Journal of Marine Science and Technology

Volume 25 | Issue 1 Article 1

NOVEL INTELLIGENT NEURAL GUIDANCE LAW BY USING MULTI-OPTIMIZATION ALGORITHMS

Jium-Ming Lin
Dept. of Communication Engineering, Chung-Hua University, Hsin-Chu, Taiwan, R.O.C., jmlin@chu.edu.tw

Cheng-Hung Lin
Ph.D. Program in Engineering Science College of Engineering, Chung-Hua University, Hsin-Chu, Taiwan, R.O.C.

Journal of Marine Science and Technology, Vol. 25, No. 1, pp. 1-14 (2017)
DOI: 10.6119/JMST-016-0216-1

NOVEL INTELLIGENT NEURAL GUIDANCE LAW BY USING MULTI-OPTIMIZATION ALGORITHMS

Jium-Ming Lin[1] and Cheng-Hung Lin[2]

Key words: neural guidance, proportional navigation, radome slope error, turning rate time constant, miss distance.

ABSTRACT

There are parameter variation effects that would reduce the performance of missile terminal guidance system, e.g., target maneuverability, missile autopilot time constant, turning rate time constant as well as radome slope error effects. To solve this problem this research proposed a novel neural-fuzzy missile terminal guidance law by applying three different neural network optimization algorithms alternatively in each step, such as the Gradient Descent (GD), SCG (Scaled Conjugate Gradient), and Levenberg-Marquardt (LM) methods. Moreover, the missile turning rate time constant, autopilot time delay, target maneuverability, glint and fading noises, radome slope error, missile initial heading error as well as acceleration limits were taken into consideration. On the other hand, performance comparisons with the proportional navigation (PN) method for not only the lower and higher altitudes but the lateral and head-on interceptions were also made. One can see that the miss distances, acceleration commands and engagement times by using the proposed guidance law are lower than the other methods for the encountered engagement conditions.

I. INTRODUCTION

In general, the terminal guidance laws of tactical missiles are derived on some classical and optimal control techniques (Nesline and Zarchan, 1983). However, they are suffered from parameter variations of the system, such as target maneuverability, missile autopilot time constant, turning rate time constant and radome slope error effect (as defined in Fig. 1). Although some neural and/or fuzzy methods had been proposed for the design, but they didn't consider the missile turning rate time constant and radome slope error (Nesline and Zarchan, 1979; Pastrick et al., 1981; Hull et al., 1985). From the missile aerodynamic requirement, the radome shape is very sharp to reduce the drag, and then the target line-of-sight (LOS) refraction error would be increased as in Fig. 1, so do the miss distances for larger missile turning rate time constants.

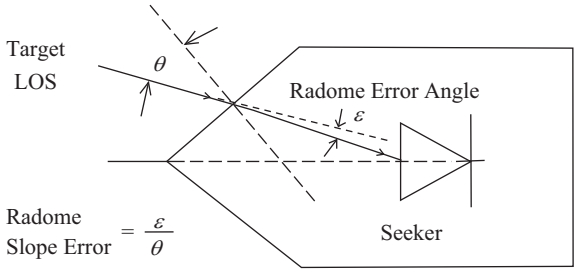

Fig. 1. Definitions of radome slope error.

Fig. 2 is the block diagram of a guidance loop, in which the missile autopilot time lag, turning rate time delay, initial heading error, and radome slope error (R) are considered. By the way in general the seeker tracking and stabilization loop gain are 10 and 100, respectively. To reduce the interception miss distance, it is important to design a suitable guidance law for the guidance system (Lin and Chau, 1995). In general, the guidance system performances can be adjusted by using proportional navigation (PN) method. In this paper the PN method for two cases (lower and higher altitudes) with two scenarios (lateral and head-on interceptions) are made in Section 2. The Scenario A is a lateral interception of missile and target, to save the simulation time only the noise effect such as the target lateral maneuver is considered. The Scenario B is a head-on interception of missile and target, since the guidance system can be approximated by a linear system, thus one can apply the adjoint technique as shown in Fig. 3 to reduce the simulation time (Perry, 1978; Rajagopalan and Bucco, 2010), to make miss distance sensitivity analyses with those effects such as target maneuver, glint, and fading noises. Note the PN guidance law cannot meet the miss distance requirement for all the cases and scenarios. So a PD-type fuzzy controller (Lin and Mon, 1999; Lin et al., 2004; Yang et al., 2005) is applied in Section 3 for the guidance law design. It was found that the fuzzy guidance laws cannot meet the miss distance requirement for some cases. Thus a novel neural guidance law is proposed in Section 4, the key point to reduce the missile performance is to apply several neural opti-

Paper submitted 03/03/15; revised 12/09/15; accepted 02/15/16. Author for correspondence: Jium-Ming Lin (e-mail: jmlin@chu.edu.tw).
[1] Dept. of Communication Engineering, Chung-Hua University, Hsin-Chu, Taiwan, R.O.C.
[2] Ph.D. Program in Engineering Science College of Engineering, Chung-Hua University, Hsin-Chu, Taiwan, R.O.C.

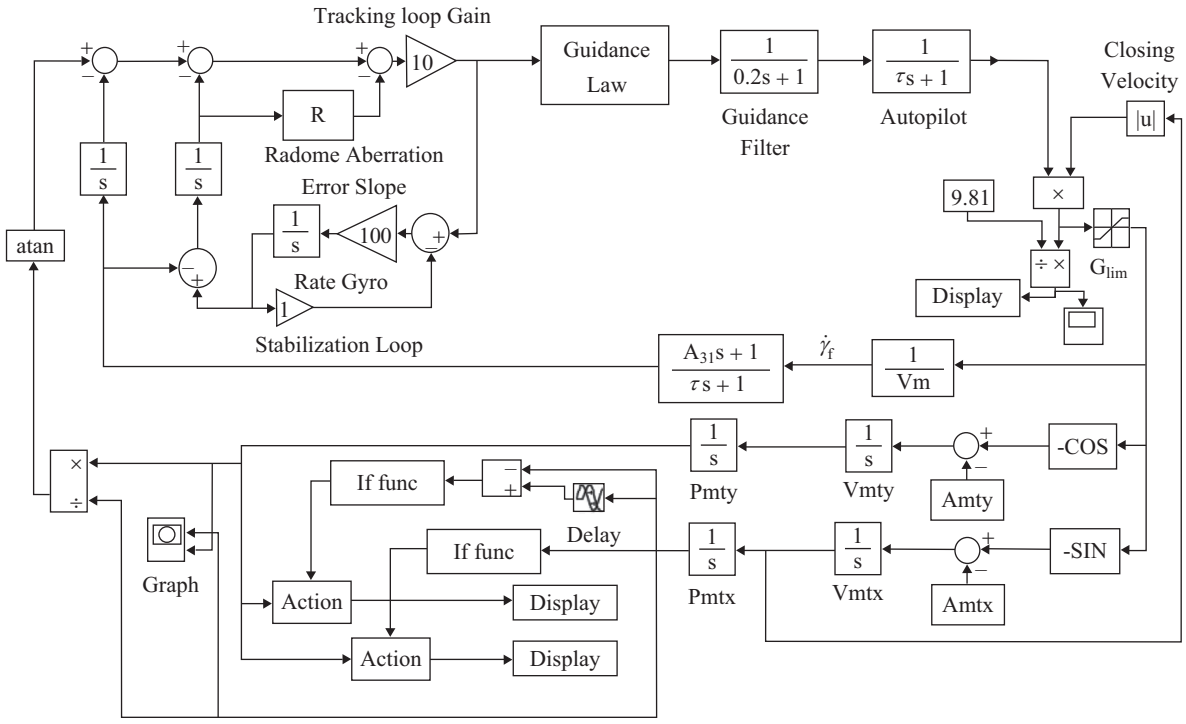

Fig. 2. Block diagram of the guidance loop for simulation.

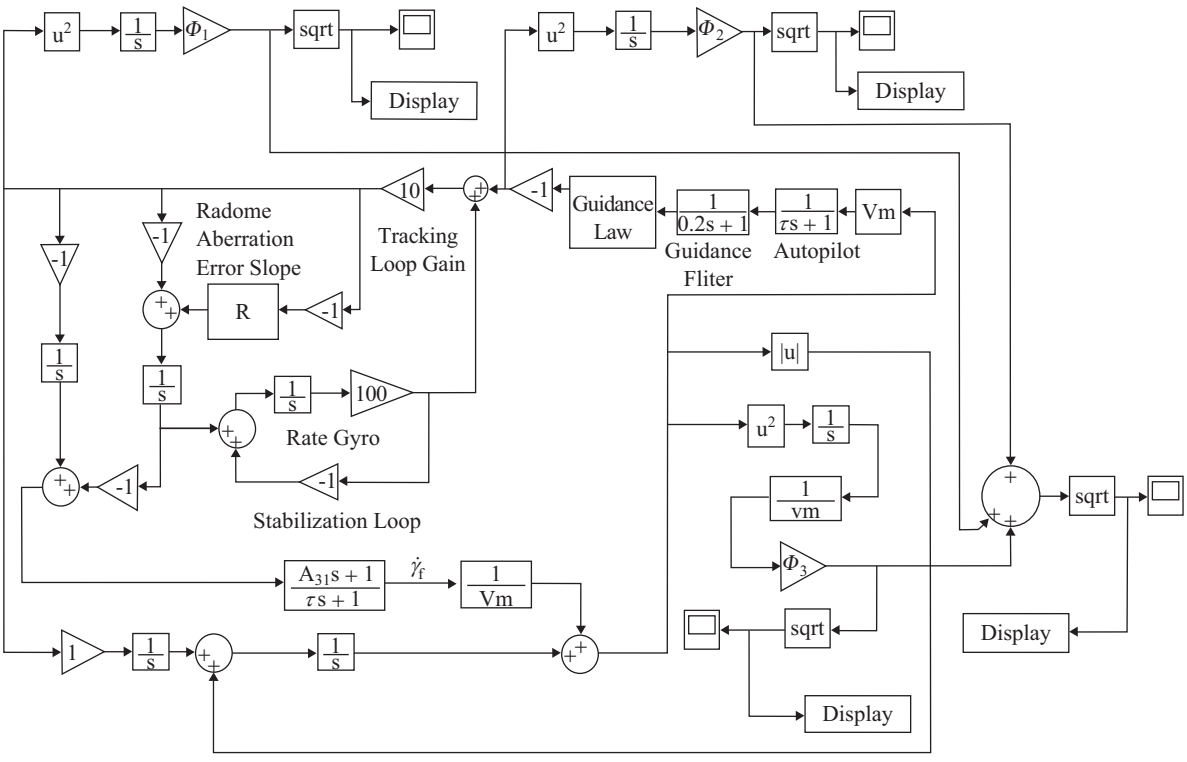

Fig. 3. Block diagram of the guidance loop for adjoint simulation.

mization algorithms alternatively in each step (such as Gradient Descent (GD) (Zaheeruddin and Garima, 2006), Levenberg-Marquardt (LM) (Hestenes and Stiefel, 1952; 1987; Fletcher, Glunt et al., 1993; Gonsalves and Caglyan, 1995; Esfahanipour and Aghamiri, 2010) and Scale Conjugate Gradient (SCG) (Shanno, 1978; Luengo et al., 1996; Chen et al., 2002; Rajagopalan and

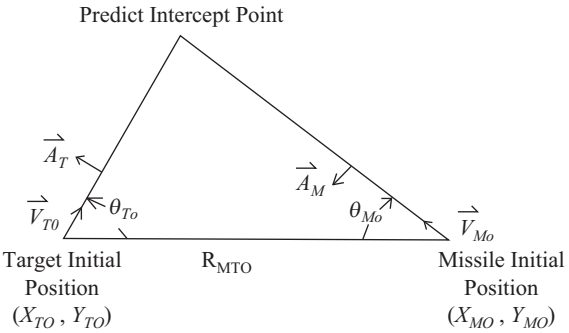

Fig. 4. Scenario A: Trajectories of lateral interception.

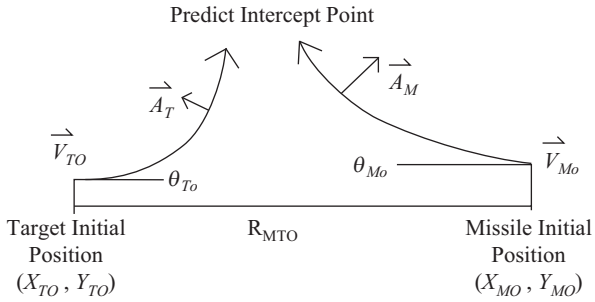

Fig. 5. Scenario B: Trajectories of head-on interception.

Bucco, 2010) methods to deal with the guidance parameter variations aforementioned. Performance comparisons with the PN and adjoint simulation method are also made in Sections 4 and 5. Note that the miss distances, acceleration commands and engagement times obtained by the proposed neural guidance law are lower and can meet the requirements for the encountered engagement conditions. Finally, a conclusion is given.

III. PROBLEM FORMULATION AND PERFORMANCE ANALYSES OF PROPORTIONAL NAVIGATION LAW

In general, the two-degree-of-dimension (2D) terminal engagement geometry of target and missile can be as shown in Fig. 4. The target (missile) initial coordinate and velocity are respectively (X_{To}, Y_{To}), and V_{To}, ((X_{Mo}, Y_{Mo}) and V_{Mo}). Let the target (missile) initial heading angle relative to the initial Line-of-Sight (LOS) and maneuver acceleration be respectively as θ_{To} (θ_{Mo}) and A_{To} (A_{Mo}), respectively. The velocity of missile and target are 600 and 400 m/sec, respectively. The initial range of missile to target is 10 km. As shown in Fig. 5 the target maneuver is assumed to be a unit-step lateral acceleration with amplitude A_T turning in a direction toward the missile to reduce the period of engagement. The miss distance requirement is less than 20 m.

This paper considers two cases of target maneuverability A_T, autopilot time constant τ, missile turning rate time constant (A_{31}), and missile acceleration limit (G_{lim}) as follows:

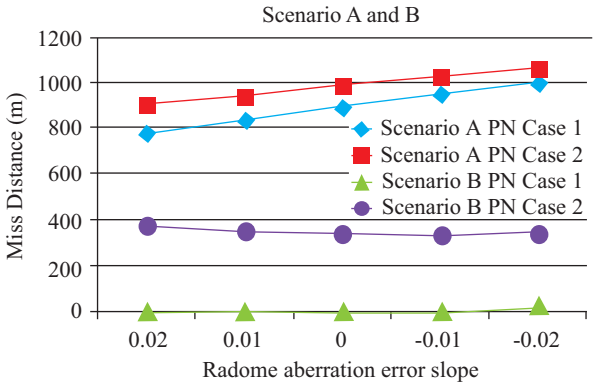

Fig. 6. Miss distances by using PN guidance law for all the engagement conditions.

(1) Case 1: $A_T = 6$ G, $\tau = 0.2$ sec, $A_{31} = 0.2$ sec, $G_{lim} = 24$ G.
(2) Case 2: $A_T = 4$ G, $\tau = 0.5$ sec, $A_{31} = 0.5$ sec, $G_{lim} = 15$ G.

Besides, two scenarios are also includes for comparison:

1. Scenario A: Lateral Interception

The lateral interception scenario is as shown in Fig. 4. The initial aspect angles (relative to the initial LOS) of target and missile are respectively as θ_{To} (30°) and θ_{Mo} (135°).

2. Scenario B: Head-On Interception

The head-on interception scenario is as shown in Fig. 5. The initial aspect angles of target (θ_{To}) and missile (θ_{Mo}) are respectively 0° and 180°, and with 10 m offset distance in the y-axis perpendicular to the line-of-sight.

Fig. 6 shows the miss distances vs. R by using the traditional PN guidance law for Cases1 and 2 of Scenarios A and B. Note that the traditional PN guidance law cannot meet the miss distance requirement for all the conditions. Moreover, the acceleration commands at sometimes are much larger than the acceleration limits as shown in Fig. 7 of Scenario B for Case 2.

III. PD-TYPE FUZZY CONTROLLER DESIGN

This section applies an intelligent PD-type fuzzy controller for the guidance system design (Gonsalves and Caglayan, 1995; Lin and Mon, 1999; Akbari amd Menhaj, 2001; Chen et al., 2002; Lin et al., 2004). The fuzzy logic is modeled by human linguistic thinking/reasoning and rule-based structure, it is able to deal with inexact information in system modeling, identification, and control (Shanno, 1978; Shi and Mizumoto, 2000a, 2000b). One of the major advantages of fuzzy logic is that it does not require mathematical model. Its performance, however, strongly depends on the selection of the membership functions and fuzzy rules, which conventionally are determined by experts knowledge or experiences. But for those systems with practical complexity and/or uncertainty, it is often quite difficult to determine the adequate fuzzy logic structure, membership functions, and logic rules.

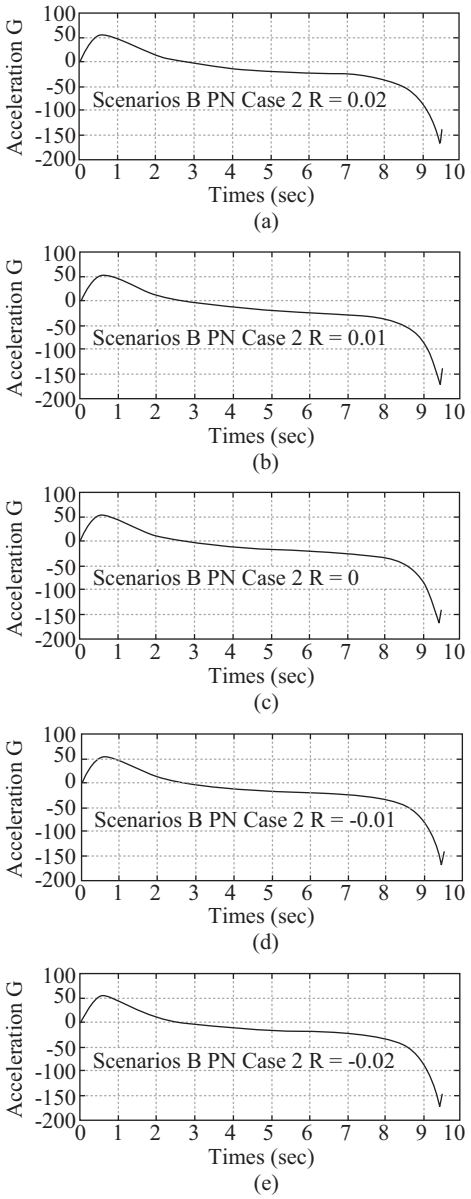

Fig. 7. Acceleration commands of Scenario B for Case 2 by PN controller.

Some algorithms had been proposed to optimize the membership functions and fuzzy logic rules of neuro-fuzzy model (Zaheeruddin and Garima, 2006; Esfahanipour and Aghamiri, 2010). A Takagi-Sugeno-Kang (TSK) type of fuzzy rule and an adaptive neuro-fuzzy model for vehicle control was applied for stock market analysis (Chen, 2011, 2013; Chen and Wang, 2011). The cross reference rules and the membership functions for the seeker tracking error E, error rate ΔE (deviations of present E and the previous E), and U (control input) are respectively defined and listed in Tables 1-4 (Holder and Sylvester, 1990; Jinho and Jungsoon, 1995; Zaheeruddin and Garima, 2006) and Fig. 8, the outputs of PD-type fuzzy controllers are defined in Table 4, where NB, NM, NS, ZE, PS, PM, and PB respectively stand for negative big, negative middle, negative small, zero, positive small, positive middle, and positive big.

The guidance performances are obtained by simulation. By trial-and-error the proportion and derivative gains are respectively set to 10 and 0.115 to speed up the response of guidance.

Table 1. Cross reference rules of E, ΔE and U in tabular form.

E, ΔE	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NM	NM	NS	NS	ZE
NM	NB	NM	NM	NS	NS	ZE	PS
NS	NM	NM	NS	NS	ZE	PS	PS
ZE	NM	NS	NS	ZE	PS	PS	PM
PS	NS	NS	ZE	PS	PS	PM	PM
PM	NS	ZE	PS	PS	PM	PM	PB
PB	ZE	PS	PS	PM	PM	PB	PB

Table 2. Membership functions of E in tabular form.

Item	Type	Parameter
Negative Big (NB)	Trapmf	[-1 -1 -0.75 -0.3]
Negative Medium (NM)	Trimf	[-0.75 -0.3 -0.15]
Negative Small (NS)	Trimf	[-0.15 -0.1 0]
Zero (ZE)	Trimf	[-0.05 0 0.05]
Positive Big(PB)	Trimf	[0 0.1 0.15]
Positive Medium (PM)	Trimf	[0.15 0.3 0.75]
Positive Small(PS)	Trapmf	[0.3 0.75 1 1]

Table 3. Membership functions of ΔE in tabular form.

Item	Type	Parameter
Negative Big (NB)	Trapmf	[-4.5 -4.5 -3.375 -1.35]
Negative Medium (NM)	Trimf	[-3.375 -1.35 -0.72]
Negative Small (NS)	Trimf	[-1 -0.5 0]
Zero (ZE)	Trimf	[-0.25 0 0.25]
Positive Big(PB)	Trimf	[0 0.5 1]
Positive Medium (PM)	Trimf	[0.72 1.35 3.375]
Positive Small(PS)	Trapmf	[1.35 3.375 4.5 4.5]

Table 4. Membership functions of U in tabular form.

Item	Type	Parameter
Negative Big (NB)	Trapmf	[-12 -12 -9.6 -8.4]
Negative Medium (NM)	Trimf	[-9.6 -8.4 -7.2]
Negative Small (NS)	Trimf	[-8.4 -4.8 0]
Zero (ZE)	Trimf	[-4.8 0 4.8]
Positive Big(PB)	Trimf	[0 4.8 8.4]
Positive Medium (PM)	Trimf	[7.2 8.4 9.6]
Positive Small(PS)	Trapmf	[8.4 9.6 12 12]

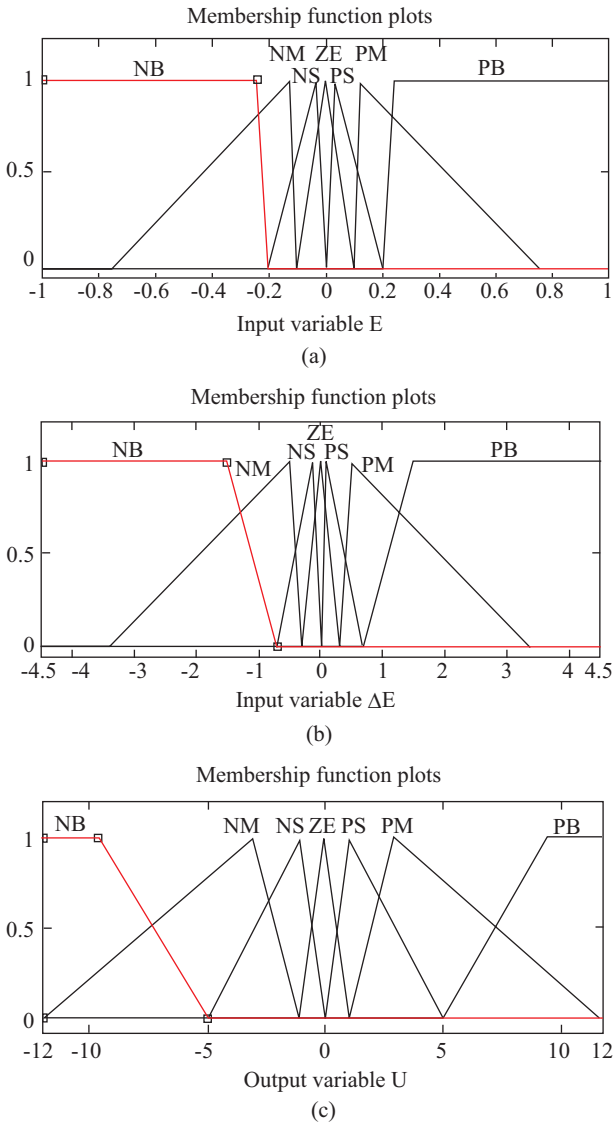

Fig. 8. Membership functions of (a) fuzzy controller error (E), (b) error rate (ΔE), and (c) output U.

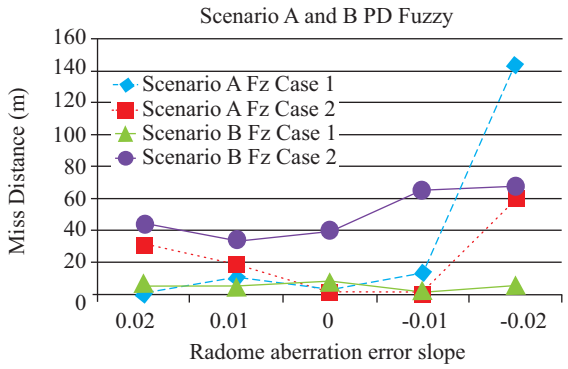

Fig. 9. Miss distances by using PD-type fuzzy controller for all the engagement conditions.

Fig. 9 shows the performances, (miss distances vs. radome aberration error slope) by using PD-fuzzy guidance law. Note that some conditions obtained by the PD-fuzzy guidance law cannot meet the miss distance requirement.

IV. NEURAL CONTROLLER DESIGN

Artificial Neural Network (ANN) is a method in computing where computers try to mimic biological neural networks in terms of how they learn and adapt. The neuron is the building block of the human brain, and how they work is exactly what ANNs aim at simulation. Like the network of neurons in the human brain, ANNs have the ability to learn through experience, and also make basic decisions. This learning process can be accomplished through training. A comprehensive introduction to ANNs can be found in the references (Shi and Mizumoto, 2000; Cheng, 2008). The NN is organized in a feed-forward manner, each neuron in a given layer is connected to all other neurons in the next layer. No connections are allowed in the backward direction. This type of ANN is made up of a series of layers including input, optional hidden and output. The number of input neurons depends on the task at hand. The number of hidden layers is determined by the complexity of the task that is to be carried out by the ANN. The values of the outputs are dependent on the factors such as the inputs, weights, thresholds and the error values. The number of output neurons is determined by what it is intended to be used. In the feed forward process, an activation function is used to scale the output of the NN within the desired ranges. Moreover, a five-layer neuro-fuzzy system was also developed by using Mamdani model for vibration control (Zaheeruddin and Garima, 2006). Most of the above studies were based on Mamdani or Sugeno fuzzy logic model with a number of IF-THEN rules and antecedent/consequent linguistic terms, which may require many computer times for real time operations (Gerdhenson, 2003; Cheng, 2008).

In practice, using only one hidden layer and a reasonable number of neurons would be able to estimate the problem at hand. For continuous functions, two hidden layers may be sufficient, but they may slow down the training process, it also has the potential to worsen the problem of local minima (Shi and Mizumoto, 2000a). However, using fewer hidden neurons than required will result in under-fitting (Shi and Mizumoto, 2000b); this is a bad situation where it becomes difficult to accurately detect the signals in a complicated data set. On the other hand, using more hidden neurons than required will result in over-fitting and prolong training time. So researchers have proposed a compromise (Cheng, 2008); the number of hidden neurons can be set as according to either two-thirds for the sum of the sizes for the input the output layers, or less than twice the size of the input layer.

In this paper a neuro-fuzzy system with a three-layer feed forward NN Mamdani model is applied as shown in Fig. 10 to determine the fuzzy logic rules and optimize the membership functions of the guidance system as shown in Fig. 1. Layers 1 and 3 define the input and output nodes, respectively. Layer 2

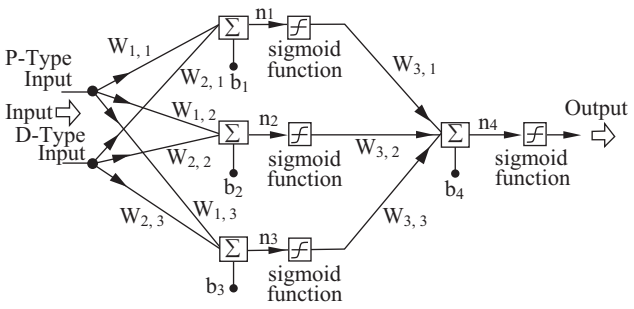

Fig. 10. Structures of neural controller.

defines the nodes to represent the fuzzy rules. Each input to the NN represents a variable that has influence on the output of the network in one way or the other.

The activation function A_j of the artificial neurons in ANNs implementing the j^{th} back propagation algorithm is a weighted sum as follows (Gershenson, 2003):

$$A_j\left(\bar{x},\bar{\omega}\right)=\sum_{i=0}^{n}x_i\omega_{ji} \tag{1}$$

If the output functions would be the identity equations i.e., (output = activation), then the neuron would be called linear. But these have severe limitations. The most common output function is the sigmoid function:

$$O_j\left(\bar{x},\bar{\omega}\right)=\frac{1}{1+e^{A_j\left(\bar{x},\bar{\omega}\right)}} \tag{2}$$

The sigmoid function is very close to one (zero) for large positive (negative) numbers, and 0.5 at zero. This allows a smooth transition between the outputs of the neurons (close to zero or close to one). One can see that anyone of the output is limited by the activation function, which in turn depends on the values of the inputs and their respective weights.

Now, the goal of the training process is to obtain a desired output when certain type of inputs are given. Since the error for the j^{th} channel is the difference between the actual O_j and the desired output d_j defined as follows:

$$E_j\left(\bar{x},\bar{\omega},d\right)=O_j\left(\bar{x},\bar{\omega}\right)-d_j \tag{3}$$

It is necessary to take the square of the difference between the output and the desired target to make it always positive, so the error for the whole network can be defined as the sum of the errors for all the neurons in the output layer:

$$E\left(\bar{x},\bar{\omega},d\right)=\sum_j\left(O_j\left(\bar{x},\bar{\omega}\right)-d_j\right)^2 \tag{4}$$

Thus the error depends on the weights, and one should adjust the weights to minimize the error. The back propagation algorithm now calculates how the error depends on the output, inputs, and weights, and then one can adjust the weights using the method of gradient descendent as follows:

$$\Delta\omega_{ji}=-\eta\frac{\delta E}{\delta\omega_{ji}} \tag{5}$$

The meaning of this formula can be interpreted in the following way: the adjustment of each weight ($\Delta\omega_{ij}$) is equal to the negative of a constant (η) multiplied by the previous weight on the error of the network, which is the derivative of E in respect to ω_{ji}. The value of the adjustment depends on η, and on the contribution of the weight to the error of the function. This is, if the weight contributes a lot to the error, the adjustment will be greater than if it contributes in a smaller amount. Eq. (5) is used iteratively until one finds appropriate weights (the error is minimal).

So, the next is to find the derivative of E in respect to ω_{ji}. This is the goal of the back propagation algorithm. First, one must calculate how much the error depends on the output, which is the derivative of E in respect to O_j by Eqs. (3) and (4):

$$\frac{\delta E}{\delta O_j}=2\left(O_j-d_j\right) \tag{6}$$

And then, how much the output depends on the activation, which in turn depends on the weights by Eqs. (1) and (2):

$$\frac{\delta O_j}{\delta\omega_{ji}}=\frac{\delta O_j}{\delta A_j}\frac{\delta A_j}{\delta\omega_{ji}}=O_j(1-O_j)x_i \tag{7}$$

And from Eqs. (6) and (7) one has:

$$\frac{\delta E}{\delta\omega_{ji}}=\frac{\delta E}{\delta O_j}\frac{\delta O_j}{\delta\omega_{ji}}=2\left(O_j-d_j\right)O_j\left(1-O_j\right)x_i \tag{8}$$

Finally, by Eqs. (5) and (8) the adjustment to each weight is obtained as:

$$\Delta\omega_{ji}=-2\eta\left(O_j-d_j\right)O_j\left(1-O_j\right)x_i \tag{9}$$

Therefore, one can use Eq. (9) to train an ANN.

Thus the neural-fuzzy (PD) controller is applied for the guidance system design with three inputs and one hidden layer as show in Fig. 9. Moreover, the optimization algorithms such as GD, LM, and SCG are applied to train the neural network parameters, such as weighting factors and biases. These methods are briefed as follows:

1. Gradient Descent (GD) Algorithm

In this method all data samples are processed at each step of

iteration to determine the steepest descent vector h. If the data samples are linearly separable, then the global minimum point can be reached. If not, it will never converge to the optimum point. This method has another weakness of computational complexity; it requires all data samples to process at each step of iteration. It starts from an initial point, and moving in the direction defined by αh vector, the relatively gain of the objective function should follow (Song, 2013):

$$\begin{aligned}\lim_{\alpha\to 0} &= \frac{F(x)-F(x+\alpha h)}{\alpha \| h \|} \\ &= -\frac{1}{\| h \|} h^T F'(x) = \| F'(x) \| \cos\theta\end{aligned} \tag{10}$$

where (α) is the step size of iterative interval and θ is the angle of vector. $F'(x)$ (means gain) would be maximum if $\theta = \pi$, then the descent vector is:

$$h_{sd} = -F'(x) \tag{11}$$

Furthermore, if α is adjusted according to the exact line search method, then the objective function can be converged toward the minimum value. Since the tuning direction is along the negative direction of the function slope, that the convergent rate would be very slow near the local minimum. On the other hand, it couldn't avoid the saddle point problem.

2. Newton Method

The main point of Newton method is to reduce the iteration times by using the characteristic property near the optimal point, i.e., if x^* is the optimal point, then $F'(x^*) = 0$, so the descending vector can be obtained by using Taylor series expansion:

$$\begin{aligned}F'(x+h) &= F'(x) + F''(x)h + 0\left(\| h \|^2\right) \\ &\cong F'(x) + F''(x)h\end{aligned} \tag{12}$$

For Eq. (12) to be true, the descending vector of Newton method (or Hh_n) must be very small, and then one has:

$$Hh_n = -F'(x),\ if\ H = F''(x),\ x = x + h_n \tag{13}$$

So that one can find the value of x for the next iteration. In the final stage of iteration, Newton method is a better choice.

3. Gauss-Newton Method

Gauss-Newton Method is generally the basis to derived other more effective algorithms, it is based on the first order Taylor series expansion of $f(x)$, i.e., to solve the problem by using the linear model of $f(x)$, the convergent rate of this method would be in quadratic manner for some special cases. The detail steps are as follows. If $||h||$ is very small, then one can make Taylor series expansion of $f(x + h)$:

$$f(x+h) \cong \ell(h) \equiv f(x) + J(x)h \tag{14}$$

Let:

$$\begin{aligned}F(x) &= \frac{1}{2}\sum_{i=1}^{m}\left(f_i(x)\right)^2 \\ &= \frac{1}{2}\| f(x) \|^2 = \frac{1}{2} f(x)^T f(x)^T\end{aligned} \tag{15}$$

And then:

$$\begin{aligned}F(x+h) &\cong L(h) \equiv \frac{1}{2}\ell(h)^T \ell(h) \\ &= \frac{1}{2} f^T f + h^T J^T f + \frac{1}{2} h^T J^T J h \\ &= F(x) + h^T J^T f + \frac{1}{2} h^T J^T J h\end{aligned} \tag{16}$$

The purpose of this algorithm is to find the best descending vector of Gauss-Newton method h_{gn} to make $L(h)$ be minimum, i.e.,

$$h_{gn} = \arg\min_h \left\{ L(h) \right\} \tag{17}$$

By Eqs. (16) and (17) one has:

$$L'(h) = J^T f + J^T J h \tag{18}$$

Let $L'(h) = 0$ in Eq. (18), one can get the descending vector h_{gn} of Gauss-Newton method by the following equations:

$$\left(J^T J\right) h_{gn} = -J^T f \tag{19}$$

4. Levenberg-Marquardt (LM) Algorithm

The LM algorithm is an iterative technique that locates the minimum of a multivariate function expressed as the sum of squares of non-linear functions. It is a standard technique for nonlinear least-squares problems. LM can be thought of as a combination of steepest descent and Gauss-Newton methods. When the current solution is far away from the correct one, the algorithm behaves like a steepest descent method. This method is slow, but guaranteed to converge.

Levenberg and Marquardt suggested to use the damped Gauss-Newton method for solving nonlinear least squares problems, that's to change Eq. (19) to the following equation:

$$\left(J^T J + \mu I\right) h_{lm} = -g,\ g = J^T f,\ \text{and}\ \mu \geq 0 \tag{20}$$

There are some improvements by introducing the damping parameter μI in Eq. (20), the details are as follows:

(1) Since the value of damping parameter (μ) at the optimal point has the property $\mu > 0$, so the coefficient matrix in Eq. (20) is positive definite, and one can assure the existence for the descending vector of Levenberg-Marquardt (LM) algorithm defined by h_{lm}.
(2) If the value of damping parameter μ is very large, so J^TJ can be neglected, one has the descending vector (h_{lm}) as follows:

$$h_{lm} \cong -\frac{1}{\mu} g = -\frac{1}{\mu} F'(x) \tag{21}$$

Then the function would be decreasing following the descending vector by a small step h_{lm}, so it is better and suitable at the initial iteration period.

(3) If the value of damping parameter μ is very small, then $h_{lm} \fallingdotseq h_{gn}$, so this case is in the final iteration period as x approaching x^*, and this result is very reasonable. Moreover, if $F'(x^*)$ is equal to zero or its value is very small, so the decreasing speed is in quadratic manner.

Therefore, the value of damping parameter μ determines the descending step size and direction, it also absolve to find the α ratio by using the original Newton's line search method. The initial value of μ (μ_0) can use the maximum value of the diagonal elements for the matrix $A_0 = J(X_0)^T J(X_0)$, it is defined by:

$$\mu_0 = \tau \cdot \max\left\{a_{ii}^{(0)}\right\} \tag{22}$$

To set the value of τ is as follows, if the initial value x_0 would be approach to the final solution x^*, then τ can be set as 10^{-6}. On the other hand, the suggest value is either 1 or 10^{-3}. The method to update μ is by computing the gain ratio ρ in the iteration process defined by Eq. (16), the gain ratio is defined as:

$$\rho = \frac{F(x) - F(x + h_{lm})}{L(0) - L(h_{lm})} \tag{23}$$

The denominator in Eq. (23) would be derived by the linear model as follows:

$$\begin{aligned} L(0) - L(h_{lm}) &= -h_{lm}^T J^T f - \frac{1}{2} h_{lm}^T J^T J h_{lm} \\ &= -\frac{1}{2} h_{lm}^T \left(2g + \left(J^TJ + \mu I - \mu I\right) h_{lm}\right) \\ &= \frac{1}{2} h_{lm}^T \left(\mu h_{lm} - g\right) \end{aligned} \tag{24}$$

Since $h^T_{lm}\, h_{lm}$ and $-h^T_{lmg}$ in Eq. (23) are positive definite, so $L(0) - L(h_{lm})$ is also positive definite.

The meaning of ρ defined by Eqs. (23) and (24) is discussed as follows. If ρ is a larger value, then $L(h_{lm})$ is a good approximation value of $F(x + h_{lm})$, so one can make μ be a lower value for approaching h_{lm} to h_{gn} in both step size and direction. But if ρ is smaller or a negative value, i.e., $L(h_{lm})$ is not a good approximation, so one would increase μ to make h_{lm} approach the solution of steepest descent method. The stop criteria depend on some choices as follows:

(1) If the step is approaching the local minimum, the gradient value should be zero, that means $F'(x^*) = g(x^*) = 0$, so one can set:

$$\|g\|_\infty \le \varepsilon_1 \tag{25}$$

In Eq. (25) ε_1 is a very small positive number.
(2) Considering the condition for approaching the local minimum point, x would be taken small changes, so one can define:

$$\|x_{\text{new}} - x\| \le \varepsilon_2 \left(\|x\| + \varepsilon_2\right) \tag{26}$$

(3) Finally, a safety stop condition to avoid the infinite loops problem is:

$$k \ge k_{\max} \tag{27}$$

5. Conjugate Gradient (CG) Method

There is observed that the conjugate gradient iteration is one of the most important methods in scientific computation for rapidly solving large linear systems of equations with symmetric positive definite coefficient matrices (Hestenes and Stiefel, 1952; Song, 2013). Since the convergence of an iterative method depends on the eigenvalues of the coefficient matrix, it is often better to use a preconditioner that transforms the system to one with a better distribution of eigenvalues. Therefore, preconditioning is the key to a successful iterative solver (Gershenson, 2003).

In general, the CG algorithm to solve the system $Ax = b$, starts with an initial guess of the solution x_0, with an initial residual r_0, and with an initial search direction that is equal to the initial residual: $p_0 = r_0$ (Caraba, 2008). The idea behind the conjugate gradient method is that the residual $r_k = b - Ax_k$ is orthogonal to the Krylov subspace generated by b, and therefore each residual is perpendicular to all the previous residuals. The residual is computed at each step. The solution at the next step is found using a search direction that is only a linear combination of the previous search directions, which for x_1 is just a combination between the previous and the current residual.

A more visual explanation of how the CG algorithm finds the approximate solution to the exact solution is as shown in Fig. 11.

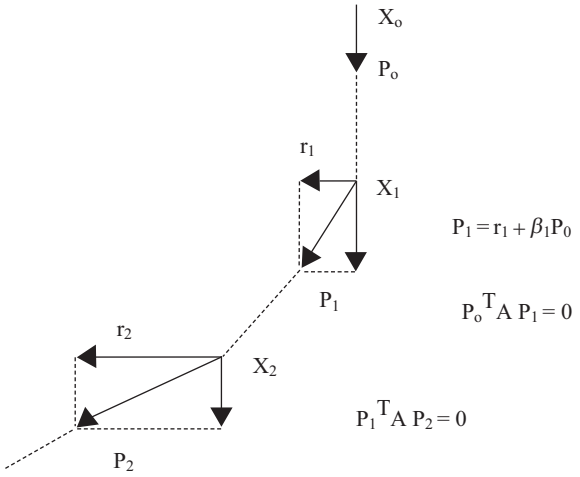

Fig. 11. Searching direction of the CG algorithm.

The iterative formulas of CG are given below:

(1) approximate solution:

$$x_k = x_{k-1} + \alpha_k p_{k-1} \tag{25}$$

(2) residual calculation:

$$r_k = r_{k-1} - \alpha_k A p_{k-1} \tag{26}$$

(3) search direction:

$$p_k = r_k + \beta_k p_{k-1} \tag{27}$$

(4) improvement at step k:

$$\beta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} \tag{28}$$

(5) step size:

$$\alpha_k = \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}} \tag{29}$$

6. SCG (Scaled Conjugate Gradient) Algorithm

The SCG algorithm is a scaled memoryless Broyden-Fletcher-Goldfarb-Shanno (BFGS) preconditioned conjugate gradient algorithm. Had mentioned that the key point is to combine the scaled BFGS method and the preconditioning technique in the frame of conjugate gradient method (Hestenes and Stiefel, 1952). This method is also extended based on the above general optimization strategy, but chooses the search direction and the step size more carefully by using information from Taylor's second order approximation (Yang et al., 2005).

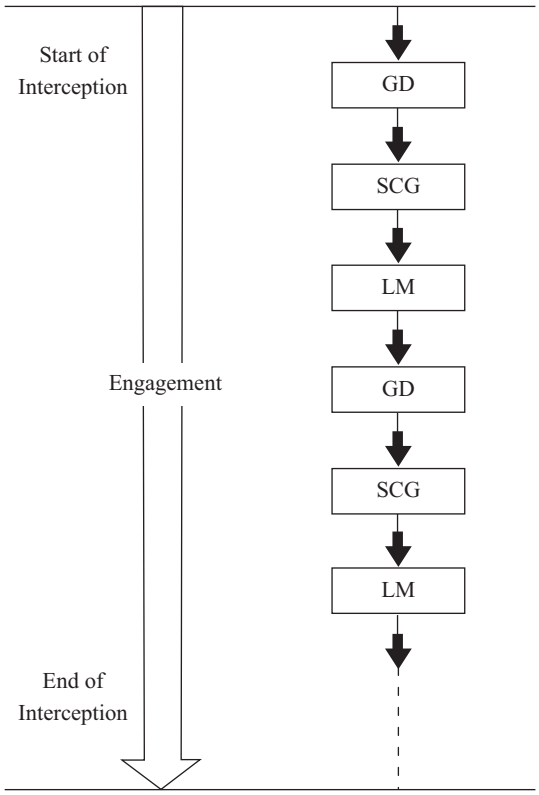

Fig. 12. The optimization methods such as GD, SCG and LM are applied alternatively in the guidance engagement.

$$E(w+y) \approx E(w) + E'(w)^T y + \frac{1}{2} y^T E''(w) y \tag{30}$$

If one denotes the quadratic approximation to E in a neighborhood of a point w by $E_{qw}(y)$, so that $E_{qw}(y)$ is given by:

$$E_{qw}(y) = E(w) + E'(w)^T y + \frac{1}{2} y^T E''(w) y \tag{31}$$

In order to determine the minima of $E_{qw}(y)$, the critical points for $E_{qw}(y)$ must be found, i.e., the points where the first derivative of Eq. (34) with respective to w must be zero (neglect the derivative of the third order term):

$$E'_{qw}(y) = E''(w) y + E'(w) = 0 \tag{32}$$

SCG can yields a speed-up than the standard back propagation algorithm (BP). SCG is fully automated, the user need not to provide the dependent parameters and avoids a time consuming line-search, the conjugate gradient back propagation (CGB) and BFGS uses this SCG algorithm in each iteration in order to determine an appropriate step size. Moreover, neural network can often reduce the overall complexity without knowing the system exact structure. Thus the smaller the complexity of the neural network, the bigger the possibility

Table 5. Weighting factors and biases of neural controller.

Input Layer to Hidden Layer Weighting Factors		
P input (W1, j)	D input (W2, j)	
-7.3266	2.3793e-07	
16.7042	2.4022e-07	
-4.3741	-1.8535e-08	
Input Layer to Hidden Layer Bias (B1, B2, B3)		
2.9383	0.0102	-2.6458
Hidden Layer to Output Layer Weighting Factors (W3, j)		
-0.3328	0.2019	-1.7852
Hidden Layer to Output Layer Bias (B4)		
-1.43		

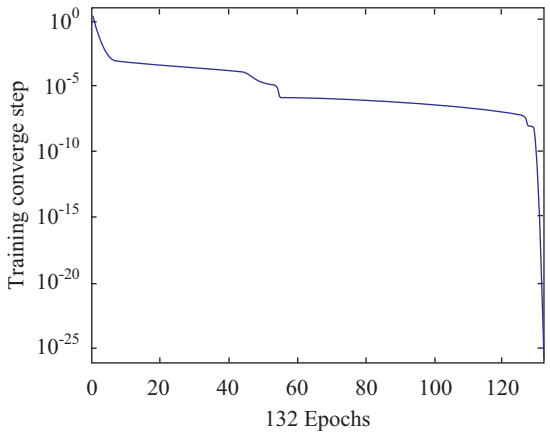

Fig. 13. The training curve of convergence for neural controller.

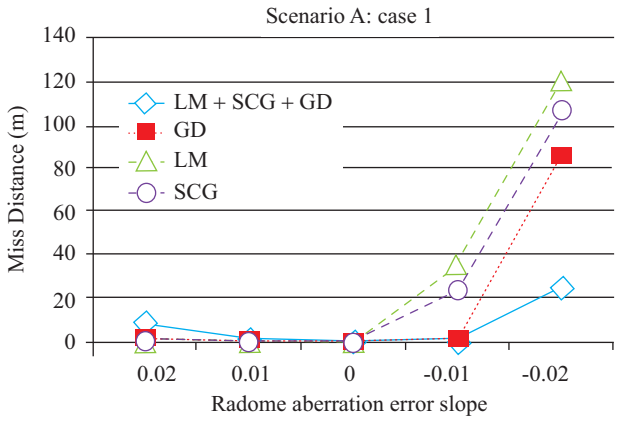

Fig. 14. Miss distances of Case 1 for Scenario A by using neural controller with single and multi-optimization optimization algorithms.

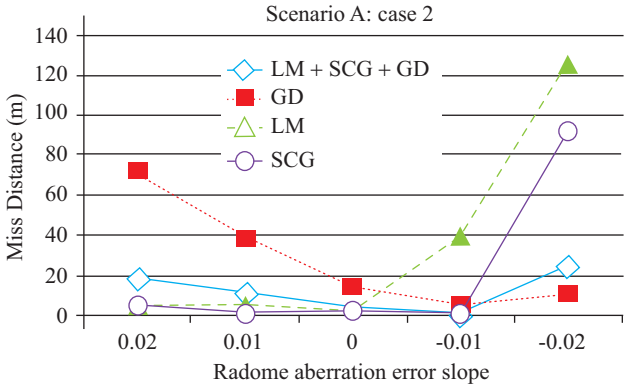

Fig. 15. Miss distances of Case 2 for Scenario A by using neural controller with single and multi-optimization algorithms.

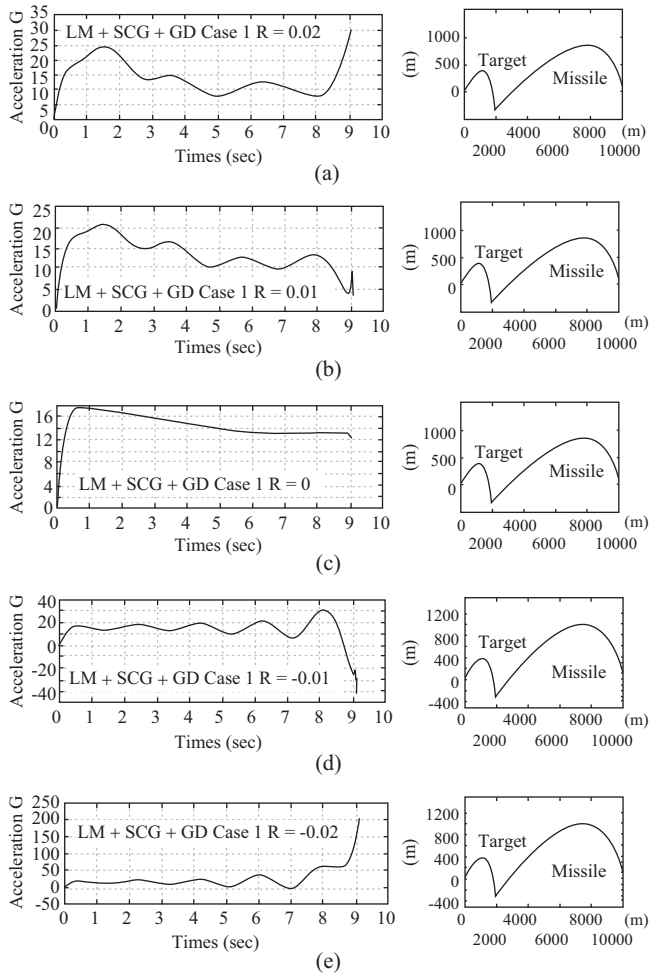

Fig. 16. Acceleration commands and trajectories of Scenario A for Case 1 by using the proposed method. (a) R = 0.02, (b) R = 0.01, (c) R = 0, (d) R = -0.01, (e) R = -0.02.

that the weight space contains long ravines with sharp curvature. While BP is inefficient in this area, it shows that SCG handles them effectively.

To speed up the guidance law design, the optimization methods such as GD, SCG and LM are applied alternatively in the guidance engagement as shown in Fig. 12 to consider those guidance effects such as target maneuver, missile autopilot time lag, turning rate time delay, initial heading error, and radome slope error. Then one can obtain the weighting factors and biases of neural controller as listed in Table 5. The training curve

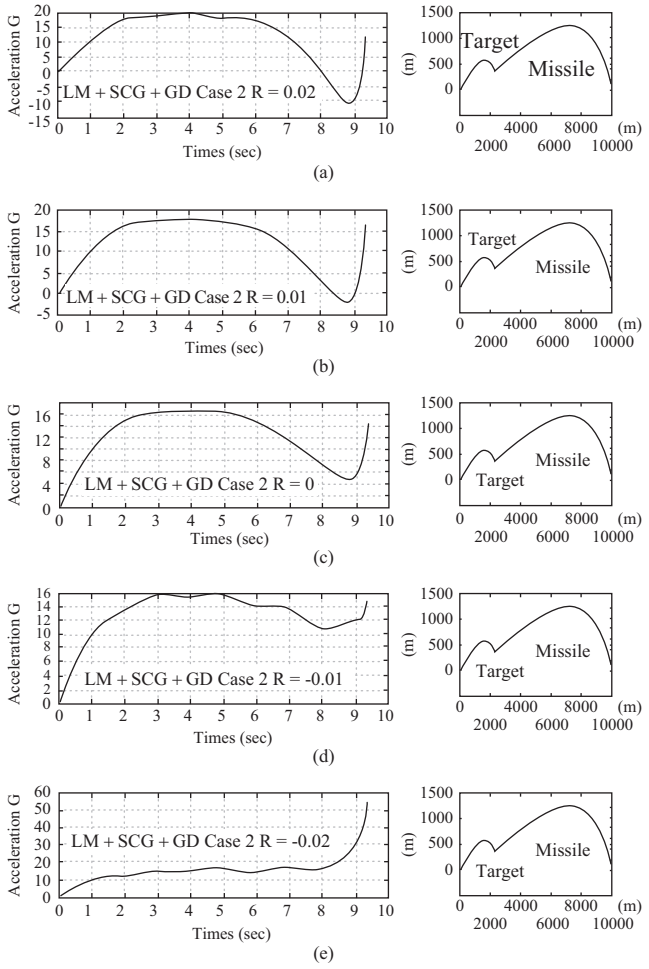

Fig. 17. Acceleration commands and trajectories of Scenario A for Case 2 by using the proposed method. (a) R = 0.02, (b) R = 0.01, (c) R = 0, (d) R = -0.01, (e) R = -0.02.

of convergence for neural controller is as shown in Fig. 13, thus the proposed method may be applied.

7. Simulation Results of Scenario A

Fig. 14 shows the miss distances of Case 1 by using single and multi-optimization algorithms. Note that only the proposed method can satisfy all the values of R. So do the result as shown in Fig. 15 for Case 2. On the other hand, the acceleration commands and trajectories for Cases 1 and 2 are also as shown in Figs. 16 and 17, respectively. Note that all the acceleration commands have some oscillations, this is due to as times go by and at the interception times, the guidance loops would become unstable (Lin et al., 1991; Zaheeruddin and Garima, 2006). However, the neural guidance laws can still hit the target by using the proposed multi-optimization algorithm for the encountered conditions. Moreover, the accelerations as shown in Figs. 16 and 17 are much lower than those obtained by using the PN guidance law as shown in Fig. 7.

8. Simulation Results of Scenario B

Figs. 18 and 19 show the miss distances by using single and

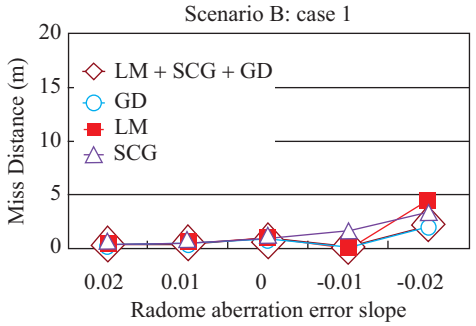

Fig. 18. Miss distances of Scenario B for Case 1 by using neural controller with single and multi-optimization algorithm.

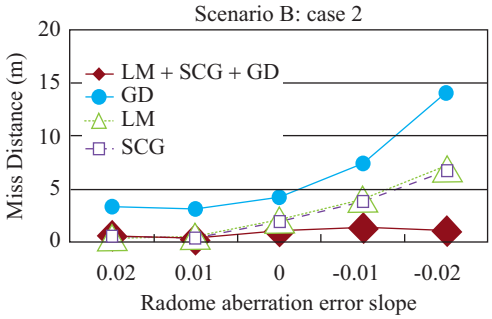

Fig. 19. Miss distances of Scenario B for Case 2 by using neural controller with single and multi-optimization algorithms.

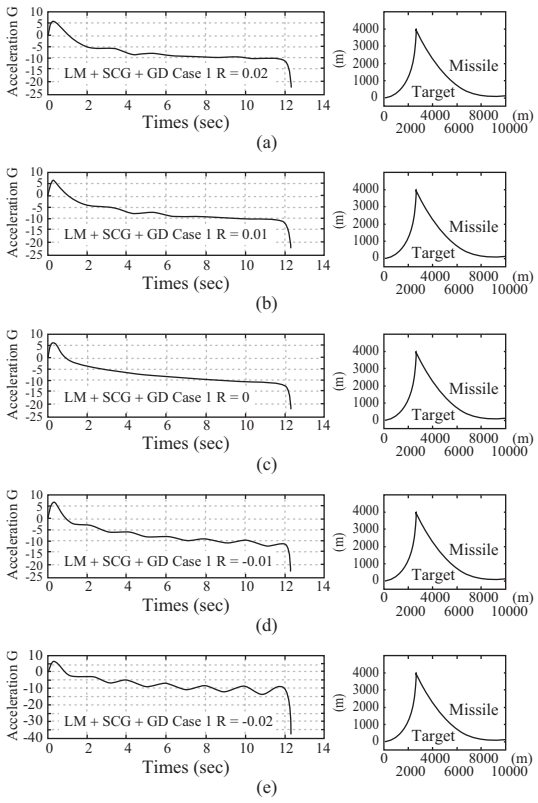

Fig. 20. Acceleration and trajectory of Scenario B for Case 1 by using the proposed method. (a) R = 0.02, (b) R = 0.01, (c) R = 0, (d) R = -0.01, (e) R = -0.02.

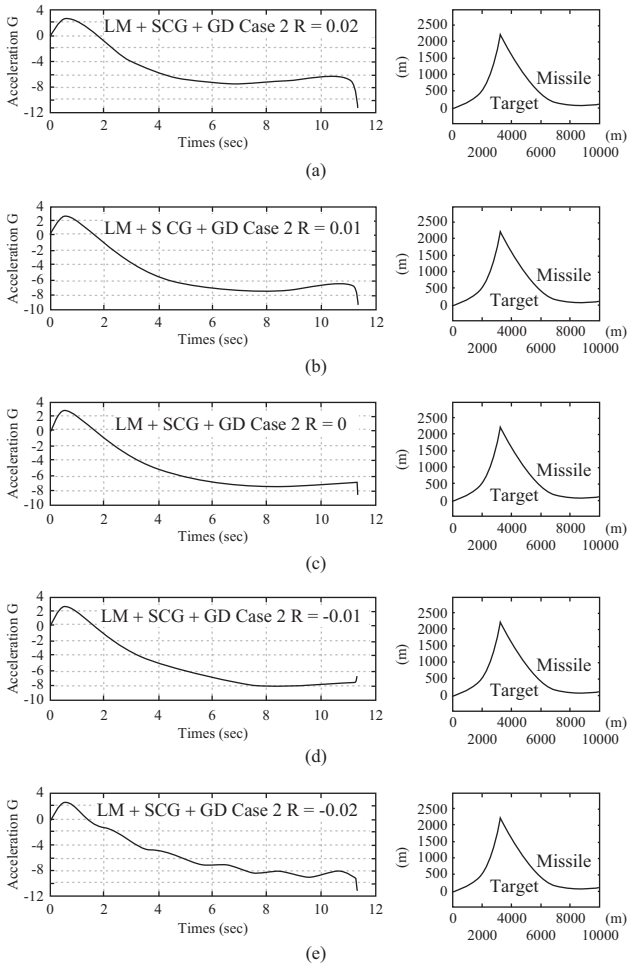

Fig. 21. Acceleration and trajectory of Scenario B for Case 2 by using the proposed method. (a) R = 0.02, (b) R = 0.01, (c) R = 0, (d) R = -0.01, (e) R = -0.02.

multi-optimization algorithms for Cases 1 and 2. One can see all the miss distances are smaller for this scenario. On the other hand, the acceleration commands and trajectories for Cases 1 and 2 are also as shown in Figs. 20 and 21, respectively. Note that the oscillation effects in the final periods of acceleration commands are lower for this condition.

9. Comparisons of Interception Times

The interception times for the guidance laws and engagement conditions are listed in Table 6 for comparison. Note that the interception times of Scenario B for Cases 1 and 2 are respectively 12.34 and 11.34 seconds, so there are no difference of interception times among the guidance laws. However, the longest interception times by using the proposed method of Scenario A for Cases 1 and 2 are respectively 9.11 and 9.37 seconds, they are always lower than the other methods. So one can conclude that the miss distances, acceleration commands, and interception times of the proposed method are lower than the other methods for the encountered engagement condition.

Table 6. Times of interceptions for all Cases.

Times (sec) / Guidance Law	Scenario A		Scenario B	
	Case 1	Case 2	Case 1	Case 2
PN	9.73-9.92	10.15-10.29	12.34	11.34
PD Fuzzy	9.08-9.21	9.34-9.42	12.34	11.34
Neural GD	9.09-9.16	9.36-9.42	12.34	11.34
Neural LM	9.08-9.18	9.35-9.46	12.34	11.34
Neural SCG	9.08-9.17	9.35-9.43	12.34	11.34
Proposed Method	9.09-9.11	9.35-9.37	12.34	11.34

Table 7. Parameters of noise spectral density.

Type of Noises	Spectral Density
Glint (Φ1)	2 (m^2/Hz)
Fading (Φ2)	1*10e-6 (Rad^2/Hz)
Target maneuverability of case 1 (Φ3)	7.2 (G^2/sec)
Target maneuverability of case 2 (Φ3)	3.2 (G^2/sec)

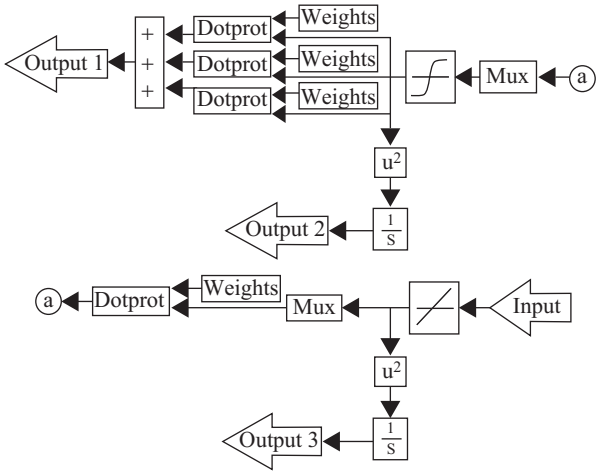

Fig. 22. Block diagram of adjoint system for neural controller.

V. ADJOINT SIMULATION RESULTS (SCENARIO B)

Since the adjoint method can be applied for only linear system, so the head-on engagement (Scenario B) is taken into consideration in this section. The advantage of the adjoint technique is that one can take the target glint and fading noises into simulation more easily, and the computation time can be reduced. The adjoint simulation block diagram of neural controller is as shown in Fig. 22. The parameters of the noise spectral densities such as target glint, fading, and maneuverability are listed in Table 7 (Rajagopalan and Bucco, 2010). The results obtained by using PN and NC guidance laws of Cases 1 and 2 are respectively as shown in Figs. 23-26.

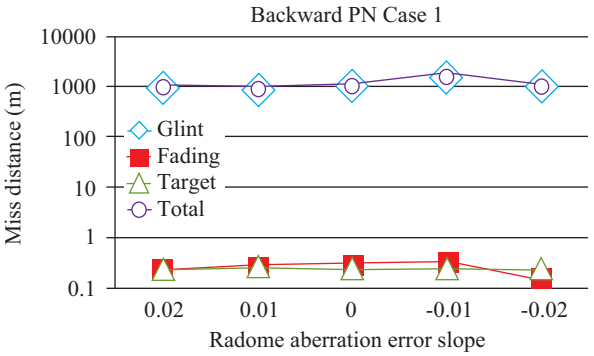

Fig. 23. Miss distance for PN of Case 1 (Scenario B).

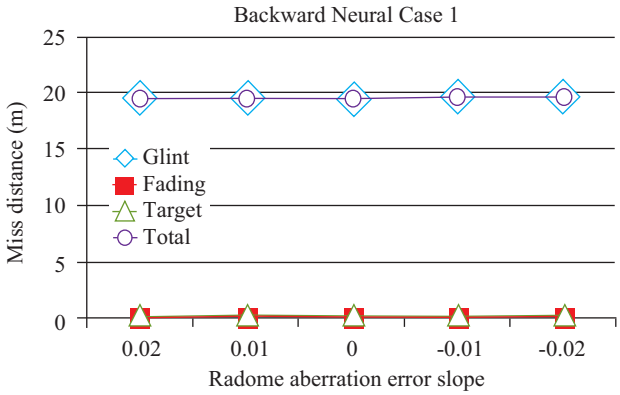

Fig. 24. Miss distance of the proposed method for Case 1 (Scenario B).

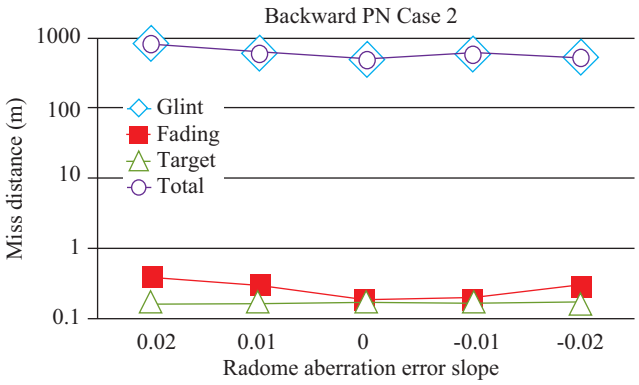

Fig. 25. Miss distance for PN of Case 2.

Note that the total miss distances obtained by using the proposed method for Cases 1 and 2 are increased respectively from 2 m and 1 m (as shown in Figs. 17 and 18) to 20 m and 11 m. This is true because in this section more noises are taken into consideration, but the proposed guidance law can still meet the requirement. On the other hand, the miss distances by using PN of Cases 1 and 2 are largely increased respectively from 30 m and 400 m (as shown in Fig. 6) to 1000 m and 800 m by adding more noises into consideration.

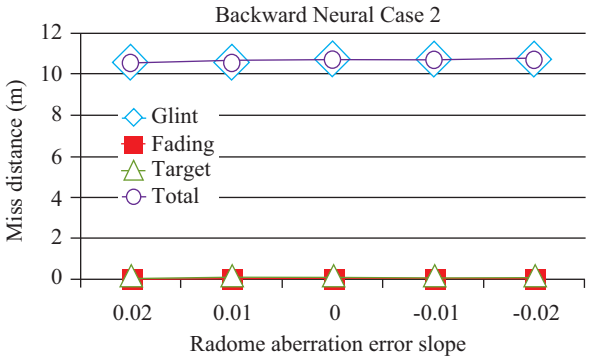

Fig. 26. Miss distance of the proposed method for Case 2.

VI. CONCLUSION

This research applies a novel intelligent neural guidance law by applying several neural network optimization algorithms (such GD, LM and SCG) alternatively in each step for terminal guidance law design of a surface-to-air missile. Not only the lower and higher altitudes but the lateral and head-on interceptions are adopted for comparison. Besides, both the missile autopilot delay, turning rate time constant, target maneuverability, glint and fading noises, radome slope error, missile initial heading error as well as acceleration limits were taken into consideration. are taken into consideration, which are scarcely considered in the previous literatures. Note that the miss distances, acceleration commands and engagement times obtained by using the proposed method are lower than the other methods for the encountered conditions.

REFERENCES

Akbari, S. and M. B. Menhaj (2001). A fuzzy guidance law for modeling offensive air-to-air combat maneuver. Joint 9th. FSA World Congress and 20th NAFIPS International Conference 5(25-28), 3027-3031.

Andrei, N. (2007). A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Applied Mathematics Letters 20, 645-650.

Andrei, N. (2007). Scaled conjugate gradient algorithms for unconstrained optimization. Computational Optimization and Applications 38(3), 401-416.

Andrei, N. (2007). Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. Optimization Methods and Software 22, 561-571.

Andrei, N. (2008). A scaled nonlinear conjugate gradient algorithm for unconstrained optimization. Optimization. A Journal of Mathematical Programming and Operations Research 57(4), 549-570.

Birgin, E. and M. Martínez (2001). A spectral conjugate gradient method for unconstrained optimization. Applied Math. and Optimization 43, 117-128.

Gershenson, C. (2003). Artificial neural networks for beginners, arXiv preprint.

Chen, B. S., Y. Y. Chen and C. L. Lin (2002). Nonlinear fuzzy H∞ guidance law with saturation of actuators against maneuvering targets. IEEE Trans. Control Syst. Technol. 10, 769-779.

Chen, C. J. (2011). Structural vibration suppression by using neural classifier with genetic algorithm. International Journal of Machine Learning and Cybernetics 3(3), 215-221.

Chen, C. J. (2013). Development of a rule selection mechanism by using neuro-fuzzy methodology for structural vibration suppression. Journal of Intelligent and Fuzzy Systems 25(4), 881-892.

Chen, T. and Y. C. Wang (2011). A hybrid fuzzy and neural approach for fore-

casting the book-to-bill ratio in the semiconductor manufacturing industry. International Journal of Advanced Manufacturing Technology 52(1-4), 377-389.

Cheng, K. H. (2008). Hybrid learning-based neuro-fuzzy inference system: a new approach for system modeling. International Journal Systems Science 39(6), 583-600.

Cottrell, R. G. (1971). Optimal intercept guidance for short-range tactical missiles. AIAA Journal 9(7), 1414-1415.

Dennis, Jr., J. E. and R. B. Schnabel (1983). Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Englewood Cliffs, NJ: Prentice-Hall.

Caraba, E. (2008). Preconditioned Conjugate Gradient Algorithm, Doctoral dissertation, Louisiana State University.

Esfahanipour, A. and W. Aghamiri (2010). Adapted neuro-fuzzy inference system on indirect approach TSK fuzzy rule base for stock market analysis. Expert Systems with Applications 37, 4742-4748.

Fletcher, R. (1987). Practical Methods of Optimization, 2nd ed., Chichester: Wiley.

Glunt, W., T. L. Hayden and M. Raydan (1993). Molecular conformations from distance matrices. Journal of Computational Chemistry 14, 114-120.

Gonsalves, P. G. and A. K. Caglayan (1995). Fuzzy logic PID controller for missile terminal guidance. Proceedings of the 1995 IEEE International Symposium on Intelligent Control, 377-382.

Hagan, M. T. and M. B. Menhaj (1994). Training feed forward networks with the Marquardt algorithm. IEEE Trans. Neural Networks 6, 861-867.

Hestenes, M. R. and E. Stiefel (1952). Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards 49(6), 409-436.

Holder, E. J. and V. B. Sylvester (1990). An analysis of modern versus classical homing guidance. IEEE Trans. Aerospace and Electronic Systems 26(4), 599-606.

Hull, D. G., J. L. Speyer and C. Y. Teng (1985). Maximum-information guidance for homing missiles. J. Guidance Contr., Dyn. 8(4), 494-497.

Jinho, K. and J. Jungsoon (1995). Nonlinear model inversion control for bank-to-turn missile. AIAA Guidance, Navigation and Control Conference, Baltimore, MD.

Lin, C. F. (1991). Modern Navigation, Guidance, and Control Processing. Englewood Cliffs, NJ: Prentice-Hall.

Lin, C. L., H. Z. Hung, Y. Y. Chen and B. S. Chen (2004). Development of an integrated fuzzy-logic-based missile guidance law against high speed target. IEEE Trans. on Fuzzy systems 12(2), 157-169.

Lin, C. M. and Y. J. Mon (1999). Fuzzy-logic-based guidance law design for missile systems. Proc. IEEE Control Applications Conf., 421-426.

Lin, J. M. and Y. F. Chau (1995). Radome Slope Compensation Using Multi-Model Kalman Filters. AIAA Journal of Guidance, Control, and Dynamics 18(3), 637-640.

Luengo, F., M. Raydan, W. Glunt and T. L. Hayden (1996). Preconditioned spectral gradient method for unconstrained optimization problems. Technical Report 96-08, Escuela de Computaci´on, Facultad de Ciencias: Universidad Central de Venezuela, 47002 Caracas 1041-A.

Mishra, S. K., I. G. Sarma and K. N. Swamy (2006). Performance evaluation of two fuzzy-logic-based homing guidance schemes. Journal of Guidance, Control, and Dynamics 17(6), 1389-1391.

Mitchell, T. M. (1997), Machine Learning, McGraw-Hill.

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. Neural networks 6(4), 525-533.

Nesline, F. W. and P. Zarchan (1979). A new look at classical versus modern homing missile guidance. Guidance and Control Conference, Boulder, Colo.

Nesline, F. W. and P. Zarchan (1983). Missile guidance design tradeoffs for high-altitude air defense. Journal of Guidance, Control, and Dynamics 6, 207-212.

Nocedal, J. (1992). Theory of algorithms for unconstrained optimization. Acta Numerica, 199-242.

Pastrick, H. L., S. M. Seltzer and M. E. Warren (1981). Guidance laws for short-range tactical missiles. J. Guidance and Control 4(2), 98-108.

Perry, A. (1978). A modified conjugate gradient algorithm. Operations Research 26, 1073-1078.

Rajagopalan, A. and D. Bucco (2010). Applications of adjoint theory to problems in aerospace and defence science. ANZIAM J 51, C697-C714.

Raydan, M. (1997). The Barzilain and Borwein gradient method for the large unconstrained minimization problem. SIAM Journal on Optimization 7, 26-33.

Shanno, D. F. (1978). Conjugate gradient methods with inexact searches. Mathematics of Operations Research 3, 244-256.

Shi, Y. and M. Mizumoto (2000a). Some considerations on conventional neuro-fuzzy learning algorithms by gradient descent method. Fuzzy Sets and Systems 112, 51-63.

Shi, Y. and M. Mizumoto (2000b). A new approach of neuro-fuzzy learning algorithm for tuning fuzzy rules. Fuzzy Sets and Systems 112, 99-116.

Shi, Y. and M. Mizumoto (2001). An improvement of neuro-fuzzy learning algorithm for tuning fuzzy rules. Fuzzy Sets and Systems 118(2), 339-350.

Song, H. (2013). Preconditioning techniques analysis for CG method, ECS 231 Large-Scale Scientific Computation Course, College of Engineering, University of California, Davis.

Yang, S. M., Y. J. Tung and Y. C. Liu (2005). A neuro-fuzzy system design methodology for vibration control. Asian Journal of Control 7(4), 393-1400.

Zaheeruddin and Garima (2006). A neuro-fuzzy approach for prediction of human work efficiency in noisy environment. Journal of Applied Soft Computing 6(3), 283-294.

Zarchan, P. (1994). Tactical and Strategic Aeronautic Missile Guidance, 2nd ed., Washington, DC: AIAA, Inc.