdots,i_N}=1, \qquad \text{for } i_k=1,\cdots,n_k \text{ and } k=2,3,\cdots,N-1$$

$$\sum_{i_2=0}^{n_2}\cdots\sum_{i_{N-1}=0}^{n_{N-1}}\chi_{i_1,i_2,\cdots,i_N}=1, \qquad i_N=1,\cdots,n_N,\ \chi_{i_1,i_2,\cdots,i_N}\subset\{0,1\} \text{ for all } i_1,\cdots,i_N$$

where $\chi_{i_1,i_2,\cdots,i_N}$ is the associated binary variable, such that $\chi_{i_1,i_2,\cdots,i_N}=1$ if the measurement of N-tuple is associated with a candidate target. Otherwise, if the N-tuple is not a candidate target then $\chi_{i_1,i_2,\cdots,i_N}=0$.

3. ANT COLONY OPTIMIZATION FOR MULTI-SENSOR MULTI-TARGET DATA ASSOCIATION PROBLEM

In the data association problem, the goal is to find the most likelihood that each measurement is assigned to a target or declared as a false alarm. The most likelihood that the measurement is a target is stated by a small cost. Hence, in order to solve the data association problem using ant colony optimization, ants need to move toward the points which consent to achieve minimum total cost. The search process by artificial ants (Dorigo, 1992; Dorigo et al., 1996) is described as follows.

In the first cycle, the initial step is by placing the artificial ants (number of ants can be arbitraly chosen) randomly on different points. Next, until the end of the first cycle, the ant will move from point i to point j (which is allowed) and a visibility function given by the following equation is considered

$$\eta_j=\frac{1}{C_j} \tag{7}$$

where C_j is the cost of point *j*. Ants will eventually reach the points that have largest visibility value. The points are not allowed to be stored in the tabu list. After completing one cycle, the contents of the tabu list is reset. Each completed cycle, the ants will leave a pheromone trail on every visited point. After completing one cycle, the ants will die and be replaced with new ants with the same number.

In the second cycle and afterward, the new ants placed on the points that has been visited by ants in the previous cycle, and will move from point i to point j (which allowed) based on a probability function, named as status transition rule

$$p_j(t) = \begin{cases} \dfrac{[\tau_j(t)]^{\alpha}[\eta_j]^{\beta}}{\sum_{j \in allowed}[\tau_j(t)]^{\alpha}[\eta_j]^{\beta}}; & if \quad j \in allowed \\ 0; & otherwise \end{cases} \tag{8}$$

where $\tau_j(t)$ is amount of pheromone of the ant at point j at time t. The parameters α and β are used to control the relative importance of pheromone and visibility. Thus, after an ant finishes its tour in one cycle, the amount of pheromone will be updated as follows:

$$\tau_j(t+N) = \rho\tau_j(t) + \Delta\tau_j(t, t+N) \tag{9}$$

where ρ is a coefficient with a value between 0 to 1, and $(1-\rho)$ indicates the evaporation of pheromone, and

$$\Delta\tau_j(t, t+N) = \sum_{k=1}^{m} \Delta\tau_j^k(t, t+N) \tag{10}$$

where $\Delta\tau_j^k(t, t+N)$ is the pheromone left by ant k at point j, by the time between t to $(t+N)$, which is defined as follows:

$$\Delta\tau_j^k(t+t+N) = \begin{cases} \dfrac{1}{L^k}; & if \quad j \in tour \quad k \\ 0; & otherwise \end{cases} \tag{11}$$

where L^k is the total cost of the tour by the k-th ant and N is the number of points visited by each ant in a cycle. Total points visited by each ant in one cycle (N) are given by $N = 1 + \sum_{i=1}^{s} n_i$. Where n_i denotes number of measurements on the i-th sensor and s is the number of sensors. The amount of pheromone at $t = 0$ for every point is represented by $\tau_j(0)$. One possibility of the selection is to set the value equals to the value of its visibility. The algorithm stops when all ants have the same route.

4. ARTIFICIAL NEURAL NETWORK

The artificial neural network (ANN) architecture which is used to improve the estimation results from the Kalman filter is a feed forward neural network with supervised learning rule. The learning algorithm uses a back-propagation method. The artificial neural network is used to improve the estimation quality of the states. In other words, it is employed to gain further reduction of the tracking error. Consequently, the parameters that have direct influence to the error reduction are tuned as ANN inputs. Figure 2 shows diagram block of the learning process of the ANN. For each target, it is utilized nine input signals (Turkmen and Guney, 2006). The selected signals consist of:

(1) The position difference (δ_1) between the measurement vector and the estimation vectors in Cartesian coordinate system, $(z_k - [\hat{x}_{k|k}]_p)$.

(2) The position difference (δ_2) between the estimation vectors and the prediction vectors in Cartesian coordinate system, $([\hat{x}_{k|k}]_p - [\hat{x}_{k|k-1}]_p)$.

(3) The velocity difference (δ_3) between the estimation and the prediction vectors in Cartesian coordinate system, $([\hat{x}_{k|k}]_v - [\hat{x}_{k|k-1}]_v)$.

During the learning phase, all the array of input parameters becomes input to the ANN.

If x_k is the true state vector, and let this vector provides a correct measurement in the Cartesian coordinate system (x, y and z), then the error from a track in the coordinates x, y and z after the process in the Kalman filter is given by

$$(E_r)_x = (x_k - \hat{x}_{k|k})_x ; \quad (E_r)_y = (x_k - \hat{x}_{k|k})_y ; \quad (E_r)_z = (x_k - \hat{x}_{k|k})_z$$

Measurement inaccuracy in tracking E_r is expressed by the above equations will be corrected by using the neural network (Vaidehi et al., 2001). After the learning process, the output of ANN is ideally describing the exact difference between the estimation position by the Kalman filter and the actual position of the target. Therefore, in performing supervised learning, the output of the artificial neural network will be compared with the value of E_r. In the simulation, the value of states x_k is assumed to be known.

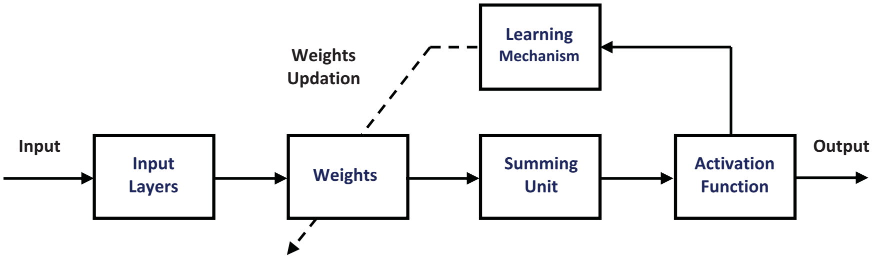

Figure 2 Learning in Artificial Neural Network

5. SIMULATION

Simulation is carried out to test the performance of the proposed target tracking system. There are two difference cases in the simulation, i.e. two targets in case-I and three targets in case-II. To each target, it is given additional zero mean Gaussian noise. Measurement is taken for 100 scans, where each scan has 4 seconds time interval. The motion of targets being tracked is modeled by the following discrete-time state space equation

$$x_{k+1} = F_k x_k + G_k w_k \tag{12}$$

where the state vector x_k represents the position and velocity of each target in the Cartesian coordinates (x , y , z), F_k denotes the state transition matrix, given by the following matrix

$$F_k = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

and G_k is the process noise matrix, given by

$$G_k = \begin{bmatrix} \Delta t^2/2 & 0 & 0 \\ \Delta t & 0 & 0 \\ 0 & \Delta t^2/2 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t^2/2 \\ 0 & 0 & \Delta t \end{bmatrix} \tag{14}$$

In equation (14), Δt is the sampling time interval at which the measurement data are received. The time interval is assumed to be uniform. w_k denotes the process noise which is assumed Gaussian, zero mean and covariance Q_k . Process noise covariance matrix, assuming that all noise processes are not correlated, is given by the following matrix

$$Q_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \tag{15}$$

The measurement vector z_k is modeled as

$$z_k = H_k x_k + v_k \tag{16}$$

where H_k is the measurement matrix, given by

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{17}$$

v_k is the measurement noise is assumed Gaussian, zero mean with known covariance R_k. Covariance matrix of measurement noise, assuming that all noise measurements are not correlated, is given by

$$R_k = \begin{bmatrix} \sigma_{xm}^2 & 0 & 0 \\ 0 & \sigma_{ym}^2 & 0 \\ 0 & 0 & \sigma_{zm}^2 \end{bmatrix} \tag{18}$$

The parameters of the ant colony optimization were selected as $\alpha = 1$, $\beta = 1$ and $\rho = 0.9$ for both cases. However, the number of ants was selected differently (6 ants in case-I and 10 ants in case-II). The process noise covariance matrix and the measurement noise covariance matrix were selected as Q_k = diag(0.00001, 0.00001, 0.00001)km^{2}s^{-4} and R_k = diag(0.1, 0.1, 0.1)km^2 respectively. The initial covariance matrix of the Kalman filter was chosen $P_{0|0} = I_{6\times6}$. The sensors were assumed to have detection probability of $P_{ds} = 0.98$. The false alarm rate of the sensors was assumed to be 1.

During the artificial neural network learning phase, the outputs of the the Kalman filter for two targets in case-I and three targets in case-II were given as the input patterns of ANN after normalization. There were 80 input patterns of the targets without maneuver and 100 input patterns of the target with maneuver. These became input patterns in the learning process of the ANN. Therefore, the total input patterns were 420 inputs. Experimentally, it was found that the architecture of the artificial neural network that provides the best convergence results in order to achieve the specified error limit is given by the network that has 9 inputs and 3 outputs with 4 hidden layers. Each hidden layer had 45 neurons.

Tangent-sigmoid activation function was used for as transfer function from inputs until the fourth hidden layer. A linear activation function was used as transfer function from the fourth hidden layer to the outputs. Error of the learning process was specified to achieve the mean square error of 0.002. Learning speed for each layer was selected $\eta = 0.01$. After the learning process was completed, the outputs of the neural network were de-normalized and summed at the position components of the target state vector of the Kalman filter in order to improve estimation results. The process of targets tracking for case-I is shown in Figure 3 to 6 and for case-II is shown in Figure 7 to 10 respectively.

Case-I:

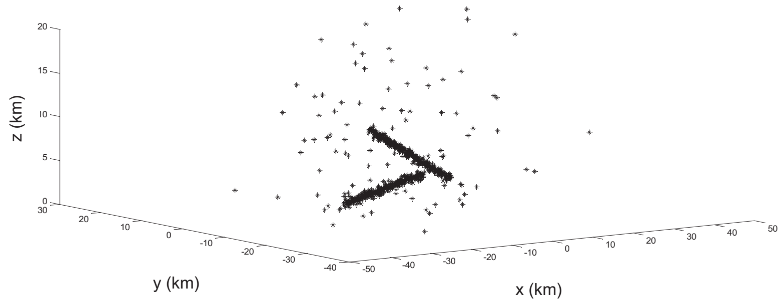

Figure 3 Noisy measurements with false alarm

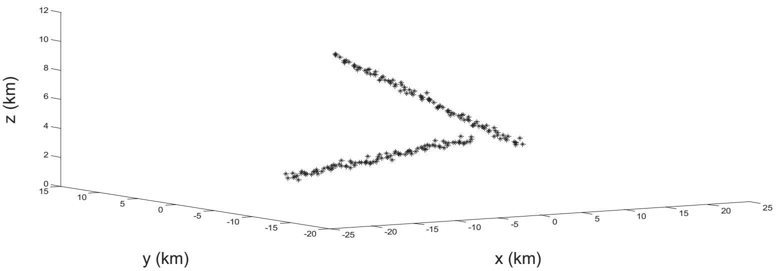

Figure 4 Selected measurements by ACO

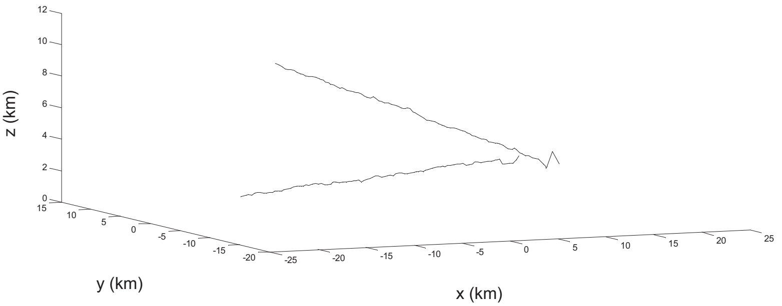

Figure 5 Tracking using Kalman filter

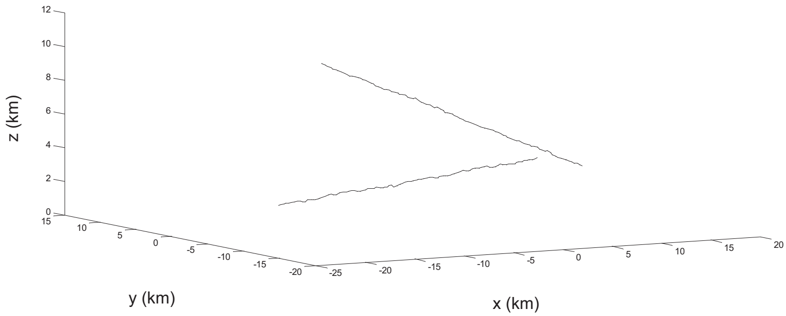

Figure 6 Tracking using Kalman filter and ANN

Case-II:

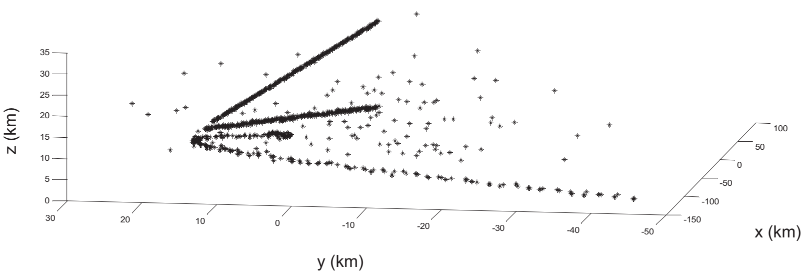

Figure 7 Noisy measurements with false alarm

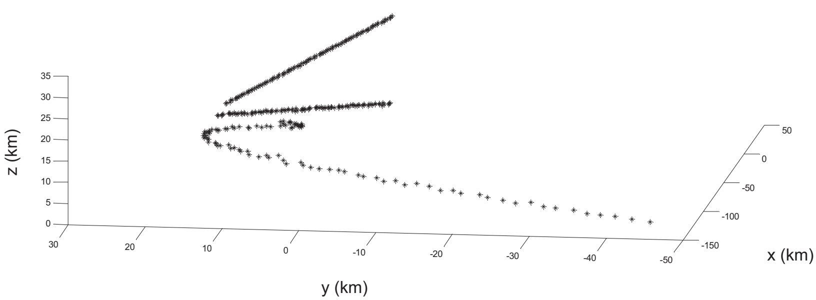

Figure 8 Selected measurements by ACO

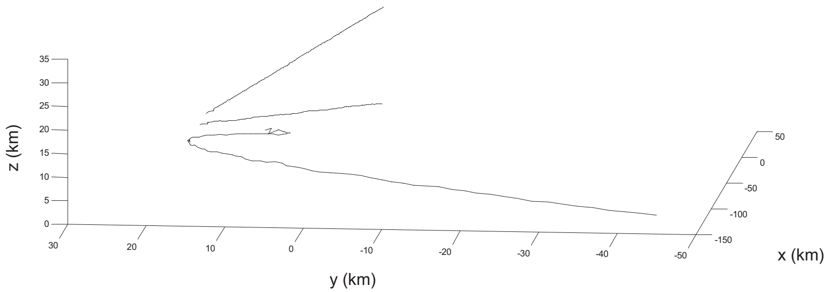

Figure 9 Tracking using Kalman filter

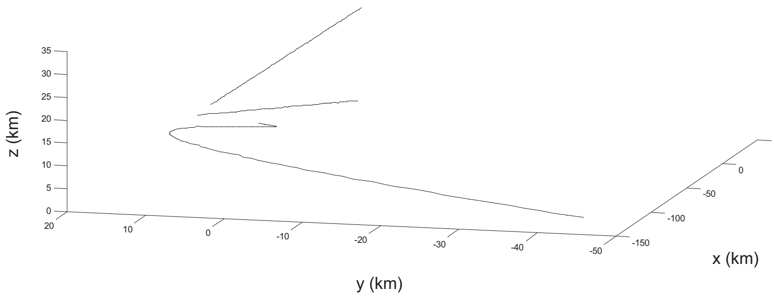

Figure 10 Tracking using Kalman filter and ANN

Table 1 Performance comparison between KF and KF-ANN

Case	Target	RMSE (km)		% improvement with ANN
		KF	KF-ANN	
1	1	0.155	0.090	42
	2	0.190	0.079	58
2	1	0.148	0.088	41
	2	0.530	0.079	85
	3	0.101	0.050	51

Table 1 shows comparison of RMSE value of performance between the results of estimation by using Kalman filter and the Kalman filter-ANN. Results show that estimation improvement using the artificial neural network yields 55% average of the estimation by only using the Kalman filter. The artificial neural network leads to better responses if the input patterns are selected similar to the patterns of the inputs given to the learning of neural network.

6. EFFECTS OF PARAMETERS AND NUMBER OF ANTS

In the ant colony optimization (ACO), ρ and α are parameters which influence the pheromone. Parameter ρ affects the evaporation of pheromone. Increasing the value of ρ would make the evaporation of pheromone will be slower. Conversely, a small ρ value will make the evaporation of pheromone will be faster. The parameter α determines the sensitivity of ants to pheromone. Small value of α makes the ants are less sensitive to pheromone, whereas a large value of α that makes the ants are more sensitive to pheromone.

Table 2 Test results for different values of α , β and ρ

α	β	ρ	Average of iteration	Tracking results
1	1	0.3	4.05	Match
1	1	0.5	4.06	Match
1	1	0.7	4.07	Match
1	1	0.9	4.08	Match
1	2	0.3	4.09	Match
1	2	0.5	4.10	Match
1	2	0.7	4.08	Match
1	2	0.9	4.08	Not match
2	1	0.3	4.03	Match
2	1	0.5	4.06	Match
2	1	0.7	4.06	Match
2	1	0.9	4.06	Match
2	2	0.3	4.05	Match
2	2	0.5	4.06	Match
2	2	0.7	4.07	Match
2	2	0.9	4.08	Match
2	3	0.3	4.06	Match
2	3	0.5	4.08	Match
2	3	0.7	4.11	Match
2	3	0.9	4.07	Not match
3	2	0.3	4.05	Match
3	2	0.5	4.06	Match
3	2	0.7	4.06	Match
3	2	0.9	4.06	Match

The parameter β determines the sensitivity of ants to the heuristic factor in state transition rule. Large value of β will increase the sensitivity of ants to the visibility factor, otherwise the small value of β which will reduce the sensitivity of ants to the visibility factor. Increasing the number of ants caused more points explored by the ants in a cycle. Thus it will improve the quality of solutions. Conversely, a large number of ants increased the time required to compute one cycle.

Table 2 shows the test results for different values of α, β and ρ to the number of iterations with 15 ants. Testing is performed in the case-II. In this algorithm, the second cycle and afterward, the position of the end visited of ants in the previous cycle becomes the initial position of the new ants. Based on Table 2, it is found that better results are achieved by taking $\alpha > \beta$. Figure 11 illustrates one case where the tracking result does not match when $\alpha > \beta$. For faster computation time, the optimum value is achieved by using $\alpha = 2$, $\beta = 1$ and $\rho = 0.3$. A three dimensional plot of the results in Table 2 is shown in Figure 12. It can be seen that the faster iteration is indicated by the dark blue color. The figure shows a pattern of colours as a function of varying values of α, β and ρ.

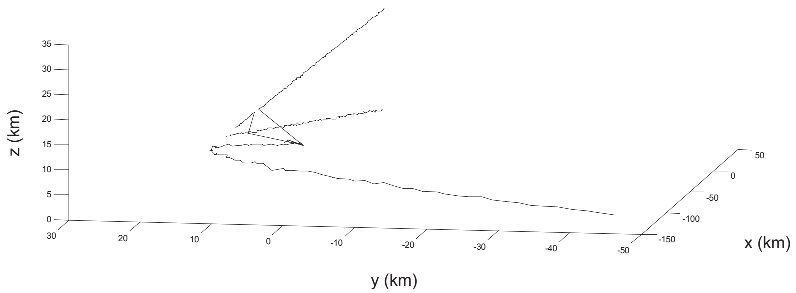

Figure 11 An example of not match tracking

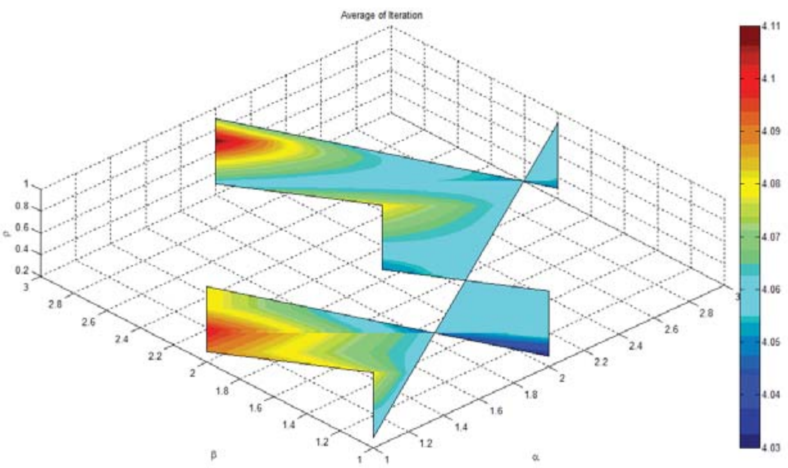

Figure 12 3-D Plot of Average of Iteration

In the next test, the placement of new ant initial points for the second cycle and afterward will be compared between the initial points taken from final visit of the previous cycle (Method-I) and the initial points are taken from the second visit of the previous cycle (Method-II). In the testing, the algorithm is run by using MATLAB program in a computer with the Intel Core2Duo processor (2.33 GHz). Table 3 shows the comparison of search results by the method-I and the method-II which are shown as a bar graph in Figure 13, by using parameter values $\alpha = 2$, $\beta = 1$ and $\rho = 0.3$.

Table 3 Comparison of results between Method-I and Method-II

	Method-I		Method-II	
Number of ants	Average of iteration	Computation time (seconds)	Average of iteration	Computation time (seconds)
10	4.06	0.249	3	0.137
9	4.11	0.232	3	0.132
8	4.05	0.207	3	0.114
7	4.03	0.149	3	0.091
6	4.01	0.137	3	0.089
5	3.91	0.131	3	0.080

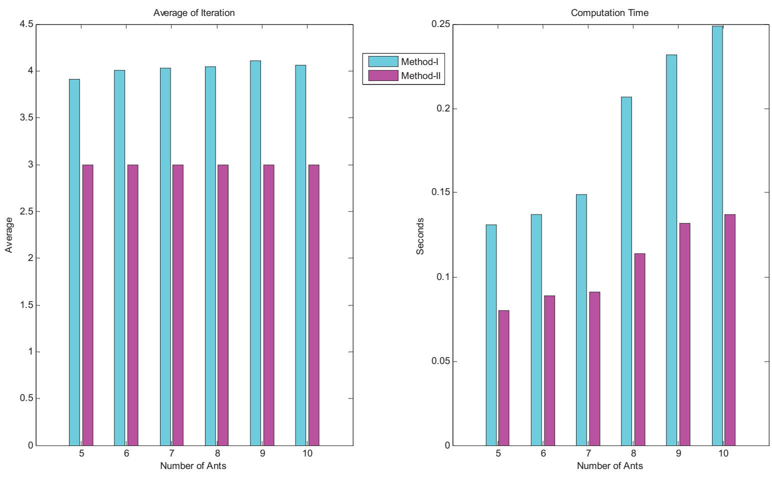

Figure 13 Comparison of Search Results

Based on the comparison as shown in Figure 13, it can be observed that placing the initial points of new ants at the point of the second visit of ants from the previous cycle will reduce the number of iterations, which purported to shorten the computation time. Search results are faster with the Method-II. This can be explained as follows. The initial pheromone value equals to the value of visibility, then for the first cycle, the ants are placed at arbitrary points. Next, the ants will travel to the second point. This is the first chance for the ants to determine the point of the visit and to have a great opportunity that the second visited point is the correct point. For the second cycle and subsequently, the initial points of new ants are placed at the point of the second visit of the previous cycle. Thus, in the next step of the visit, the ants will move at the right points if the initial points are the correct points.

7. CONCLUSIONS

In this paper, it has been developed a system for multiple target tracking by combining the ant colony optimization, the Kalman filter and the artificial neural network. Simulation showed that the proposed system can identify the number of true targets and can improve the states estimation. Processing stages of this target tracking can be explained as follows: measurement data from sensors are processed by the observation of the ant colony optimization algorithm to obtain the right target number. Next, the states of the targets are estimated by using the Kalman filter. After that, the estimation results are improved by applying the artificial neural network. The developed algorithm was tested with a system model for multi sensors multiple target tracking. Simulation showed that the ant colony optimization algorithm can effectively separate the measurements originated from the true targets and the false alarms.

In the ant colony optimization algorithm, it has been investigated effects on selecting initial pheromone value of ants, and ant assignment in the second cycle and afterward. Based on simulation results, the ant colony optimization algorithm produced better and faster search results if the initial pheromone value is taken to be equal to the visibility, the initial position of new ants is selected at the second cycle and afterward on the point of the second visit of the previous cycle of ants. The simulation results also showed that the Kalman filter gave fairly good estimation. The application of the artificial neural network improved significantly the results obtained by the Kalman filter.

REFERENCES

Badamchizadeh M.A., Nikdel, A., and Kouzehgar, M., 2010, Comparison of Genetic Algorithm and Particle Swarm Optimization for Data Fusion Method Based on Kalman Filter, *International Journal of Artificial Intelligence (IJAI)*, 5 (A10), 67-78.

Bar-Shalom, Y., Li., X.R., Kirubarajan, T., 2001, *Estimation with Applications to Tracking dan Navigation, Theory, Algorithms and Software*, John Wiley & Sons Inc., New York.

Bonabeau, E., Dorigo, M., and Theraulaz, G., 1999, *Swarm Intelligence: From Natural to Artificial Systems*, New York, Oxford University Press.

Chen G., and Hong, L., 1997, A Genetic Algorithm Based Multi-dimensional Data Association Algorithm for Multi-sensor Multi-target Tracking, *Mathl. Comput. Modelling,* 26 (4), 57-69.

Di Caro, G., and Dorigo, M., 1998, AntNet: Distributed Stigmergetic Control for Communications Networks, *Journal of Artificial Intelligence Research,* 9, 317-365.

Dorigo, M., 1992, *Optimization, Learning and Natural Algorithms (in Italian)*, PhD Thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy.

Dorigo, M., Maniezzo, V., and Colorni, A., 1996, Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26 (1), 29-41.

Dorigo, M., and Stützle, T., 2004, *Ant Colony Optimization*, MIT Press, Cambridge, MA.

Duan, H.B., Wang, D.B., and Yu, X.F., 2006, Novel Approach to Nonlinear PID Parameter Optimization Using Ant Colony Optimization Algorithm*, Journal of Bionic Engineering*, 3, 73-78.

Grewal, M.S., and Andrews, A.P., 2008, *Kalman Filtering, Theory and Practice Using MATLAB,* New Jersey, John Wiley & Sons Inc.

Kalman, R.E., 1960*,* Contribution to The Theory of Optimal Control, *Bol. Soc. Matem. Mex.,* 5, 102-119.

Kalman, R.E., and Bucy, R.S, 1961, New Results in Linear Filtering and Prediction Theory, *Transactions ASME J. Basic Eng.*, 8, 95-107.

Kammerdiner A.R., Krokhmal, P.A., and Pardalos, P.M., 2007, *Characteristics of the Distribution of Hamming Distance Values Between Multidimensional Assignment Problem Solutions,* LNCIS 369, Spinger-Verlag, Berlin Heidelberg, 339-352.

Kosko, B., 1992, *Neural Networks and Fuzzy Systems A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall International Inc., USA.

Krose B., and van der Smagt, P., 1996, *An Introduction to Neural Networks,* The University of Amsterdam.

Naidu, D.S., 2003, *Optimal Control Systems*, CRC Press LLC, USA.

Popp R.L., Pattipati, K.R., and Bar-Shalom, Y., 2001, m-Best S-D Assignment Algorithm with Application to Multitarget Tracking, *IEEE Transactions on Aerospace and Electronic Systems,* 37 (1), 22-39.

Stimson, G.W., 1998, *Introduction to Airborne Radar, 2nd Ed.,* New York, Scitech Publishing Inc.

Sumathi, S., and Paneerselvam, S., 2010, *Computational Intelligence Paradigms: Theory and Applications using MATLAB*, CRC Press, Boca Raton – USA.

Turkmen, I., and Guney, K., 2008, *Genetic Tracker with Adaptive Neuro-Fuzzy Inference System for Multiple Target Tracking,* Expert System with Applications, 35, 1657-1667.

Turkmen, I., and Guney, K., 2006, *Tabu Search Tracker with Adaptive Neuro-Fuzzy Inference System for Multiple Target Tracking*, Progress In Electromagnetics Research, PIER, 65, 169-185.

Vaidehi, V., Chitra, N., and Chokkalingam, M., Krishnan, C.N., 2001, Neural Network Aided Kalman Filtering for Multitarget Tracking Applications, *Computer and Electrical Engineerin*g, 27, 217-228.

Wiranto, I., and Joelianto, E., 2010, Multiple Target Tracking Using Ant Colony Optimization and Artificial Neural Network, *Proc. 6th International Conference on Intelligent Unmanned Systems (ICIUS*), Nov. 3-5, 2010, Bali, Indonesia, 548-554.