An Intelligent Tracking Method for a Maneuvering Target

Bum-Jik Lee, Young-Hoon Joo, and Jin Bae Park

Abstract: Accuracy in maneuvering target tracking using multiple models relies upon the suitability of each target motion model to be used. To construct multiple models, the interacting multiple model (IMM) algorithm and the adaptive IMM (AIMM) algorithm require predefined sub-models and predetermined acceleration intervals, respectively, in consideration of the properties of maneuvers. To solve these problems, this paper proposes the GA-based IMM method as an intelligent tracking method for a maneuvering target. In the proposed method, the acceleration input is regarded as an additive process noise, a sub-model is represented as a fuzzy system to compute the time-varying variance of the overall process noise, and, to optimize the employed fuzzy system, the genetic algorithm (GA) is utilized. The simulation results show that the proposed method has a better tracking performance than the AIMM algorithm.

Keywords: Maneuvering target tracking, IMM algorithm, AIMM algorithm, GA-based IMM method, fuzzy system.

1. INTRODUCTION

The problem of tracking a maneuvering target has been studied in state estimation over decades. The Kalman filter has been widely used to estimate the state of the target, but in the presence of a maneuver, its performance may be seriously degraded. To solve this problem, various techniques have been investigated and applied. First, in 1970, Singer proposed a target tracking model whose maneuver was assumed as a first order Markov process with time correlation [1]. Since the Singer's method, recent research has been roughly divided into two main approaches. One approach is to detect a maneuver and then to cope with it effectively. Examples of this approach include the input estimation (IE) technique [2], the variable state dimension (VSD) approach [3], and so on. The other approach is to describe the motion of the target with multiple models. The generalized pseudo-Bayesian (GPB) approach [4], the interacting multiple model (IMM) algorithm [5], and the adaptive IMM (AIMM) algorithm [6] are included in this approach. In this paper, the second approach is mainly discussed.

Accuracy in maneuvering target tracking using multiple models relies upon the suitability of each target motion model to be used for a maneuver. In the IMM algorithm, the estimate is obtained by a weighted sum of the estimates from sub-models in accordance with the probability of each model being effective. But, to construct multiple models, the algorithm requires predefined sub-models with the different dimensions or process noise levels in consideration of the properties of the maneuvers. On the other hand, the AIMM algorithm needs no predefined sub-models because it estimates the acceleration of the target adaptively and constructs multiple models using this estimated acceleration. In this algorithm, a two-stage Kalman estimator [7], which has a bias-free filter and a bias filter, is used only in estimating the acceleration. However, the acceleration intervals, which are symmetrically added to or subtracted from the estimated acceleration value to construct multiple models, should also be determined by the properties of the maneuvers.

To solve these problems and track a maneuvering target effectively, the GA-based IMM method is proposed in this paper as an intelligent tracking method for a maneuvering target. In the maneuvering target model, the acceleration input is regarded as an additive process noise and the time-varying variance of this overall process noise is calculated using the relations between the filter residual and the process noise variance. Because the filter residual increases in the presence of a maneuver, we can treat a target maneuver by adjusting the process noise variance. In the proposed method, a sub-model is represented as a fuzzy system to compute this time-varying variance. The GA is applied to identify the parameters and the structure of this fuzzy system for a certain maneuver

Manuscript received October 14, 2002; accepted February 19, 2003

Bum-Jik Lee is with the Department of Electrical and Electronic Engineering, Yonsei University, Korea. (e-mail: bumjik@control.yonsei.ac.kr).

Young-Hoon Joo is with the School of Electronic and Information Engineering, Kunsan University, Korea. (e-mail: yhjoo@kunsan.ac.kr).

Jin Bae Park is with the Department of Electrical and Electronic Engineering, Yonsei University, Korea. (e-mail: jbpark@yonsei.ac.kr).

input within the assumed maximum acceleration input. The GA is the method used to obtain an optimal solution based on the principles of natural population genetics and natural selection although the mathematical relationship between the parameter to be identified and the nonlinear cost function to be optimized is not known exactly. Then, multiple models are composed of these fuzzy systems, which are optimized for different acceleration inputs.

Section 2 shows the maneuvering target model and summarizes the AIMM algorithm as previous works, and the details of the proposed method are described in Section 3. In Section 4, the computer simulation results prove the better tracking performance of the proposed method than that of the AIMM algorithm. Conclusions are finally drawn in Section 5.

2. PREVIOUS WORKS

2.1. Maneuvering target model

The linear discrete time model for a maneuvering target is described for each axis by

$$X(k+1) = FX(k) + G[u(k) + w(k)], \quad (1)$$

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} T^2/2 \\ T \end{bmatrix},$$

where $X(k) = [p \ \dot{p}]^T = [p \ v]^T$ is the state vector, F and G are the transition matrix and the excitation matrix, respectively, $w(k)$ is the process noise, and $u(k)$ is the unknown acceleration input. The measurement equation is

$$Z(k) = HX(k) + v(k), \quad (2)$$

where $H = [1 \ 0]$ is the measurement matrix and $v(k)$ is the measurement noise. $w(k)$ and $v(k)$ are considered as white Gaussian noise sequences with zero-mean and variances q and r, respectively, and their correlation is assumed to be zero.

2.2. AIMM algorithm

The AIMM algorithm has a limited number of sub-models for each axis, and each sub-model is represented as the estimated acceleration or the acceleration levels distributed symmetrically about the estimated one [6]. In the case of N sub-models for each axis, the set of multiple models is represented as

$$\begin{aligned} M_A &= \{\hat{a}_1(k),\ \hat{a}_2(k),\ \cdots,\hat{a}_N(k)\} \\ &= \{\hat{a}(k), \hat{a}(k) \pm \varepsilon_1, \cdots, \hat{a}(k) \pm \varepsilon_{(N-1)/2}\} \end{aligned}$$

where $\hat{a}(k)$ is the estimated acceleration and $\varepsilon_{(N-1)/2}$ is the predetermined acceleration interval. The actual number of sub-models is determined according to the computational power.

In the AIMM algorithm, the acceleration of the target is estimated in parallel for each axis by a two-stage Kalman estimator, which consists of a bias-free filter and a bias filter [6, 7]. The bias filter equations to estimate the acceleration of the target are

$$\hat{a}(k \mid k-1) = \hat{a}(k-1 \mid k-1) \quad (3a)$$

$$P^a(k \mid k-1) = P^a(k-1 \mid k-1) + q^a, \quad (3b)$$

$$S(k) = HU(k-1), \quad (3c)$$

$$V(k) = (I - K(k)H)U(k), \quad (3d)$$

$$U(k) = FV(k-1) + G, \quad (3e)$$

$$\begin{aligned} K^a(k) = {} & P^a(k \mid k-1)S^T(k)[S(k)P^a(k \mid k-1)S^T(k) \\ & + HP(k \mid k-1)H^T + r]^{-1}, \end{aligned} \quad (3f)$$

$$\hat{a}(k \mid k) = \hat{a}(k \mid k-1) + K^a(k)[\upsilon(k) - S(k)\hat{a}(k \mid k-1)] \quad (3g)$$

$$P^a(k \mid k) = [I - K^a(k)S(k)]P^a(k \mid k-1), \quad (3h)$$

where $\hat{a}(k \mid k)$ is the bias vector, $P^a(k)$ is the covariance of the bias, q^a is the process noise for the bias vector, $U(k)$ and $V(k)$ are the sensitivity matrices, $K^a(k)$ is the Kalman gain of the bias filter, and $S(k)$, $K(k)$, and $\upsilon(k)$ are obtained from the bias-free filter.

The AIMM algorithm to be represented by the estimated acceleration and the acceleration intervals follows.

Interaction of the estimates (mixing)

$$\hat{X}_{0m}(k-1 \mid k-1) = \sum_{n=1}^{N} \mu_{n|m}(k-1 \mid k-1)\hat{X}_n(k-1 \mid k-1), \quad (4a)$$

$$\begin{aligned} P_{0m}(k-1 \mid k-1) = \sum_{n=1}^{N} \mu_{n|m}(k-1 \mid k-1)\{ & P_n(k-1 \mid k-1) \\ & + [\hat{X}_n(k-1 \mid k-1) - \hat{X}_{0m}(k-1 \mid k-1)] \bullet \\ & [\hat{X}_n(k-1 \mid k-1) - \hat{X}_{0m}(k-1 \mid k-1)]^T \}, \end{aligned} \quad (4b)$$

where the mixing probability $\mu_{n|m}$ and the normalization constant α_m are

$$\mu_{n|m}(k-1 \mid k-1) = \frac{1}{\alpha_m}\phi_{nm}\mu_n(k-1), \quad (5a)$$

$$\alpha_m = \sum_{n=1}^{N} \phi_{nm}\mu_n(k-1), \quad (5b)$$

where ϕ_{nm} is the known model transition probability from the nth sub-model to the mth sub-model and $\mu_n(k-1)$ is the model probability of the nth sub-model at the scan $k-1$.

Filtering algorithm

$$\hat{X}_m(k \mid k-1) = F\hat{X}_{0m}(k-1 \mid k-1) + G\hat{a}_m(k-1), \quad (6a)$$

$$\upsilon_m(k) = Z(k) - H\hat{X}_m(k \mid k-1), \quad (6b)$$

$$P_m(k \mid k-1) = FP_{0m}(k-1 \mid k-1)F^T + GqG^T; \quad (6c)$$

$$S_m(k) = HP_m(k \mid k-1)H^T + r, \quad (6d)$$

$$K_m(k) = P_m(k \mid k-1)H^T S_m^{-1}(k), \quad (6e)$$

$$\hat{X}_m(k \mid k) = \hat{X}_m(k \mid k-1) + K_m(k)\upsilon_m(k), \quad (6f)$$

$$P_m(k \mid k) = P_m(k \mid k-1) - K_m(k)S_m(k)K_m^T(k). \quad (6g)$$

Update of model probability
likelihood function:

$$\Lambda_m(k) = \mathcal{N}[\upsilon_m(k); 0, S_m(k)] = \frac{1}{\sqrt{2\pi \mid S_m(k) \mid}} \exp\left(-\frac{1}{2}\upsilon_m^T(k)S_m^{-1}(k)\upsilon_m(k)\right). \quad (7a)$$

model probability update:

$$\mu_m(k) = \frac{\Lambda_m(k)\alpha_m}{\sum_{n=1}^{N}\Lambda_n(k)\alpha_n}. \quad (7b)$$

Estimate combination
state estimate:

$$\hat{X}(k \mid k) = \sum_{m=1}^{N}\mu_m(k)\hat{X}_m(k \mid k). \quad (8a)$$

estimate covariance matrix:

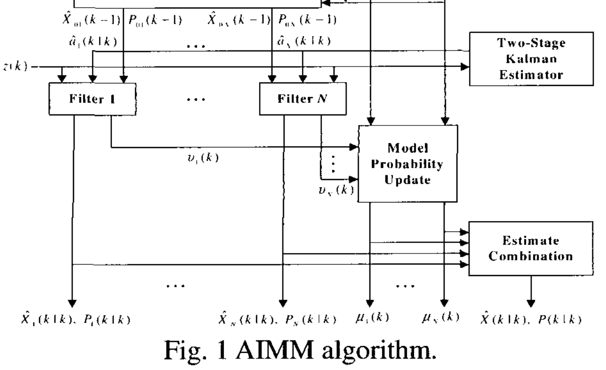


Fig. 1 AIMM algorithm.

$$P(k \mid k) = \sum_{m=1}^{N}\mu_m(k)\{P_m(k \mid k) + [\hat{X}_m(k \mid k) - \hat{X}(k \mid k)][\hat{X}_m(k \mid k) - \hat{X}(k \mid k)]^T\} \quad (8b)$$

Fig. 1 describes the AIMM algorithm with N sub-models.

3. GA-BASED IMM METHOD USING FUZZY LOGIC

This paper proposes the GA-based IMM method using fuzzy logic for a maneuvering target. In the maneuvering target model (1), the acceleration input $u(k)$ is regarded as an additive process noise. A new piecewise constant white acceleration model is

$$X(k+1) = FX(k) + G\overline{w}(k), \quad (9)$$

where $\overline{w}(k)$ is the overall process noise with the time-varying variance $\overline{q}(k)$, which can be determined from the filter residual and its variation. Because the filter residual increases in the presence of a maneuver, we can treat a target maneuver by adjusting the process noise variance. However, analyzing the mathematical relations between the residual and the process noise variance is very difficult. To alleviate such a difficulty, we use the fuzzy system, which can approximate an unknown or highly nonlinear system well. Therefore, in the proposed method, a sub-model is represented as a fuzzy system to calculate this time-varying process noise variance and the GA is applied to identify the parameters and the structure of the fuzzy system. Multiple models for tracking a maneuvering target are finally composed of these fuzzy models, which are identified for various maneuver inputs. The proposed method is illustrated in Fig. 2. In the case of N sub-models, the set of multiple models are represented as
$M_G = \{\text{fuzzy rules } 1, \text{ fuzzy rules } 2, \cdots, \text{ fuzzy rules } N\}$.

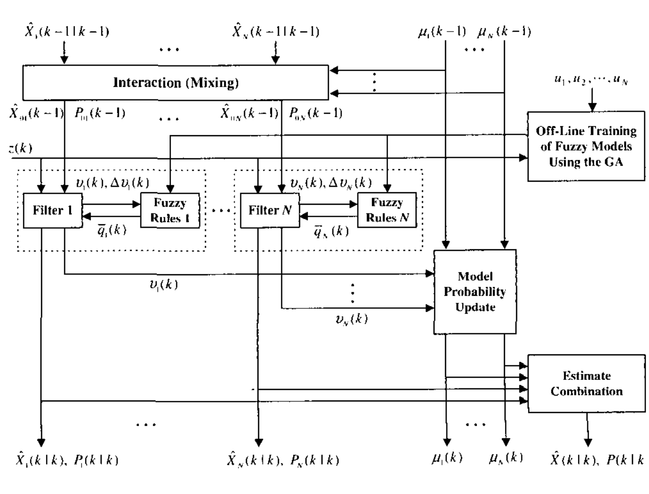


Fig. 2. GA-based IMM method using fuzzy logic.

3.1. GA-based IMM method

The fuzzy inference rules for calculating the time-varying variance have the scatter-partitioned structure within the range of input or output space. The j th fuzzy rule for the m th sub-model is represented by

$$R_j^m\ (j=1,\cdots,M): \text{If } \chi_1 \text{ is } A_{1j}^m \text{ and } \chi_2 \text{ is } A_{2j}^m, \text{ then } y^m \text{ is } q_j^m$$

where two input variables, χ_1 and χ_2 are the mth filter residual $\upsilon_m(k)$ and its variation $\Delta\upsilon_m(k)$, respectively. A consequent variable y^m is the process noise variance q_j^m for the j th fuzzy rule. The Gaussian membership function A_{ij}^m with the center c_{ij}^m and the standard deviation σ_{ij}^m has the following membership grade.

$$\theta_{A_{ij}^m}^m(\chi_i) = \exp\left[-\frac{1}{2}\left(\frac{\chi_i - c_{ij}^m}{\sigma_{ij}^m}\right)^2\right]. \quad (10)$$

The unknown time-varying variance $\bar{q}_m(k)$ for the m th sub-model can be approximated in the following form.

$$\bar{q}_m(k) = \frac{\sum_{j=1}^{M} q_j^m\left(\prod_{i=1}^{2}\phi_{A_{ij}^m}^m(\chi_i(k))\right)}{\sum_{j=1}^{M}\left(\prod_{i=1}^{2}\phi_{A_{ij}^m}^m(\chi_i(k))\right)}. \quad (11)$$

According to the universal approximation theorem [8], there exist optimal parameters c_{ij}^m, σ_{ij}^m, and q_j^m, which can approximate $\bar{q}_m(k)$ as closely as possible. In this paper, we use the GA to optimize these parameters, as will be presented in the next subsection. By using (11), the filtering algorithm (6.a)–(6.g) can be replaced for the proposed GA-based IMM method as follows.

$$\hat{X}_m(k\mid k-1) = F\hat{X}_{0m}(k-1\mid k-1), \quad (12a)$$

$$\upsilon_m(k) = Z(k) - H\hat{X}_m(k\mid k-1), \quad (12b)$$

$$P_m(k\mid k-1) = FP_{0m}(k-1\mid k-1)F^T + G\bar{q}_m(k)G^T, \quad (12c)$$

$$S_m(k) = HP_m(k\mid k-1)H^T + r, \quad (12d)$$

$$K_m(k) = P_m(k\mid k-1)H^T S_m^{-1}(k), \quad (12e)$$

$$\hat{X}_m(k\mid k) = \hat{X}_m(k\mid k-1) + K_m(k)\upsilon_m(k), \quad (12f)$$

$$P_m(k\mid k) = P_m(k\mid k-1) - K_m(k)S_m(k)K_m^T(k). \quad (12g)$$

The proposed method has the following advantages compared with the IMM algorithm or the AIMM algorithm.

1. Unlike an IMM algorithm, no sub-models predefined in consideration of the property of the maneuver are required.

2. Unlike an AIMM algorithm, no estimation of the target acceleration or adjustment for different acceleration levels in accordance with the property of maneuver is required to construct sub-models.

3. Although the property of the maneuver is unknown, the proposed method can be applied if the maneuver is within the assumed maximum acceleration input because the fuzzy systems can effectively infer the time-varying variances.

3.2. Optimization of fuzzy system using the GA

In this paper, the GA is applied to optimize the parameters in both the premise part and the consequence part of the fuzzy system simultaneously. Obviously the fuzzy system should be designed such that the following objective function is minimized.

$$J = \sqrt{(\text{sum of position error})^2 + (\text{sum of velocity error})^2}. \quad (13)$$

The GA represents the searching variables of the given optimization problem as a chromosome containing one or more substrings. In this case, the searching variables are the center c_{ij}^m and the standard deviation σ_{ij}^m for a Gaussian membership function of the fuzzy set A_{ij}^m and the singleton output q_j^m. A convenient way to convey the searching variables into a chromosome is to gather all searching variables associated with the j th fuzzy rule in the m th sub-model into a string and to concatenate the strings as

$$S_j^m = \{c_{1j}^m, \sigma_{1j}^m, c_{2j}^m, \sigma_{2j}^m, q_j^m\},$$
$$S^m = \{S_1^m, S_2^m, \cdots, S_M^m\},$$

where S_j^m is the real coded parameter substring of the j th fuzzy rule in an individual and S^m denotes an individual for the m th sub-model. At the same time and to identify the number of fuzzy rules, we utilize the binary coded rule number string, which assigns a 1 or 0 for a valid or invalid rule, respectively. Fig. 3. illustrates the structure of the chromosome.

The initial population is made up of initial individuals to the extent of the population size. The premise string of each initial individual is determined at random within the given search space, i.e., the range of residual $\upsilon_m(k)$ and the range of its variation $\Delta\upsilon_m(k)$. The corresponding consequent string is determined at random using the possible range of the standard deviation of the overall process noise [9]:

$$0.5\left(a_{\max} + \sqrt{q}\right) \le \bar{\sigma}_m(k) \le \left(a_{\max} + \sqrt{q}\right), \quad (14)$$

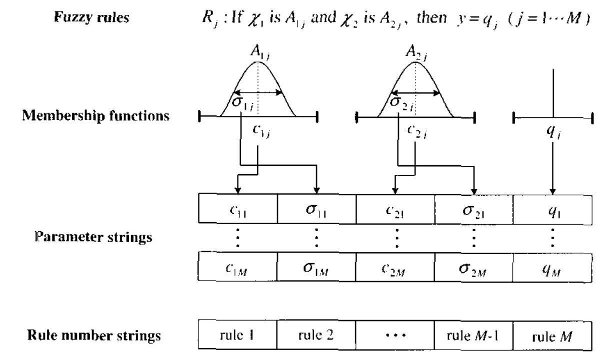


Fig. 3. Structure of chromosome.

where $\bar{\sigma}_m(k)$ is the standard deviation of the process noise variance and a_{max} is the maximum value of the acceleration input.

Each individual is evaluated by a fitness function. Since the GA originally searches the optimal solution so that the fitness function value is maximized, mapping the objective function (13) to the fitness function is necessary. Furthermore, since reducing the number of the fuzzy IF-THEN rules in a hardware implementation and a computation resource point of view is strongly desired, we use the fitness function of the form

$$f(J) = \frac{\lambda}{J+1} + \frac{1-\lambda}{M+1}, \qquad (15)$$

where λ is a positive scalar, to adjust the weight between the objective function and the rule number.

The GA that optimally identifies the time-varying variance of the proposed method is summarized as follows [10-13]:

Step 1: Set the parameters for the GA (maximum generation number, maximum rule number, population size, crossover rate, and mutation rate).

Step 2: Randomly generate the initial population such that all searching variables exist within the search space.

Step 3: Decode the chromosome of each individual in the population and determine the fuzzy systems for sub-models. Evaluate the determined fuzzy systems by (13) and give a fitness value to each individual in the population by (15).

Step 4: Evolve a new population by reproduction, crossover, and mutation.

Step 5: Increase the generation number by one, and replace the old generation with the new one. During the replacement, preserve an individual that has the maximum fitness value by the elitist reproduction.

Step 6: Repeat Steps 3 through 5 until one of the following is satisfied:
(1) the satisfactory population shows up,
(2) the generation number reaches the maximum generation number, or
(3) the fitness function value is not increased for the predetermined generations.

4. SIMULATION RESULTS

In this section, the simulations are divided in two parts: a simulation for searching the optimal fuzzy rules off-line and a simulation for tracking a maneuvering target. The tracking performance of the proposed method is compared with that of the AIMM algorithm.

The initial parameters of the GA are presented in Table 1. The maximum acceleration input for whole simulations is assumed to be $0.1 km/s^2$. The fuzzy rules identified off-line for the acceleration input $u_1 = 0.001 km/s^2$ are shown in Table 2, for $u_2 = 0.01 km/s^2$ in Table 3, and for $u_3 = 0.1 km/s^2$ in Table 4.

The target is assumed as an incoming anti-ship missile on the $x-y$ plane [14]. The initial position of the target is at [$72.9 km$ $21.5 km$], and it moves with a constant velocity of $0.3 km/s$ along a -150° line to the x-axis. The target has the lateral maneuvers as shown in Fig. 4, and the corresponding target motion is illustrated in Fig. 5. For both axes, the standard deviation of the zero mean white Gaussian measurement noise is $0.5 km$ and that of a random acceleration noise is $0.001 km/s^2$. The standard deviations of the bias filter and the bias-free filter for a two-stage Kalman estimator are 0.01 km/s^2 and $0.001 km/s^2$, respectively. The switching probability matrix of the sub-model, ϕ_{nm}, was taken by

Table 1. The initial parameters of the GA.

Parameters	Values
Maximum Generation	200
Maximum Rule Number	50
Population Size	500
Crossover Rate	0.9
Mutation Rate	0.01
λ	0.75

Table 2. The fuzzy rules identified for the acceleration input $u_1 = 0.001 km/s^2$.

no. of rule	parameters identified for $u_1 = 0.001 km/s^2$				
	c_1	σ_1	c_2	σ_2	w
1	0.9088	0.1426	0.2152	3.1791	0.3984×10^5
2	1.0168	0.0662	1.2639	3.2672	0.3477×10^5
3	1.3824	1.5158	-0.4991	0.5773	0.1040×10^5
4	-0.1311	0.5958	1.0934	0.5674	0.1057×10^5
5	1.4090	0.0295	0.2804	2.0269	0.3014×10^5
6	-1.2545	0.1582	1.1365	0.2148	0.2836×10^5
7	0.3876	0.0198	1.4585	0.0809	0.2605×10^5

Table 3. The fuzzy rules identified for the acceleration input $u_2 = 0.01 km/s^2$.

no. of rule	parameters identified for $u_2 = 0.01 km/s^2$				
	c_1	σ_1	c_2	σ_2	w
1	0.1197	0.0796	-1.5367	0.7641	0.1123×10^3
2	-0.0836	1.1625	-0.2359	0.5158	0.1207×10^3
3	0.8457	0.9523	-0.7970	2.5884	0.1202×10^3
4	1.4667	0.0370	-0.8562	1.6529	0.1086×10^3
5	0.7429	2.3806	-0.6072	2.5407	0.1205×10^3
6	-1.2103	0.3070	-1.0827	2.2768	0.1186×10^3
7	-0.2791	0.0920	-1.5338	1.9053	0.1185×10^3
8	-0.5224	2.7356	-0.8323	2.7159	0.1208×10^3
9	0.1966	1.9984	-1.1328	1.0014	0.1205×10^3

Table 4. The fuzzy rules identified for the acceleration input $u_3 = 0.1 km/s^2$.

no. of rule	parameters identified for $u_3 = 0.1 km/s^2$				
	c_1	σ_1	c_2	σ_2	w
1	1.5042	2.4868	-0.2898	0.5533	0.0102
2	3.5042	0.8063	0.6460	0.6212	0.0047
3	-1.4802	0.3118	0.2204	3.1265	0.0044
4	4.0089	0.4257	-1.6225	0.7013	0.0098
5	4.1174	0.8075	0.5442	2.7186	0.0094
6	4.3489	1.0112	-0.1821	3.0504	0.0101
7	3.7746	0.1013	-1.2030	0.7165	0.0053

$$\phi_{nm} = \begin{cases} 0.97 & \text{if} \quad n = m \\ \dfrac{1-0.97}{N-1} & \text{otherwise} \end{cases}, \tag{16}$$

where N means the number of sub-models. Assuming that the first sub-model is nearer the motion model of the target, the initial model probability for sub-models was selected by

$$\mu_m(0) = \begin{cases} 0.6 & \text{if} \quad m = 1 \\ \dfrac{0.4}{N-1} & \text{otherwise,} \end{cases} \tag{17}$$

The acceleration levels of the sub-models for the AIMM algorithm are shown in Table 5, where AIMM3 and AIMM5 mean the AIMM algorithm with 3 and 5 sub-models, respectively. These values are determined in consideration of the properties of the target maneuvers.

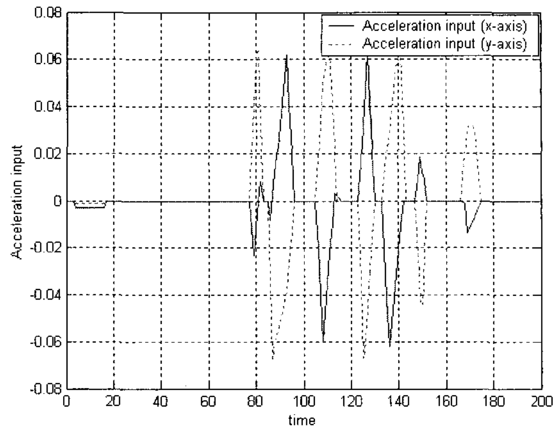


Fig. 4. The acceleration inputs (km/s^2).

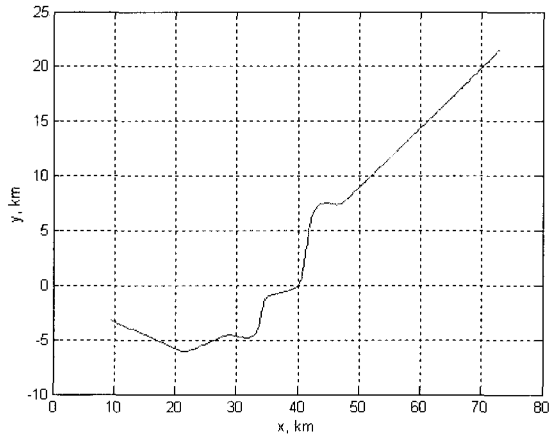


Fig. 5. The motion of incoming anti-ship missile.

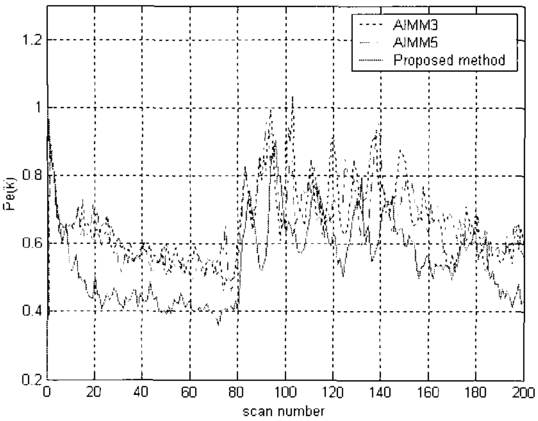
Table 5. The acceleration levels for the AIMM algorithm.

Configuration		The acceleration levels for sub-models (km/s^2)		
		m_1	m_2, m_3	m_4, m_5
1	AIMM3	$\hat{a}(k)$	$\hat{a}(k) \pm 0.04$	-
2	AIMM5	$\hat{a}(k)$	$\hat{a}(k) \pm 0.02$	$\hat{a}(k) \pm 0.04$

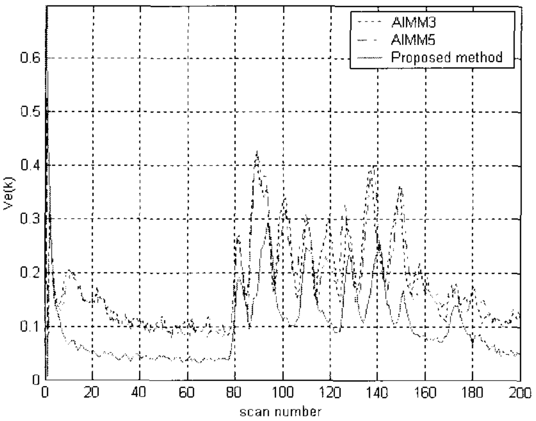
Comparisons between the performances of two algorithms are made on the basis of the normalized position and velocity errors, $P_e(k)$ and $V_e(k)$, as follows:

$$P_e(k) = \frac{\sqrt{\sum_{i=1}^{N} [(x^i(k) - \hat{x}^i(k))^2 + (y^i(k) - \hat{y}^i(k))^2]}}{\sqrt{\sum_{i=1}^{N} [(x^i(k) - z_x^i(k))^2 + (y^i(k) - z_y^i(k))^2]}}, \tag{18}$$

$$V_e(k) = \frac{\sqrt{\sum_{i=1}^{N} [(v_x^i(k) - \hat{v}_x^i(k))^2 + (v_y^i(k) - \hat{v}_y^i(k))^2]}}{\sqrt{\sum_{i=1}^{N} [(v_x^i(k) - v_{mx}^i(k))^2 + (v_y^i(k) - v_{my}^i(k))^2]}}, \tag{19}$$



(a) Normalized position error.



(b) Normalized velocity error.

Fig. 6. The simulation results.

Table 6. The numerical results.

Configurations		No. of sub-models	The results		CPU time (sec.)
			ζ_p	ζ_v	
1	AIMM3	3	0.6633	0.1818	65.91
2	AIMM5	5	0.6593	0.1784	122.05
3	Proposed method	3	0.5527	0.1063	146.86

where $x^i(k)$, $y^i(k)$ and $\hat{x}^i(k)$, $\hat{y}^i(k)$ are the true and estimated positions of the target, respectively, and z^i_x, z^i_y is the measured position of the target. $v^i_x(k)$, $v^i_y(k)$ and $\hat{v}^i_x(k)$, $\hat{v}^i_y(k)$ are the true and estimated velocities of the target, and $v^i_{mx}(k)$, $v^i_{my}(k)$ are the velocities corresponding to the measured position of the target. To compare the performance of two algorithms numerically, the following equations are used.

$$\zeta_p = \frac{\sum_{k=1}^{S} P_e(k)}{S}, \tag{20}$$

$$\zeta_v = \frac{\sum_{k=1}^{S} V_e(k)}{S}, \tag{21}$$

where ζ_p and ζ_v are the averages of the normalized position and velocity errors, $P_e(k)$ and $V_e(k)$, over the total number of scan S.

The simulation results and the numerical results over 100 runs are shown in Fig. 6 and Table 6. As shown in Fig. 6, the proposed method had much better tracking performance than the AIMM algorithm. Table 6 indicates that the normalized position and velocity errors of the proposed method were reduced by 16.67% and 41.53%, respectively, compared with the AIMM3 and 16.17% and 40.42%, respectively, compared with AIMM5. Although its CPU time over 100 runs was increased to some extent, we could overcome the mathematical limits of the conventional methods.

5. CONCLUSIONS

In this paper, we have developed the GA-based IMM method as an intelligent tracking method for a maneuvering target. In the proposed method, a submodel was represented as a fuzzy system to compute the time-varying variances of the overall process noises of a new piecewise constant white acceleration model and the GA was applied to optimize this fuzzy system for a certain maneuver input. Multiple models were then composed of these fuzzy models. Compared to the IMM algorithm and the AIMM algorithm, the proposed method has three advantages. First, unlike an IMM algorithm, no sub-models predefined in consideration of the property of maneuver are required. Secondly, unlike an AIMM algorithm, no estimation of the target acceleration or adjustment for different acceleration levels in accordance with the property of maneuver is required to construct submodels. Thirdly, although the property of the maneuver is unknown, the proposed method can be applied if the maneuver is within the assumed maximum acceleration input. The simulation results have shown that the proposed method has much better tracking performance than the AIMM algorithm.

REFERENCES

[1] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. on Aerosp. Electron. Syst.*, vol. AES-6, no. 4, pp. 473-483, July 1970.

[2] P. L. Bogler, "Tracking a maneuvering target using input estimation," *IEEE Trans. on Aerosp. Electron. Syst.*, vol. AES-23, no. 3, pp. 298-310, May 1987.

[3] Y. Bar-Shalom and K. Birmiwal, "Variable dimension filter for maneuvering target tracking," *IEEE Trans. on Aerosp. Electron. Syst.*, vol. AES-18, no. 5, pp. 621-629, September 1982.

[4] J. K. Tugnait, "Detection and estimation for

abruptly changing systems," *Automatica*, vol. 18, no. 5, pp. 607-615, 1982.

[5] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with a jump-linear smoothing application", *IEEE Trans. on Autom. Control*, vol. AC-33, no. 8, pp. 780-783, August 1988.

[6] A. Munir and D. P. Atherton, "Adaptive interacting multiple model algorithm for tracking a maneuvering target," *IEE Proc. of Radar, Sonar, and Navigation*, vol. 142, no. 1, pp. 11-17, 1995.

[7] A. T. Alouani, A. T. Xia, P. Price, and W. D. Blair, "A two-stage Kalman estimator for state estimation in the presence of random bias for tracking maneuvering targets," *Proc. of 30th IEEE Conference on Decision and Control*, pp. 2059-2062, December 1991.

[8] L. X. Wang, *A Course in Fuzzy Systems and Control*, Prentice Hall, 1998.

[9] Y. Bar-Shalom and X. Li, *Estimation and Tracking; Principles, Techniques and Software*, Norwood, MA: Arteck House, 1993.

[10] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[11] Y. H. Joo, H. S. Hwang, K. B. Kim, and K. B. Woo, "Linguistic model identification for fuzzy system," *Electron. Letter*, vol. 31, pp. 330-331, 1995.

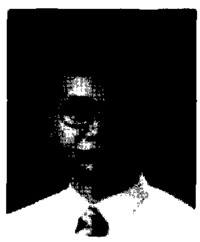
[12] Y. H. Joo, H. S. Hwang, K. B. Kim, and K. B. Woo, "Fuzzy system modeling and its application to mobile robot control," *Fuzzy Logic and its Applications to Engineering, Information Sciences, and Intelligent Systems*, ed. Z. Bien and K C. Min, Kluwer Academic Publisher, pp. 147-156, 1995.

[13] Y. H. Joo, H. S. Hwang, K. B. Kim, and K. B. Woo, "Fuzzy system modeling by fuzzy partition and GA hybrid schemes," *Fuzzy Sets and Systems*, vol. 86, pp. 279-288, 1997.

[14] S. McGinnity and G. W. Irwin, "Fuzzy logic approach to maneuvering target tracking," *IEE Proc. of Radar, Sonar, and Navigation*, vol. 145, no. 6, pp. 337-341, 1998.

Bum-Jik Lee received the B.S. degree in Mechanical Engineering from R.O.K. Naval Academy in 1993, and the M.S. degree in Electrical and Computer Engineering from Yonsei University in 2000. He is presently working toward the Ph.D. degree in Electrical and Electronic Engineering at Yonsei University. His research interests are fuzzy modeling, filtering, maneuvering target tracking, genetic algorithms, and DNA coding methods. nonlinear control, mobile robots, fuzzy logic control, neural networks, genetic algorithms, and Hadamard-transform spectroscopy.

Young-Hoon Joo received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from Yonsei University in 1982, 1984, and 1995, respectively. He is currently an associate professor in the School of Electronic and Information Engineering, Kunsan National University. His major interests are mobile robots, fuzzy modeling and control, genetic algorithms, intelligent control, and nonlinear systems control.



Jin Bae Park received the B.S. degree in Electrical Engineering from Yonsei University in 1977, and the M.S. and Ph.D. degrees in Electrical Engineering from Kansas State University, Manhattan, in 1985 and 1990, respectively. Since 1992, he has been with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea, where he is currently a professor. His research interests include robust control and filtering.