

An Optimal Fuzzy Logic Guidance Law using Particle Swarm Optimization

Labeed Hassan

Department of Aerospace Engineering
Maleke-Ashtar University
Islamic Republic of Iran

Seyed Hossein Sadati

Space Research Institute
Islamic Republic of Iran

Jalal Karimi

Department of Aerospace Engineering
Sharif University
Islamic Republic of Iran

ABSTRACT

An Optimal fuzzy logic guidance (OFLG) law for a surface to air homing missile is introduced. The introduced approach is based on the well-known proportional navigation guidance (PNG) law. Particle Swarm Optimization (PSO) is used to optimize the of the membership functions' (MFs) parameters of the proposed design. The distribution of the MFs is obtained by minimizing a nonlinear constrained multi-objective optimization problem where; control effort and miss distance are treated as competing objectives. The performance of the introduced guidance law is compared with classical fuzzy logic guidance (FLG) law as well as PNG one. The simulation results show that OFLG performs better than other guidance laws. Moreover, the introduced design is shown to perform well with the existence of noisy measurements.

Keywords:

Fuzzy Logic, Particle Swarm Optimization, Proportional Navigation Guidance.

1. INTRODUCTION

Guidance is considered the key to the successful performance of any missile. The conventional PNG law, which considered the backbone of guidance laws, becomes more difficult to accomplish the interception task because of the improvement of aircraft maneuverability. Moreover, this operation is hampered by the presence of large number of uncertainties ranging from atmospheric turbulence to variations in radar parameters. The presence of several other factors like sensor noise, inaccurate representation of missile flight control, system dynamics, components driven to saturation etc, forms the problem [1]. It is difficult to formulate what the true missile model is and what the behavior of the missile change. It is therefore obvious to design the robust controller which solves the problem.

In recent years fuzzy logic controllers (FLCs) have been used in missile guidance wherever the system under consideration is not well-defined, uncertain or model free. There are, however, a limited number of published works available that discuss the subject of fuzzy guidance for aerial vehicles in a systematic manner [2, 3, 4, 5].

FLCs are developed to utilize human expert knowledge in controlling various systems, in particular those ill-defined and nonlinear systems [6]. It is well known that while fuzzy rules are relatively easy to derive from human experts, the fuzzy MFs are difficult to obtain. Tuning of MFs is a time consuming and often frustrating exercise. To overcome these difficulties various techniques have been reported to automate the tuning process of MFs. An adaptive network based fuzzy

inference system (ANFIS) was introduced [7], where an adaptive neural network was used to learn the mapping between the inputs and outputs and a Sugeno-type of fuzzy system could be generated based on the neural network. A quantum neural network was also used to learn the data space of a Tagaki-Sugeno FLC [8]. Genetic algorithm has been used in the automatic design of FLCs [9, 10] in the areas of mobile robotics.

In the current work, we introduce a fuzzy guidance law with rules obtained based upon the conception of the PNG law. Then, the MFs' parameters of the introduced design will be optimized using PSO technique. In fact, PSO is a population based stochastic optimization technique developed by Eberhart and Kennedy [11, 12]. Based on their description, particle swarm optimization imitates human (or insects) social behavior. Individuals interact with one another while learning from their own experience, and gradually the population members move into better regions of the problem space. The swarm of PSO can be envisioned as multiple birds (particles) that search for the best food source (optimum) by using their inertia, their knowledge, and the knowledge of the swarm. Single particles behave similarly because traditionally they share the same configuration. While searching for food, the birds are either scattered or go together before they locate the place where they can find the food. While the birds are searching for food from one place to another, there is always a bird that can smell the food very well, that is, the bird is perceptible of the place where the food can be found, having the better food resource information, Figure 1.

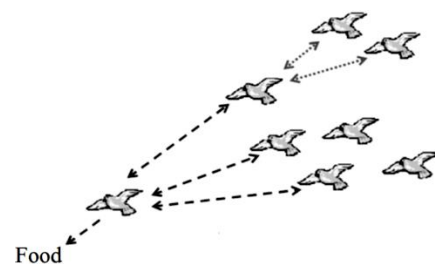


Fig. 1: The flock swarms using PSO.

Because they are transmitting the information, especially the good information at any time while searching the food from one place to another, conducted by the good information, the birds will eventually flock to the place where food can be found.

Due to its simplicity in implementation, PSO has gained popularities in engineering applications, such as in image processing [13] and in system modeling [14]. A number of publications have also been reported in using PSO to automatically tune the FLC parameters [15, 16, 17, 18]. These publications are focused on tuning the parameters involved in the TS-type fuzzy controllers. In general, the PSO is used to perform the learning tasks that are usually associated with the NN in the TS FLCs. A PSO based fuzzy MFs optimizing method is also introduced to a fixed point control problem, i.e. parking a car into a predefined garage location [19, 20].

Although there are research results in the area of automatic fuzzy MFs optimizing, most of them are in the area of TS type of fuzzy controllers. To the best of our knowledge, there is no report on using PSO for the Mamdani-type of fuzzy controller MFs tuning.

In this paper, we use PSO to automatically tune MFs of the proposed guidance law. The OFLG then is compared with PNG law and FLG law which have been introduced by Rajasekhar and Sreenatha [21].

The paper is organized as follows: in Section 2 we proceed with a brief overview of the PNG law, whereas; OFLG is explained in Section 3. We provide some case-studies in Section 4 and the differences of all guidance laws are discussed. The noise affecting on the proposed design is discussed in chapter 5 while the conclusions are provided in section 6.

2. PNG GUIDANCE LAW

Theoretically; PNG as used in many missiles gives the commanded acceleration perpendicular to the instantaneous line-of-sight (LOS), the magnitude being proportional to the LOS angle rate ($\dot{\lambda}$) and the closing velocity (V_c) as:

$$A_c = NV_c \dot{\lambda} \quad (1)$$

Figure 2 shows two-dimensions, point mass missile-target engagement geometry.

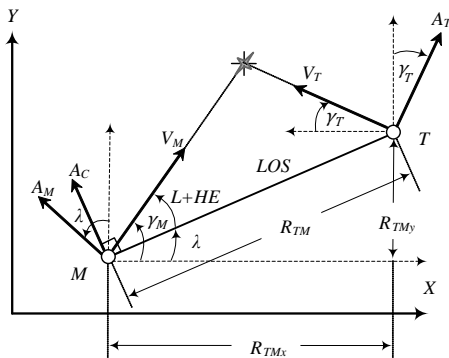


Fig. 2: Missile-target engagement geometry.

Where, λ , L and HE are LOS angle, lead angle and head angle respectively. Whilst, $\gamma_M, \gamma_T, V_M, V_T, A_M$ and A_T are flight path angles, velocities, accelerations of the missile and the target respectively. In addition, R_{TMx} and R_{TMy} are horizontal and vertical relative distances and R_{TM} is the relative target-

missile distance. The gravitational forces and drag effects have been neglected for simplicity.

A missile that employs PNG is not fired at the exact direction of the target but is fired in a direction of expected interception point. The missile will hit target if both of missile and target continue to fly along a straight-line path at constant velocities.

Anyway, PNG does not always perform well because the measurement of V_c and $\dot{\lambda}$ are easily and frequently contaminated with noises. In addition N is usually set as constant value which leads to low performance of PNG against maneuvering targets.

3. OFLG USING PSO

Each FLC has the structure as shown in the following:

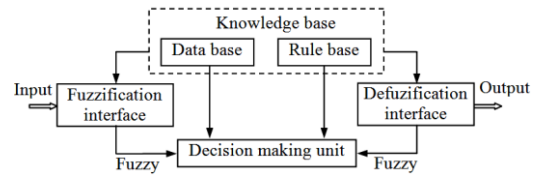


Fig. 3: The structure of FLC.

The important components of a FLC are the Fuzzifier, the Inference engine, the Fuzzy Knowledge base, and the Defuzzifier. According to Figure 3, the Fuzzifier converts the crisp input to a linguistic variable using the MFs stored in the fuzzy knowledge base. Using *If-Then* type fuzzy rules the Inference engine converts the fuzzy input to a fuzzy output. The Defuzzifier converts the fuzzy output of the inference engine to crisp one using MFs analogous to the ones used by the Fuzzifier. In our design the centre of area (CoA) method, which supplies defuzzified output with better continuity is used for defuzzification. In general, CoA method with the output u^* is calculated as:

$$u^* = \frac{\int u m_O(u) du}{\int m_O(u) du} \quad (2)$$

Where; u is the output variable, O is the output fuzzy set and m_O is the MFs of the output fuzzy set.

Minimum Mamdani (AND method), the most popular inference engine, is chosen to obtain the best possible conclusion. This type of inference engine allows easy and effective computation and it is appropriate for the real time control application [22].

The starting point of a fuzzy controller design is to choose the number and the shape of the MFs for input and output variables.

Our fuzzy controller is similar to PNG controller. It has two inputs (V_c and $\dot{\lambda}$) and single output (A_c). So that, three groups of MFs are chosen for the three corresponding variables $V_c, \dot{\lambda}$ and A_c . Each group has 7 triangular MFs as shown in Figure 4. It has been found that using complex forms of MFs cannot

bring any advantage over the triangular ones where this kind of MFs gives faster response [23].

In fact, appropriate number and shapes of membership functions are usually result of different compromises among contradicting factors; such as accuracy, hardware and computation complexities. The ones, we are mostly concerned in this work, are relative accuracy, computation time, and complexity.

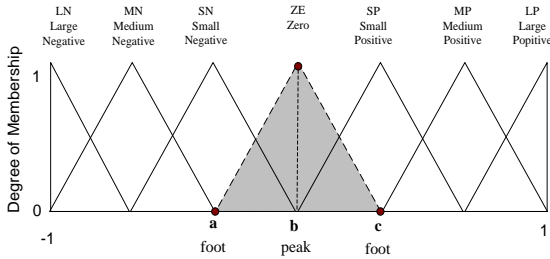


Fig. 4: A typical set of MFs.

Before defining the rules we normalize the input and output data of the controller, the normalization procedures are required to transform performance ratings with different data measurement units into a decision matrix with compatible unit. So that, the variables V_C , λ and A_C are quantized and normalized within $[-1, 1]$, according to the following Equation:

$$X_{norm} = \frac{x}{x_{max}} \quad (3)$$

Where, X is the variable has to be normalized, X_{max} its maximum value. The maximum values of the controller's variables can be obtained from the engineering experience about the missile besides helping of other classical guidance laws in PNG's class.

The maximum values those used in our design are shown in the following Table:

Table 2. Maximum values used in normalization.

V_{Cmax}	λ_{max}	A_{Cmax}
1300 [m/sec]	0.05 [rad/sec]	200 [m/sec ²]

Each normalized variable then is replaced by a set of linguistic values as shown in Table 3.

Table 3: Term sets adopted.

V_C	\equiv	LN MN SN ZE SP MP LP
λ	\equiv	LN MN SN ZE SP MP LP
A_C	\equiv	LN MN SN ZE SP MP LP

Where; the linguistic values $\{LN, MN... LP\}$ are abbreviations of $\{Large\ Negative, Medium\ Negative \dots Large\ Positive\}$ respectively as shown in Figure 3. Now, we can obtain the rules based upon PNG's conception.

3.1 Defining of the Rules

In the classical PNG law, the acceleration command A_C is proportional to multiplication of the two variables of λ and V_C . So that, it is trivial that the sign of A_C will be negative (N) if only one of (λ or V_C)'s signs were negative and positive (P) in the other cases, therefore, the sign of A_C could simply define as following:

Table 4: Defining the sign of A_C

If V_C Is N	And λ is P	Then A_C Is N
If V_C Is N	And λ is N	Then A_C Is P
If V_C Is P	And λ is P	Then A_C Is P
If V_C Is P	And λ is N	Then A_C Is N

As it is explained erstwhile the data have to be normalized before feeding to the controller input, this means they have to be in the interval $[-1, 1]$, and it is known that multiplication of any two variables in this interval would be smaller than the smallest one between them, also it is closer to the smaller value than the bigger one. In addition the output would be zero if any of the inputs were zero. Adopting this concept the linguistic value A_C could be defined as described in Table 5:

Table 5: Defining the values of A_C

If V_C Is L	And λ Is L	Then A_C Is L
If V_C Is L	And λ Is M	Then A_C Is M
If V_C Is L	And λ Is S	Then A_C Is S
If V_C Is L	And λ Is ZE	Then A_C Is ZE
If V_C Is M	And λ Is L	Then A_C Is M
If V_C Is M	And λ Is M	Then A_C Is M
If V_C Is M	And λ Is S	Then A_C Is S
If V_C Is M	And λ Is ZE	Then A_C Is ZE
If V_C Is S	And λ Is L	Then A_C Is S
If V_C Is S	And λ Is M	Then A_C Is S
If V_C Is S	And λ Is S	Then A_C Is S
If V_C Is S	And λ Is ZE	Then A_C Is ZE
If V_C Is ZE	And λ Is L	Then A_C Is ZE
If \square_\square Is ZE	And $\dot{\square}$ Is M	Then \square_\square Is ZE
If \square_\square Is ZE	And $\dot{\square}$ Is S	Then \square_\square Is ZE
If \square_\square Is ZE	And $\dot{\square}$ Is ZE	Then \square_\square Is ZE

Taken the two previous conceptions into account, enables to find the entire rules of the proposed design as declared in the following:

Table 6: The entire Rules of OFLG.

		LP	MP	SP	ZE	SN	MN	LN
LP	LP	MP	SP	ZE	SN	MN	LN	
MP	MP	MP	SP	ZE	SN	MN	MN	
SP	SP	SP	SP	ZE	SN	SN	SN	
ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE	
SN	SN	SN	SN	ZE	SP	SP	SP	
MN	MN	MN	SN	ZE	SP	MP	MP	
LN	LN	MN	SN	ZE	SP	MP	LP	

The last remained step is to optimize the MFs helping of PSO.

3.2 PSO

The PSO is a population based stochastic optimization technique, consists of a swarm of particles flying through the search space. Every individual in the swarm contains parameters for position and velocity. The position of each particle represents a potential solution to the optimisation problem. The dynamic of the swarm is governed by a set of rules that modify the velocity of each particle according to the experience of the particle and that of its neighbors depending on the social network structure within the swarm. By adding a velocity to the current position, the position of each particle is modified.

As the particles move around the space, different fitness values are given to the particles at different locations according to how the current positions of particles satisfy the objective. At the iteration each particle keeps track of its personal best position. Depending on the social network structure of the swarm, the global best position, and/or the local best position, is used to influence the swarm dynamic. After a number of iterations, the particles will eventually cluster around the area where fittest solutions are.

The swarm behavior is influenced by the number of particles, the neighbourhood size, the inertia weight, the maximum velocity, and the acceleration calculation ω, ω that modifies the velocity. The larger the number of particles in the swarm, the more likely the swarm will converge on the global optimum, because the social information exchange is increased. The influence of the current velocity on the new velocity can be controlled by the inertia weight. The influence of the particle's experience and that of its neighbor is governed by the acceleration calculation. The acceleration limits the trajectory of the particle oscillation. The new velocity is limited by the given maximum velocity to prevent particles from moving too fast in the space.

In particular, the velocity associated with each particle in PSO is calculated as:

$$v_i(t+1) = \omega v_i(t) + \omega_1 \cdot r_1(t) (p_{best} - v_i(t)) + \omega_2 \cdot r_2(t) (g_{best} - v_i(t))$$

(4)

Where; ω is the momentum or inertia weight of the particle, $v_i(t)$ is the velocity of the particle i at time step t , p_{best} is the global best performing particle up to time step t in the entire population, p_{best} is the best experience particle i has had up to time step t . $v_i(t)$ is the current location of particle i , and r_1, r_2 are constants usually equal each to other, r_1, r_2 are random numbers within [0, 1] those represent random fiction [24]. To limit the searching space $v_i(t+1)$ is limited to be within a certain range of $-v_{max} \leq v_i \leq v_{max}$. The new location of particle i can be calculated as:

$$p_i(t+1) = p_i(t) + v_i(t+1)$$

(5)

The evaluation of the particle performance is based on a problem specific objective function that decides the 'closeness' of the particle to the optimal solution. With Figure 5, the optimization process is started with random initial values then the object function is calculated. The first positions are automatically the best values. Based on Equations (4 and 5) the PSO updates both velocity and position vectors. Object function is calculated again, if the new value of the object function is smaller than the old one, the corresponding position vector is replaced by the old one, else it remained. The process is re-run until a termination criterion, such as a limit on the number of iterations or satisfactory results, is reached.

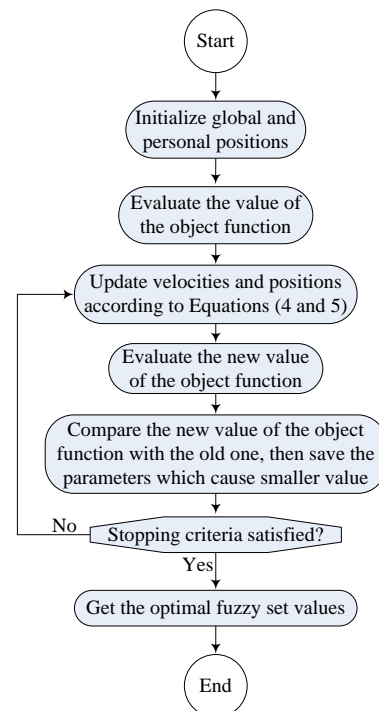


Fig. 5: Flowchart of PSO.

3.3 Optimization of the MFs

Each triangular MFs is determined by three parameters such as **a**, **b** and **c**, where **a**, **c** locate the "feet" of the triangle and **b**

locates the peak (see Figure 4). Anyway the triangular MF has the form:

$$\mu_{\text{triangular}}(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \geq c \end{cases} \quad (6)$$

The corresponding parameters have to satisfy the inequality $a < b < c$.

Since there are 14 MFs in the inputs and 7 MFs in the output, in addition, each MF has its three own parameters. In total we have 63 parameters for optimization, so that, a vector of dimension $n = 63$ particles is adopted in the optimization process. The population is set as $N = 200$ vectors, whilst the total searching iterations is set to be $iter = 1000$. With the help of [24], the inertia weight w was set to be 0.9 decreased linearly to 0.4 and the weighting factors w_1, w_2 were set to be 0.5 for both, whilst p_1, p_2 are randomized within [0, 1].

Therefore, it is possible to use PSO as a global optimization search method to find a set of such parameters that will produce the best control performance of the FLC.

Throughout the optimization process we try to minimize the control effort and the miss distance of the FLC, the object function is defined as:

$$J(\mathbf{p}) = w_1 \int_0^t \mu^2 dt + w_2 \cdot \mu(t) \quad (7)$$

Where; \mathbf{p} is a vector denotes the MFs parameters, t is the time of flight, w_1 and w_2 are constants refer to the importance of the objectives; Miss Distance (μ) and Control Effort ($\int \mu^2 dt$). In our design we considered that both μ and $\int \mu^2 dt$ have similar significance, so that; the constants here are calculated as follows:

$$w_1 = \frac{\mu}{\max(\mu)}, \quad w_2 = \frac{\int \mu^2 dt}{\max(\int \mu^2 dt)} \quad (8)$$

For the optimization process, we assumed the following data:

Table 7: Data used for simulation.

Target position	$[x_t = 10, y_t = 3]$ km.
Missile position	$[x_m = 0, y_m = 0]$ km.
Target velocity	$v_t = 300$ m/sec.
Missile velocity	$v_m = 1000$ m/sec.
Missile Dynamic saturation	± 200 m/sec ² .

The target is supposed to have a changeable maneuvering as shown in Figure 6:

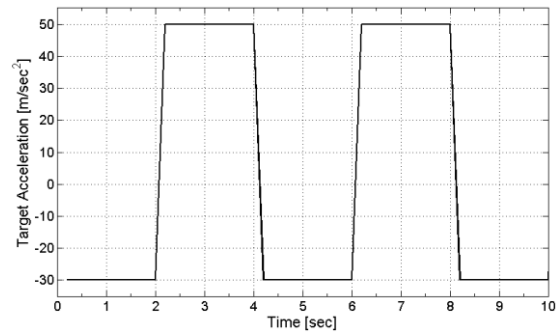


Fig. 6: The target's maneuver.

The general arrangement of the guidance loop is illustrated in Figure 7. Where, a transfer function of flight control system (FCS) and the plant is considered as:

$$\frac{A_c}{A_m} = \frac{-0.006s^2 + 11.296}{0.003s^3 + 0.139s^2 + 3.42s + 11.313} \quad (9)$$

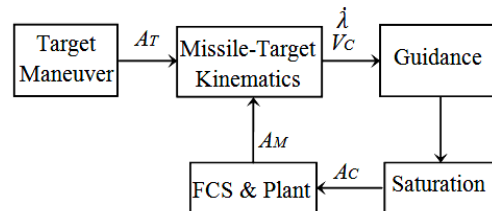


Fig. 7: Homing Guidance Loop.

One of the important factors in the simulation process is usually the integration time-step. This is normally chosen based on nature of the problem or experience. Here, we use a time step equal to 0.01 second, mainly because a typical missile-gyro gyrates around 100 cycles per second.

The resulted values of the object function during the optimization process are declared as shown in Figure 8. As it can be seen, the searching can be terminated after about 600 iterations when there is no reduction in total value of J was observed.

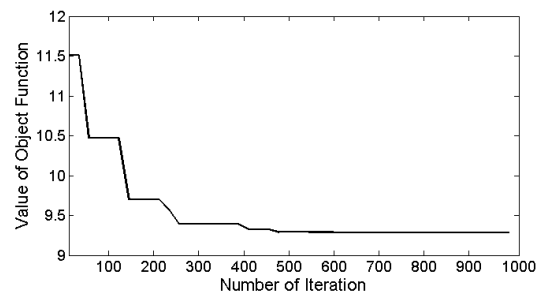


Fig. 8: The reduction of J during PSO searching process.

At the completion of the optimization, MFs of the inputs $\dot{\alpha}$, \dot{r} and the output a are extracted and plotted in Figure 9.

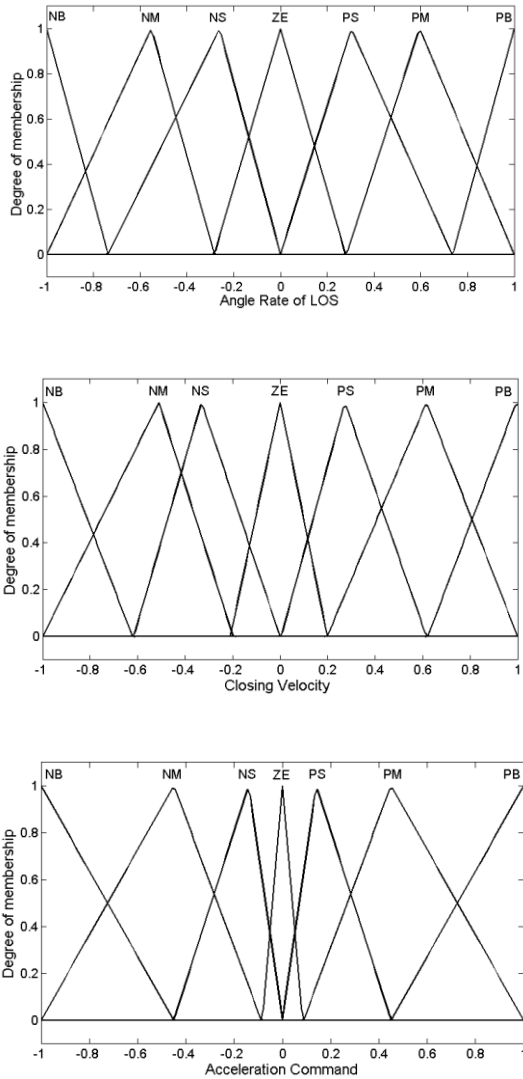


Fig. 9: Optimized MFs of $\dot{\alpha}$, \dot{r} and a for OFLG.

4. RESULT AND ANALYSIS

Engagement accuracy of PNG, FLG [21] and OFLG laws for many different scenarios of a maneuvering target is fully examined. The scenarios are: Incoming target with upward/downward maneuvering, and Escaping target with upward/downward maneuvering. The PNG's navigation ratio is chosen as $N=4$.

Figures (10, 11, 12, and 13) show the four scenarios respectively. In these Figures, the trajectories and the acceleration's histories are plotted where the target is considered to maneuver in its maximum ability (-3 or $+5$)g.

Further scenarios are also examined when the values of $\dot{r} = [-2.5, -2, -1.5, \dots, 3.5, 4, 4.5]$ m/s. Since there is no ability to show all the scenarios, a Root Mean Square value (RMS) of \dot{r} and \ddot{r} for the all scenarios are calculated and listed as shown in Table 8.

Table 8: RMS of \dot{r} and \ddot{r} for the all scenarios.

RMS	\dot{r} [m]	\ddot{r} [m ² /sec ³] $\times 10^{-4}$
PNG	7.58	5.36
FLG	7.21	5.14
OFLG	6.43	4.62

From Table 8 we can note that both FLG and OFLG laws outweigh the PNG one in terms of \dot{r} and \ddot{r} . On the other hand; OFLG gives the best performance. OFLG shows noticeable improvements in both terms where it insures 15%, 9% improvement in term of \dot{r} and 14%, 10% increment in term of \ddot{r} when compared with PNG and FLG respectively.

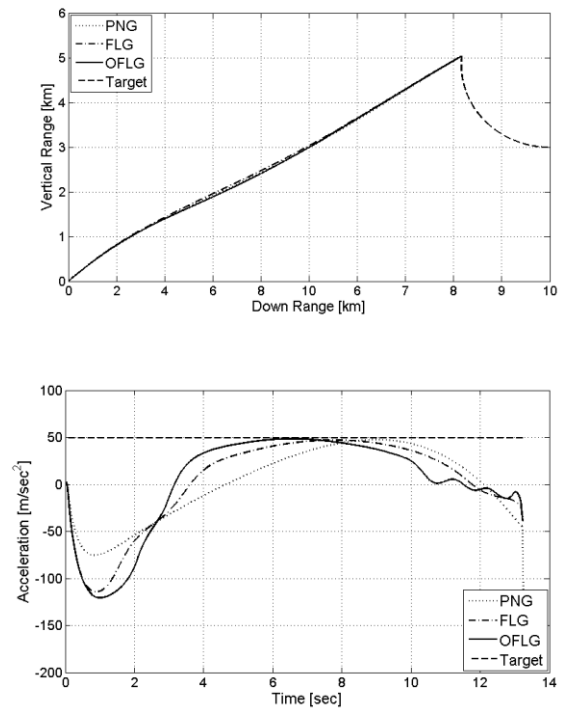
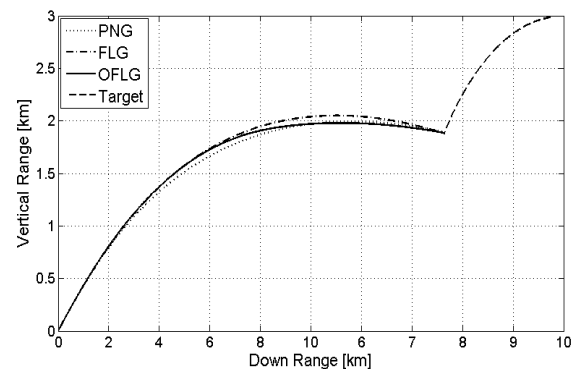


Fig. 10: Trajectories and Accelerations for Incoming and Up-ward maneuvering Target.



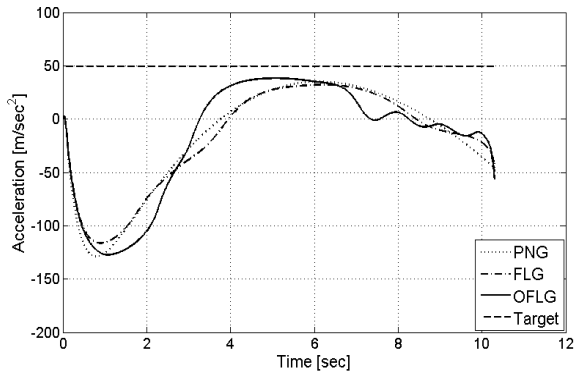


Fig. 11: Trajectories and Accelerations for Incoming and Down-ward maneuvering Target.

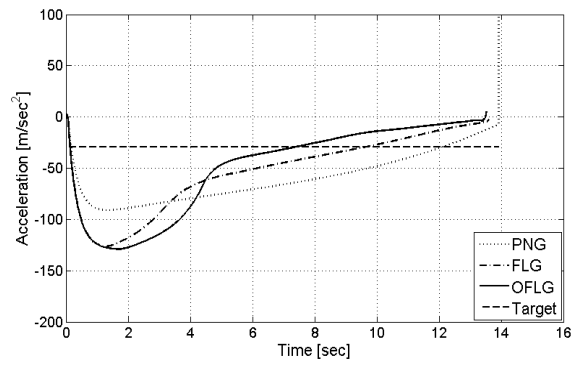


Fig. 13: Trajectories and Accelerations for Escaping and Down-ward maneuvering Target.

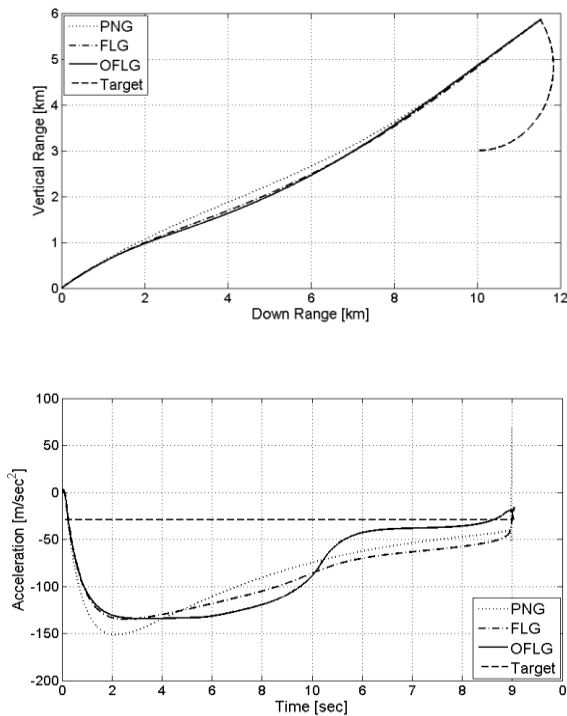
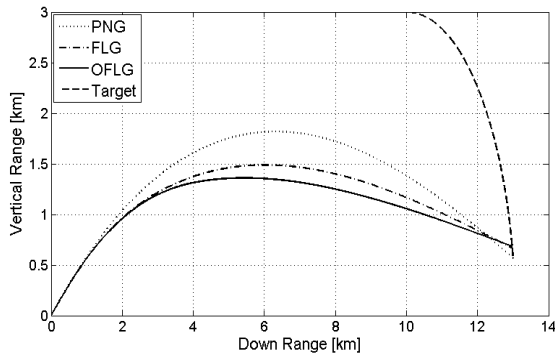


Fig. 12: Trajectories and Accelerations for Escaping and Up-ward maneuvering Target.



5. NOISE AFFECTING ON OFLG:

In fact, measuring data follows a random distribution due to thermal and radar noises; therefore, white noise is added to the measured signals to account for the disturbances. Such effects can be modeled as Gaussian density function (\square_{\square}) declared as:

$$\square_{\square}(\square) = \frac{1}{\sqrt{2\square\square^2}} \cdot \square^{-\frac{(\square-\square)^2}{2\square^2}} \quad (11)$$

Where; \square is the mean value and \square is the standard deviation. We added white Gaussian noise to the input signals (\square_{\square} and \square). The results listed in Table 9 show that the proposed OFLG law performs well with the existence of noisy measurements when the noise is low, in other words, high signal to noise ratio (SNR).

Table 9: Performance of OFLG with noise existence.

Case	RMS of \square_{\square} [m]	RMS of $\square_{\square\square\square}$ [m^2/sec^3] $\times 10^{-4}$
SNR=100	8.92	6.07
SNR=50	11.71	10.26
SNR=25	1217.65	5.32

It is obvious that when noise becomes very high the missile cannot intercept the target, actually this is because of the wrong information which sent to the guidance controller which in turn sends wrong commands to the actuator.

6. CONCLUSION

In the current study, we introduce an OFLG law based on the classical PNG conception. The MFs of the proposed design are optimized using PSO algorithm. Many cases are studied respect to different scenarios of target maneuvers. RMS values of miss-distance and control effort for all scenarios are calculated. The results show that OFLG performs better than FLG and PNG in all scenarios. Further investigation shows that the designed guidance law is a robust where it shows to perform well with the existence of noise measurements.

7. REFERENCES

- [1] P. Zachan, "Tactical and Strategic Missile Guidance." 5th edition, AIAA, 2007.
- [2] Chun Ling Lin, "On the Design of a Fuzzy-PD Based Missile Guidance Law." IEEE, The Fourth International Conference on Control and Automation, vol. 10, no. 12, pp. 7-11, 2003.
- [3] Chun Ling Lin, Tzu Chiang Kao, and Meng Tzong Wu, "Design of a Fuzzified Terminal Guidance Law." International Journal of Fuzzy Systems, vol. 9, no. 2, pp. 110-115, 2007
- [4] C. H. Chen, Y W Ling, D. C. Liaw, S. T. Chang, and S. D. Xu, "Design of Midcourse Guidance Laws via a Combination of Fuzzy and SMC Approaches." International Journal of Control, Automation and Systems, vol. 8, no. 2, pp. 272-278, 2010.
- [5] Hanafy M. Omar, M. A. Abido, "Designing integrated guidance law for aerodynamic missiles by hybrid multi-objective evolutionary algorithm and tabu search." Aerospace Science and Technology, vol. 14, no. 5, pp. 356-363, 2010.
- [6] Lin C.T. and Lee C.S.G. "Neural fuzzy systems." Upper Saddle River, Prentice Hall, Inc.1996.
- [7] Jang, J.S.R., "ANFIS: Adaptive-Network-Based Fuzzy Inference System." IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 665- 685. 1993.
- [8] Lin C.J., Chen C.H., Lee C. Y. "TSK-type quantum neural fuzzy network for temperature control." International Mathematical Forum, Vol. 1, No 18, pp. 853 - 866. 2006.
- [9] Pratihar D.K., Deb K., Ghosh A. "A genetic-fuzzy approach for mobile robot navigation among moving obstacles." International Journal of Approximate Reasoning, Vol 20, No. 2, pp. 145–172. 1999.
- [10] Mucientes M., Moreno D.L., Bugarin A., and Barro S., "Design of a fuzzy controller in mobile robotics using genetic algorithms." Applied Soft Computing. 7, pp. 540-546. 2007.
- [11] Kennedy, J. and Eberhart, R. "Swarm Intelligence," San Francisco, CA: Morgan Kaufmann Publishers, Inc. 2001.
- [12] Clerc, M. and Kennedy, J. "The particle swarm - explosion, stability, and convergence in a multidimensional complex space." IEEE Transactions on Evolutionary Computation, Vol 6, No. 1, pp. 58-73. 2002.
- [13] Kwok, N.M., Ha, Q.P., Liu, D.K. and Fang, G. "Intensity-preserving contrast enhancement for gray-level images using multi-objective particle swarm optimization." IEEE Conf. on Automation Science and Engineering, Shanghai, P. R. China, pp. 19-24. 2006.
- [14] Kwok, N.M., Ha, Q.P., Nguyen, T.H., Li, J. and Samali, B. "A novel hysteretic model for magneto rheological fluid dampers and parameter identification using particle swarm optimization." Sensors & Actuators: A. Physical, Vol 132, No 2, pp. 441-451. 2006.
- [15] Karakuzu C. "Fuzzy controller training using particle swarm optimisation for nonlinear system control." ISA Transactions. Vol. 47, pp. 229 – 239. 2008.
- [16] Niu B., Zhu Y., He X., and Shen H., "A multi-swarm optimizer based fuzzy modeling approach for dynamic systems processing." Neuro-computing. Vol 71. pp. 1436-1448. 2008.
- [17] Mukherjee V., Ghoshal S.P. "Intelligent particle swarm optimized fuzzy PID controller for AVR system." Electric Power Systems Research, Vol. 77, pp. 1689-1698. 2007.
- [18] Lin C., Hong S. "The design of neuro-fuzzy networks using particle swarm optimisation and recursive singular value decomposition." Neuro-computing, Vol 71, pp. 271-310. 2007.
- [19] Esmine A., Aoki A., Lambert T. G. "Particle swarm optimisation for fuzzy membership functions optimisation" IEEE, International Conference on Systems, Man, and Cybernetics, Hammamet, Tunisia. 2002.
- [20] Esmine A., Lambert T. G. "Fitting fuzzy membership functions using hybrid particle swarm optimization." IEEE, International Conference on Fuzzy Systems, Vancouver, BC, Canada. pp. 2112 – 2119. 2006.
- [21] V. Rajasekhar and A. G. Sreenatha, "Fuzzy Logic Implementation of Proportional Navigation Guidance." Acta Astronautica, vol. 46, no 1, pp17-24, 2000.
- [22] Werner Van Leekwijck and Etienne E. Kerre, "Defuzzification: criteria and classification." Fuzzy Sets and Systems, vol. 108, pp. 159-178. 1999.
- [23] Chun Liang Lin, Hao Zhen Hung, Yung Yue Chen, and Bor Sen Chen, "Development of an Integrated Fuzzy-logic based missile guidance law against high speed target." IEEE, Transaction on Fuzzy System, vol. 12, no. 2, 2004.
- [24] Qinghai Bai, "Analysis of Particle Swarm Optimization Algorithm." CCSE, computer and information science, vol. 3, no. 1, 2010.