MissileSIM: A System for Missile Simulation

Yi Ru Liu [1,*], Yui-Liang Chen [2] and Limin Liu [2]

[1] Acton-Boxborough Regional High School, Acton, MA; 24liuy@abschools.org
[2] Department of Information Management, Shih Hsin University, Taipei, Taiwan; ychen@mail.shu.edu.tw, liminliu@mail.shu.edu.tw
* Correspondence: 24liuy@abschools.org

Abstract: In this study, we build an app which provides an environment for simulating guidance and navigation of interceptors to a desired target location. We provide multiple modes of visualization and simulation of our proposed guidance method which is derived from modifying the lead collision method. Different scenarios of target-interceptor guidance are also discussed in this paper. Such a system can help to visualize basic guidance principles and can also be adapted for different application domains such as video games.

Keywords: Navigation; Guidance; Lead Collision; Proportional Navigation Guidance; Missile Interception

1. Introduction

For the past few decades, many studies regarding the various methods of guiding an interceptor to a target have been conducted. As technology advances, aerial interceptions have gone through different phases where the relative speed and agility difference of the interceptor and target have drastically changed. At the dawn of military aviation, defending fighters had to be guided to intercept inbound enemy planes; in this case, the target group of enemy planes and defending interceptors had relatively similar speeds and maneuverability. Whether or not the interception was successful usually came down to an aerial dogfight between the defenders and the attackers, in which interceptors generally preferred to arrive behind the targets, placing themselves in a more advantageous position.

As military technology matured, rockets, missiles, and other projectile interceptors were developed. Projectiles of this kind were smaller and lighter, making them much faster and nimbler than propeller or jet powered aircraft; these obvious advantages eventually led to development of offensive guided missiles and intercontinental ballistic missiles. Under these circumstances, defending missiles have to intercept projectiles high up in the atmosphere or even re-entering the atmosphere from outer space. The interceptors, in this situation, would have much greater maneuverability but much slower speeds [1].

With missile interceptions, the target only has to be within the blast radius of the intercepting missile for a successful interception. Different guidance algorithms have been developed to ensure that the interceptor successfully reaches its desired target location. In this study, we develop a missile navigation simulator with Java and JavaFX called MissileSIM. We also implement a guidance method which adopts the concept of lead collision based on proportional navigation. An acceleration modification is proposed for future works, which could further improve the accuracy of the simulator. This paper is organized as follows: Two fundamental guidance techniques and our own proposed method are explained in Section 2. Experimental results are illustrated in Section 3, followed by conclusions in Section 4.

2. Overview of Missile Interception Sequence

Before an interceptor can guide itself to the target, it must first acquire the location, velocity, and other critical information about the target. There are three different categories of target acquisition: *active*, *semi-active*, and *passive* [2]. Active missiles track on to a target with its own onboard seekers, while semi-active missiles lock on to the radiation reflected off the target provided by an external source. Examples of active missiles would include the AIM-120 AMRAAM and the AIM-9 Sidewinders, while semi-active missiles include the AIM-7 Sparrows. Passive missiles lock on to enemy targets emitting radiation, an example being the AGM-88 HARM.

After acquiring a target, the missile now has to guide itself to the target location for a successful interception. The two most fundamental guidance techniques are *Pure Pursuit* and *Proportional Navigation*. In this section we discuss these two techniques as well as our own proposed guiding method.

2.1. Pure Pursuit

An interceptor following Pure Pursuit points directly at the target at all times as shown in Figure 1 [3]. The line between the interceptor (red triangle) and the target (blue diamond) is called the Line of Sight (LOS). In this figure at time t_0, the target is moving left directly perpendicular to LOS (λ_0=90 degrees). The interceptor is therefore pointed downwards (γ_0=-90 degrees) directly at the target in Figure 1.

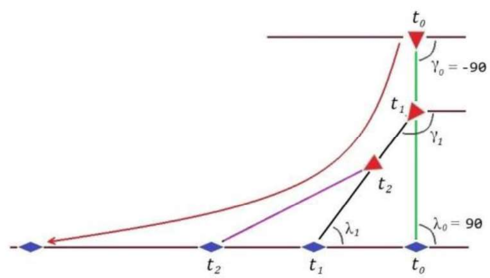

Figure 1. Pure Pursuit.

After a certain time period Δt, the interceptor and target move to new locations designated as t_1 in **Figure 1**. The same principle is then applied to find the new LOS (black line), which is then used to calculate the new λ_1 and γ_1 to guide the interceptor. Such a technique may lead to a *tail chase* scenario if the interceptor misses the target on its first pass. As long as the interceptor has unlimited energy, a speed faster than the target, and a spontaneous turn rate, the interceptor will eventually reach the target. However, Pure Pursuit may not be the most effective at hitting the target since upon final approach, sharp angles are created between the target heading and the interceptor heading. In reality, missiles have a finite amount of energy, so if the missile misses on its first pass, it most likely will not have enough energy to turn back at the target.

2.2. Proportional Navigation

In order to address this flaw of the Pure Pursuit method, another fundamental guidance algorithm known as Proportional Navigation was developed. This method of missile guidance can be expressed by the following:

$$a_n = NV_c\left(\frac{d\lambda}{dt}\right) \tag{1}$$

where a_n is the acceleration perpendicular to the interceptor's instantaneous velocity vector, N is the proportionality constant, V_c is the closing velocity, and $d\lambda/dt$ is the LOS rate measured by the interceptor [2, 4]. The trajectory angle of the interceptor can be illustrated as follows:

$$\dot{\gamma}(t) = N\dot{\lambda}(t) \tag{2}$$

where $\dot{\gamma}(t)$ is the interceptor flight path angle rate, N is the proportionality constant identical to formula (1), and $\dot{\lambda}(t)$ is rate of change in LOS angle. Figure 2 illustrates the interceptor trajectory using the proportional navigation technique with initial conditions equal to Figure 1. Assuming γ_0=-100, γ_1 will be -110 degrees, -100+1x(-100+90), when N=1; γ_1 will be -120 degrees if N=2, and γ_1=-130 if N=3. The larger N is, the faster the interceptor reaches a static angle. The value of N is usually set between 3 and 5 to minimize the interceptor acceleration (a_n) as shown in formula (1) [4]. The trajectory of an interceptor (red triangle) following Proportional Navigation is illustrated by the blue line, which eventually collides with the target (blue diamond) at C_1.

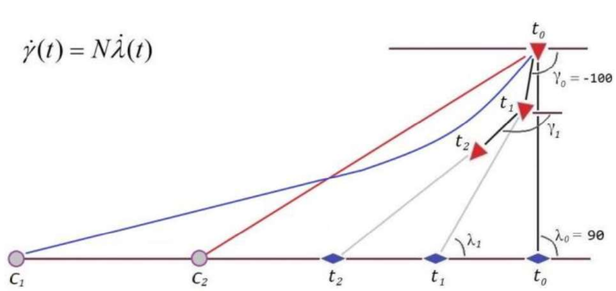

Figure 2. Proportional Navigation.

2.3. Proposed Guidance Law

As shown in Figure 2, the trajectory produced by following Proportional Navigation turns the interceptor at the beginning to eventually meet the ideal intercept angle. Our proposed guidance method decreases turn time of the missile by modifying the Proportional Navigation method. Rather than turning slowly towards the final interception angle, the collision angle is calculated and the interceptor is turned immediately on to a collision course as shown with the green line, which ends at the collision point of P_c in Figure 3.

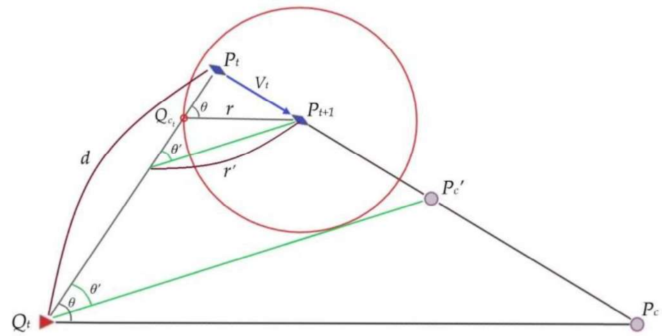

Figure 3. Proposed Guidance System.

The interceptor in Figure 3 is located at Q_t with a distance of d to P_t, the target, on LOS. The target, located at P_t, has a heading shown as the blue arrow and a velocity of V_t. After a time period of Δt, the target will travel to P_{t+1}. If Δt equals one unit time of V_t, then the traveling distance will equal V_t. The intercept window is shown as the red circle with center P_{t+1} and radius r, the traveling distance of the interceptor over a time period of Δt. The location where the intercept window intersects LOS is designated Q_{ct} and is where the interceptor should be Δt before interception. If the intercept window does not intersect or is tangent to LOS, then interception is not possible. To prevent a tail-chase scenario where the missile can never catch up to the target, the proposed technique signals the interceptor to speed up to a velocity of C_iV_t, where C_i is a constant greater than 1. In the scenario discussed next, C_i is set to 1.2. The angle formed between point P_t, Q_{ct} and P_{t+1} is the angle of interception expressed as θ. If the interceptor follows a path of the line originating from P_t with angle θ and the target maintains speed and direction, they will collide at point P_c. If the interceptor has a larger speed r', the intercept angle will be θ' and it will collide with the target at P_c'.

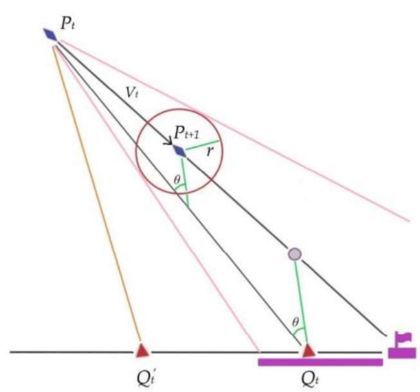

Figure 4. Proposed Guidance System.

In the scenario where an interceptor is defending nearby assets (purple flag) from incoming threats, the target is heading towards the asset and the interceptor launch site as shown in Figure 4. The target moves from P_t to P_{t+1} after Δt. In such situations, even low speed interceptors can still successfully intercept the target. Locations within the purple bar are all possible launch locations assuming spontaneous turn rate. These are determined by tangents from the intercept window to point P_t, shown as pink lines. Any launch location outside of the purple bar will result in failed interceptions because the LOS does not intersect with the intercept window.

3. Experimental Results

In this study, we use Java and JavaFX to implement our proposed guidance method [5]. The system follows the producer-consumer design pattern where the target produces the necessary data consumed by the interceptor(s).

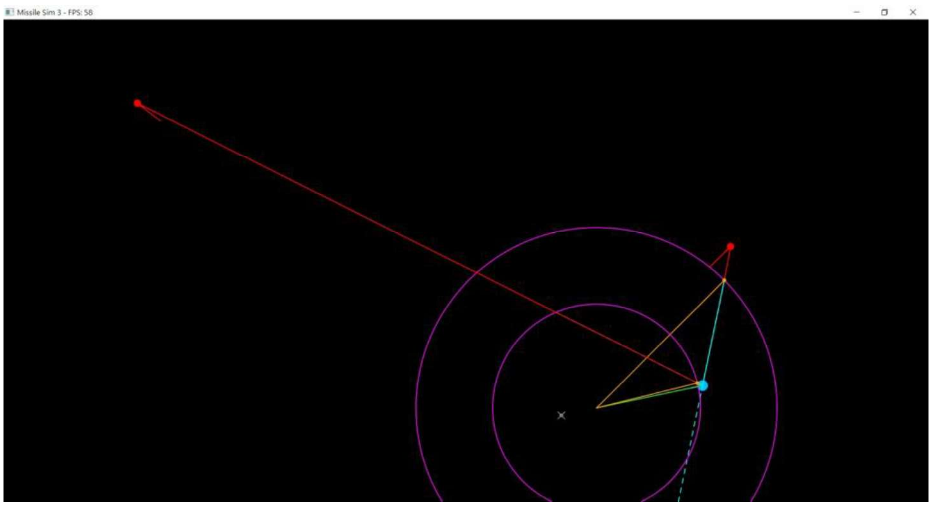

Figure 5. Direct-intercept mode.

This simulation provides various modes for the user to choose. Figure 5 shows an example of Direct-intercept mode, in which the interceptor points directly at the target when launching. The magenta circle represents the intercept window and grows as the interceptor's speed increases. The blue dot represents the target and the red dots represent the interceptors. Auxiliary lines in the small colored indicate headings for both the target and the interceptor, and the red line from the interceptor to the target represents LOS. The green line represents V_t from P_t to P_{t+1} and the blue line represents P_t to Q_{ct} in Figure 3. The gray cross indicates the intercept point for each interceptor.

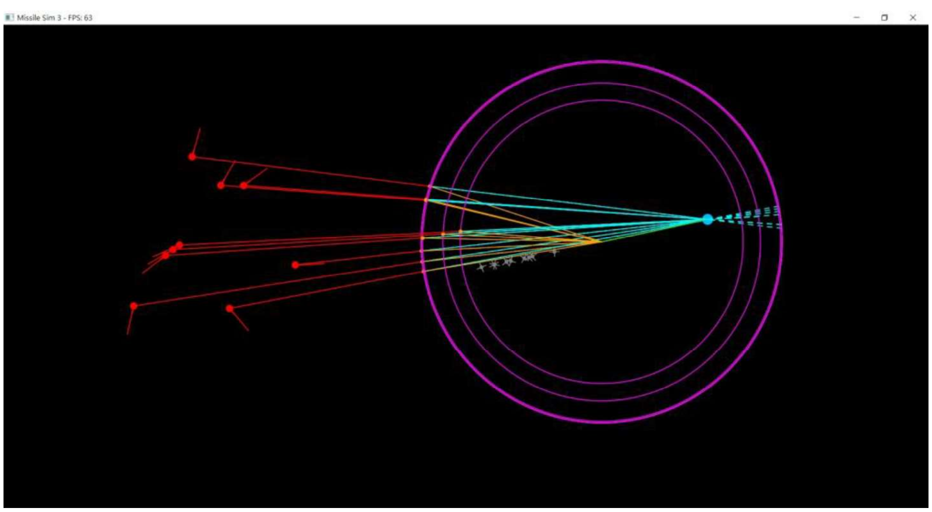

Figure 6. Burst mode.

Figure 6 shows an example of Burst mode, in which all interceptors launch at a random angle. This mode creates an interesting scenario where if the target detects the first missile launch and maneuvers to evade the incoming interceptor, interceptors that were launched later or pointed in a different direction may now be ahead or facing the target from a different angle, increasing the probability of successful interception.

In Trail mode, users can control the target's movement by keystroke A and D as demonstrated in Figure 7, in which the target makes several turns during the simulation. In such mode, the flight path of both target and interceptors are shown as trails and are more easily visualized. Note that the target is bound by the canvas and will bounce back after colliding with the borders. Interceptors, however, are not bound by the canvas and can turn freely in and out of the canvas. This creates another interesting scenario where the border acts as an artificial and imaginary force that turns the target sharply, which then forces the target onto a path the missile cannot follow. This illustrates a real life problem not discussed in this paper of missile turn rate, where targets can out turn missiles and force overshoots that missiles are not able to recover from. Several video clips of the simulation demonstrations are provided in [6].

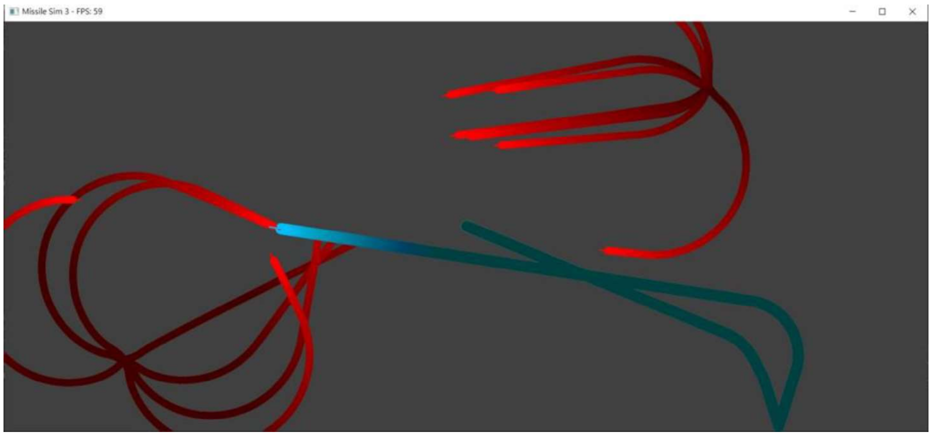

Figure 7. Trail mode.

4. Conclusions

In this study, we develop the app MissileSIM and implement our own proposed guidance method based on proportional navigation and lead collision. MissileSIM offers many execution modes: Direct-intercept, Burst and Trail mode, where Direct-intercept mode launches the interceptor directly pointing at the target; Burst mode launches interceptors in all directions that will intercept the target in a surrounding manner; and Trail mode provides an enhanced visualization of the entire interception sequence by displaying the trajectory of both the target and the interceptor(s). In addition, users can control the movement of the target and modify several other parameters in a configuration file such as time scale, trail length, movement behaviors of both target and interceptor, and visualization colors.

MissileSIM provides a simulation environment for a target-interceptor guidance scenario that can not only be used to help understand the basic guidance principles but also be applied in computer games, especially in evasion maneuver scenarios.

References

1. Shin H.; Lee J., Kim H.; Shim D.H. Design the Guidance Law for Formation Flight of Multiple UAVs. In Proceedings of the 30th Congress of the International Council of the Aeronautical Sciences, Daejeon, Korea, 28 Sept. 2016.
2. Siouris G.M. *Missile Guidance and Control Systems*, Springer, New York, 2004. pp. 155–268.
3. Scharf L.L.; Harthill W.P.; Moose P.H. A Comparison of Expected Flight Times for Intercept and Pure-pursuit Missiles. *IEEE Tran. on Aero. and Elec. Sys.* 1969, 5(4): 672–673.
4. Raj S.D. Performance Evaluation of Proportional Navigation Guidance for Low-Maneuvering Targets. *Intl. J. of Sci. & Eng. Res.*, 2014, 5(9): 93–99.
5. Oracle Java Software: https://www.oracle.com/ java/ (2022/11/10).
6. MissileSIM Demo Video Clips: https://forms.gle/WXJvXzsPEDXjUxDLA (2022/11/10).