

The Design of a Modified PSO Guidance Law Using Predictor and LOS Rate Evaluation

Yung-Lung Lee¹, Kuei-Yi Chen^{2,a}, Yi-Wei Chen³, Yen-Bin Chen¹ and Sheng-Ju Liao²

¹Department of Power Vehicle and Systems Engineering, C.C.I.T., National Defense University, Taoyuan County 33551, Taiwan, ROC.

²School of Defense Science, C.C.I.T., National Defense University, Taoyuan County 33551, Taiwan, ROC.

³Department of Aircraft Engineering, Air Force Institute of Technology, Kaohsiung 82047, Taiwan, ROC.

Abstract. This paper proposes a modified particle swarm optimization guidance (MPSOG) for the pursuit-evasion optimization problem. If the line-of-sight (LOS) rate equals zero, the curvature of the missile's trajectory would be smaller and the probability of intercept greater. Thus, we propose using a MPSOG to improve the guidance performance. The MPSOG uses a Kalman filter to predict a target's dynamic. The lateral acceleration commands of the y - and z -axis are optimized by the particle swarm optimization (PSO) algorithm, respectively. Numerical simulation results show that the MPSOG have better guidance performance than proportional navigation guidance and particle swarm optimization guidance in miss distance, time-to-go, and final lateral acceleration commands.

1 Submitting the manuscript

Since the inception of missile technology, the research of guidance laws, which guide a missile to attack its target effectively, has attracted a lot of attention. Many research achievements have been made, including proportional navigation guidance (PNG) [1], augmented PNG [2], pure PNG [3], and command to line-of-sight (CLOS) guidance [4]. With the development of control theory, optimal control theory has also been utilized in missile guidance design. This includes optimal intercept guidance [5], optimal missile guidance [6], and modified receding horizon guidance [7].

In recent years, artificial intelligence algorithms have successfully been applied in many fields which started the research into intelligent guidance. For example, Rajasekhar et al. [8] proposed a fuzzy logic controller to adjust the navigational constant of the PNG. Deskovski et al. [9] developed a fuzzy-rule-based system which can calculate the closing velocity and lead angle in order to compute the navigation constant of the PNG. Song et al. [10] proposed an on-line suboptimal midcourse guidance law which used neural networks for air-to-air medium-range missiles. Choi et al. [11] presented a differential game missile guidance which used neural networks.

Since the inception of a particle swarm optimization (PSO) algorithm, its many advantages which include its few parameters, fast convergence, simple scheme and easy implementation have caused concern. Some studies, such as Parsopoulos et al. [12] and Kuo et al. [13] have also pointed out that the capability of a PSO is similar to other AI algorithms. In particular, some studies, such as Kung et al. [14], have been used in missile guidance.

In this paper, we propose a modified PSO guidance (MPSOG) to improve the guidance performance. The MPSOG uses a Kalman filter to predict the target's dynamic at the first step, and the line-of-sight (LOS) rates of the y - and z -axis are taken as the objective functions. Then the fitness functions are evaluated according to the defined objective function. Therefore, the lateral acceleration commands of the y - and z -axis can be optimized by the MPSOG, respectively.

In the remainder of this paper, Section 2 briefly describes the background of the PSO, and describes the MPSOG in detail. After presenting the experimental design and discussing the results in Section 3, finally, Section 4 gives conclusions.

2 PSO algorithm and MPSOG

2.1 PSO algorithm

The PSO algorithm was proposed in 1995, by Kennedy et al. [15]. Since the proposal of the PSO algorithm, a number of more advanced research papers have been published. The improved formula, described in Eqs. (1)-(2), was proposed by Shi et al. [16, 17], and is now called the standard PSO.

$$V_{ij}^{new} = wV_{ij}^{old} + c_1rand_1(Pbest_{ij} - X_{ij}^{old}) + c_2rand_2(Gbest_i - X_{ij}^{old}) \quad (1)$$

$$X_{ij}^{new} = X_{ij}^{old} + V_{ij}^{new} \quad (2)$$

where V_{ij}^{old} is the previous velocity of a particle; V_{ij}^{new} is the new velocity of a particle after updating; $Pbest_{ij}$ is the self-experience of an individual particle; $Gbest_i$ is the neighbouring experience of particle swarm; w is the

^a Corresponding author: KueiYi.Chen@gmail.com

inertial weight factor; Both c_1 and c_2 are learning factors; Both $rand_1$ and $rand_2$ are random numbers between $[0, 1]$; X_{ij}^{old} is a current value of a particle; X_{ij}^{new} is a new value of a particle after updating; the subscript i is the iteration index, and the subscript j is the particle index.

The PSO algorithm implementation initializes all parameters which the PSO algorithm needs at first. Second, objective values are calculated by the objective function, and the fitness function is evaluated according to the defined objective function. The algorithm uses Eqs. (1)-(2) to update the states of the particle swarm after the fitness function. Finally, the algorithm should stop and output a solution if the stop condition is satisfied, otherwise goes to the second step.

2.2 MPSOG

At each time-step, the MPSOG decides the best guidance commands. The inputs are states of the missile and the target, and the outputs are the best guidance commands that are the y - and z -axis lateral acceleration commands. According to the symbol definitions of the PSO algorithm described in the previous section, the details of the MPSOG can be described in the following:

- Step 1. (**Prediction**): In the first step, the Kalman filter is used to predict the target's dynamic.
- Step 2. (**Initialization**): Preset the w , c_1 , c_2 , maximum iteration index i_{max} , swarm size n , global best fitness value $GbestJ_{\phi_i}$ and $GbestJ_{\theta_i}$, respectively. According to n , preset the velocities of particle $V_{\phi_{ij}}^{old}$ and $V_{\theta_{ij}}^{old}$, and the local best fitness values $PbestJ_{\phi_{ij}}$ and $PbestJ_{\theta_{ij}}$. Let particle X_{yij}^{old} and X_{zij}^{old} be the sets of the y - and z -axis lateral acceleration commands, respectively. We initialize the particles within a random distribution between the limits of lateral acceleration command.
- Step 3. (**Calculating objective values and fitness values**): We assume that the missile is modeled as a point-mass, and the motion equations refer

$$\begin{cases} Pbest_{\phi_{ij}} = X_{yij}^{old} (\min(J_{\phi_{ij}})) & \& PbestJ_{\phi_{ij}} = \min(J_{\phi_{ij}}), \text{ if } PbestJ_{\phi_{ij}} > \min(J_{\phi_{ij}}) \\ Pbest_{\theta_{ij}} = X_{zij}^{old} (\min(J_{\theta_{ij}})) & \& PbestJ_{\theta_{ij}} = \min(J_{\theta_{ij}}), \text{ if } PbestJ_{\theta_{ij}} > \min(J_{\theta_{ij}}) \end{cases} \quad (6)$$

$$\begin{cases} Gbest_{\phi_i} = Pbest_{\phi_{ij}} (\min(PbestJ_{\phi_{ij}})) & \& GbestJ_{\phi_i} = \min(PbestJ_{\phi_{ij}}), \text{ if } Gbest_{\phi_i} > Pbest_{\phi_{ij}} (\min(PbestJ_{\phi_{ij}})) \\ Gbest_{\theta_i} = Pbest_{\theta_{ij}} (\min(PbestJ_{\theta_{ij}})) & \& GbestJ_{\theta_i} = \min(PbestJ_{\theta_{ij}}), \text{ if } Gbest_{\theta_i} > Pbest_{\theta_{ij}} (\min(PbestJ_{\theta_{ij}})) \end{cases} \quad (7)$$

$$\begin{cases} V_{\phi_{ij}}^{new} = wV_{\phi_{ij}}^{old} + c_1rand_1 (Pbest_{\phi_{ij}} - X_{yij}^{old}) + c_2rand_2 (Gbest_{\phi_i} - X_{yij}^{old}) \\ V_{\theta_{ij}}^{new} = wV_{\theta_{ij}}^{old} + c_1rand_1 (Pbest_{\theta_{ij}} - X_{zij}^{old}) + c_2rand_2 (Gbest_{\theta_i} - X_{zij}^{old}) \end{cases} \quad (8)$$

where $PbestJ_{\phi_{ij}}$ and $PbestJ_{\theta_{ij}}$ are the local best fitness values, respectively; $Pbest_{\phi_{ij}}$ and $Pbest_{\theta_{ij}}$ are the local best y - and z -axis lateral acceleration commands, respectively; $GbestJ_{\phi_i}$ and $GbestJ_{\theta_i}$ are the global best fitness values, respectively; $Gbest_{\phi_i}$ and $Gbest_{\theta_i}$ are the global

to Imado [18]. Let the a_{pc} equal a particle X_{yij}^{old} , and a_{yc} equal a particle X_{zij}^{old} . We can then predict the missile's behavior when we utilize the Runge-Kutta numerical method to solve the point-mass model of the missile. Also, we assume that the target is modeled as a point-mass. Therefore, LOS can be defined as a virtual straight line between the missile and target. The LOS rate is the angular rate of the angle between LOS and horizontal reference line in each time-step. Let LOS rate be the objective value. The Eqs. (3)-(4) of the three-dimensional LOS rate refer to Yang [19], and the objective values $J_{\phi_{ij}}$ and $J_{\theta_{ij}}$ for each particle are calculated by Eq (5).

$$\dot{\phi}_{ij} = \frac{\dot{z}_{ij} (x_{ij}^2 + y_{ij}^2 + z_{ij}^2) - z_{ij} (x_{ij}\dot{x}_{ij} + y_{ij}\dot{y}_{ij} + z_{ij}\dot{z}_{ij})}{\sqrt{x_{ij}^2 + y_{ij}^2} (x_{ij}^2 + y_{ij}^2 + z_{ij}^2)} \quad (3)$$

$$\dot{\theta}_{ij} = \frac{x_{ij}\dot{y}_{ij} - \dot{x}_{ij}y_{ij}}{x_{ij}^2 + y_{ij}^2} \quad (4)$$

$$\begin{cases} J_{\phi_{ij}} = \sqrt{\dot{\phi}_{ij}^2} \\ J_{\theta_{ij}} = \sqrt{\dot{\theta}_{ij}^2} \end{cases} \quad (5)$$

where x , y , and z are the relative distance of x -, y -, and z -axis between the missile and the target, respectively; $\dot{\theta}$ and $\dot{\phi}$ are the horizontal and vertical LOS rate, respectively.

- Step 4. (**Individual best and swarm best evaluating**): In a pursuit-evasion game, the variation of the LOS rate will vary with the dynamic of the missile and target. If the LOS rate equals zero at each step-time, namely the LOS parallels with each other, the curvature of the missile's trajectory will be smaller. The probability of intercept will be greater. Therefore, the two fitness functions, Eqs. (6)-(7), are based on these concepts.

best y - and z -axis lateral acceleration commands, respectively.

- Step 5. (**Velocity and Position updating**): We use Eq. (8) to update the $V_{\phi_{ij}}^{new}$ and $V_{\theta_{ij}}^{new}$, and use Eq. (9) to update the X_{yij}^{old} and X_{zij}^{old} , respectively.

$$\begin{cases} X_{yij}^{new} = X_{yij}^{old} + V_{\phi ij}^{new} \\ X_{zij}^{new} = X_{zij}^{old} + V_{\theta ij}^{new} \end{cases} \quad (9)$$

Step 6. (**Stop condition checking**): Calculate the standard deviations of the X_{yij}^{new} and X_{zij}^{new} , respectively. If the standard deviations are both greater than the convergent tolerance, set $V_{\phi ij}^{old}$ and $V_{\theta ij}^{old}$ equal to $V_{\phi ij}^{new}$ and $V_{\theta ij}^{new}$, respectively. We also set $X_{\phi ij}^{old}$ and $X_{\theta ij}^{old}$ equal to $X_{\phi ij}^{new}$ and $X_{\theta ij}^{new}$, respectively, and then return to Step 3. If the standard deviations are both less than the convergent tolerance or i equals to i_{max} , and output the $Gbest_{\phi i}$ and $Gbest_{\theta i}$.

3 Numerical simulation and results

The maximum g -force left turn flight is used to evaluate the guidance performance of the MPSOG. The results would compare with the PNG and the PSO guidance (PSOG). [14].

The parameter set of the MPSOG are as follows: n is set as 100; i_{max} is set as 700; w is set as 0.5; Both c_1 and c_2 are set as 1; The convergent tolerance is set as 10^{-2} ; The $X_{\phi ij}^{old}$, $X_{\theta ij}^{old}$, $V_{\phi ij}^{old}$, and $V_{\theta ij}^{old}$ are initialized randomly $\pm 25g$. The simulation stops at the minimal miss distance (MD), otherwise, the initial conditions of missile and target are shown in Table 1.

Table 1. The initial conditions of missile and target.

| Item | Missile | Target |
|----------------------|---------|--------|
| x -axis (m) | 0 | 4,500 |
| y -axis (m) | 0 | 3,500 |
| z -axis (m) | 3,000 | 3,100 |
| Initial speed (Mach) | 2 | 0.8 |
| Time-step (sec) | 0.02 | 0.02 |

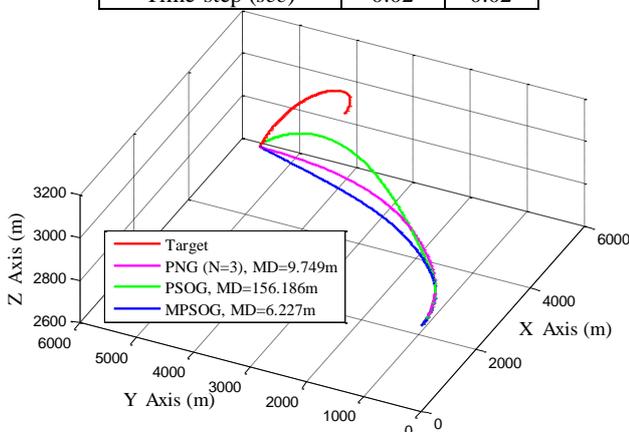


Figure 1. The simulation results where the maneuver of the target is a maximum g -force left turn flight.

The simulation trajectories are illustrated in Figure 1, and Table 2 collates the simulation results. According to the simulation results, the guidance performance of the MD makes it quite clear that the MPSOG is better than other methods. Also, the guidance performance of the

time-to-go (T_{go}) of the MPSOG is better than other methods. These results revealed that the MPSOG has a higher tracking ability and accuracy. As a brief summary, the simulation results revealed that the MPSOG had a better guidance performance than the PNG and the PSOG.

Table 2. The simulation results of the evasion scenario.

| Item | PNG | PSOG | MPSOG | |
|--|-----------|--------|--------|-------|
| MD (m) | 9.75 | 156.19 | 6.23 | |
| T_{go} (sec) | 8.84 | 8.96 | 8.82 | |
| The lateral acceleration commands at the final stage (g) | y -axis | -0.65 | -25.00 | 0.48 |
| | z -axis | -6.15 | 25.00 | -5.33 |

4 Conclusion

In this study, we propose a MPSOG and demonstrate its validity. We use a Kalman filter to predict the target's dynamic at first. Then the LOS rate of the y - and z -axis are taken as the objective functions, and the lateral acceleration commands of the y - and z -axis are optimized by a PSO algorithm, respectively. Simulation results show that the MPSOG had a better guidance performance in MD, T_{go} , and final lateral acceleration commands.

References

1. S. A. Murtaugh and H. E. Criel, IEEE Spectrum **3**, 75 (1966)
2. G. M. Siouris, NTZ-Commun J. **27**, 278 (1974)
3. K. Becker, IEEE T. Aero. Elec. Sys. **26**, 526 (1990)
4. I.J. Ha and S. Chong, IEEE T. Aero. Elec. Sys. **28**, 51 (1992)
5. R. G. Cottrell, AIAA J. **9**, 1414 (1971)
6. R. K. Aggarwal, IEEE Decis. Contr. P. **3**, 2775 (1996)
7. K. B. Kim, M. J. Kim, and J. W. Choi, IEEE T. Aero. Elec. Sys. **36**, 725 (2000)
8. V. Rajasekhar and A. G. Sreenatha, Acta Astronaut. **46**, 17 (2000)
9. S. Deskovski, Z. Gacovski, and G. Dimirovski, RAST, 680 (2003)
10. E.J. Song and M.J. Tahk, Control Eng. Pract. **6**, 957 (1998)
11. H.L. Choi, Y. Park, H.G. Lee, and M.J. Tahk, AIAA GNC., 1 (2001)
12. K. E. Parsopoulos and M. N. Vrahatis, Intelligent Technologies—Theory and Application: New Trends in Intelligent Technologies **76**, 214 (2002)
13. H.C. Kuo, J.R. Chang, and C.H. Liu, J. Mar. Sci. Technol. **14**, 170 (2006)
14. C.C. Kung and K.Y. Chen, T. Can. Soc. Mech. Eng. **37**, 971 (2013)
15. J. Kennedy and R. Eberhart, IEEE ICNN **4**, 1942 (1995)
16. Y. Shi and R. Eberhart, IEEE ICEC, 69 (1998)
17. Y. Shi and R. C. Eberhart, IEEE CEC **3**, 1945 (1999)
18. F. Imado, J. Guid. Control Dynam. **16**, 289 (1993)
19. C.C. Yang, Master Thesis, Tainan, (1993)