Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Control Engineering Practice 14 (2006) 735–742
CONTROL ENGINEERING PRACTICE
www.elsevier.com/locate/conengprac

Neural network guidance based on pursuit-evasion games with enhanced performance ☆

Han-Lim Choi, Min-Jea Tahk*, Hyochoong Bang

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

Received 9 August 2003; accepted 11 March 2005
Available online 3 May 2005

Abstract

This paper addresses a neural network guidance based on pursuit-evasion games, and performance enhancing methods for it. Two-dimensional pursuit-evasion games solved by the gradient-based method are considered. The neural network guidance law employs the range, range rate, line-of-sight rate, and heading error as its input variables. Additional pattern selection methods 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 approximation, and of the improved interception performance by the proposed methods. Moreover, all proposed guidance laws are compared with proportional navigation.

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 Isaacs (1967) in the first place, has become an attractive concept in missile guidance, as the need for a guidance law guaranteeing good interception performance against a smart target increased. (Ehtamo & Raivio, 2001; Faber & Shinar, 1980; Shima & Shinar, 2002) Since pursuit-evasion game considers the worst-case design, it is expected to warrant acceptable interception performance even when a target aircraft maneuvers in a very intelligent way. Pursuit-evasion game considers a minimax optimization problem between the missile and the target. In other words, the missile makes an effort to minimize a specified payoff function, while the target maximizes it. Interception time and miss distance are frequently chosen as the payoff of the game. (Breitner, Pesch, & Grimm, 1993; Shima & Shinar, 2002; Tahk, Ryu, & Kim, 1998; Ehtamo & Raivio, 2001) Intercept time has been preferred as the payoff if the model dynamics are complicated, since it entails easier mathematical formulation.

It is needed to obtain a feedback guidance law for real-time implementation of pursuit-evasion game. No one can expect good interception performance when using pre-programmed open-loop guidance, since real engagement situations are not exactly same as those one considered before. Unfortunately, many solvers for pursuit-evasion game merely give open-loop solutions. Some researches have been conducted for obtaining feedback type game solutions. (Ben-Asher, 1996; Faber & Shinar, 1980; Menon, 1989) However, all these works led to complicated problem formulations, which inevitably limited the extension of each idea to very simple cases. A neural network can be a good help, since it is a universal approximator (Hornik, Stinchcombe, &

☆This is the revised version of the paper with Editorial Reference Number CD 2379.

*Corresponding author. Tel.: +82 42 869 3718; fax: +82 42 869 3710.
E-mail address: mjtahk@fdcl.kaist.ac.kr (M.-J. Tahk).

0967-0661/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2005.03.001

White, 1989). It can yield an approximate functional relation between the state variables and the game-optimal control inputs. In addition, Song and Tahk (1998, 1999, 2001, 2002) have substantiated the feasibility of this concept in missile midcourse guidance, although they considered one-sided optimal problems rather than game-optimal problems. For this reason, in this work, a neural network is employed to synthesize a feedback guidance law from open-loop solutions.

The authors have studied neural network guidance law based on the pursuit-evasion game solutions obtained by using the gradient-based method for both two-dimensional and three-dimensional situations. (Choi, Park, Lee, & Tahk, 2001; Choi, Tahk, Bang, & Lee, 2001; Lee, Choi, Tahk, & Bang, 2001). While investigating the outcomes, however, it is observed that to select neural network input variable plays a key role in determining the performance of the guidance law, and it is also observed that the performance of the neural network guidance law degrades too much, when the target does not maneuver along the game-optimal trajectory obtained in advance. Based on these two observations, this paper focuses on the selection of the network input variables and on the ways of overcoming the undesirable feature above.

This paper derives a neural network guidance law from two-dimensional pursuit-evasion games. This study focuses on only two-dimensional situations, since they are more appropriate for elucidating the qualitative features of the pursuit-evasion game and the neural network guidance. Four variables, i.e. the range, range rate, heading error, and line-of-sight (LOS) rate are selected as neural network input variables. Two methods are also proposed for the sake of improving the interception performance against not game-optimally maneuvering targets: additional pattern scenario selection, and hybrid guidance. In addition, performance of the neural network guidance laws is compared with proportional navigation.

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 and the target are expressed as follows.

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

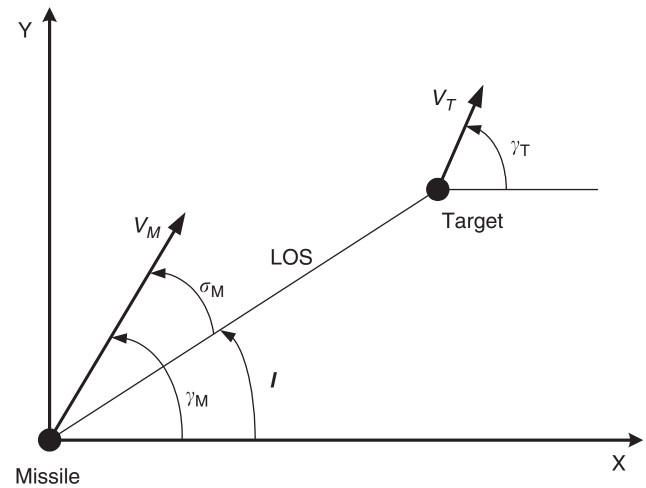

Fig. 1. Two-dimensional pursuit-evasion situation.

where x, y are the missile's or the target's position, v is the speed and γ is 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. α and β are related to aerodynamic coefficients. The values of R, α, and β for each player are given as follows: $\alpha_M = 0.0875$, $\beta_M = 0.40$, $R_M = 1515.15\,\text{m}$, $\alpha_T = 0$, $\beta_T = 0.40$, and $R_T = 600\,\text{m}$. The subscript 'M' denotes the missile, and 'T' the target.

With these dynamic models of both players, the authors take into account a time-optimal differential game, which can be expressed as

$$\max_{u_T(t)} \min_{u_M(t)} J = t_f,$$

where

$$t_f \triangleq \inf\{t \in [0, \infty) : r(t) = 0\} \tag{2}$$

$u_M(t)$ or $u_T(t)$ implies the time history of the missile's or target's normalized control input described in Eq. (1), and $r(t)$ is the range between the missile and the target at time t.

This kind of differential game can be solved by some numerical algorithms, such as indirect methods, the gradient-based method (Tahk et al., 1998), the bilevel programming (Ehtamo & Raivio, 2001), and co-evolutionary methods (Kim & Tahk, 2001; Choi, Ryu, Tahk, & Bang, 2004). 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 discretize with time step $\delta t (= t_f/N)$ as the following:

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

$u_{i,k}(i = M, T)$ is the control input during the kth interval, which is assumed constant during the corresponding time interval. Hence, the gradient-based method offers game-optimal parameterized control inputs. In other words, the open-loop solutions for game-optimal control are available by using the gradient-based method. Moreover, optimal trajectories obtained by numerical methods for finding open-loop represented solutions are equivalent to those constructed by optimal feedback game strategies, if the dynamics of each player is decoupled and the payoff function is terminal. (Basar & Olsder, 1999) Thus, it is theoretically possible to obtain both open-loop and feedback game solutions that result in exactly same trajectories.

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 a sub-optimal guidance command to the missile. If it is possible to gather all the state information, the best choice for the NN inputs of the guidance NN is to select all the state variables both of the missile and of the target. However, unfortunately, this choice is impossible in real implementation, since all the state values cannot 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 variables that can be measured or at least can easily be estimated. It is also important not to sacrifice the approximation accuracy, though. The basic architecture of the NN feedback guidance loop is given in Fig. 2. For a designer of the guidance law, selection of the neural network input vector, or X_{NN}, is the most important issue.

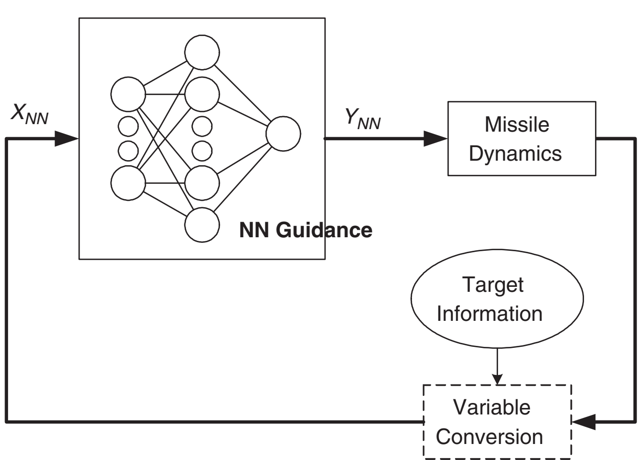

Fig. 2. Basic architecture of neural network feedback guidance.

In this paper, it is assumed that the game-optimal guidance law mainly depends on 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, LOS angle, and LOS angular rate.

However, when the lateral acceleration normal to the velocity vector not to the LOS vector is considered as the missile's guidance command, the absolute value of LOS angle (λ in Fig. 1) matters less 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, 0°–$30^\circ \times 10^\circ$, and the initial γ_T varies 5 times, 30°–$150^\circ \times 30^\circ$, while the initial positions and the speeds are fixed as $(x_M, y_M, v_M) = (0\,\text{m}, 0\,\text{m}, 600\,\text{m/s})$, and $(x_T, y_T, v_T) = (5000\,\text{m}, 0\,\text{m}, 200\,\text{m/s})$ (Fig. 3).

Afterwards, a neural network with 2 hidden layers is developed for training; the number of neurons is 10 and 6 for the first and second hidden layer, respectively. The activation function of each neuron is hyperbolic tangent function: $f_{out} = C_1 \tanh(C_2 f_{in})$, where $C_1 = 1.0$, $C_2 = 0.5$. Learning continues until the normalized output MSE (mean squared error) decreases to 6×10^{-7} by using the Levenberg–Marquardt algorithm (Hagan & Menhaj, 1994).

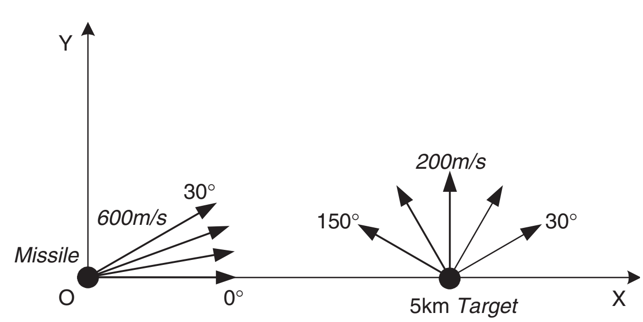

Fig. 3. Training scenarios.

4.2. Verification of neural network approximation

Since the low value of MSE cannot guarantee the approximation performance of the NN guidance law, it is needed to simulate the trajectories by the NN feedback guidance law in order to examine the approximation performance of it. The authors calculate 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—40°, 50°, 70°, 80°, 100°, 110°, 130°, and 140°—with the same configuration of the initial position, speed, and the missile's path angle as those of the pattern scenarios.

Fig. 4 shows the trajectories for representative four scenarios described in Table 1; Fig. 5 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. With respect to the control history, two histories are about the same for three scenarios. For 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 0.2 m, and the final time error is less than 2×10^{-3} s. In the consequence, the guidance NN approximates the game-optimal solutions to a satisfactory extent.

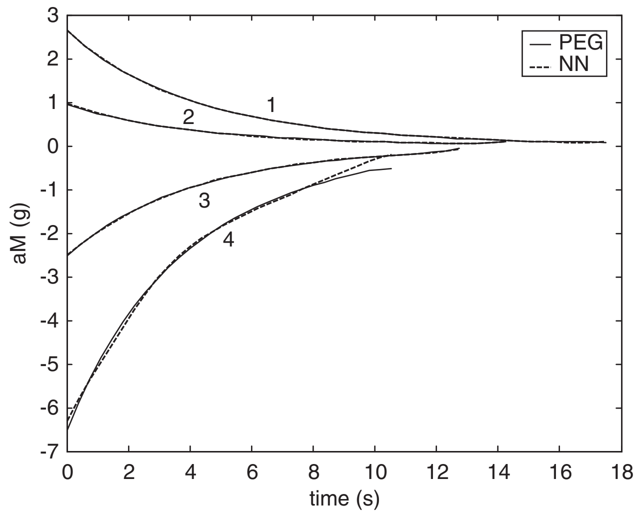

Fig. 5. Missile accelerations for the verification of neural network approximation.

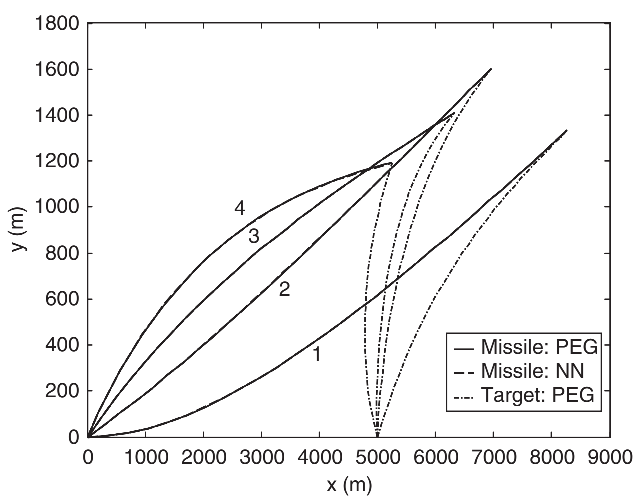

Fig. 4. Trajectories for the verification of neural network approximation.

Table 1
Path angles for illustrated scenarios

Scenario	γ_M°	γ_T°
1	0	40
2	10	80
3	20	100
4	30	140

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 adopt 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 guidance error, thereby leading to the success in the interception. However, this is not the case, when the target maneuvers in a disparate way; the interception performance greatly degrades in this situation. For example, sometimes the missile using NN guidance law even fails to capture a dumb target.

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

5.1. Additional network training

It is obvious that an almost perfect neural network guidance law would be obtained, if the network training could cover all the possible engagement situations. Unfortunately, this is impossible in the real design process. Instead, a designer selects some scenarios standing for the engagement situations that he/she 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 similarly to the game solutions; otherwise, the interception performance is not desirable. This means that the scenarios selected before does not contain all the 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 replenishing the pattern scenarios.

5.1.1. Game solutions along the fictitious trajectories

First of all, let us assume that the authors are interested in improving the interception performance of the NN guidance law for a specific engagement: the target maneuvers with constant u_T; initial γ_M and γ_T are 0° and 90°, 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 can be computed 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 s. In other words, the missile is open-loop guided using the game solution for 2 s, and then the game solution with a new initial condition is solved; this solution is used for guiding the missile during the next 2 s.

In this way, the authors can obtain 22 more game solutions. Started at the marked positions (○: missile, Δ: target) in Fig. 6, 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; therefore, total 42 pattern scenarios are trained. The network training proceeds until the MSE converges to 2×10^{-5}. Let denote this NN as NN_B, while denote the original NN constructed in the previous section as NN_A.

5.1.2. 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, 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

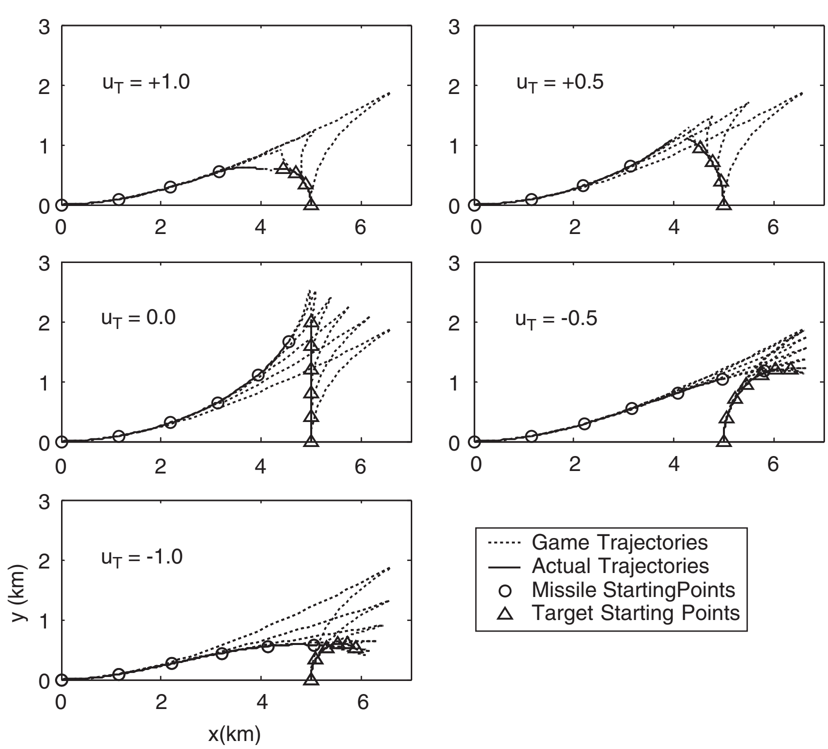

Fig. 6. Additional training scenarios for intercepting the target with constant control input.

Table 2
Interception performance improvement by additional network training

u_T	Performance criteria	NN_A	NN_B	NN_C
1.0	t_f (s)	8.313	8.279	8.274
	r_f (m)	17.159	4.377	0.176
0.5	t_f (s)	8.542	8.503	8.486
	r_f (m)	11.037	**9.825**	0.923
0.0	t_f (s)	12.186	12.269	12.536
	r_f (m)	**52.537**	4.4901	0.230
−0.5	t_f (s)	13.952	13.965	14.042
	r_f (m)	8.024	0.263	0.599
−1.0	t_f (s)	12.333	12.373	12.324
	r_f (m)	1.749	0.132	**15.140**

Note: The underlined bold figures imply the worst case for each neural network guidance law.

law to compensate the guidance errors more promptly, if it contains the information of shorter-range engagements.

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

5.1.3. Performance comparison

Table 2 shows the interception results— final time and miss distance—of the three NNs against constant-radius turning targets. The capture radius of the missile's warhead is assumed to be 10.0 m. It is found that NN_B provides good interception performance as a whole, while NN_A does not give good performance except when u_T is negative. Although NN_C fails to intercept the target when u_T is -1.0, it is, taken altogether, much better than NN_A. The bold number implies 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. Hybrid guidance 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 that 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 PN guidance. This paper proposes the following adaptation scheme:

```
G(0) = NN
    t := 1
    while (c(t) ≠ true) do
      if G(t − 1) == NN,
        if |Δa_T(t − i)| > ε   ∀ i = 1, 2, ···, n
          G(t) = PN
        endif
      else
        if |Δa_T(t − i)| < ε   ∀i = 1, 2, ···, n
          G(t) = NN
        endif
      t = t + 1
      endif
    enddo
```

where t is the current guidance step, $G(t)$ is the current guidance scheme that is initially set as NN, $c(t)$ is the termination criteria of an engagement, Δa_T is the difference in target acceleration between the game-optimal command and the actual command ($\Delta a_T = a_T - \bar{a}_T$), ε is the allowable threshold of Δa_T, and n is a specified integer. In other words, if the missile successively observes the target command differs from (or is similar to) the game-optimal one for n guidance steps, then it changes the guidance scheme from NN to PN (or from PN to NN). Here, to select ε and n is a critical issue. ε can be chosen after testing the NN guidance law, and n has to be selected to accomplish the interception not causing a chattering problem. In this work, ε is 0.5 g and n is 3.

In addition, in order to evaluate Δa_T, target's actual and game-optimal acceleration should be prepared in advance. It can be assumed that the missile uses a target tracking filter; therefore, it can estimate the target's actual acceleration. In actual situations, the tracking filter would suffer from the estimation noise; nevertheless, this work ignores it in order to analyze the feasibility and validity of the idea rather conceptually and qualitatively. The missile can estimate target's game-optimal acceleration, if adopting one more NN. The authors assume the game-optimal target acceleration to be a function of the range, range rate, LOS, and LOS rate. Namely, $\bar{a}_T \approx \bar{a}_{T,NN}(r, \dot{r}, \lambda, \dot{\lambda})$.

With the same pattern scenarios for NN_A, the authors train target acceleration estimating NN; a two-hidden-layered NN with 10 and 6 neurons is trained until the training error decreases to 3×10^{-6}. Numerical simulation shows that this NN approximates the target's game-optimal maneuver to a sufficient extent.

6. Comparison with proportional navigation

Neural network guidance laws described heretofore are compared with PN guidance. For the initial condition of $(x_M, y_M, \gamma_M, v_M) = (0\,\mathrm{m}, 0\,\mathrm{m}, 0^\circ, 600\,\mathrm{m/s})$, and $(x_T, y_T, \gamma_T, v_T) = (5\,\mathrm{k}, 0\,\mathrm{m}, 90^\circ, 200\,\mathrm{m/s})$, six guidance laws ($NN_A$, NN_B, NN_C, Hybrid, PN_3, PN_4) are compared with each other. PN_3 and PN_4 denote PN guidance with gain 3 and 4, respectively. In other words, $a_M = N_{PN}\dot{\lambda}v_M$ ($N_{PN} = 3$ or 4) for there cases. 'Hybrid' guidance law is a combination of NN_A and 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 PN_3 (Opt_3), time-optimal maneuver against PN_4 (Opt_4), and anti-PN maneuver with gain 5 ($Anti_5$). For 'Dumb,' $u_T = 0$, while for 'Max,' $u_T = -1$. 'Opt_3' and 'Opt_4' are optimal target trajectories for the following problem:

$$\max_{u_T(t)} J = t_f$$
$$\text{when } a_M = N_{PN}\dot{\lambda}v_M \ (N_{PN} = 3 \text{ or } 4). \tag{4}$$

These optimization problems are solved by using co-evolutionary augmented Lagrangian method (CEALM) (Tahk & Sun, 2000; Choi, Bang, & Tahk, 2001). For '$Anti_5$,' $a_T = -N_{APN}\dot{\lambda}v_T$ where a_T is the target's lateral acceleration, and the value of N_{APN} is 5.

Table 3 shows the final time for each engagement. The bold character denotes the worst case of the corresponding guidance law; the underline means that the missile fails to intercept the target. Obviously, failure of interception is worse situation for the missile than any other situation resulting in success of interception. The miss distance for 'NN_A vs. Dumb' is 52.539 m, and that for 'NN_C vs. Max' is 15.140 m, while for all other cases miss distances are less than 10.0 m. Among the NN guidance laws, NN_B provides the best performance, succeeding in interception all the targets. It is noticeable 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 means that NN_B approximates the differential game strategy very well. It is also found that NN guidance laws excel PN guidance laws, when the target maneuvers time-optimally against PN. In addition, it should not 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. Figs. 7 and 8 demonstrate the simulation results—trajectories and control histories, respectively,—of 3 engagements: 'NN_B, Hybrid, PN_3 vs. Opt_3'. It is observed that 'Hybrid' guided missile switches its guidance scheme halfway, and that 'PN_3' guided missile goes a long way round until it intercepts the target.

7. Conclusions

A neural network guidance law adopting the range, range rate, LOS rate, and heading error as its input variables is established based on the two-dimensional game solutions solved by the gradient-based method. In order to enhance the interception performance of the neural network guidance law, two techniques for selecting additional training scenarios and a hybrid

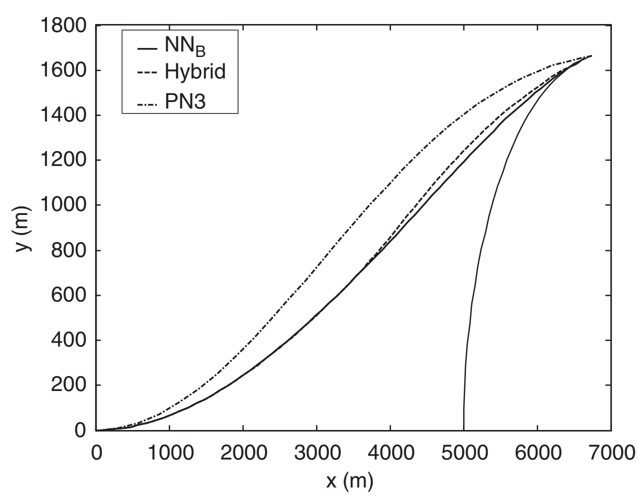

Fig. 7. Trajectories against a time-optimally maneuvering target.

Table 3
Final times for engagement between each guidance law and each maneuver law

Missile/Target	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	**<u>12.186</u>**	12.269	12.536	11.641	11.621	11.587
Max	12.333	12.373	**<u>12.324</u>**	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

Note: The bold figures correspond to the worst cases for each guidance law of the missile; moreover, the underline means the failure of interception.

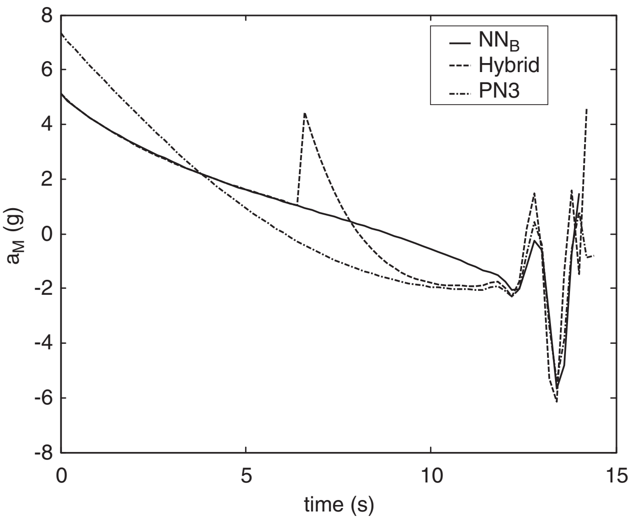

Fig. 8. Missile's acceleration histories against a time-optimal maneuvering target.

guidance scheme are proposed. Numerical simulations shows that the neural network approximation is desirable and two proposed pattern selection methods are effective. All proposed guidance methods are compared with proportional guidance in the respect of worst-case performance. The neural network guidance law reinforced by additional fictitious scenarios and the hybrid guidance law provide outstanding performance. This study only focuses on the two-dimensional problem, and many important features of neural network guidance based on pursuit-evasion games are enlightened. However, for practical application, similar research considering more complicated three-dimensional problems should be followed.

Acknowledgements

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

References

Basar, T., & Olsder, G. J. (1999). *Dynamic noncooperative game theory*. Philadelphia: SIAM.

Ben-Asher, J. Z. (1996). Linear quadratic pursuit-evasion games with terminal velocity constraints. *Journal of Guidance, Control, and Dynamics*, *19*(2), 499–501.

Breitner, M. H., Pesch, H. J., & Grimm, W. (1993). Complex differential games of pursuit-evasion type with state constraints, Part 2: Numerical computation of optimal open-loop strategies. *Journal of Optimization Theory and Applications*, *78*(3), 443–463.

Choi, H. L., Bang, H. C., & Tahk, M. J. (2001). 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., Park, Y., Lee, H. G., & Tahk, M. J. (2001). A Three-dimensional differential game missile guidance law using neural networks. *Proceedings of AIAA Guidance, Navigation, and Control Conference* (no. 2001–4343), Montreal, Canada.

Choi, H. L., Ryu, H., Tahk, M. J., & Bang, H. C. (2004). A co-evolutionary method for pursuit-evasion games with non-zero lethal radii. *Engineering Optimization*, *36*(1), 19–36.

Choi, H. L., Tahk, M. J., Bang, H. C., & Lee, H. G. (2001), Synthesis and investigation of the neural network guidance based on pursuit-evasion games. *Proceedings of International Conference on Control, Automation, and Systems* (pp. 1133–1136), Jeju, Korea.

Ehtamo, H., & Raivio, T. (2001). On applied nonlinear and bilevel programming for pursuit-evasion games. *Journal of Optimization Theory and Applications*, *108*(1), 65–96.

Faber, N., & Shinar, J. (1980). Approximate solution of singularly perturbed nonlinear pursuit-evasion game. *Journal of Optimization Theory and Applications*, *32*(1), 39–73.

Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with Marquardt algorithm. *IEEE Transactions of Neural Networks*, *5*(6), 989–993.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

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

Kim, J. G., & Tahk, M. J. (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.

Lee, K. H., Choi, H. L., Tahk, M. J., & Bang, H. C. (2001). An improved neural network guidance law using a network pruning technique. *Proceedings of JSASS 15th International Sessions in 39th Aircraft Symposium* (pp. 5–8), Kifu, Japan.

Menon, P. K. A. (1989). Short-range nonlinear feedback strategies for aircraft pursuit-evasion. *Journal of Guidance, Control, and Dynamics*, *12*(1), 27–32.

Shima, T., & Shinar, J. (2002). Time-varying linear pursuit-evasion game modles with bounded controls. *Journal of Guidance, Control, and Dynamics*, *25*(3), 425–432.

Song, E. J., & Tahk, M. J. (1998). Real-time midcourse guidance with intercept point prediction. *Control Engineering Practice*, *6*(8), 957–967.

Song, E. J., & Tahk, M. J. (1999). Real-time midcourse missile guidance robust against launch conditions. *Control Engineering Practice*, *7*(4), 507–515.

Song, E. J., & Tahk, M. J. (2001). Real-time neural-network midcourse guidance. *Control Engineering Practice*, *9*(10), 1145–1154.

Song, E. J., & Tahk, M. J. (2002). Three-dimensional midcourse guidance using neural networks for interception of ballistic targets. *IEEE Transactions on Aerospace and Electronic Systems*, *38*(2), 404–414.

Tahk, M. J., Ryu, H., & Kim, J. G. (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, USA.

Tahk, M. J., & Sun, B. C. (2000). Coevolutionary augmented Lagrangian methods for constrained optimization. *IEEE Transactions on Evolutionary Computation*, *4*(2), 114–124.