Copyright © 2002 IFAC
15th Triennial World Congress, Barcelona, Spain

NEURAL NETWORK GUIDANCE BASED ON PURSUIT-EVASION GAMES WITH ENHANCED PERFORMANCE

Han-Lim Choi, Min-Jea Tahk, and Hyo-Choong Bang

*Division of Aerospace Engineering, Korea Advanced Institute of Science and Technology,
373-1, Guseong-dong, Yuseong-gu, Daejeon,305-701, Republic of Korea.*

Abstract: This paper deals with a neural network guidance law based on pursuit-evasion games, and performance enhancing methods for the neural network guidance. Two-dimensional pursuit-evasion games solved by using the gradient method are considered. The neural network guidance law in this work employs the range, range rate, line-of-sight rate, and heading error as its input variables. An additional network training method and a hybrid guidance method are proposed for the sake of the interception performance enhancement. Numerical simulations are accompanied for the verification of the neural network guidance law, and validation of the performance enhancement achieved by the proposed methods. Moreover, all proposed guidance laws are compared with proportional navigation. *Copyright © 2002 IFAC*

Keywords: Missile, Guidance system, Differential games, Neural networks, Feedback control

1. INTRODUCTION

This study deals with missile guidance based on pursuit-evasion games. Pursuit-evasion game, which was introduced by Issacs (1967), became an attractive concept in missile guidance, as the need for developing a guidance law that provides good interception performance against an smart target increased. Since pursuit-evasion game considers the worst-case design, it is expected to guarantee acceptable interception performance even when the target aircraft maneuvers in a very intelligent way. Pursuit-evasion game deals with a minimax optimization problem between the missile and the target. Namely, the missile makes an effort to minimize a specified payoff function, while the target maximizes it. Intercept time and/or miss distance is frequently chosen as the payoff of the game.

It is needed to obtain a feedback guidance law for real-time implementation of the pursuit-evasion game. No one can expect a good interception performance using programmed open-loop guidance when real engagement situation is not exactly same as that one considered before. Unfortunately, many solvers for pursuit-evasion game just give open-loop solutions. In this paper, a neural network is employed to synthesize a feedback guidance law from open-loop solutions. The neural network provides an approximate functional relationship between the state variables and the game-optimal control inputs, throughout network training.

This work shares the same basic idea with Choi *et al.* (2001a), in which a three-dimensional missile guidance law is constructed using a neural network. They employed ten variables - three relative position components, three relative velocity components, two line-of-sight rates, missile's speed, and target's speed - as the neural network input variables. The performance of the neural network guidance law was compared with that of proportional navigation guidance. Although results compared are very informative and impressive, most of the network input variables adopted are not measurable, or hardly obtainable. Moreover, they focused only on verifying the similarity of the neural network guidance law with the original optimal solutions.

This paper makes up for insufficiencies of Choi *et al.* (2001a) in that the neural network input variables are selected more reasonably, and the interception performance is enhanced even against the target maneuvering very strangely. This work establishes a neural network guidance law based on the two-dimensional pursuit-evasion game. Four variables, i.e. the range, range rate, heading error, and line-of-sight rate are selected for neural network input variables. In addition, two methods such as additional network training and hybrid guidance are proposed for the sake of improving the interception performance against non game-optimally maneuvering targets. Moreover, proposed guidance methods are compared with proportional navigation in terms of the worst-case performance.

2. TWO-DIMENSIONAL PURSUIT-EVASION GAME

Two-dimensional pursuit-evasion situation is considered as described in the fig. 1. The equations of motion of the missile or the target are expressed as follows:

$$\begin{aligned}\dot{x}_i &= v_i \cos\gamma_i \\ \dot{y}_i &= v_i \sin\gamma_i \\ \dot{\gamma}_i &= \frac{v_i}{R_i}u_i = \frac{1}{v_i}\left(\frac{v_i^2}{R_i}u_i\right) = \frac{a_i}{v_i}, \quad |u_i| \le 1 \qquad (1) \\ \dot{v}_i &= -\frac{v_i^2}{R_i}(b_i + c_i u_i^2) \\ &(i = M, T)\end{aligned}$$

where x, y are the missile's or the target's position and v is the speed and γ the flight path angle, respectively. u is the normalized control input, and R is the minimum turn radius. In addition, a is the lateral acceleration command. b and c are related to the lift and/or drag coefficients. The values of R, b, and c for each player are given as follows: $b_M = 0.0875$, $c_M = 0.40$, $R_M = 1515.15$m, $b_T = 0$, $c_T = 0.40$, and $R_T = 600$m. The subscript *'M'* denotes the missile, and *'T'* the target.

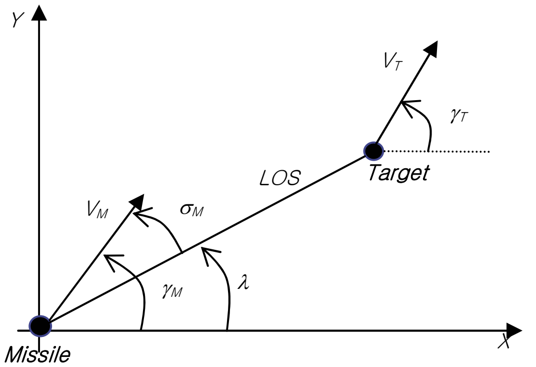

Fig. 1. Two-dimensional pursuit-evasion situation

With the missile's and the target's dynamic models, the author takes into account a time-optimal differential game, which can be expressed as

$$\begin{aligned}&\max_{u_T(t)} \min_{u_M(t)} J = t_f \\ &\text{s.t. equations of motion}\end{aligned} \qquad (2)$$

$u_M(t)$ or $u_T(t)$ implies the time history of the missile's or target's control input.

This kind of differential game can be solved by some numerical algorithms, such as indirect method, gradient-based method (Tahk *et al.*, 1998), bilevel programming (Ehtamo and Raivio, 2001), and co-evolutionary methods (Kim and Tahk, 2001; Choi *et al.*, 2002). This work employs the gradient-based method devised by Tahk *et al.* (1998), which is a direct optimization method based on control input parameterization. The control inputs of the missile and the target are discretized with time step δt (= t_f/N) as the following:

$$\begin{aligned}\mathbf{u}_M &= [u_{M,1}, u_{M,2}, \cdots, u_{M,N}]^T \\ \mathbf{u}_T &= [u_{T,1}, u_{T,2}, \cdots, u_{T,N}]^T\end{aligned} \qquad (3)$$

$u_{M,k}$ and $u_{T,k}$ are the control input in the k-th interval, which are assumed constant during the corresponding time interval. Hence, the gradient-based method provides game-optimal parameterized control inputs. In other words, the open-loop solutions for game-optimal control are available by using the gradient-based method.

3. STRUCTURE OF NEURAL NETWORK GUIDANCE LAW

The neural network (NN) feedback guidance law implies an approximate functional relation between the state variables and the game-optimal control inputs. The guidance NN takes current state information as its input and provides an optimal - strictly speaking, sub-optimal - guidance command to the missile. If it is possible to gather all the state information, the best choice for the NN inputs are all, that is, both the missile's and the target's state variables. However, unfortunately, this choice is impossible in real implementation, since not all the state values can be measured. Instead, just a few variables are measured and the other variables are estimated based on the measurement. Therefore, it is reasonable to select NN input variables as those that can be measured or at least easily estimated. It is also important keep the approximation accuracy, though. The basic architecture of the NN feedback guidance loop is given in fig. 2. To a designer of the guidance law, selecting the neural network input vector, or X_{NN}, is the most important issue.

In this paper, it is assumed that the game-optimal guidance law mainly depends on the relative motion between the missile and the target. The key variables to represent the relative motion are the range, rate of change of the range, line-of-sight (LOS) angle, and LOS angular rate.

However, when the missile's guidance command is the lateral acceleration normal to the velocity vector, the absolute value of LOS angle matters little in determining the guidance command. Instead, the heading error, $\sigma_M = \gamma_M - \lambda$, is much more important, since it contains the information of the velocity direction. Therefore, the heading error replaces the LOS angle in this paper. Thus, the neural network input vector consists of the range, range rate, heading error, and LOS angular rate.

In addition, the lateral acceleration a_M is chosen as the NN output variable instead of u_M, since the former contains more physical meaning.

4. SYNTHESIS OF NEURAL NETWORK GUIDANCE

4.1 Neural Network Training

Pursuit-evasion games are solved by the gradient-based method for 20 engagement scenarios, in which the initial γ_M varies 4 times, 0deg thru 30degs by 10degs, and the initial γ_T varies 5 times, 30degs thru 150degs by 30degs, while the initial positions and the speeds are fixed as (x_M, y_M, v_M) = (0m, 0m, 600m/s), and (x_T, y_T, v_T) = (5000m, 0m, 200m/s).

Afterwards, a neural network with 2 hidden layers, each of which has 10/6 neurons, is developed for training. Learning lasts until the normalized output MSE (mean squared error) decreases to 6×10^{-7} by using the Levenberg-Marquardt algorithm (Hagan and Menhaj, 1994).

4.2 Verification of Neural Network Approximation

Since the value of MSE can seldom say about the approximation performance of the NN guidance law, it is needed to reconstruct the trajectories by adopting the NN feedback guidance law in order to examine the approximation performance of it. The authors reconstruct the trajectories for 52 scenarios, 20 pattern scenarios and 32 off-trained scenarios. The off-trained scenarios are selected by changing the target's initial path angle 8 times - 40degs, 50degs, 70degs, 80degs, 100degs, 110degs, 130degs, and 140degs - with the same configuration of the initial position, speed, and the missile's path angle as in the pattern scenarios.

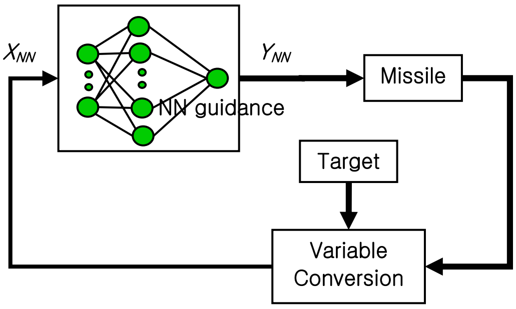

Fig. 2. Basic architecture of neural network feedback guidance

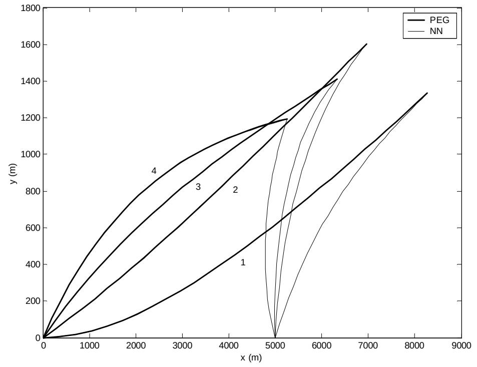

Fig. 3. Trajectories for the verification of neural network approximation

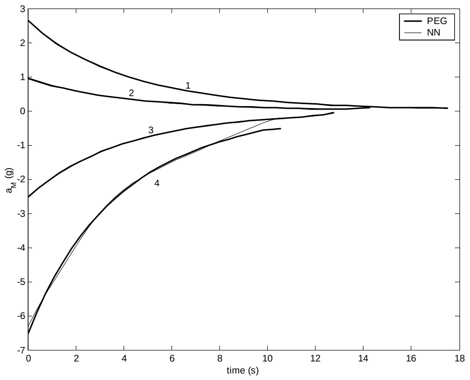

Fig. 4. Missile accelerations for the verification of neural network approximation

Fig. 3 shows the trajectories for representative four scenarios, and fig. 4 depicts the acceleration histories for the same scenarios. It is found that the trajectories constructed by using the NN guidance law are very similar to the original pursuit-evasion game trajectories. As for the control history, two histories are about the same for three scenarios. For the scenario 4, slight difference in the acceleration command is found near the final time; nevertheless, this amount is not so much that determines the success or failure in the interception. For all 52 testing scenarios, the miss distance is less than 0.2m, and the final time error is less than 2×10^{-3} sec. In the consequence, the guidance NN approximates the game-optimal solution to a satisfactory extent.

5. PERFORMANCE ENHANCEMENT OF THE GUIDANCE LAW

Although the NN guidance law described in the previous sections copies the game-optimal solutions well, it does not guarantee good interception performance for all the engagement situations. Since the NN is trained using the trajectory data for the situations in which both the missile and the target take the game-optimal strategies, the missile often fails to generate appropriate guidance command if the target maneuvers in a different way from the game-optimal law. When the target maneuvers slightly differently from the game-optimal law, the feedback structure of the NN guidance law satisfactorily compensates the error, and leads to the success in the interception. However, the interception performance of the NN guidance law greatly degrades when the target maneuvers in a very different way. For example, the missile using NN guidance law even fails to capture a dumb target in some cases.

There might be two approaches to overcome this defect above; one is training the NN using additional pattern scenarios, and the other is to compensate or aid the NN guidance law in some ways. As for the first approach, this paper proposes two ways of selection of additional training patters. For the second approach, a hybrid guidance scheme is proposed.

5.1 Additional Network Training

It is obvious that a nearly perfect neural network guidance law is obtained, if the network is trained taken into account all possible engagement situations. Unfortunately, this is impossible in the real design process. Instead, the designer selects some scenarios standing for the engagement situations that one wants to deal with. Actually, the pattern scenarios in the previous section were selected in this manner. However, the performance of the NN guidance law constructed from those scenarios is only guaranteed when the target maneuvers similar to the game solutions, and the interception performance is not satisfactory. This means the scenarios selected before does not contain all information that the authors wanted to consider. To supplement the lack of information, additional scenarios are required in the NN training. This section proposes two ways of supplementing the pattern scenarios.

Game Solutions along the Fictitious Trajectories; First of all, let assume that the authors are interested in improving the interception performance of the NN guidance law for a specific engagement in which the target maneuvers with constant u_T, and initial γ_M and γ_T are 0 deg and 90 deg, respectively. Since the target's game solution for that initial engagement is not a constant input maneuver, it is impossible to obtain the exact trajectories expressing what happens when the game-optimally guided missile chases the target. Instead, the trajectories are available if the missile is assumed to take sampled-feedback guidance with a finite sampling step.

Five cases of target's control commands – 1.0, 0.5, 0, -0.5, and -1.0 – are considered, and the missile is assumed to update its strategy every 2 seconds. In other words, the missile is open-loop guided using the game solution for 2 seconds, and then the game solution with a new initial condition is solved, and this solution is used for guiding the missile for the next 2 seconds.

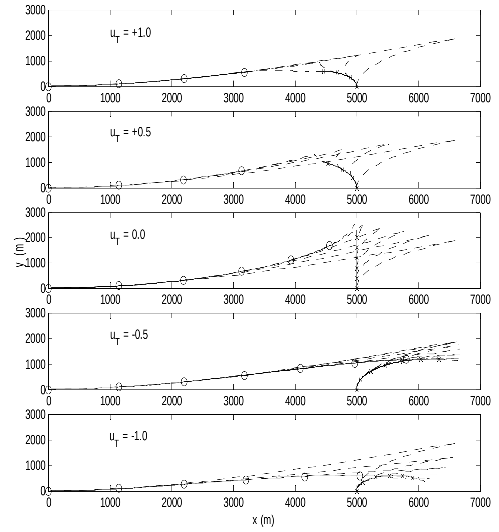

Fig. 5. Additional training scenarios for intercepting the target with constant control input

In this way, the authors can obtain 22 more game solutions. In fig. 5, at the marked position (o : missile, x : target), the pursuit-evasion game solutions are evaluated using the gradient-based method. These 22 scenarios – 3 for u_T is 1.0, 3 for u_T is 0.5, 5 for u_T is 0.0, 6 for u_T is -0.5, and 5 for u_T is -1.0 – are added to the training patterns, and total 42 pattern scenarios are trained. The network training proceeds until the MSE converges to 2×10^{-5}. Let denote this NN as $\mathrm{NN_B}$, while denote the original NN constructed in the previous section as $\mathrm{NN_A}$.

General Geometries for Shorter-Range Engagements; Although selecting the additional pattern scenarios along the fictitious trajectory for intercepting a specific target is reasonable approach, it requires some tedious labors; solve the game solution and propagate it for one guidance step, and solve a new game solution at that position, and so on. Instead of this, just choosing more scenarios in shorter-range engagements can be helpful. For the shorter-range cases, the guidance commands vary more rapidly than for the longer-range ones. Thus, it is expected for the NN guidance law to compensate the errors more promptly if it contains the information of shorter-range engagements.

Twenty scenarios are selected in the engagements with initial range of 3km, while the missile's and the target's path angles change in the same manner with the engagements with initial range of 5km. Hence, total 40 pattern scenarios are trained until the MSE reaches 2×10^{-5}. Let denote this network as $\mathrm{NN_C}$.

Performance Comparison; Table 1 shows the interception results (final time and miss distance) of the three NNs against a constant-radius turning target. It is found that $\mathrm{NN_B}$ provides good intercepting performance as a whole, while $\mathrm{NN_A}$ does not give good performance when u_T is 0.0 and 1.0. Although $\mathrm{NN_C}$ fails to intercept the target when u_T is -1.0, it is much better than $\mathrm{NN_A}$. The bold number imply the worst-case for each NN in the manner of the miss distance.

5.2 Hybrid Guidance

This section introduces another algorithm, called as a hybrid guidance method, for enhancing the interception performance of the NN guidance law. The hybrid guidance (HG) means a combination of the NN guidance law and an existing guidance law, such as PN (proportional navigation) and APN (augmented PN). The reason why the NN guidance fails to intercept is the target moves very differently from what the missile expects. Therefore, it is reasonable to adapt the missile's guidance algorithm to the target's maneuvering technique; if the target seems to behave game-optimally, then use the NN guidance, if not, use the PN guidance. This paper proposes the following adaptation scheme:

```
if $G(t-1) == NN$,
   if $|\Delta a_T(t-i)| > a_a,\ \forall i = 1,2,\cdots,n$
      $G(t) = PN$
   endif
else
   if $|\Delta a_T(t-i)| < a_a,\ \forall i = 1,2,\cdots,n$
      $G(t) = NN$
   endif
endif
```

where t is current guidance step, $G(t)$ is current guidance scheme that is initially set as NN, Δa_T is the difference in target acceleration between the game-optimal command and the actual command ($\Delta a_T = a_T - \bar{a}_T$), a_a is the allowable threshold of Δa_T , and n is a specified integer. In other words, if the missile successively observes the target command differ from (or is similar to) the game-optimal one for n guidance steps, then change the guidance scheme from NN to PN (or from PN to NN). Here, to select α and n is a critical issue. α can be chosen after some testing of the NN guidance law, and n has to be selected to accomplishing the interception not causing a chattering problem. In this work, α is 0.5g and n is 3. Fig. 6 illustrates basic structure of the proposed hybrid guidance.

Table 1. Interception performance improvement by additional network training

u_T	$\mathrm{NN_A}$		$\mathrm{NN_B}$		$\mathrm{NN_C}$	
	t_f(s)	r_f(m)	t_f(s)	r_f(m)	t_f(s)	r_f(m)
1.0	8.313	17.159	8.279	4.377	8.274	0.176
0.5	8.542	11.037	8.503	**9.825**	8.486	0.923
0.0	12.186	**52.537**	12.269	4.4901	12.537	0.230
-0.5	13.952	8.024	13.965	0.263	14.042	0.599
-1.0	12.333	1.749	12.373	0.132	12.324	**15.140**

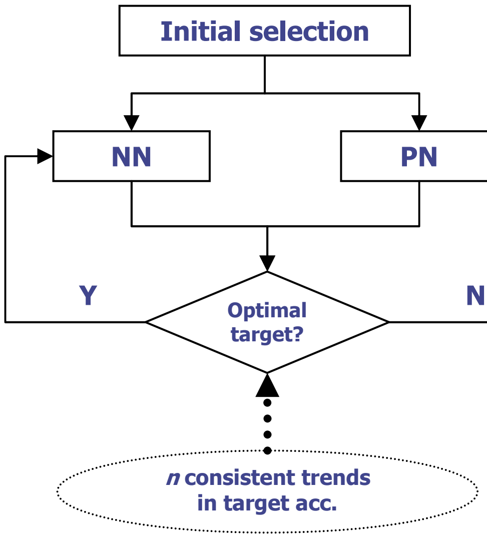

Fig. 6. Basic structure of the hybrid guidance

In addition, in order to evaluate Δa_T, target's actual and game optimal acceleration are required. When using a target tracking filter, the actual acceleration is available after 1 guidance step. As for the game-optimal acceleration, this thesis uses one more NN for the missile to estimate the target's game-optimal acceleration command. The game-optimal target acceleration is assumed as a function of the range, range rate, LOS, and LOS rate. Namely, $\bar{a}_T \approx \bar{a}_{T,NN}(r,\dot{r},\lambda,\dot{\lambda})$. Then, an approximate value of the nominal target acceleration is available using one more neural network.

For the same pattern scenarios with $\mathrm{NN_A}$, the target acceleration estimating NN are trained. A two-hidden-layer NN with 10/6 neurons is trained until the training error decreases to 3×10^{-6}.

6. COMPARISON WITH PROPORTIONAL NAVIGATION

Neural network guidance laws described so far are compared with PN guidance. For the initial condition of (x_M, y_M, γ_M, v_M) = (0m, 0m, 0deg, 600m/s), and (x_T, y_T, γ_T, v_T) = (5km, 0m, 90degs, 200m/s), six guidance laws ($\mathrm{NN_A}$, $\mathrm{NN_B}$, $\mathrm{NN_C}$, Hybrid, $\mathrm{PN_3}$, $\mathrm{PN_4}$) are compared with each other. $\mathrm{PN_3}$ and $\mathrm{PN_4}$ denote the PN guidance with gain 3 and 4, respectively, while 'Hybrid' guidance law is a combination of $\mathrm{NN_A}$ and $\mathrm{PN_4}$. In order to test the performance, six target maneuvers are considered; differential game maneuver (DG), a dumb target (Dumb), maximum turn maneuver (Max), time-optimal maneuver against $\mathrm{PN_3}$ ($\mathrm{Opt_3}$), time-optimal maneuver against $\mathrm{PN_4}$ ($\mathrm{Opt_4}$), and anti-PN maneuver with gain 5 ($\mathrm{Anti_5}$). The time-optimal solutions against PN guidance are obtained using co-evolutionary augmented Lagrangian method (CEALM) (Tahk and Sun, 2000; Choi et al, 2001b).

Table 2 shows the final time for each engagement. The bold character denotes the worst case performance of the corresponding guidance law, and the underline means the failure in the interception. The miss distance for 'NN$_A$ vs Dumb' is 52.539m, and that for 'NN$_C$ vs Max' is 15.140m, while for all other cases miss distances are less than 10.0m. Among the NN guidance laws, NN$_B$ provides the best performance succeeding in interception all the targets. It is also noted that the NN$_B$ provides the best worst-case performance among all the guidance laws. Moreover, the fact that the worst case of NN$_B$ occurred in the engagement versus DG implies that NN$_B$ approximates the differential game strategy very well. It cannot be overlooked that the hybrid guidance guarantees good performance as a whole, providing better worst-case performance than PN guidance. This shows the feasibility of application of hybrid guidance scheme in real situations.

7. CONCLUSIONS

A neural network guidance law adopting the range, range rate, line-of-sight rate, and heading error as its input variables is established by using the game solutions solved by the gradient-based method. In order to enhance the interception performance of the neural network guidance law against non game-optimal target maneuvers, two techniques for selecting additional training scenarios and a hybrid guidance scheme are proposed. Numerical simulations are performed for verifying the neural network approximation, and for examining the performance improvement due to the proposed performance enhancement schemes. All proposed neural network guidance methods are compared with proportional guidance in the aspect of worst-case performance. The neural network guidance law reinforced by additional fictitious scenarios and the hybrid guidance law provide outstanding performance.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support by Agency for Defense Development and by Automatic Control Research Center, Seoul National University.

Table 2. Performance comparison of neural network guidance laws and PN guidance (final time)

	NN$_A$	NN$_B$	NN$_C$	Hybrid	PN$_3$	PN$_4$
DG	14.391	**14.391**	14.391	14.391	14.634	14.846
Dumb	**12.186**	12.269	12.536	11.641	11.621	11.587
Max	12.333	12.373	**12.324**	12.630	12.581	12.725
Opt$_3$	14.295	14.300	14.298	14.423	**14.946**	14.924
Opt$_4$	14.270	14.273	14.274	14.521	14.658	**15.353**
Anti$_5$	14.289	14.291	14.241	**14.526**	14.157	13.947

REFERENCES

Basar, T. and G.J. Olsder (1999). Dynamic Noncooperative Game Theory, SIAM, Philadelphia

Choi, H.L., Y. Park, H.G. Lee and M.J. Tahk (2001a). A Three-dimensional differential game missile guidance law using neural networks, AIAA-2001-4343, *AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada.

Choi, H.L., H.C. Bang and M.J. Tahk (2001b). Co-evolutionary optimization of three-dimensional target evasive maneuver against a proportionally guided missile, *Proceedings of Congress on Evolutionary Computation*, pp. 1406-1413, Seoul, Korea.

Choi, H.L., M.J. Tahk and H.C. Bang (2002). A novel co-evolutionary method for solving pursuit-evasion games, *Submitted to 10th International Symposium on Dynamic Games and Applications*, Petersburg, Russia.

Ehtamo, H. and T. Raivio (2001). On applied nonlinear and bilevel programming for pursuit-evasion games, *Journal of Optimization Theory and Applications*, **Vol. 108**, **No. 1**, pp. 65-96.

Hagan, M.T. and M.B. Menhaj (1994). Training feedforward networks with Marquardt algorithm, *IEEE Transactions of Neural Networks*, **Vol. 5**, **No. 6**, pp. 989-993.

Isaacs, R. (1967), *Differential Games*, John Wiley and Sons, New York.

Kim, J.G. and M.J. Tahk (2001), Co-evolutionary computation for constrained min-max problems and its applications for pursuit-evasion games, *Proceedings of Congress on Evolutionary Computation*, pp. 1205-1212, Seoul, Korea.

Tahk, M.J., H. Ryu and J.G. Kim (1998). An iterative numerical method for a class of quantitative pursuit-evasion games, *Proceedings of AIAA Conference on Guidance, Navigation, and Control*, pp. 175-182, Boston, Mass., USA.

Tahk, M.J. and B.C. Sun (2000). Coevolutionary augmented Lagrangian methods for constrained optimization, *IEEE Transactions on Evolutionary Computation*, **Vol. 4, No. 2**, pp. 114-124.