ANALYSIS AND SIMULATION OF THE TARGET-ATTACKER AND THE TARGET-ATTACKER-DEFENDER PROBLEMS_Literature Review

By Mostafa Ali Rushdi

Abstract

This thesis deals with two scenarios of target defense. We first present several methodologies to find the optimal escape maneuver for a target against an attacking missile. We simulate two-dimensional proportional-navigation using MATLAB and Simulink. Optimization is achieved via the techniques of Monte-Carlo simulation and genetic algorithms. We establish a Graphic User Interface (GUI) Guidance toolbox containing our guidance law and several types of maneuvers. This toolbox is an open-source program for the development and addition of other guidance laws and maneuvers. We also construct a mathematically-correct game of target-attacker and let many people play it taking the target side. We find the best escape maneuver by collecting and analyzing data of the human escape maneuver. The game is developed using Unity, a free readily-available cross-platform game engine. We then consider the case when the target is being helped by a defender. We offer a unified analytic treatment of this active defense problem via the construction of two Apollonius circles, considering all possibilities of the ratio between the speeds of the attacker and defender. A criticality condition is derived, from which we obtain the critical target speed and the Voronoi diagram bordering the safe or escape region for the target optimal strategies. Our numerical results and plots allow useful and insightful qualitative interpretations. Next, we find the optimal heading angles to be followed by the target so as to stay in the safe region. We use Hamiltonian equations to formulate an exact two-point boundary value problem that is solved numerically, yielding results verifying our earlier results.

Part I

INTRODUCTION AND LITERATURE REVIEW

Chapter 1: Introduction and Literature Review

This thesis deals with two scenarios of active target defense involving:

1. Two-agent pursuit-evasion: a Target (aircraft) in opposition to an Attacker (missile). The Target tries to evade (avoid being captured) the Attacker. This problem will be referred to herein as the TA problem as it concerns the Target (T) and Attacker (A).

2. Three-agent pursuit-evasion where each agent has a specific role. A two-agent team consists of a Target (aircraft) and a Defender (missile) cooperating in opposition to an Attacker (missile). The Target tries to evade the Attacker and avoid being captured by him. The Defender cooperates with and assists the Target by trying to intercept (capture and destroy) the Attacker before the latter captures the Target. This problem will be referred to herein as the TAD problem as it concerns a triad consisting of the Target (T), Attacker (A), and Defender (D).

The following two sections pertain to a brief introduction and literature survey for each of the TA and TAD problems.

1.1 The TA problem

Solutions to the TA problem are called homing missile guidance laws, and the most popular among them is a control law called proportional navigation [1, 2, 3], which is analogous to classical proportional control. The underlying basic concept is that if the direct Target-Attacker line-of-sight time rate is diminished, i.e., if the direct TA Line-of-Sight ceases to change its direction then (for a non-maneuvering, constant velocity Target) the Attacker is on a collision course. If the Target is considered smart or maneuvering, then variations to the proportional navigation are needed so as to minimize the miss distance. These variations have been given optimal-control interpretations through linear quadratic Gaussian (LQG) formulations, and a variety of other techniques [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31].

The literature has investigated manifold variations of the TA problem (including the TAD problem, producing a large amount of results, and a huge number of publications (see, *e.g.*, [9, 10, **?**, 32, 33, 34]. The study of pursuit evasion games can be traced back to the von Neumanns hide-and-seek game [35], where a hider chooses one cell of a two-dimensional grid in which to hide himself and a seeker chooses a subset of cells of the grid (usually one row and one column) in which to seek the hider. If the seeker selects the cell chosen by the hider, then the hider is captured and the seeker wins the game. Otherwise, the hider wins the game. Starting from this seminal work, adversarial pursuit evasion settings have been modeled by a variety of mathematical techniques. Notable among these are the optimal-control formulation and the differential-game one [7, 36, 37].

The fundamental conceptual difference between missile guidance laws based on optimal control theory and those based on differential game theory is in the assumptions made by the guidance laws on the future trajectory and maneuvering capabilities of the Target. Optimal control theory assumes that the future maneuver strategy of the Target is completely defined, either in open-loop or closed-loop form [7]. The feedback nature of missile guidance laws allows the missile to make corrections for inaccurate predictions of the maneuvers of the Target. The optimal-control formulation is appropriate only when future maneuver time history, or strategy, of the Target is known or can be justifiably assumed or accurately predicted. One possible perspective is to design strategies that maximize performance of the Attacker against a worst-case Target [32]. In such a setting, the Target is usually considered to be of a finite speed, complete awareness of the location and intent of the Attacker, and full knowledge of the conflict environment. Such a method guarantees the success of the pursuit, defined, for example, by capture of the Target in a finite time. However, this powerful-adversary model may yield solutions that are too conservative in practical applications. A better alternative is to use probabilistic formulations addressing average-case behaviours [32].

The most suitable (and presumably most successful) mathematical framework for analyzing conflicts controlled by two independent agents is in the realm of dynamic or differential games. Thus, the scenario of intercepting a maneuverable Target has to be formulated as a zero-sum pursuitevasion game [36]. The roles of the game players are clearly defined, the interceptor (Attacker) is the pursuer and the Target is the evader. The natural cost function of such a zero-sum game is the miss distance (the distance of the closest approach, or in other words the smallest norm of the separation vector), to be minimized by the pursuer and maximized by the evader. The game solution provides simultaneously the missiles guidance law (the optimal pursuer strategy), the best Target maneuver (the optimal evader strategy), and the resulting guaranteed miss distance (the value of the game). As a consequence, the game solution provides a guidance law that is robust with respect to the Target maneuver structure.

Many prominent extensions of the TA problem are currently problems of hot research. In addition to the celebrated TAD problem (to be fully introduced in section 1.2), we give a glimpse of problems of contemporary interest in the following (far-from-conclusive) list

- Three-dimensional pursuit-evasion [1, 5, 21, 37],
- Fuzzy guidance laws [23, 25],
- Non-conventional or modern approaches for interception [6, 20],
- Lyapunov-based non-linear guidance [16].
- Incorporation of probabilistic uncertainty in the location, behavior, and/or sensor observations of the Target.

1.2 The TAD problem

The TAD problem constitutes a dynamic differential game [38, 39, 40, 41, 42, 43, 44, 45, 46] and is of interest in aerospace, control, and robotics engineering. We will consider herein recent formulations and treatments of this problem [47, 48, 49, 50, 51, 52], though there exist other formulations and treatments of it that span almost half a century [53, 54, 55, 56, 57, 58, 59, 60, 61]. Table 1.1 shows several variants of this generic problem in a variety of settings, contexts or disciplines. The TAD problem is a generalization of the classical problem of a single pursuer and a single evader [62, 63, 64, 65, 31]. In fact, the TAD is essentially

Area	Agent 1	Agent 2	Agent 3	References
Aerospace	Target	Attacker (missile)	Defender (missiles)	[47, 48, 49, 50, 51]
Biology	Prey	Predator	Protector	[56, 61]
Society	Lady	Bandits	Bodyguards	[55]
Criminology	Robber	Policemen/Cops	Gangsters	[77]

Table 1.1: A variety of settings or contexts for the same generic problem

a duplication of this classical problem, since in the TAD problem, the Attacker plays the double role of being a pursuer for the Target, and at the same time an evader for the Defender [19]. The TAD problem is also a special case of a more general pursuit-evasion problem in which there are multiple Attackers and multiple Defenders [66, 67, 68, 69, 70]. Many common threads are shared by all these problems, such as the solution of differential games [38, 39, 40, 41, 42, 43, 44, 45, 46] and the construction of Apollonius circles [71, 72, 73, 74] and Voroni diagrams [75, 76, 77, 78, 79, 80, 81, 82].

1.3 Thesis Contribution

As stated earlier, this thesis has contributions to both the TA and the TAD problems.

In the TA problem, we search for a path that the Target can move on to escape from the Attacker. All the evasion techniques depend on the time of the turn that the Target makes when it detects the Attacker (Missile) and the objective is to maximize the Missile acceleration till the Missile power bleed. We choose the escaping trajectory as a polynomial with unknown coefficients, then decide the values of these coefficients so as to make the Missile exert a maximum acceleration to bleed its power as fast as possible before it reaches the Target. We explain the meaning of proportional navigation, and subsequently simulate two- dimensional proportional-navigation equations using MATLAB and Simulink. Optimization is achieved via the techniques of Monte-Carlo simulation and genetic algorithms. We establish a Graphic User Interface (GUI) Guidance toolbox containing our guidance law and several types of maneuvers. This toolbox is an open-source program for the development and addition of other guidance laws and maneuvers. We also construct a mathematically-correct game of target-attacker and let many people play it taking the target side. We find the best escape maneuver by collecting and analyzing data of the human escape maneuver. The game is developed using Unity, a free readily-available cross-platform game engine.

This thesis also offers a unified analytic treatment of the TAD problem based on the construction of two Apollonius circles. The treatment includes all possibilities of the ratio between the speeds of the Attacker and Defender.Note that the case of a slow Defender appears here for the first time, while the treatment of the cases of fast Defender or similar Defender is extended and augmented with novel results and new insights. A criticality condition is derived from which the two following important entities are obtained

- the critical Target speed normalized w.r.t. the Attacker's speed so as to be a dimensionless quantity,
- the Voronoi diagram bordering the safe or escape region for the Target.

Optimal strategies are also studied, and are shown to obey a complex sixth-degree polynomial when the Defender differs in speed from the Attacker. This polynomial reduces to a real fourth-degree polynomial when the Defender and Attacker are similar in speed. Beside unifying previously published results in a common setting, this thesis simplifies all computations by using intuitionistic plane-geometric arguments rather than the more tedious analytic-geometric manipulations. Moreover, the thesis extends existing results by adding some novel results, thereby giving a complete picture of all cases of interest. The analysis in this thesis is supplemented by extensive computations using MATLAB to solve the complex high-order polynomial equations and to plot the Voronoi diagrams under a variety of pertinent parameters. The numerical results and plots obtained allow useful and insightful interpretations and are in exact agreement with numerical solution of the corresponding two-point boundary value problem.

1.4 Thesis Outline

This thesis consists of four parts. Part I comprises the current chapter of the introduction and literature survey. Part II pertains to the TA problem and consists of chapters 2-4. Chapter 2 is an introduction of the TA problem and devoted to a brief discussion of the guidance law used here. Chapter 3 constitutes a prominent cornerstone of the thesis as it reports extensive simulation results for the TA problem obtained via Matlab and simulink. Chapter 4 discusses a new game-based methodology for discovering optimal escape maneuver. It is followed by Part III of the thesis on the TAD problem, which consists of chapters 5-8. Chapter 5 introduces the TAD problem, lists the assumptions, notation and nomenclature used therein, discusses the general concepts and properties of an Apollonius circle [71, 72, 73, 74], and then specializes these to the cases of the AD Apollonius circle and the TA Apollonius circle. Chapter 6 derives a criticality condition which constitutes a quadratic equation that is later solved for the critical dimensionless ratio between the Target speed and Attacker speed. The same criticality condition is then rephrased in Chapter 7 as a relation between the initial coordinates x_T and y_T of the Target. This relation leads to the Voronoi diagram bordering the *safe* or *escape* region R_e of the Target. This Voronoi diagram is published for the first time when the Defender is fast or slow. It includes as a special case the much simpler diagram for a similar Defender that has already appeared in [49]. Chapter 8 explores the simulation of the optimal heading angle. The thesis is concluded with its last part (Part IV) which is a single (chapter 9) that summarizes the thesis contributions and points out possible directions for further work.

Part II

The 2-D TARGET-ATTACKER PROBLEM

Chapter 2: Introduction for TA problem

2.1 Problem Statement

In this Part, we consider a two-agent pursuit-evasion problem that involves a Target (aircraft) in opposition to an Attacker (missile). The Target tries to evade the Attacker and avoid being captured by him. We will try to find a simple technique for target evasion, so we should have a look at evasion techniques.

2.2 Evasion techniques

Each missile type has its own fundamental weaknesses and though some of these weaknesses or limitations may be kept secret by the missiles manufacturer, many of them can be easily inferred by the manufacturer's opponents through observation. Propulsion is one of the most prominent limitations of any missile. Most contemporary missiles employ rocket propulsion which enjoys the two advantages of a high thrust/weight ratio and a small size. The typical thrust profile of such a propellant rocket is that of an initial high impulse burn to accelerate up to cruise speed, which is followed by a slower sustainer burn. Once the propellant is burned out, the rocket will continue due to inertia until drag takes its toll. The maneuvering ability of such a missile depends critically on its speed and therefore the amount of wing and body lift it can generate. Immediately after launch the missile has poor maneuverability due to its low speed, but this improves as its cruise speed is reached. Maneuverability will peak at the instant when the sustainer is about to burn out as the weapon has its lowest mass and a high energy state while still possessing powerplant thrust. After sustainer burnout, the missile will bleed off its energy which reduces its ability to follow through maneuvers. Ideally, a missile airframe will employ a combination of wing and body lift to provide maximum turn rate with minimum energy bleed throughout all phases of its flight while maximizing range.

The aircraft must exploit every known weakness of the missile's system to defeat the missile. Missiles are a visible threat which act on a timescale that allows some tangible response. An aircraft maximizes its chances of survival by maintaining a high energy state, and by possessing a system of early and accurate warning. Figure 2.1 displays tree-like taxonomy of the techniques that can be used by an aircraft to avoid being hit by an attacking missile. The figure shows that there are two main classes of such techniques. The first class is that of kinematics-based tactics (maneuvers) and is, in fact, the genuine class of evasion techniques studied in aeronautical engineering and is the class of interest in this thesis. The second class of evasion techniques are not based on kinematics but are of proven practical utility. The advent of Fourth Generation Air-Air Missiles (AAMs) and 'double digit' Surface-Air missiles (SAMs) has somewhat decreased what could be achieved by using maneuver techniques. Nevertheless, these techniques remain effective against many conventional and legacy weapons, provided the aircraft has the required performance capabilities. Kinematics-based techniques are still of paramount importance form both the theoretical and practical points of view.

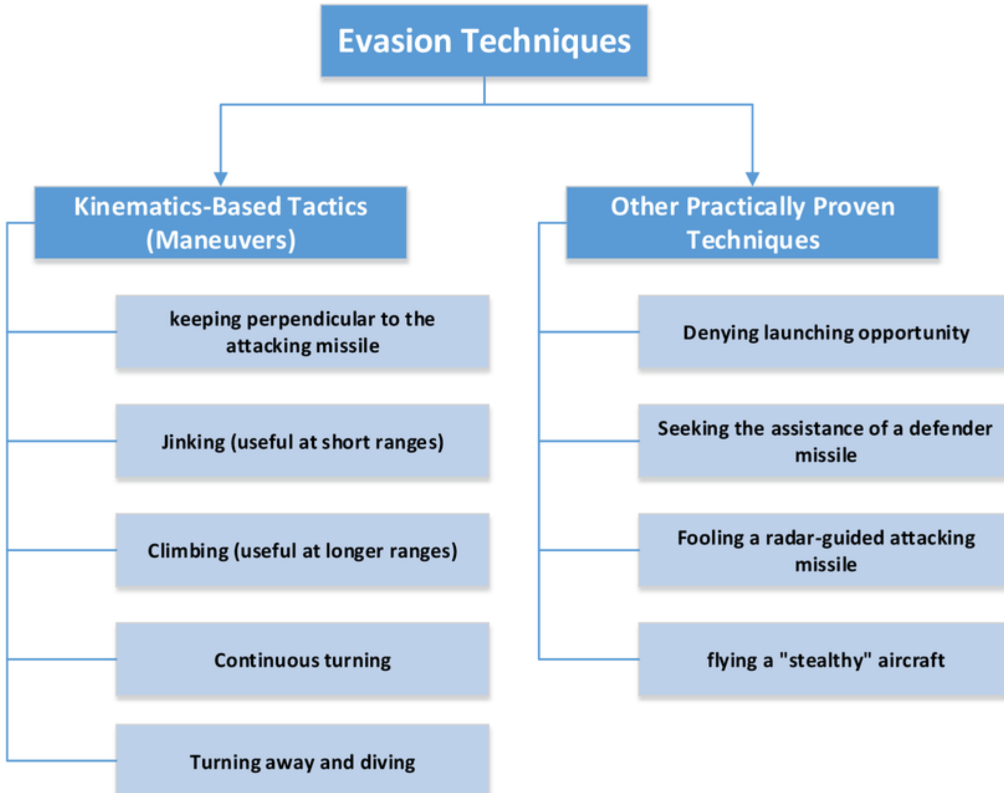


Figure 2.1: Taxonomy of evasion techniques

Before explaining the taxonomy given in Figure 2.1, we list a few useful observations, namely

- The missile is typically faster than the aircraft (otherwise, the aircraft can evade the missile by simply running away from it without having to maneuver).
- The missile cannot turn tighter than the aircraft, so it takes a longer path.
- The control mechanism of the missile is much simpler and is of less capability than that of the aircraft. In order to pull as tight turn as the aircraft, it must exercise an acceleration that is far beyond its capability.
- The missile always attempts to trace the target and not to lead it. Thus, if the target changes heading, it will be necessary for the Missile to change heading similarly, but this is too difficult for it to achieve.
- The main problem for an aircraft when evading a missile is how to compensate for the missile's speed superiority. This makes timing a somewhat critical issue for the aircraft.

The most important kinematics-based evasion techniques are :

1. **keeping perpendicular to the attacking missile:**
   Perpendicularity evading is the maneuver technique used when the missile is fired head-on at a beyond visual range (BVR). The target turns hard to either left or right so as to fly at roughly a 90 degrees angle to the attacking missile. This perpendicular orientation or beam aspect (3 o'clock position or 9 o'clock position) should allow visual acquisition and tracking of the incoming missile, particularly if it is originally fired from 6 o'clock position. Perpendicularity evading forces the missile to bleed off its energy and to possibly lead the target. Once the target aircraft makes a hard turn to reverse its direction, the missile with its far larger turn circle will be unable to compensate. The aircraft might have to constantly maneuver to keep perfectly perpendicular to the missile to keep it from locking on to it. Figure 2.2 illustrates the perpendicularity evading maneuver technique.

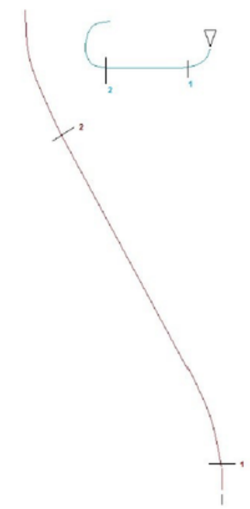


Figure 2.2: Illustration of the perpendicularity evading maneuver technique

2. **Jinking (useful at short ranges):**
   The word "jinking" means literally "to suddenly change direction" and essentially belongs to the specialized jargon of aeronautical engineering. In the jinking maneuver, the aircraft must be positioned so that it is at an acute angle (30-60 degrees are optimum) relative to the missiles flight path. Once the missile gets closer, the aircraft will make a hard turn in the opposite direction. As there is a time lag between the aircraft changing its direction and the missile following it (for several reasons, most important of which is the missiles inertia), this will cause the missile to head in the wrong direction until it manages to correct, and also to bleed off its limited energy. Usually the missile will fly past the aircraft and miss. Figure 2.3 illustrates the jinking evading maneuver technique.

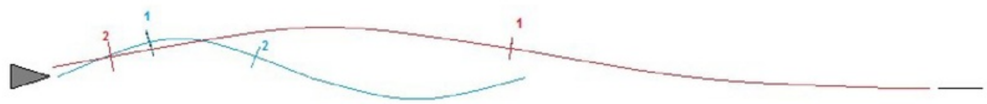


Figure 2.3: Illustration of the jinking maneuver technique

3. **Climbing (useful at longer ranges):**
   Since at a long range the missile will have burned out its engine, it will rely on inertia to keep flying, and climbing will mean that it will bleed off energy rapidly. Once the missile reaches a close range (maybe around 1,500 meters), the aircraft should dive for the ground, then pull up. This will allow the aircraft to gain kinematic energy and use it to evade the missile. Figure 2.4 illustrates the climbing evading maneuver technique.

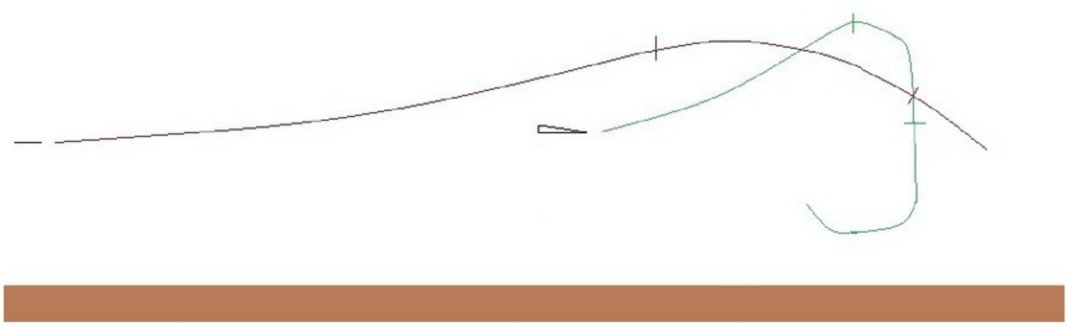


Figure 2.4: Illustration of the climbing maneuver technique

4. **Continuous turning:**
   The aircraft places the missile at a 3 oclock or a 9 oclock position. It then maintains a sufficient turn to keep the missile turning all the time. This tactic forces the missile to execute a continuous turn, bleeding its energy the entire time, making it easier for the aircraft to outturn the missile once it comes close. Figure 2.5 illustrates the continuous turning maneuver technique.

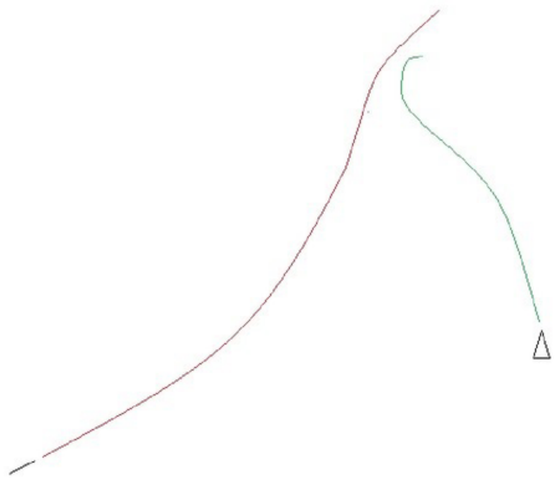


Figure 2.5: Illustration of the continuous turning maneuver technique

5. **Turning away and diving :**
   The aircraft turns away from the attacking missile and dives for the ground, gaining speed and putting as much distance as possible between itself and the missile. This tactic is not useful on its own at shorter ranges, and the pilot must evade the missile physically by pulling the aircraft into the turn and forcing the missile to overshoot once the missile comes sufficiently close. This ”Going to ground” tactic also confuses the missile’s guidance via ”back scatter” from objects on the ground

To make this thesis self-contained, we now add a brief description of other practical evasion techniques that are not of kinematic origin, and are therefore outside the main scope of this thesis:

1. **Denying launching opportunity:**
   The famous proverb ”*prevention is better than cure*” indicates that it is better to stop something undesirable from happening than it is to deal with it after it has already happened. Therefore, the best tactic for the aircraft when dealing with Surface-Air missile (SAM) sites is to prevent the launching of a missile by remaining safely outside of the known envelope or launching cone of the missile. This tactic is called the maneuver of denying the missile-launching site an opportunity to shoot. In fact, flying over SAM-infested hostile territory is asking to get hit, and should be avoided by all means. Of course, the tactic of denying launching opportunity is not an evasion one per se, albeit it is a very prudent one, indeed. In line with this tactic is the fact that the first mission objective in any offensive air operation is to destroy every hostile radar and SAM installation.

2. **Seeking the assistance of a defender missile:**
   The aircraft (or some of its air or surface allies) might launch a defender missile to

intercept and stop the attacking missile, hopefully before this attacker reaches it. This kind of active evasion changes our current TA problem to the more sophisticated TAD problem, to be discussed in the second part of this thesis.

3. **Fooling a radar-guided attacking missile:**
   Physically avoiding the enemy missile is not the only option for a targeted aircraft. If the missile is radar-guided, it can be misguided, decoyed, fooled, or deceived by the aircraft via electronic signal jamming or by dispensing or deploying a bundle of flares and chaff (tiny metal pieces) followed by changing course. The missile is caused to lose lock on the target and have either a miss or a premature detonation (When missiles lose the lock they either self-destruct or continue flying in a straight course). The aircraft might receive assistance from nearby friendly planes for jamming enemy radar on a broad range of frequencies (typically hostile ones).

4. **Flying a ”stealthy” aircraft:**
   Modern stealthy aircrafts have extremely small radar cross-sections, are impossibly quiet, are hard to see in the day or night, and have a heat signature not that much warmer than the ground they fly over. No one can target what radars, heat sensors, and human eyes and ears cannot detect. Drones are particularly effective with stealth, as the lack of need to support a pilot can grant them a cross-section smaller than a bird on radar. A very hot area of research in electromagnetics nowadays is to develop meta-materials that are essentially invisible over a broad band of frequencies.

Now we want to search for a path that the Target can move on it to escape from the Attacker. All the evasion techniques depend on the time of the turn that the Target makes when it detects the Attacker (Missile) and the objective is to maximize the Missile acceleration till the Missile power bleed. We will choose the escaping trajectory as a polynomial with unknown coefficients, then try to find these coefficients which make the Missile exert a maximum acceleration to bleed its power as fast as we can before it reaches the Target.

2.3 Assumptions, Notation, and Nomenclature

Assumptions

1. Both the Attacker and Target travel at constant speeds.
2. The Attacker’s speed is larger than that of the Target. Otherwise, the Target will be unconditionally or trivially capable of escaping.
3. Gravitational and drag effects are neglected for simplicity.
4. The maximum g-force a typical person can handle is 5G
5. The maximum g-force a pilot fighter (with safety equipments) could resist is 10G.
6. The maximum g-force for a sprint missile is 100G.
7. The range for Tactical missile defense, which has short ranges is 18 $\rightarrow$ 20 Kilometer, approximately 59000 $\rightarrow$ 66000 feet.

Notation

- n_c : Acceleration command (for the Missile) in m/s^2.
- N' : Effective navigation ratio, a unit-less designer-chosen gain (usually in the range of $3 \rightarrow 5$).
- V_c : Missile-Target closing velocity.
- λ : line-of-sight angle.
- $\dot{\lambda}$: line of sight rate.
- R_{TM} : length of the line of sight.
- L : Missile lead angle.
- HE : Heading error.
- $\dot{\beta}$: angular velocity of the Target.
- V_{T1}, V_{T2} : Target velocity components in the Earth fixed coordinate system.
- V_{M1}, V_{M2} : Missile velocity components in the Earth fixed coordinate system.

Nomenclature

Inertial coordinate system: fixed to the surface of a flat-Earth model (the 1 axis is down-range and the 2 axis can either be altitude or cross-range).

Missile lead angle: theoretically correct angle for the missile to be on a collision triangle with the Target.

Heading error (HE) : angle representing the initial deviation of the Missile from the collision triangle.

line of sight: The imaginary line connecting the Missile and Target.

length of the line of sight (R_{TM}): Instantaneous separation between Missile and Target.

Miss distance : The point of closest approach of the Missile and Target.

Closing velocity (V_c): the negative rate of change of the distance from the Missile to the Target $Vc = -\dot{R_{TM}} = -\frac{d}{dt}R_{TM}$.

2.4 Guidance Law

In this section, we will investigate some types of guidance laws, then we will choose one type to use it. We will be in the side of the Attacker, trying to hit the target. Figure 2.6 illustrates a classification (or taxonomy) of guidance laws [83]. The taxonomy (or an outgrowth thereof) is potentially a great aid in the further development of our Guidance Toolbox.

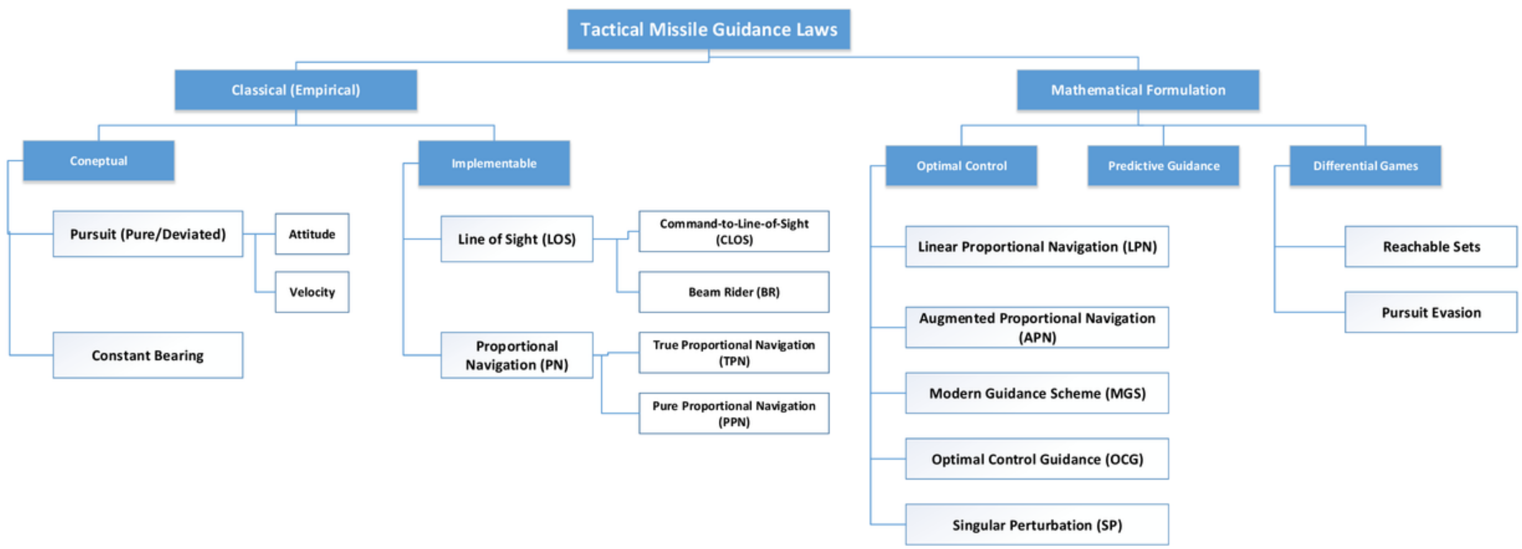


Figure 2.6: Guidance Laws Taxonomy.

2.4.1 Proportional Navigation

In this section, we will illustrate some fundamentals of missile guidance, focusing on the proportional navigation technique, which is one of the simplest guidance laws.

What is proportional navigation?

The proportional navigation guidance law issues acceleration commands, perpendicular to the instantaneous missile-target line-of-sight, which are proportional to the line-of-sight rate and closing velocity. Mathematically, the guidance law can be stated as

$$n_c = N' V_c \dot{\lambda} \tag{2.1}$$

where n_c is the acceleration command (for the missile) in (m/s^2), N' is the the effective navigation ratio, a unit-less designer-chosen gain (usually in the range of $3 \to 5$), V_c is the missile-target closing velocity in (m/s) and $\dot{\lambda} = \frac{d\lambda}{dt}$ is the rate of the line-of-sight angle and is in (rad/s). Note that (2.1) is dimensionally homogeneous, since n_c has the dimension

$$\begin{aligned}[n_c] &= [N'][V_c][\dot{\lambda}] \\ &= (1)(LT^-1)(T^-1) \\ &= LT^-2\end{aligned} \tag{2.2}$$

which is the appropriate dimension of linear acceleration.

2.4.1.1 Simulation of proportional navigation equations in 2-D

In this subsection we will introduce the equations of proportional navigation and the sequence to get a simulation for the path of the Target and the Attacker and how the missile acceleration will be affected during this simulation.

The simulation inputs are the initial location of the Missile (R_{M1}, R_{M2}) and the initial location of the Target (R_{T1}, R_{T2}), Target speed V_T, Missile speed V_M, and effective navigation ratio N'.

There are two types of **error sources** that cause the Attacker to miss the Target; they are the heading error (HE) and the acceleration of the Target (n_T) called the Target maneuver.

proportional navigation differential equations

The components of Target velocity

$$V_{T1} = -V_T \cos(\beta) \tag{2.3}$$

$$V_{T2} = V_T \sin(\beta) \tag{2.4}$$

Relative missile-target separation

$$R_{TM1} = RT1 - R_{M1} \tag{2.5}$$

$$R_{TM2} = RT2 - R_{M2} \tag{2.6}$$

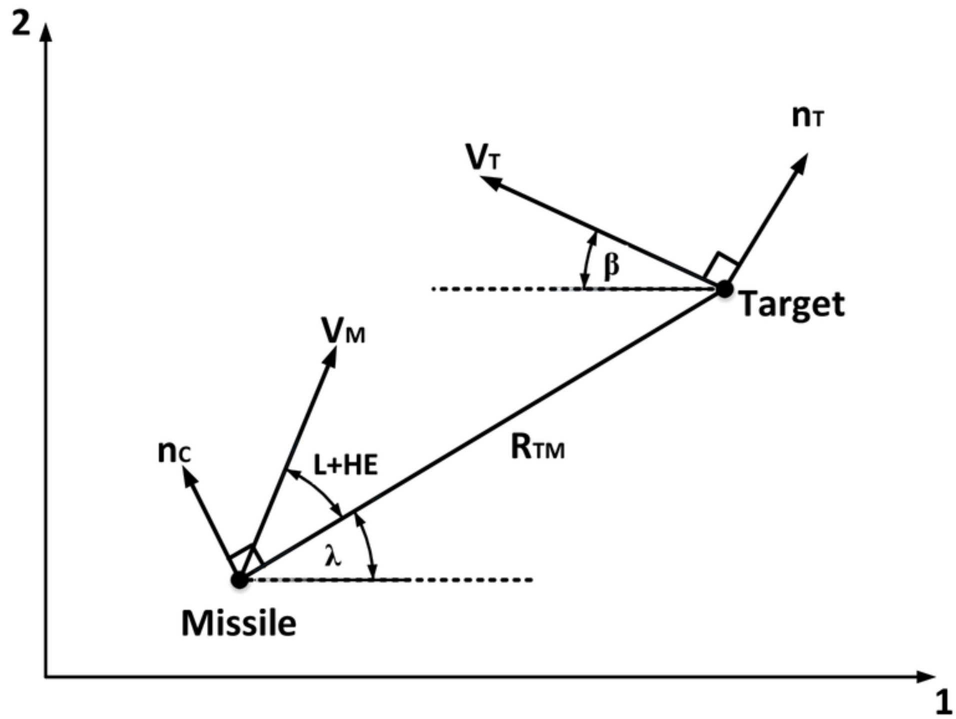


Figure 2.7: Two dimensional Missile-Target engagement geometry.

from the previous 2 equations we get

$$R_{TM} = \sqrt{R_{TM1}^2 + R_{TM2}^2} \tag{2.7}$$

line of sight angle

$$\lambda = \tan^{-1}(\frac{R_{TM2}}{R_{TM1}}) \tag{2.8}$$

missile lead angle

$$L = \sin^{-1}(\frac{V_T \sin(\beta+\lambda)}{V_M}) \tag{2.9}$$

the angle between the downrange axis and V_M vector is $\theta = \lambda + L$

Missile velocity components

$$V_{M1} = V_M \cos(\theta + HE) \tag{2.10}$$

$$V_{M2} = V_M \sin(\theta + HE) \tag{2.11}$$

Relative velocity components

$$V_{TM1} = V_{T1} - V_{M1} \tag{2.12}$$

$$V_{TM2} = V_{T2} - V_{M12} \tag{2.13}$$

closing velocity it is the negative rate of change of the distance from the missile to the target $Vc = -R_{TM}$, so we have to differentiate eq(2.7)

$$\dot{R_{TM}} = \frac{1}{2}(R_{TM1}^2 + R_{TM2}^2)^{\frac{-1}{2}}[2R_{TM1}\dot{R_{TM1}} + 2R_{TM2}\dot{R_{TM2}}]$$

we see that so we get

$$V_c = -\dot{R_{TM}} = -\frac{R_{TM1}V_{TM1} + R_{TM2}V_{TM2}}{R_{TM}} \tag{2.14}$$

line of sight rate we have to differentiate eq(2.8) using the rule $\tan^{-1} x = \frac{dx}{1+x^2}$

$$\begin{aligned}
\dot{\lambda} &= [\frac{1}{1+(\frac{R_{TM2}}{R_{TM1}})^2}](\frac{\dot{R_{TM2}}}{R_{TM1}}) \\
&= \frac{R_{TM1}^2}{R_{TM1}^2 + R_{TM2}^2}[\frac{R_{TM1}\dot{R_{TM2}} - R_{TM2}\dot{R_{TM1}}}{R_{TM1}^2}] \\
&= \frac{R_{TM1}V_{TM2} - R_{TM2}V_{TM1}}{R_{TM1}^2}
\end{aligned} \tag{2.15}$$

magnitude of the missile guidance command

$$n_c = N'V_c\dot{\lambda} \tag{2.16}$$

missile acceleration components

$$a_{M1} = -n_c \sin\lambda \tag{2.17}$$

$$a_{M2} = -n_c \cos\lambda \tag{2.18}$$

angular velocity of the target

$$\dot{\beta} = \frac{n_T}{V_T} \tag{2.19}$$

we will solve all the equations in this section using second-order Runge-Kutta numerical integration procedure. If we have a first order differential equation of the form

$$\dot{x} = f(x,t)$$

where t is time, we seek to find a recursive relationship for x as a function of time. With the second-order RungeKutta numerical technique, the value of x at the next integration interval h is given by

$$x_{k+1} = x_k + \frac{hf(x,t)}{2} + \frac{hf(x,t+h)}{2}$$

The following figure is a flowchart illustrating the steps of calculations till we plot the trajectories.

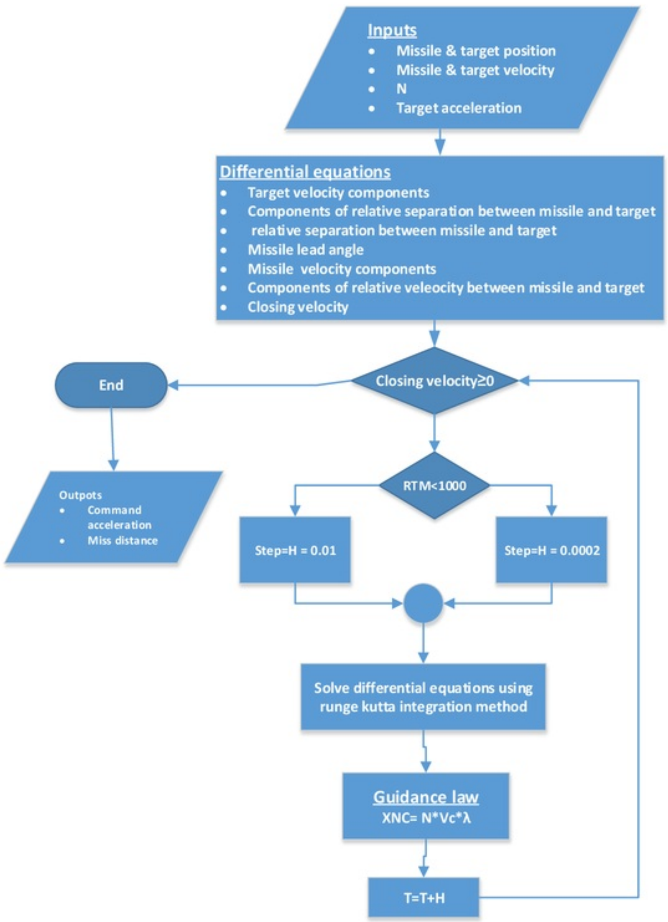


Figure 2.8: Flowchart illustrating the steps of calculations using the equations in sec. 2.3 till we plot the trajectories.

Chapter 3: Optimization for escaping maneuver

In this chapter, we will be in the Target side, trying to find an optimized escaping maneuver from the Attacker. We will use two optimization techniques, *Monte Carlo* and *Genetic algorithm*.

3.1 Models & Simulations

In this section we will simulate equations in sec. 2.4.1.1 for proportional navigation using MATLAB and Simulink.

3.1.1 Simulation for some target maneuver cases [using MATLAB]

In this subsection we will use MATLAB to simulate the equations in Sec. 2.4.1.1 for proportional navigation. We will solve the initial value problem of differential equations using second-order Runge-Kutta numerical integration technique, then we will draw the trajectories of the pursuit and evasion for 4 cases for the Target maneuver error source, and we will deduce the effect of the effective navigation ratio N' and the other type of the error source; Heading error.

3.1.1.1 Zero Target maneuver

In this case the evader (Target - plane) does not do any effort to escape, it just moves in a straight line, as we see in Fig. 3.1. So the pursuer (Attacker - Missile) does not have to bleed much energy to reach the Target.

In the case of **zero Heading error** the effective navigation ratio has no effect on the simulation engagement at all. The missile's acceleration will be zero as in Fig. 3.2.

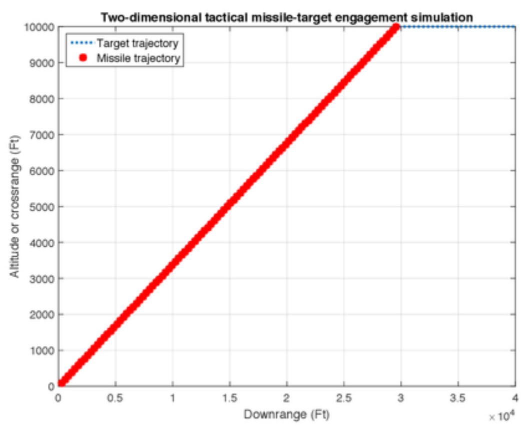


Figure 3.1: Trajectory of the target and attacker in case of zero target maneuver with zero heading error and $N' = 4$.

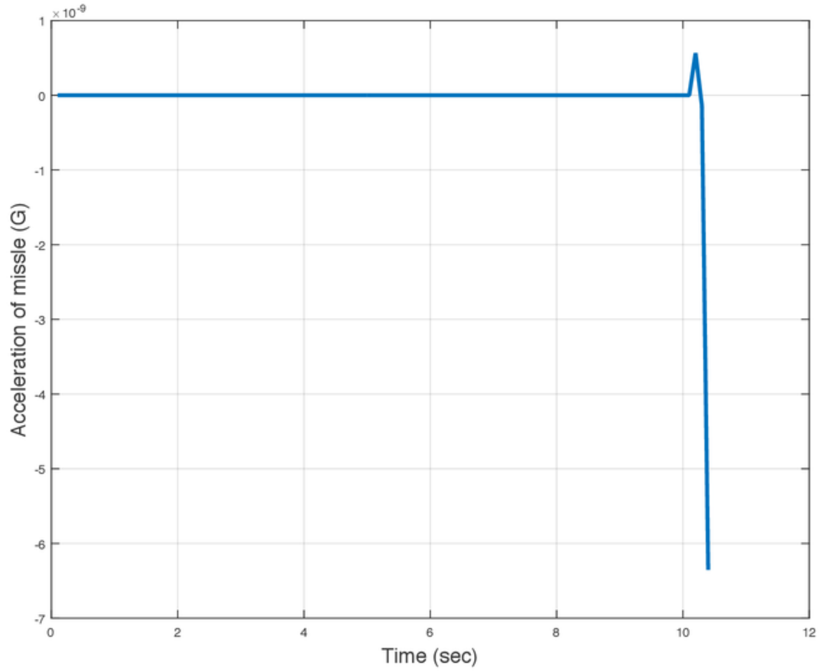


Figure 3.2: Missile acceleration in case of zero target maneuver with zero heading error and $N' = 4$.

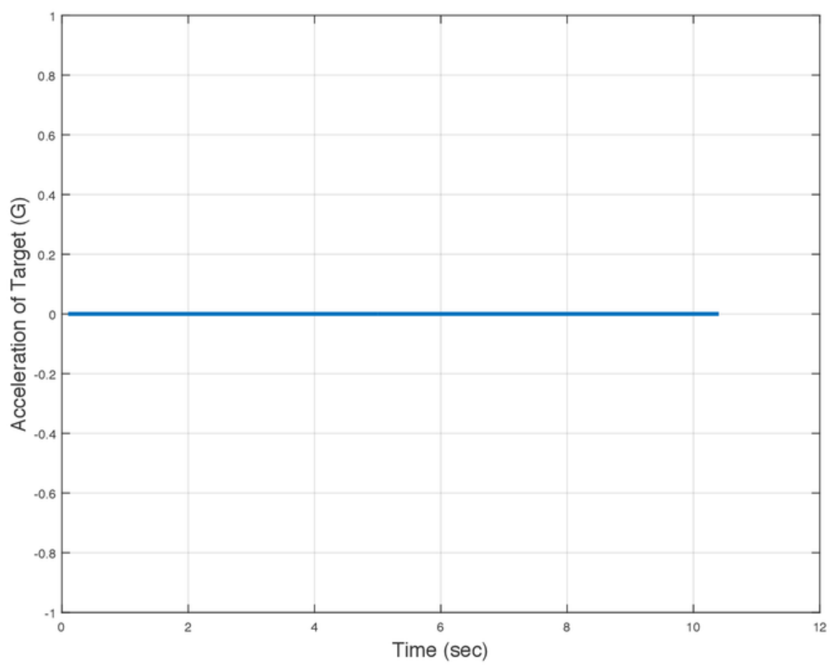


Figure 3.3: Target acceleration in case of zero target maneuver with zero heading error and $N' = 4$.

In the case of **Heading error = -20** increasing the effective navigation ratio causes heading error to be removed rapidly as we see from Fig. 3.4, Fig. 3.6 and Fig. 3.8. The effective navigation ratio has an effect on the acceleration of the missile; the way that the missile will bleed energy as we see from Fig. 3.5 , Fig. 3.7 and Fig. 3.9. The total acceleration (area under the curve) is increasingly inversely proportional with the effective navigation ratio N' .

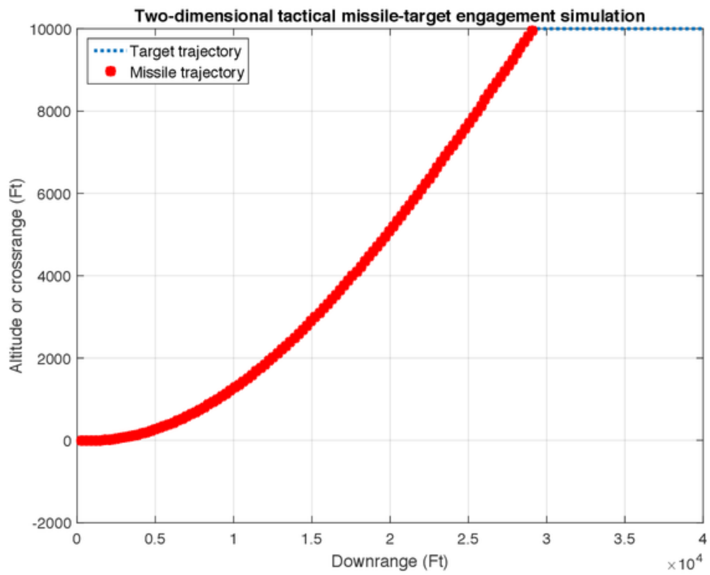


Figure 3.4: Trajectory of the target and attacker in case of zero target maneuver with heading error=-20 and $N' = 3$.

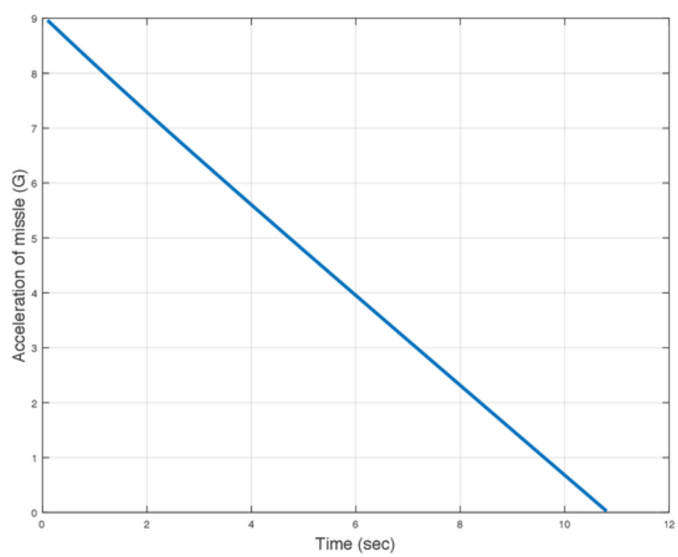


Figure 3.5: Missile acceleration in case of zero target maneuver with heading error=-20 and $N' = 3$.

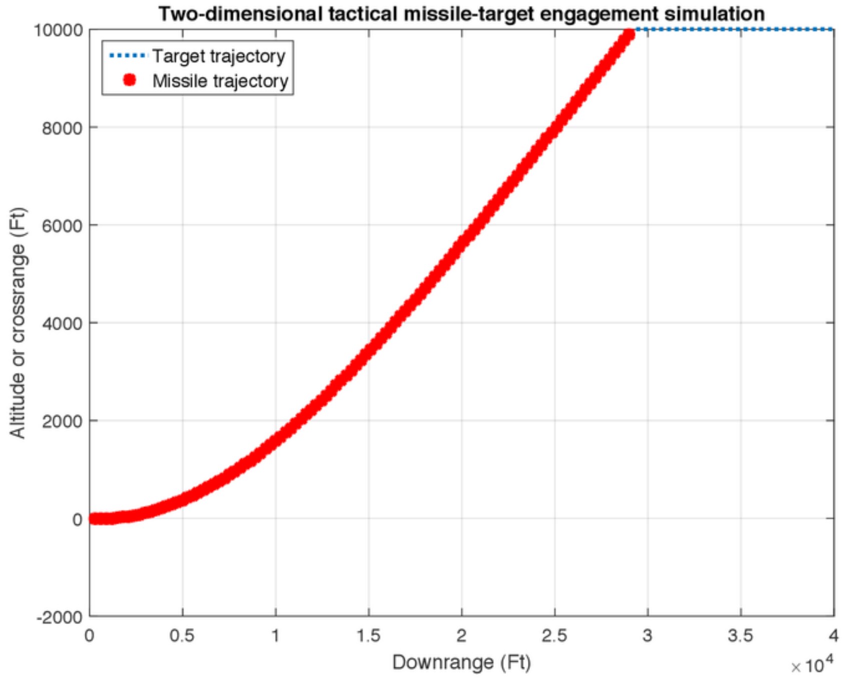


Figure 3.6: Trajectory of the target and attacker in case of zero target maneuver with heading error=-20 and $N' = 4$.

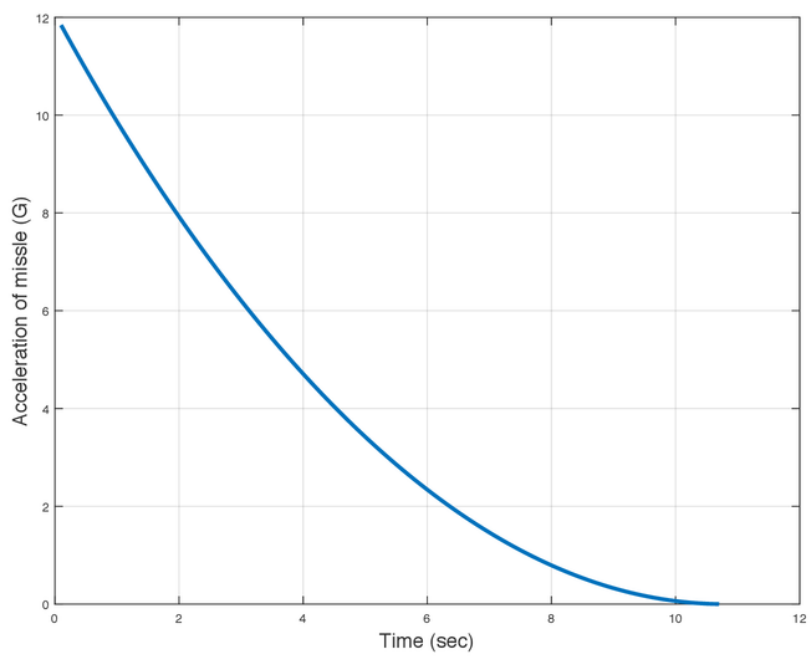


Figure 3.7: Missile acceleration in case of zero target maneuver with heading error=-20 and $N' = 4$.

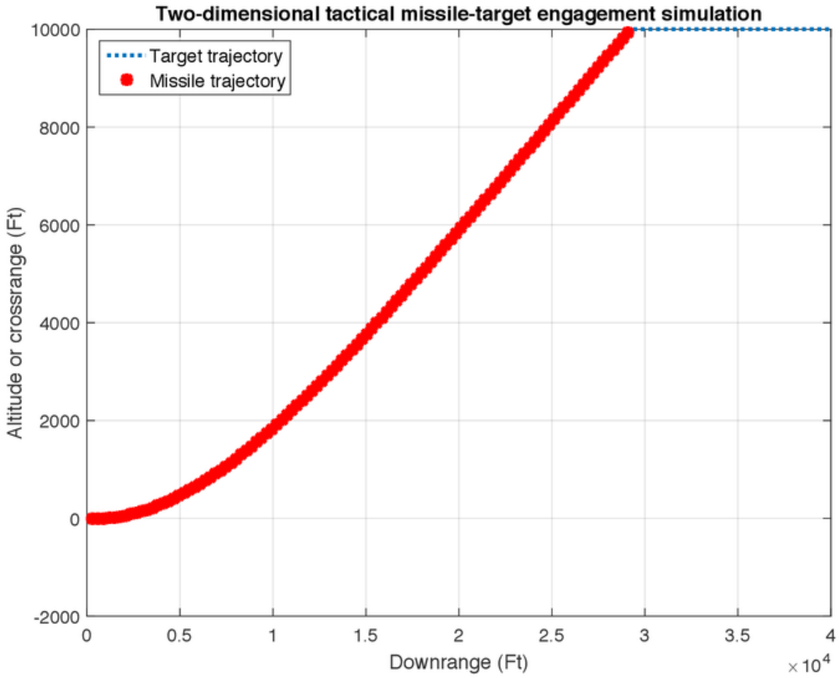


Figure 3.8: Trajectory of the target and attacker in case of zero target maneuver with heading error=-20 and $N' = 5$.

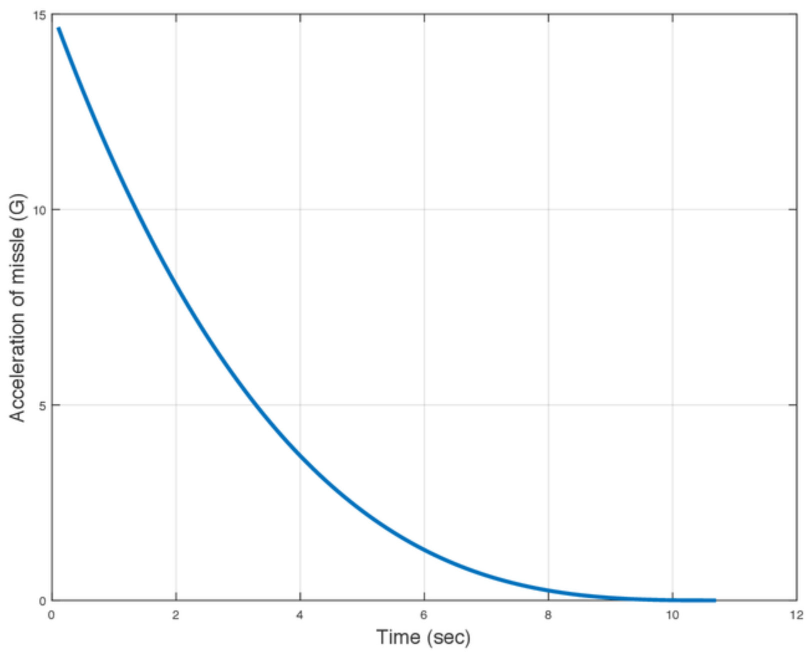


Figure 3.9: Missile acceleration in case of zero target maneuver with heading error=-20 and $N' = 5$.

3.1.1.2 Constant Target maneuver

In this case the target does some effort to escape from the attacker, in the form of constant acceleration (in our example target acceleration = 5 g). We see from Fig. 3.11 and Fig. 3.14 that a higher effective navigation ratio yields less acceleration to hit maneuvering target, and causes the missile to lead the target slightly more than a lower effective navigation ratio does, as we see from Fig. 3.10 and Fig. 3.13.

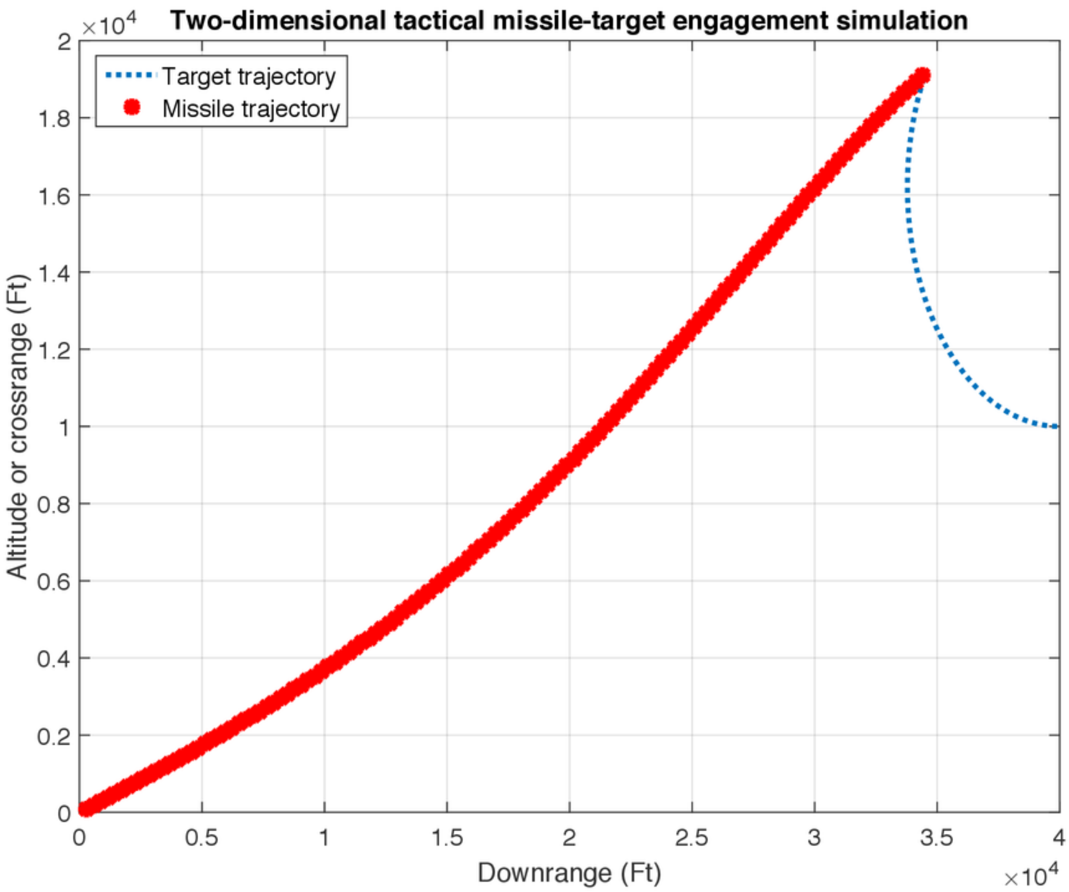


Figure 3.10: Trajectory of the target and attacker in case of constant target maneuver=5G with heading error=0 and $N' = 3$.

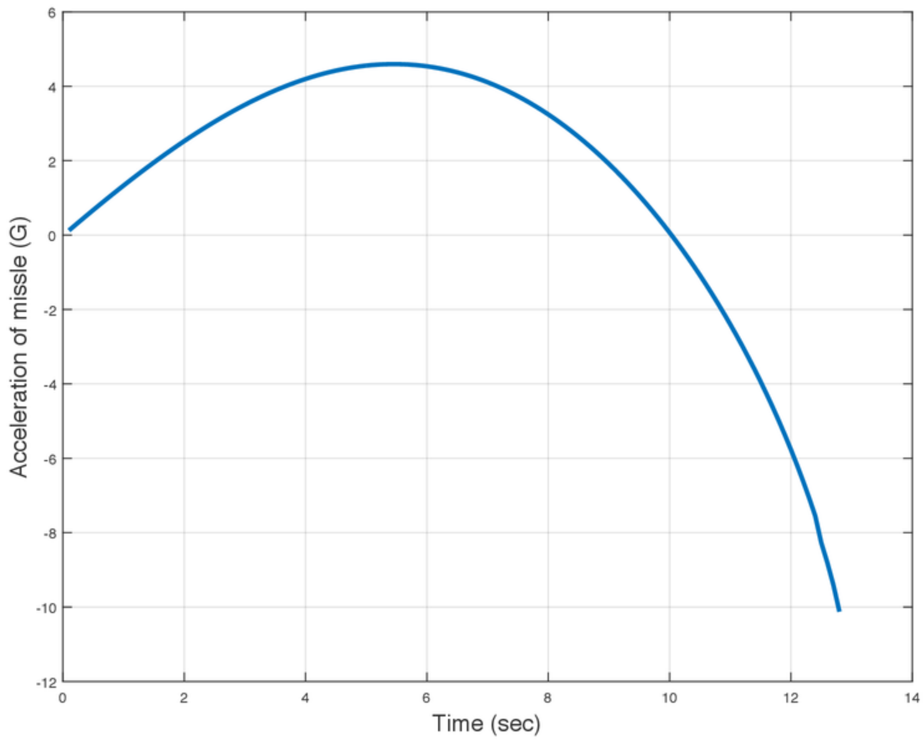


Figure 3.11: Missile acceleration in case of constant target maneuver=5G with heading error=0 and $N' = 3$.

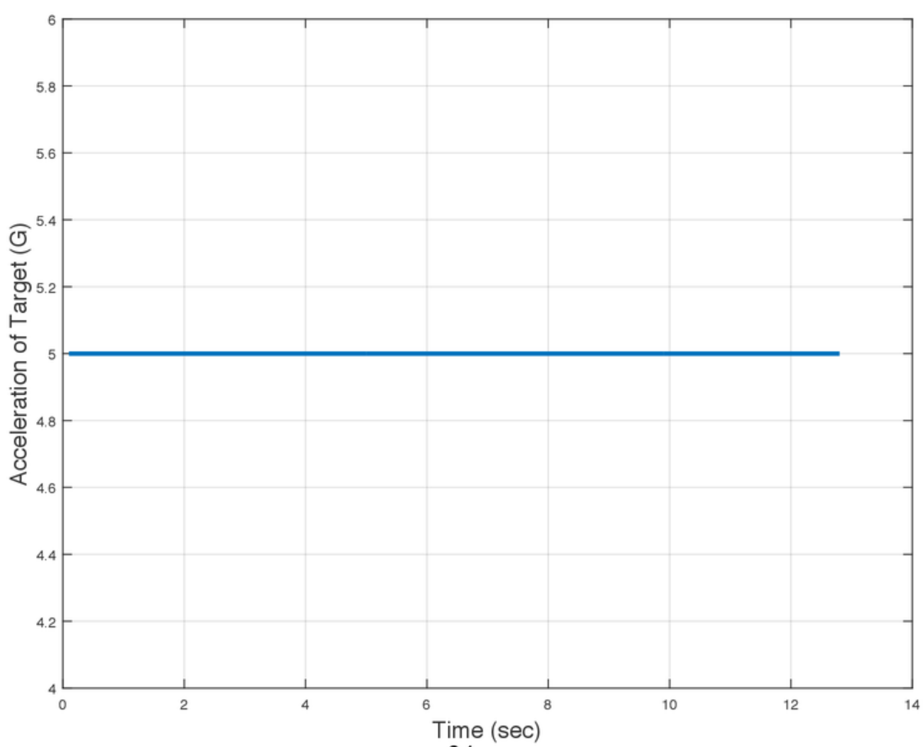


Figure 3.12: Target acceleration in case of zero target maneuver with zero heading error and $N' = 3$.

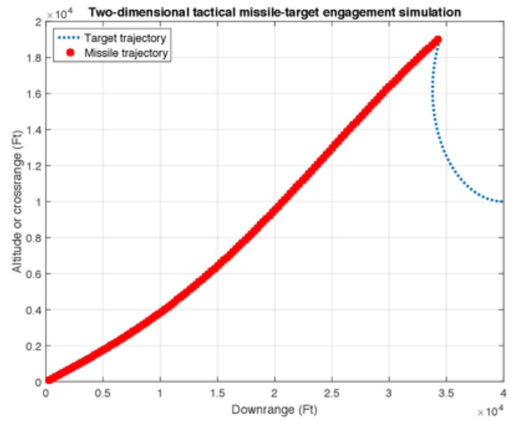


Figure 3.13: Trajectory of the target and attacker in case of constant target maneuver=5G with heading error=0 and $N' = 5$.

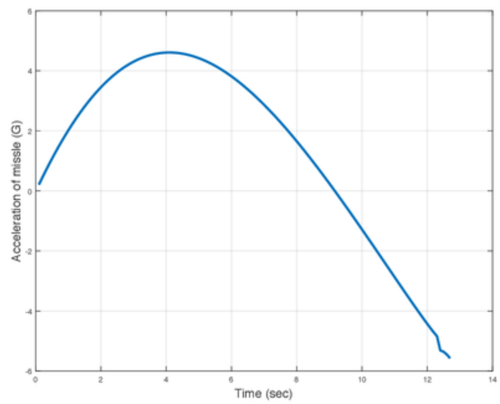


Figure 3.14: Missile acceleration in case of constant target maneuver=5G with heading error=0 and $N' = 5$.

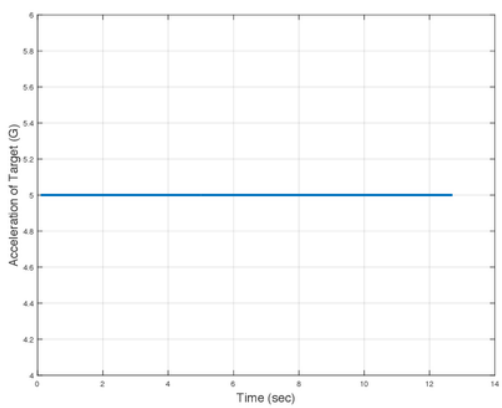


Figure 3.15: Target acceleration in case of zero target maneuver with zero heading error and $N' = 5$.

3.1.1.3 Polynomial Target maneuver

Now we will make the Target maneuver as a polynomial in the form

$$f(t) = c_0 + c_1T + c_2T^2 + c_3T^3 + ... + c_NT^N \tag{3.1}$$

where T is time, c_i's are unknown coefficients of the polynomial and N is the polynomial degree. In our simulation we choose N reasonably $3 \rightarrow 5$ and the coefficients are randomly generated then we solve the proportional navigation equations to get values for two important parameters, namely the miss distance and the command acceleration for the Missile. Then, we calculate the cost function depends on these two parameters, as our target is to maximize miss distance to make the target safe, and maximize command acceleration for the missile, so it bleeds energy before it catches the target. We will do this process many times till the number of runs we need. The next figures shows the trajectories of the Target and Missile for N=3,4 and corresponding missile and target accelerations for each case. In the first case (N=3), as we see from Fig. 3.18 the target have to exert $50G$ to escape from the attacker, these amount of G'es is very high and the pilot could not withstand all this amount. Also, form Fig. 3.17 the attacker have to exert 120 G, this will not happen and the missile's energy will bleed at 100 G after 5 sec. Here the simulation did not till us a complete vision about what will happen. We have to take in consideration the amount of G's that the pilot could resist, so we will use genetic algorithm technique to generate the polynomial because it is easy to control the maximum target acceleration in this method. The second case (N=4), is hypothetical case because the pilot will die in less than 1 sec. If the technology improved and the scientists invent a pilot suit that could resist these huge amount of G'es, then we can take this solution in consideration.

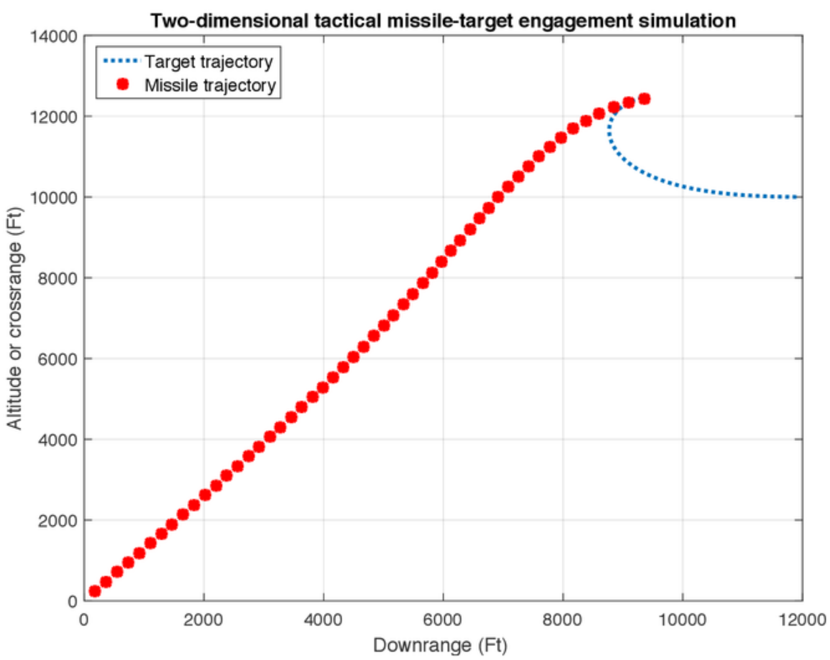


Figure 3.16: Trajectory of the target and attacker in case of polynomial of degree N=3 target maneuver with zero heading error and $N' = 3$.

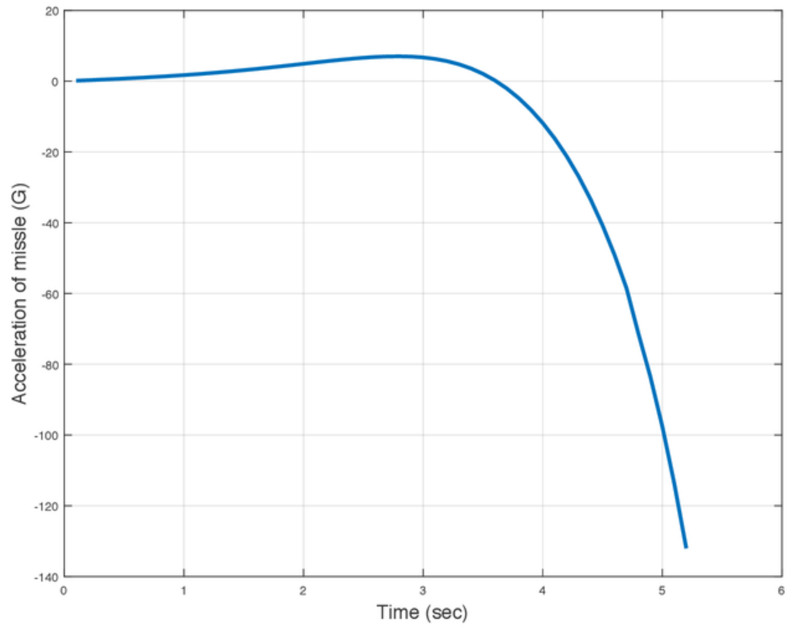


Figure 3.17: Missile acceleration in case of polynomial of degree N=3 target maneuver with zero heading error and $N' = 3$.

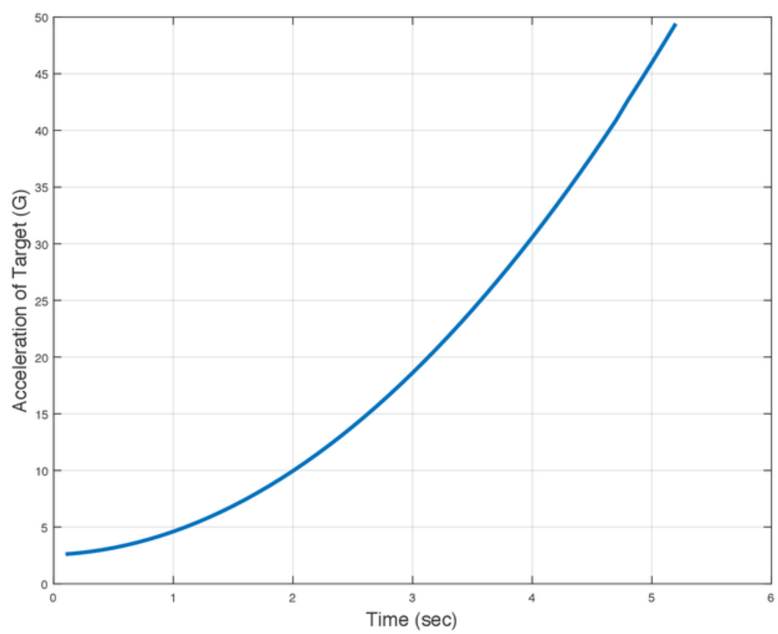


Figure 3.18: Target acceleration in case of polynomial of degree N=3 target maneuver with zero heading error and $N' = 3$.

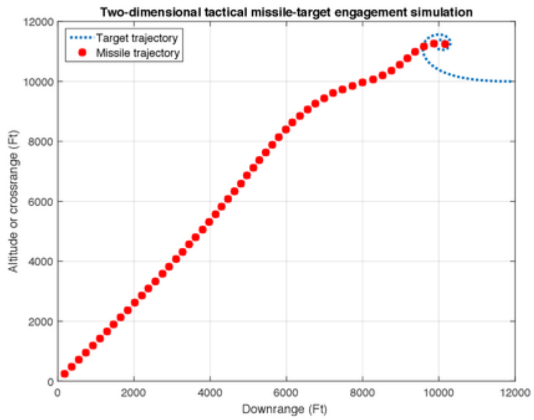


Figure 3.19: Trajectory of the target and attacker in case of polynomial of degree N=4 target maneuver with zero heading error and $N' = 3$.

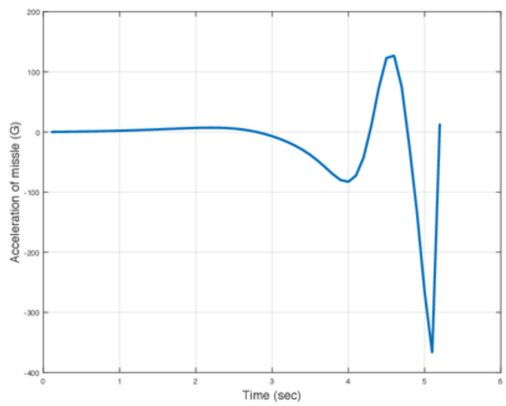


Figure 3.20: Missile acceleration in case of polynomial of degree N=4 target maneuver with zero heading error and $N' = 3$.

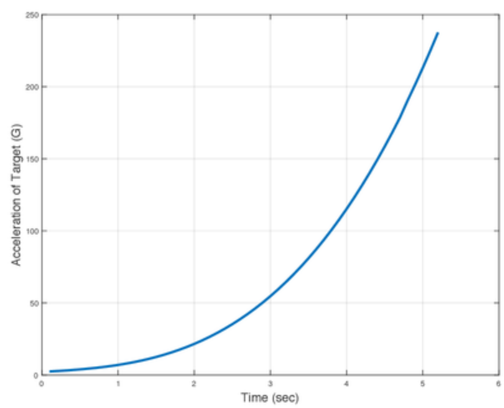


Figure 3.21: Target acceleration in case of polynomial of degree N=4 target maneuver with zero heading error and $N' = 3$.

3.1.1.4 Trapezoidal Target maneuver

Now we will make the Target maneuver as a trapezoidal function (as a general case of tooth and square maneuvers) illustrated in Fig. 3.22

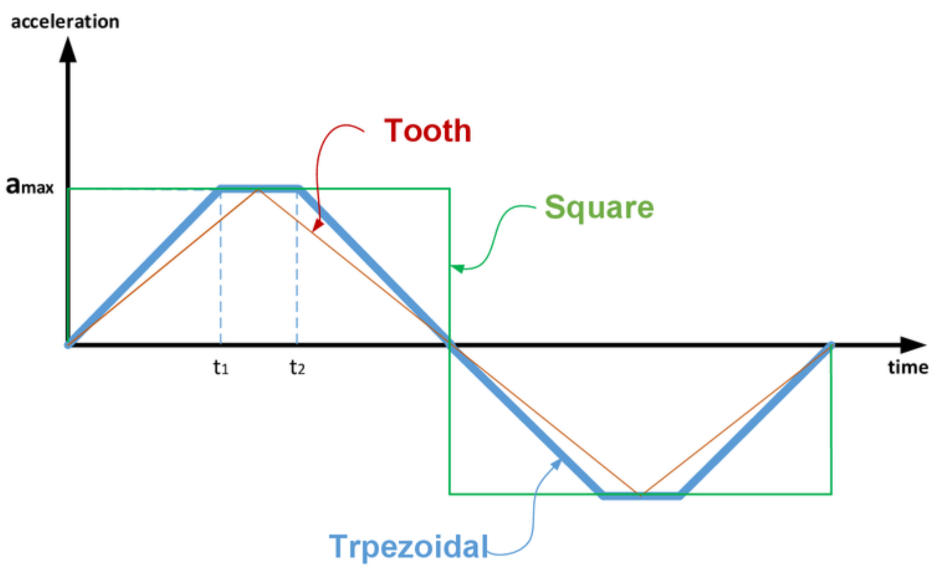


Figure 3.22: Trapezoidal Target maneuver.

In the simulation we choose the value for maximum acceleration. The time of beginning the ramp (which is 0), and the time at the end of the descent, t_1 and t_2 are variables.

We have two cases: **First case:** the maneuver is *symmetric* so we have two extreme cases.

- when t_1 is equal to zero (there is no ascend ramp) so t_2 is at the middle of the maneuver; this leads to *square* target maneuver.

- when t_1 equals t_2 at the quarter of the maneuver; this leads to *tooth* target maneuver.

Varying t_1 and t_2, starting from the square case till the tooth case, until we find the optimum solution based on or cost function which depends on the values of the miss distance and Missile acceleration.

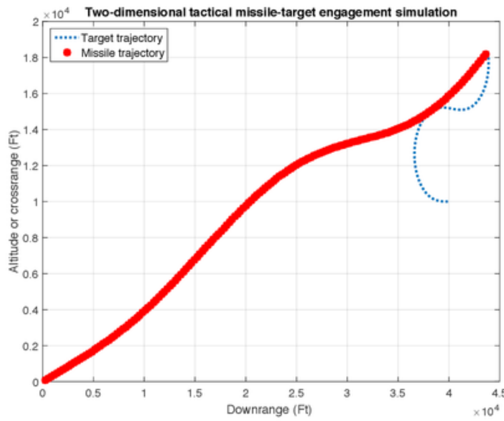


Figure 3.23: Trajectory of the target and attacker in case of symmetric trapezoidal target maneuver with zero heading error and $N' = 5$.

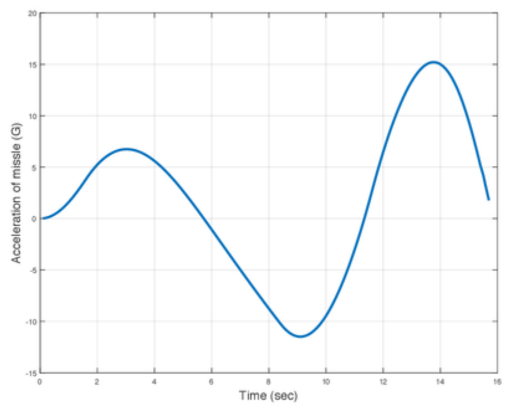


Figure 3.24: Missile acceleration in case of symmetric trapezoidal target maneuver with zero heading error and $N' = 5$.

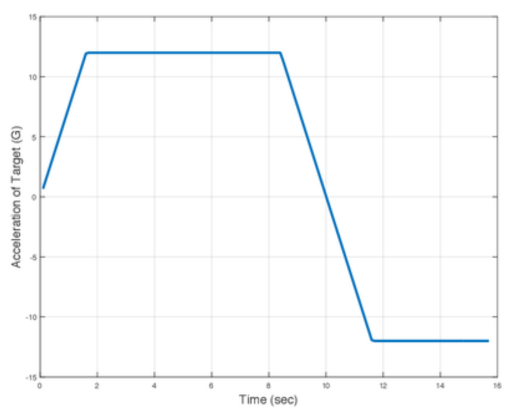


Figure 3.25: Target acceleration in case of symmetric trapezoidal N=4 target maneuver with zero heading error and $N' = 5$.

Second case: the maneuver is *not symmetric* so we put a first value for t_1 (starting with zero) and varying t_2 till we scan all the domain, then change t_1 slightly and scan all the domain again with t_2 till we find the optimum solution.

What about if the optimum solution for the target maneuver did not have Identical halves? and all the slops are not related to each other. We will try to figure that using genetic algorithm in the next section.

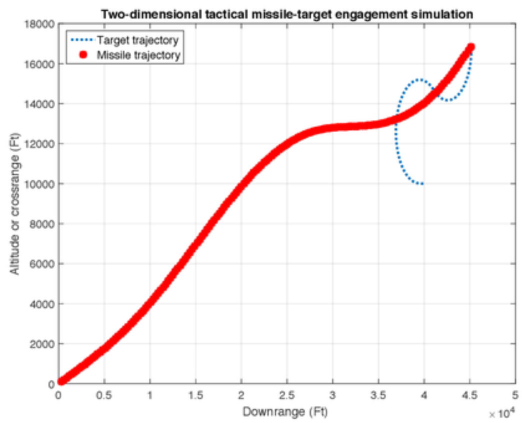


Figure 3.26: Trajectory of the target and attacker in case of trapezoidal target maneuver with zero heading error and $N' = 5$.

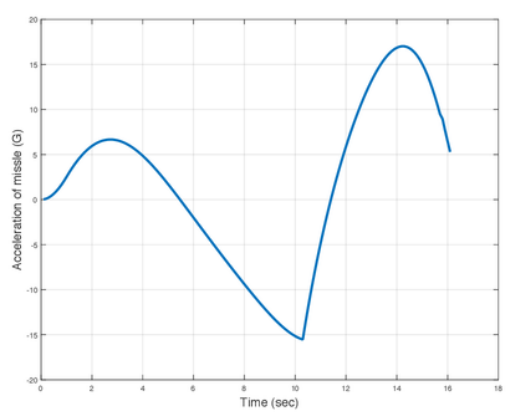


Figure 3.27: Missile acceleration in case of trapezoidal target maneuver with zero heading error and $N' = 5$.

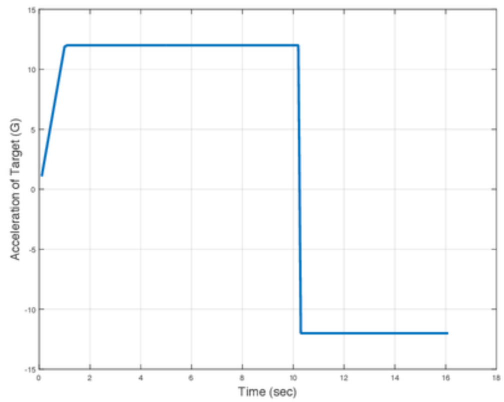


Figure 3.28: Target acceleration in case of trapezoidal N=4 target maneuver with zero heading error and $N' = 5$.

3.2 Guidance Toolbox

All the previous results could be generated from a simple GUI (graphic user interface), (Fig.3.29). The inputs for this GUI are the locations and velocities for the Target (plane) and the Attacker (missile). Then the user have to choose the guidance law (this will be as a future work, till now we have only PN guidance law), which is the way that the Attacker tracking the Target.

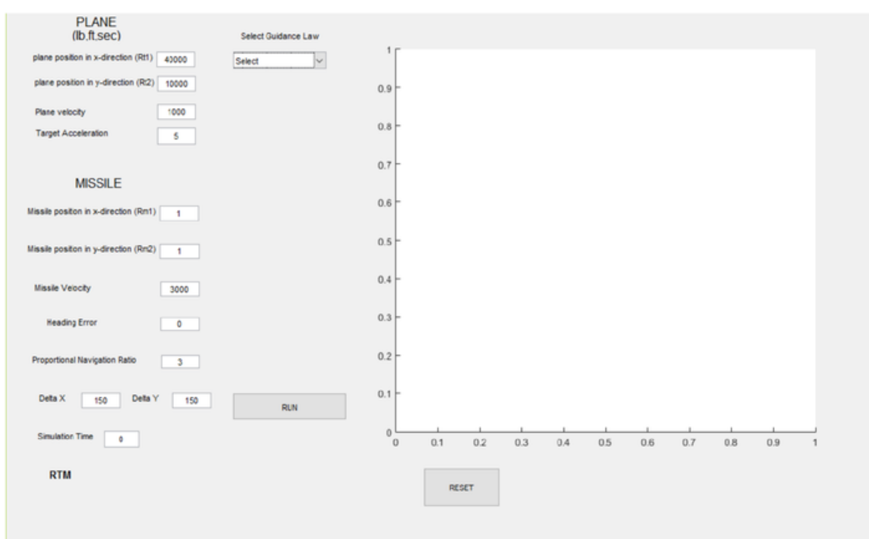


Figure 3.29: Guidance Toolbox

If the user chooses the PN guidance law (Fig.3.30) , there will be more options be available, Now he could select the type of escape maneuver:

- Polynomial

- Trapezoidal
- Symmetric Trapezoidal

and the result will be optimized escape maneuver.

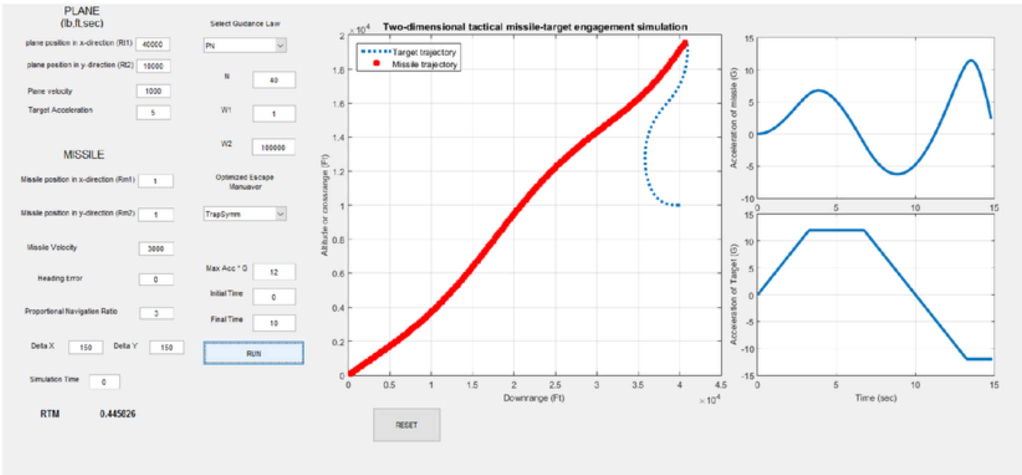


Figure 3.30: Guidance Toolbox options when PN selected

3.3 Genetic Algorithm Solution [using simulink]

In this subsection we will use the power of blocks in Simulink to solve the differential equations in section 2.4.1.1 in an easy way (Fig. 3.31). We will make these equations as a main block (Fig. 3.32) and as we change the inputs, we get the same results. The advantage of using simulink is to use the main block with an alternative optimization technique (like genetic algorithms) and we will simulate two cases, polynomial and trapezoidal. Now let us have a brief introduction to the genetic algorithm technique.

3.3.1 Introduction to Genetic Algorithm technique

Genetic algorithm is a *stochastic population-based* search method. Genetic Algorithms (GAs) are search algorithms based on the mechanics of the natural selection process (biological evolution). The most basic concept is that the strong tend to adapt and survive while the weak tend to die out. That is, optimization is based on evolution, and the ”Survival of the fittest” concept. GAs have the ability to create an initial population of feasible solutions, and then recombine them in a way to guide their search to only the most promising areas of the state space. Each feasible solution is encoded as a chromosome (string) also called a genotype, and each chromosome is given a measure of fitness via a fitness (evaluation or objective) function.

- The fitness of a chromosome determines its ability to survive and produce offspring.
- A finite population of chromosomes is maintained.

- GAs use probabilistic rules to evolve a population from one generation to the next. The generations of the new solutions are developed by genetic recombination operators:
  - Biased Reproduction: selecting the fittest to reproduce.
  - Crossover: combining parent chromosomes to produce children chromosomes.
  - Mutation: altering some genes in a chromosome.

Most Important Parameters in GAs:

- Population Size
- Evaluation Function
- Crossover Method
- Mutation Rate

Use Genetic Algorithms:

- When facing a problem of local minima.
- When a good fitness function is available.
- When a near-optimal, but not optimal solution is acceptable.
- When the state-space is too large for other methods.

3.3.2 COMPONENTS OF GENETIC ALGORITHMS

Representation (definition of individuals):

Objects forming possible solution within original problem context are called phenotypes, their encoding, the individuals within the GA, are called genotypes. The representation step specifies the mapping from the phenotypes onto a set of genotypes. Candidate solution, phenotype and individual are used to denote points of the space of possible solutions. This space is called phenotype space. Chromosome, and individual can be used for points in the genotye space. Elements of a chromosome are called genes. A value of a gene is called an allele. The GA works with a coding of the parameter rather than the actual parameter. Depending on the particular problem, the coding may be Boolean-valued, integer-valued, real-valued, complex-valued, vector-valued, symbolic valued, or multiple-valued.

Evaluation function:

This is just the cost or the objective function which represent what we want to maximize. For minimization problem use the reciprocal (i.e. 1/f).

Population:

The role of the population is to hold possible solutions. A population is a multiset of genotypes. In GA, the population size is (almost always) constant. The bigger the size, the better the final solution, and the larger the execution time.

Parent Selection Mechanism:

The role of parent selection (mating selection) is to distinguish among individuals based on their quality to allow the better individuals to become parents of the next generation. Parent selection is probabilistic. Thus, high quality individuals get a higher chance to become parents than those with low quality. Nevertheless, low quality individuals are often given a small, but positive chance; otherwise the whole search could become too greedy and get stuck in a local optimum.

Variation Operators:

The role of variation operators is to create new individuals from old ones. Variation operators form the implementation of the elementary steps with the search space.

Mutation Operator:

A unary variation operator is called *mutation*. It is applied to one genotype and delivers a modified mutant, the child or *offspring* of it. In general, mutation is supposed to cause a random unbiased change.

Crossover Operator:

A binary variation operator is called recombination or crossover. This operator merges information from two parent genotypes into one or two offspring genotypes. Similarly to mutation, crossover is a stochastic operator: the choice of what parts of each parent are combined, and the way these parts are combined, depend on random drawings. The principle behind crossover is simple: by mating two individuals with different but desirable features, we can produce an offspring which combines both of those features.

Survivor Selection Mechanism:

The role of survivor selection is to distinguish among individuals based on their quality. In GA, the population size is (almost always) constant, thus a choice has to be made on which individuals will be allowed in the next generation. This decision is based on their fitness values, favoring those with higher quality. As opposed to parent selection which is stochastic, survivor selection is often deterministic, for instance, ranking the unified multiset of parents and offspring and selecting the top segment (fitness biased), or selection only from the offspring (age-biased).

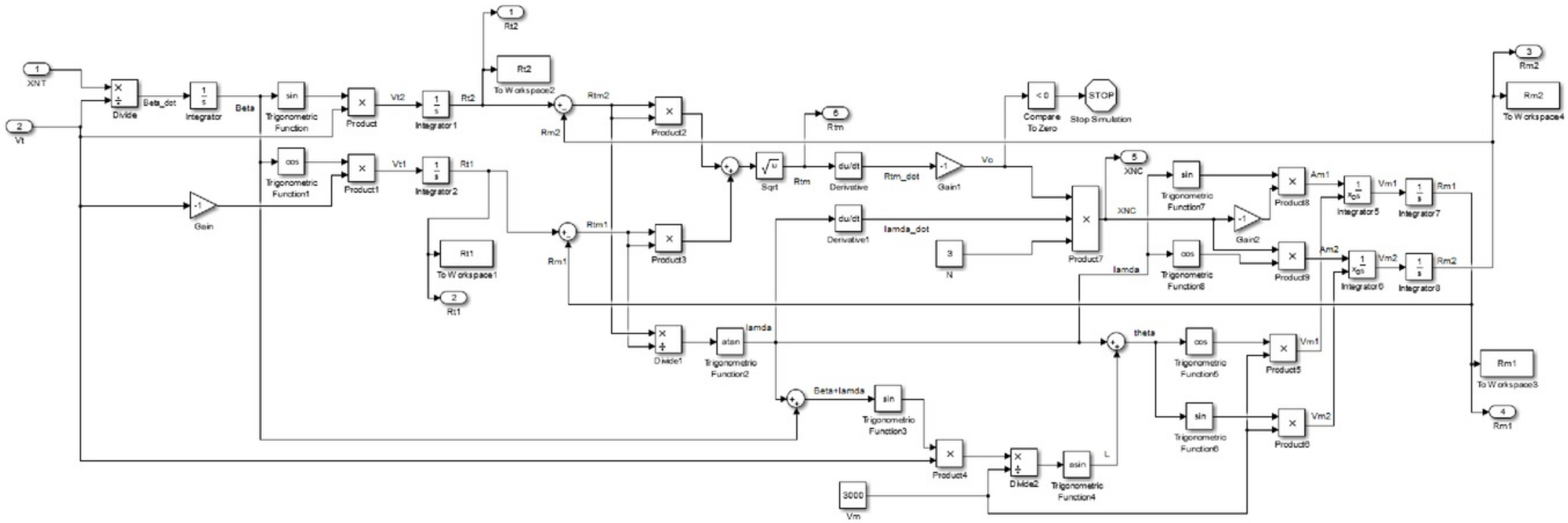


Figure 3.31: The Simulink model for proportional navigation equations in sec 2.4.1.1

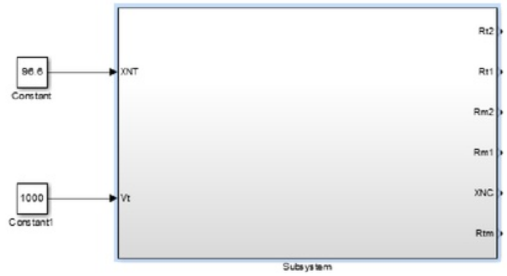


Figure 3.32: The Simulink main block for solving proportional navigation equations in sec 2.4.1.1

3.3.3 Results

We will use the main block we have created with the genetic algorithm toolbox in MATLAB in the same manner we have done, let the target maneuver unknown, then it is required to find it using genetic algorithm to maximize miss distance and missile acceleration. We have two cases of the target maneuver:

3.3.3.1 Polynomial Target Maneuver

let the target maneuver ba a polynomial with unknown coefficients, then it is required to find the values of these coefficients using genetic algorithm toolbox in Matlab (Fig. 3.33) to maximize miss distance and missile acceleration. The result for the best fitness values are in Fig.3.34. The optimized target acceleration and trajectory simulation are in figures 3.35 and 3.36.

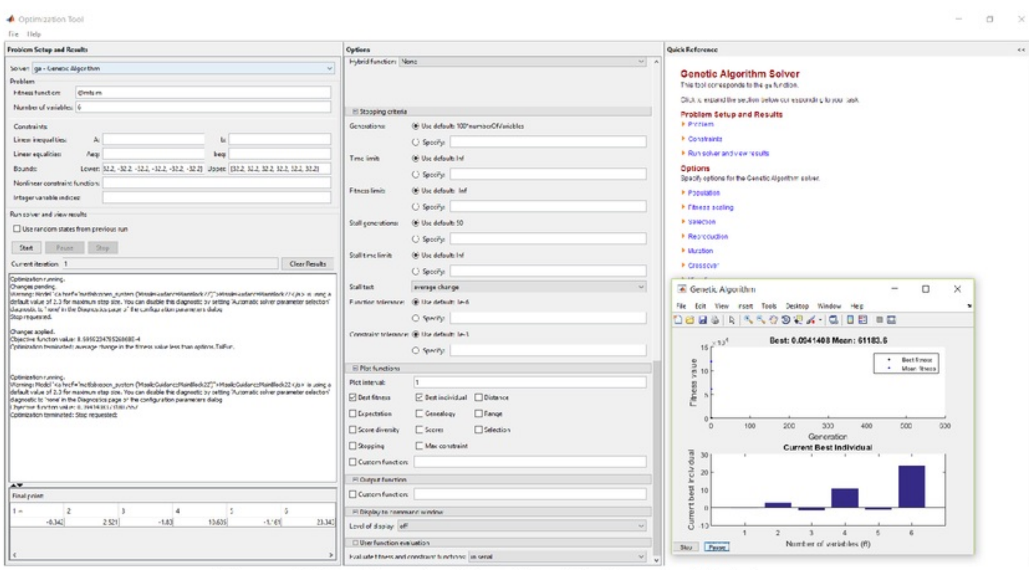


Figure 3.33: Genetic Algorithm Toolbox in Matlab.

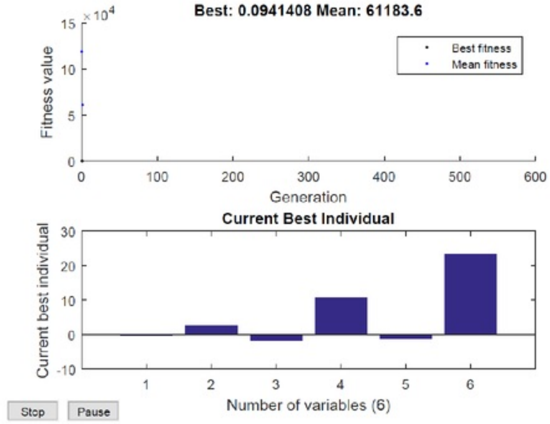


Figure 3.34: Best fitness values

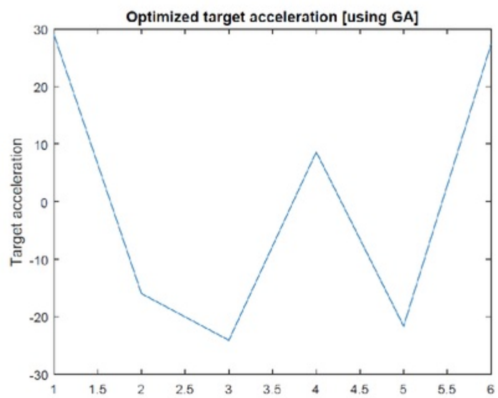


Figure 3.35: Optimized target acceleration using GA.

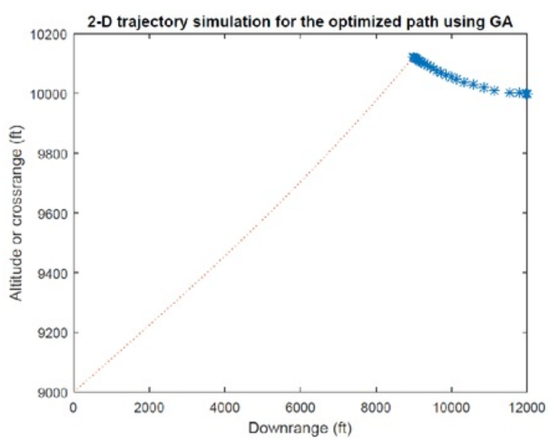


Figure 3.36: Optimized trajectory using GA.

Chapter 4: A new game-based methodology for discovering optimal escape maneuver

This chapter represents a novel game-based methodology for the possible discovery of aircraft optimal escape maneuver against an attacking missile. This discovery could be achieved through a mathematically-correct game of target-attacker where a human player controls the target, trying to evade the missile. The game is distributed to some ordinary persons (typically young kids) who are asked to play the game and to get the highest scores possible. Best escape maneuvers are collected, analyzed, and optimized, to find the optimal human-based escape maneuver. The game is based on 2D point-mass models for target and attacker and a proportional navigation law for missile guidance. The game is developed using Unity, a free cross-platform game engine. The preliminary results obtained suggest the production of enhanced versions of the game with more desirable outcomes, *viz.*, to utilize collective human-brain capability in producing a new guidance law which could possibly compete with existing ones.

4.1 Introduction

Missile avoidance is a very important element of aircraft survivability against attackers in air combat and defense. If the missile guidance law (PN for example) and its parameters (speed, location, etc.) are known, the best avoidance maneuver is the solution of an optimal control problem which can be solved using different optimization techniques such as genetic algorithms (GA) or particle swarm optimization (PSO). In this work, we try to explore the hypothesis that optimal avoidance strategy can be generated through a completely different approach. A human based approach where humans (most probably kids) are given the problem of missile avoidance in a sort of a game and they are asked to play the game and to get the highest score possible.

In this context, we try to benefit from the emergence of novel paradigms for games played by humans versus computers. These paradigms include:

- **Competitions** between humans and computers in games of increasing difficulty, including the games of Chess [84], Arimaa [85], Jeapordy! [86], and Go [87]. Typically, the rivalry is between a single human champion (such as Kasparov) against a state-of-the-art computer (such as Deep Blue) limited to a specific domain (such as Chess). The sole purpose of the competition is to decide whether the human or the computer wins more games within the same match.

- The **gamification** phenomenon [88, 89, 90], which is the use of game design elements in non-game or learning contexts, a trend related to human-computer interactions in the form of serious games, pervasive games, alternate reality games, or playful design.

- The paradigm of **games with a purpose** [91, 92, 93, 94, 95], which aims to utilize the billions of hours spent (wasted!) by contemporary humans in playing computer games. This paradigm channels game playing into useful work by directing people playing computer games to simultaneously solve large-scale problems without consciously knowing about this and, hence, without losing the element of fun or entertainment. Many large-scale open problems can be solved using collective human brainpower in this unique way. Examples include language translation, monitoring of security cameras, improving Web search, and text summarization. With the paradigm of games with a purpose, hundreds of millions of people can collaborate on the same problem via the Internet.

- **Human Computation** [96], which is the idea of using human brain processing power to perform tasks that computers cannot yet perform or solve problems still intractable for computers, usually in an enjoyable manner. Human Computation is also viewed as systems of computers and large numbers of humans that work together in order to solve problems that could not be solved by either computers or humans alone.

- **Crowdsourcing** [97], which is the act of taking a job traditionally performed by a designated agent (an employee) and outsourcing it to a generally large group of people in the form of an open call. Whereas human computation replaces computers with humans, crowdsourcing replaces traditional human workers with members of the human public.

This chapter introduces yet another novel paradigm of a game played by humans versus computers. This paradigm resembles that of games with a purpose, but it will neither handle a large-scale open problem nor require a dramatically huge number of human players. The current paradigm might also be viewed as an offshoot of that of human computation, wherein we relax the requirement of dealing with tasks that computers cannot yet perform, and relegate it to handling somewhat sophisticated and complex (but not intractable) tasks. The current paradigm is to solve problems requiring mathematically-elaborate game-theoretic techniques (that already have many automated algorithmic solutions) by requesting a relatively small number of people (50 persons) to play a computer game, constructed to result in a correct solution and, at the same time, is enjoyable. The basic idea is that people will play a game to be entertained, not to solve a problemno matter how noble and glorious the objective is. The game played constitutes a human-computer competition, but unlike the Kasparov-Deep Blue one, neither the human players are necessarily highly-competent or champions, nor is the computer necessarily of super characteristics. The task encountered herein of designing an online game is much like designing an algorithm. In fact, this game must be proven correct, and its efficiency might be analyzed, so that more efficient versions should always be sought so as to supersede less efficient ones. Instead of using a silicon processor, these algorithms run on a processor consisting of the brains of ordinary humans interacting with a computer.

The problem considered herein is an important pursuit-evasion problem that involves two agents, the target (aircraft) and the attacker (missile) [98, 99, 100, 101, 102]. The attacker missile pursues the target aircraft, and a dynamic or differential game arises in which the target tries to maximize the separation between itself and the attacker, while the attacker tries to minimize this separation. We present a game-based methodology to find the optimal escape maneuver for the target (represented by a human player) against the attacking missile (represented by the computer). The missile pursuit the target according to the 2D guidance law of proportional navigation. The cost function used maximizes the missile acceleration, time to interception and the miss distance. We construct a mathematically-correct game of target-attacker and let many ordinary people play it individually from the target side, and assign a score to every player that is proportional to the cost function. We find the best escape maneuver by collecting and analyzing data of the escape maneuver of the human players. The game is developed using Unity, a free cross-platform game engine.

Since our game is a skill-and-action game, it is characterized by real-time play, and it would definitely benefit of heavy emphasis on graphics and sound and also of the use of joysticks or paddles rather than just a keyboard [103]. Moreover, the primary skills demanded of the players of our game are psychomotor skills including hand-eye coordination and fast reaction or response time. The game is somewhat related to combat games, but it is not a combat game per se, since it does not involve a direct mutual violent confrontation between two opponents. The human player, placed in the role of the aircraft, must avoid being captured or hit by the attacking computer-controlled missile, but cannot benefit of the wisdom that retaliatory attack is the best way of evasion or defense. In fact, the game is a purely defensive one, in the sense that the player never has the opportunity to attack the enemy missile. The game seems somewhat to be a race game since it involves some sort of race between two opponents, but this race is not a straightforward one and it encompasses certain strategic elements.

The present approach can be evaluated as a sort of Reinforcement Learning (RL), a major technique of soft computing for adaptive optimal control of nonlinear systems [104, 105, 106], where a player takes a long series of actions (moving the aircraft up and down) before receiving the reward (the final score). So, it is hard to decide, mathematically, which of the actions were responsible for the eventual payoff.

In the remainder of this Chapter, we describe the game engine used explain how our game is built. The missile model to be supplied to the computer was earlier described in detail in Section 2.1. We also report preliminary results of our new methodology.

4.2 Game description and design

This Section briefly introduces the game engine Unity and explains the first version of our game (henceforth referred to as $Evasion-1$). Game design is an artistic process as well as a technical one, and hence its essence is to manage the integration of these two dissimilar processes [103]. The purpose of the design is to create outlines of three interdependent structures: the I/O structure, the game structure, and the program structure. The I/O structure constitutes the language for communicating information between the computer and the player. The game structure is a means of figuring out how to distill the goal and topic of the game into a workable system, and identifying some key element or elements around which the game is built. The program structure is the organization of main code, subroutines, interrupts, and data that make up the entire program. All three structures must be created simultaneously, for they must work in harmony. Decisions primarily relating to one structure must be checked for their impacts on the other structures. Once all three structures are made to work in compatibility with one another, it is necessary to evaluate the overall design for the most common design flaws that plague games and to assure that the design satisfies the design goals. The design phase is followed by the programming phase. Programming itself is straining, strenuous and tedious work, requiring attention to detail more than anything else. It might prove to be the hardest of all phases, though its inherent difficulty is somewhat eased by the availability of game engines such as Unity, which is free and simple-to-use beside being able to work on almost any computer and allowing any novice user to download and use its program, and to avoid some of the complex programming techniques through utilizing pre-constructed building blocks.

4.2.1 Introduction to Unity

Unity is a cross-platform game engine (Fig. 4.1), used to develop video games for PC, consoles, mobile devices and websites. It works with the C-sharp programming language, and can be downloaded free of charge from (https://unity3d.com/), which is the official site of its developer (Unity Technologies).

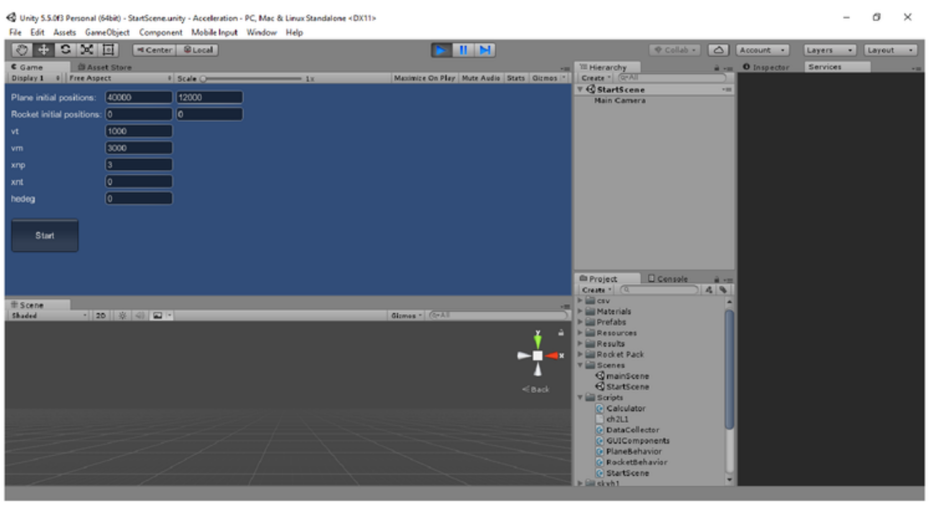


Figure 4.1: Unity game engine interface

4.2.2 Methodology of the game

Figure 4.2 illustrates a typical snapshot or computer screen for Evasion-1, showing the two objects that comprise the game, viz., the plane and the missile. The background has stationary shapes (such as clouds) that set a frame of reference allowing the human player to visually estimate the speeds of the plane and missile. The player controls the acceleration of the target (XNT) using the Up/Down arrows on the keyboard to increase/decrease this acceleration. Missile motion is computer-controlled according to proportional navigation(in sec. 2.1). The game has a few adjustable parameters (see Fig. 4.1), and hence might be adapted for other pursuit-evasion problems.

Figure 4.2: Unity game for simulating Target-Attacker engagement

Each game consists of two scenes:

1. Start or initial scene.
2. Play scene, updated each frame per second.

In the start scene, there are data fields such as position and velocity, which are destroyed at the end of the scene. Therefore, it is necessary to save all needed information in a file called "player prefs". The play scene consists of several objects (currently three). Each object usually has its own script, which must contain two points:

- Start: initialize.
- Update: every frame.

The present three objects of the game are:

1. **Plane:** with a script containing some commands to control the plane by increasing and decreasing the target acceleration by the upward and downward arrows, respectively,
2. **Missile:** which does not contain a script of its own, since the equations controlling its behavior are within the script of the fictitious "Data collector" object,
3. **Object "Data collector":** whose script consists of:
   - Start
     (a) initialization of the variables in the equations.
     (b) Setting the location of the real objects (plane and missile).
   - Update
     (a) Getting plane location,
     (b) Executing PN equations,

(c) Updating rocket location (according to PN equations),

(d) Updating information to be printed to Excel,

(e) Checking the condition for game termination, and if true, reloading the start scene.

4.3 Preliminary Results and Analysis

Our experience with Evasion-1 demonstrates that the game is working correctly. Figures 4.3-4.5 illustrate some of our preliminary results, obtained by the player with the best score so far. The target acceleration curve in Fig. 4.4 exhibits a Barrel-Roll behavior, an aerial maneuver in which an airplane makes a complete rotation on both its longitudinal and lateral axes, causing it to follow a helical path, approximately maintaining its original direction. This maneuver helps a target to force an attacker to fly out in front, i.e., to overshoot. The results is very encouraging, indeed, since a human player who is definitely unaware of the technicalities of evasion techniques, and who is playing the game just for fun, was remarkably capable of producing an evasion policy, that is considered one of the best in the literature (see, e.g. Zarchan). The results suggest the production of enhanced versions of the game with more desirable outcomes. Specifically, in our next version (Evasion-2), the plane will no longer be destined to be intercepted by the missile, since we will impose the restriction that the missile has a limited reservoir of fuel that could be drained as a result of clever target maneuver. The ultimate goal is to test human brain capability to work collectively to produce a new guidance law which could possibly compete in accuracy with the existing ones, while surpassing them in the simplicity of the way it is produced.

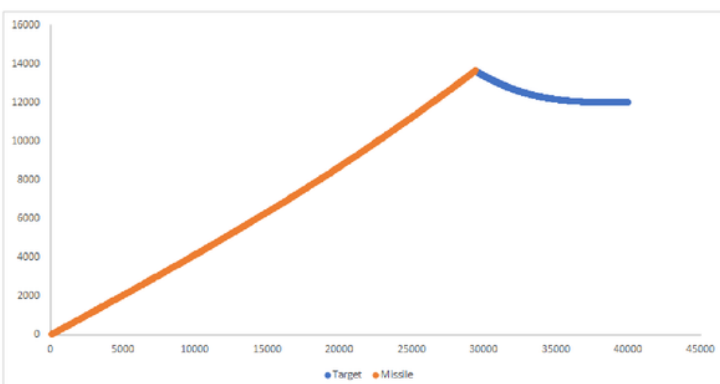


Figure 4.3: Target-Attacker 2D trajectory. The target and missile speeds are 3000 m/s,1000 m/s respectively.

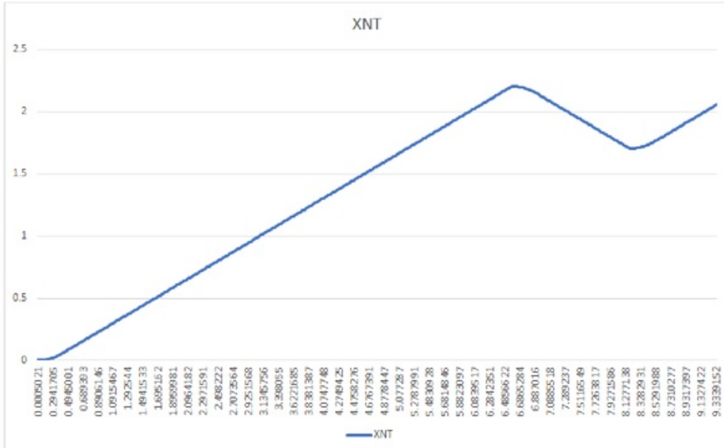


Figure 4.4: Optimized target acceleration versus time.

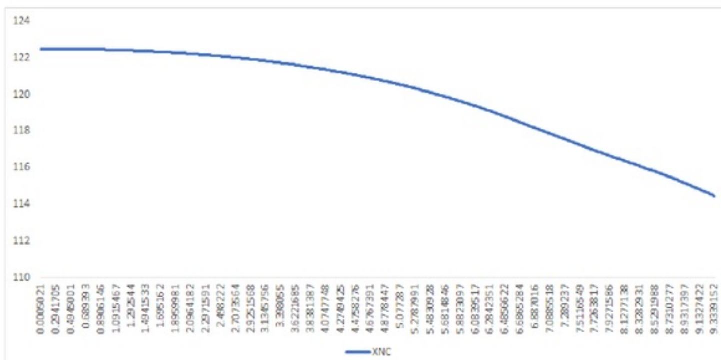


Figure 4.5: Missile acceleration versus time.

Part III

The TARGET-ATTACKER-DEFENDER PROBLEM

Chapter 5: TARGET-ATTACKER-DEFENDER PROBLEM

5.1 Problem Statement

In this part we deal with the three-agent pursuit-evasion problem where each agent has a specific role. A two agent team consists of a Target (aircraft) and a Defender (missile) cooperating in opposition to an Attacker (missile). The Target tries to evade the Attacker and avoid being hit by it. The Defender cooperates with and assists the Target by trying to intercept (hit and destroy) the Attacker before the latter captures the Target.

5.2 Assumptions, Notation, and Nomenclature

Assumptions

1. The speeds V_T, V_A, and V_D of the Target, Attacker, and Defender are constant.

2. The Attacker missile is faster than the Target aircraft, i.e., $\alpha \equiv \dfrac{V_T}{V_A} < 1$ (In the case $\alpha \geq 1$, the Target is guaranteed to survive by following an optimal strategy, even without assistance from the Defender).

3. There are three distinct cases for the ratio of the speed of the Attacker to that of the Defender $\gamma = \dfrac{1}{\beta} = \dfrac{V_A}{V_D}$

   - $\gamma < 1$ (fast Defender) discussed earlier in Garcia et el. [48], and extended, expounded, simplified and exposed herein.
   - $\gamma = 1$ (same speed Defender) discussed in Garcia et el. [47, 49], and extended, expounded, simplified and exposed herein.
   - $\gamma > 1$ (slow Defender), which is a novel case, discussed herein for the first time, and unified with the two previous cases.

4. The Defender intercepts the Attacker if their separation becomes zero (point capture).

5. The optimal trajectories of the three agents are straight lines.

6. A Cartesian frame is attached to the initial positions A and D of the Attacker and Defender in such a way the X-axis is the infinite extension of the straight segment $\overline{AD}$ and the Y-axis is the perpendicular bisector of $\overline{AD}$.

Notation

- A: Initial position of the Attacker.
- D: Initial position of the Defender.
- T: Initial position of the Target.
- T' Terminal position of the Target, i.e., its position at the time the Defender intercepts the Attacker.
- $\alpha = \dfrac{V_T}{V_A}$ (assumed < 1, otherwise the target trivially survives).
- $\gamma = \dfrac{1}{\beta} = \dfrac{V_A}{V_D}$ (studied for a fast Defender ($\gamma < 1$), a similar Defender ($\gamma = 1$), and a slow Defender ($\gamma > 1$)).
- u, v, w: Aim points on the AD Apollonius circle by the Attacker, Target and Defender, respectively.

Nomenclature

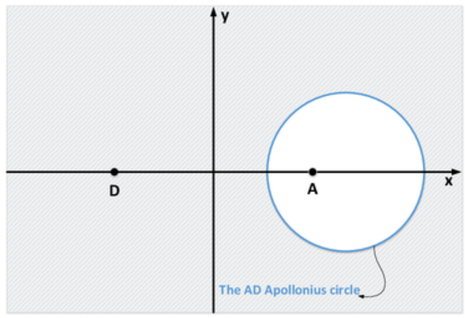
The AD Apollonius circle. The locus of a point such that the ratio of its distance from the initial positions $\boldsymbol{A} = (x_A, 0)$ and $\boldsymbol{D} = (-x_A, 0)$ of the Attacker and Defender, respectively, is a fixed ratio $\gamma = \dfrac{V_A}{V_D}$. This circle degenerates into the perpendicular bisector of $\overline{AD}$ if $\gamma = 1$. Points on the circumference of this circle are reached simultaneously by the Attacker and Defender. For $\gamma < 1$, $\boldsymbol{A}$ belongs to the interior of this circle and points within the circle are reached by the Attacker before the Defender. For $\gamma > 1$, $\boldsymbol{D}$ belongs to the interior of this circle and points within the circle are reached by the Defender before the Attacker. For $\gamma = 1$, the AD circle becomes of infinite radius and degenerates into a straight line, namely the perpendicular bisector of $\overline{AD}$. In this case, $\boldsymbol{A}$ belongs to the R.H.S. of the $XY-$plane where points are reached by the Attacker before the Defender, while $\boldsymbol{D}$ belongs to the L.H.S. of the $XY-$plane where points are reached by the Defender before the Attacker.

The TA Apollonius circle. The locus of a point such that the ratio of its distances from the initial positions $\boldsymbol{T} = (x_T, y_T)$ and $\boldsymbol{A} = (x_A, 0)$, of the Target and Attacker, respectively, is a fixed ratio $\alpha = \dfrac{V_T}{V_A}$ strictly less than 1. Naming of the AD and TA Apollonius circles herein follows a common practice in mathematical circles [71, 72, 73] and is opposite to the style used by Garcia et al. [47, 48, 49].

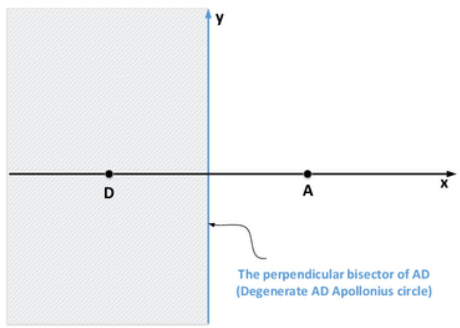
Region reachable by the Defender before the Attacker (Reachability region R_r):
For $\gamma < 1$, R_r is the exterior of the AD Apollonius circle.
For $\gamma = 1$, R_r is the L.H.S. of the $XY-$plane.
For $\gamma > 1$, R_r is the interior of the AD Apollonius circle.

Figure 5.1 illustrates this region as a shaded region to which point D belongs.

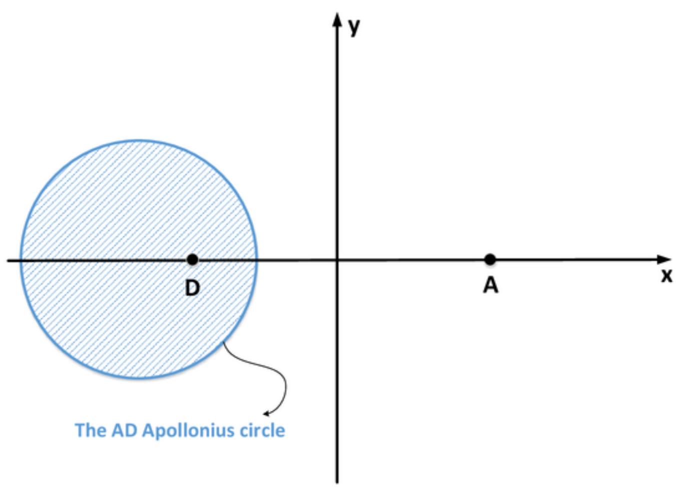
Region of guaranteed Target's escape (Escape region R_e). The escape region R_e is the set of all coordinate pairs (x, y) such that if the Target initial position $\boldsymbol{T} = (x_T, y_T)$ is inside this region, then it is guaranteed to escape the Attacker if both the Target and Defender implement their corresponding optimal strategies. This set is a strict superset of the set of



(a) $\gamma < 1$.



(b) $\gamma = 1$.



(c) $\gamma > 1$.

Figure 5.1: The reachability region R_r (one including $\boldsymbol{D}$ whose points are reached by the Defender before the Attacker) is shown shaded.

points reachable by the Defender before the Attacker ($R_e \supseteq R_r$). The boundary of this set is called a Voronoi Diagram.

The critical speed ratio $\overline{\alpha}$. A lower limit on the speed ratio $\gamma = \dfrac{V_T}{V_A}$, attained when the TA Apollonius circle is tangent to the AD Apollonius circle (or to the perpendicular bisector of $\overline{AD}$ in the degenerate case $\gamma = 1$).

Escape Condition. For a given speed ratio $\alpha = \dfrac{V_T}{V_A}$ and a given Attacker's initial position $\boldsymbol{A} = (x_A, 0)$, the escape condition is that the TA Apollonius circle (of center O_2 and radius r_2) intersects the AD Apollonius circle (of center O_1 and radius r_1). This happens for $\gamma < 1$ when

$$|\boldsymbol{O_1} - \boldsymbol{O_2}| + r_2 > r_1 \tag{5.1}$$

It happens for $\gamma > 1$ when

$$|\boldsymbol{O_1} - \boldsymbol{O_2}| - r_2 < r_1 \tag{5.2}$$

In the degenerate case of $\gamma = 1$, it is required that the TA Apollonius circle intersects the $Y-$axis, i.e.,

$$r_2 > the\ abscissa\ of\ \boldsymbol{O_2} \tag{5.3}$$

5.3 APOLLONIUS CIRCLES PERTAINING TO THE PROBLEM

A circle is the locus moving at a constant distance (called the circle's radius r) from a fixed point (called the circle's centre O). In the limit of an infinite radius ($r \longrightarrow \infty$), the circle degenerates into a straight line. Another definition of the circle is that it is the locus of a point moving such that the ratio of its distances from two fixed points A and B is a constant k. With this definition, the circle is called a circle of Apollonius (in honour of Apollonius of Perga (ca. 262-190 BC), the Great Geometer of Antiquity). In the limit ($k \longrightarrow 1$) this circle degenerates into the perpendicular bisector of $\overline{AD}$, while in the two limits ($k \longrightarrow 0$) and ($k \longrightarrow \infty$), this circle collapses to the two points A and B, respectively.

Two Apollonius circles are studied herein, namely the AD Apollonius circle, and the TA Apollonius circle. The study will be based on simple and intuitive plane-geometric arguments and will avoid the more involved treatment of analytic geometry. Figures 5.2 and 5.3 shows the Apollonius circles for a moving point P such that $\dfrac{AP}{PB} = k \neq 1$. The case $k > 1$ is shown in Fig. 5.2, while the case $k < 1$ is shwon in Fig. 5.3. The points I and E are the two special cases of P that lie on the straight line extension of the straight segment $\overline{AB}$. These points divide the straight segment $\overline{AB}$ internally and externally in the ratio k ($k \neq 1$), i.e.,

$$\boxed{\frac{AI}{IB} = \frac{AE}{EB} = k,\ \{k \neq 1\}.} \tag{5.4}$$

If we denote the position vector of a point by a bold version of its name, we can rewrite (5.4) in vectorial form as

$$(\boldsymbol{I} - \boldsymbol{A}) = k(\boldsymbol{B} - \boldsymbol{I}), \tag{5.5}$$

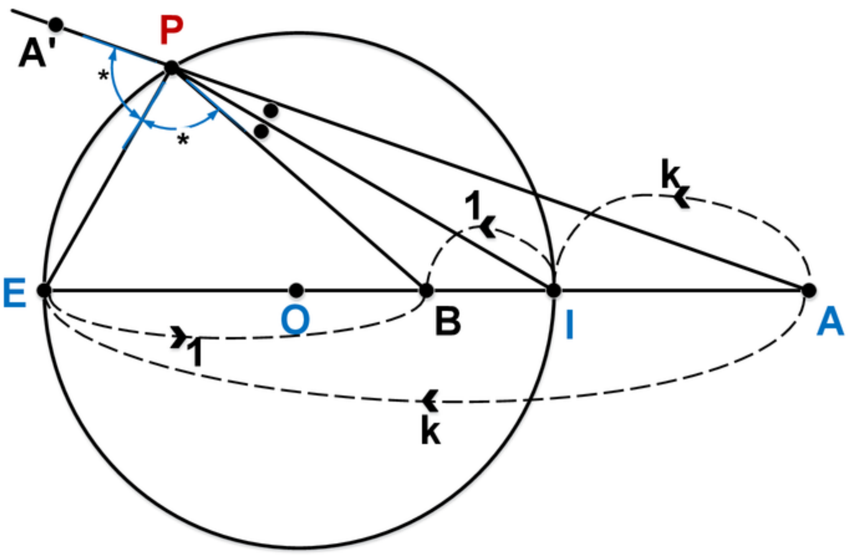


Figure 5.2: Apollonius circle for a moving point P such that $\frac{AP}{PB} = k > 1$. Here $m\angle API = m\angle BPI$ and $m\angle A'PE = m\angle BPE$.

$$(\boldsymbol{E} - \boldsymbol{A}) = k(\boldsymbol{E} - \boldsymbol{B}), \tag{5.6}$$

which can be used to express $\boldsymbol{I}$ and $\boldsymbol{E}$ in terms of $\boldsymbol{A}$ and $\boldsymbol{B}$ as

$$\boldsymbol{I} = \frac{1}{k+1}(\boldsymbol{A} + k\boldsymbol{B}), \tag{5.7}$$

$$\boldsymbol{E} = \frac{1}{k-1}(-\boldsymbol{A} + k\boldsymbol{B}). \tag{5.8}$$

The Apollonius circle is easily characterized by the points $\boldsymbol{I}$ and $\boldsymbol{E}$, since they are the two end points of one of the its diameters. The center of the circle is the midpoint of points $\boldsymbol{I}$ and $\boldsymbol{E}$, namely:

$$\begin{aligned}
\boldsymbol{O} &= \frac{1}{2}(\boldsymbol{I} + \boldsymbol{E}) \\
&= \frac{1}{2}[(\frac{1}{k+1} - \frac{1}{k-1})]\boldsymbol{A} + k(\frac{1}{k+1} + \frac{1}{k-1})\boldsymbol{B} \\
&= -\frac{1}{k^2-1}\boldsymbol{A} + \frac{k^2}{k^2-1}\boldsymbol{B},
\end{aligned} \tag{5.9}$$

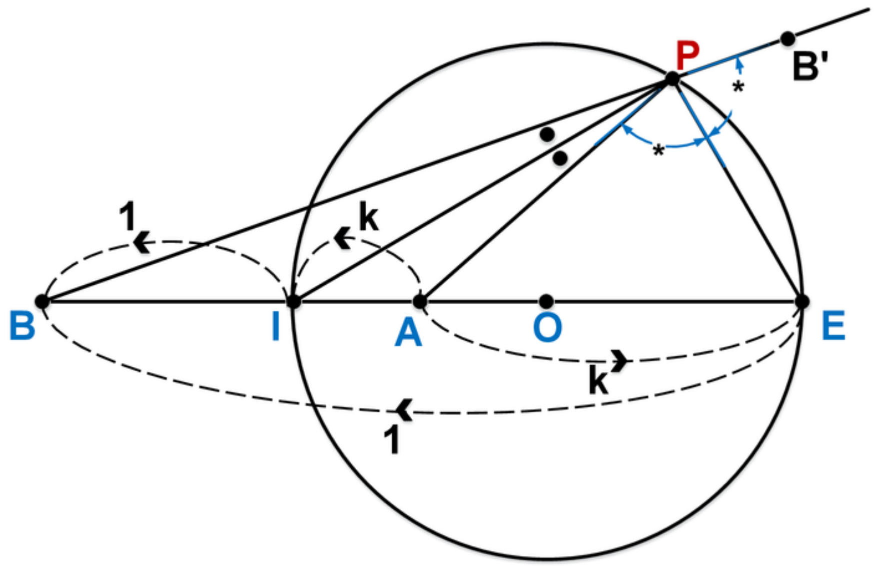


Figure 5.3: Apollonius circle for a moving point P such that $\frac{AP}{PB} = k < 1$. Here $m\angle API = m\angle BPI$ and $m\angle APE = m\angle B'PE$.

while the radius of the circle is half the length of the displacement from $\boldsymbol{I}$ to $\boldsymbol{E}$,

$$\begin{aligned}
r &= \frac{1}{2}|\boldsymbol{I} - \boldsymbol{E}| \\
&= \frac{1}{2}|(\frac{1}{k+1} + \frac{1}{k-1})\boldsymbol{A} + k(\frac{1}{k+1} - \frac{1}{k-1})\boldsymbol{B}| \\
&= |\frac{k}{k^2-1}\boldsymbol{A} - \frac{k}{k^2-1}\boldsymbol{B}| \\
&= |\frac{k}{k^2-1}||\boldsymbol{A} - \boldsymbol{B}| = \frac{k}{k^2-1}(AB).
\end{aligned} \tag{5.10}$$

It is clear from (5.9) and (5.10) that $\lim_{k\to 1}|\boldsymbol{O}| \to \infty$ and $\lim_{k\to 1} r \to \infty$, and hence for $k = 1$, the Apollonius circle degenerates into a straight line, namely the perpendicular bisector of the straight segment $\overline{AB}$ (Fig. 5.4).

5.3.1 The *AD* Apollonius circle

For the AD Apollonius circle, the two fixed points are the initial positions of the Attacker $\boldsymbol{A} = (x_A, 0)$ and the initial position of the defender $\boldsymbol{D} = (-x_A, 0)$. The fixed ratio of the circle k is replaced by the following dimensionless ratio which is the Attacker's speed normalized w.r.t the Defender speed:

$$\gamma = \frac{V_A}{V_D}. \tag{5.11}$$

We will consider the three cases of

1. $\gamma < 1$ (fast Defender) discussed in Garcia et al. [48].

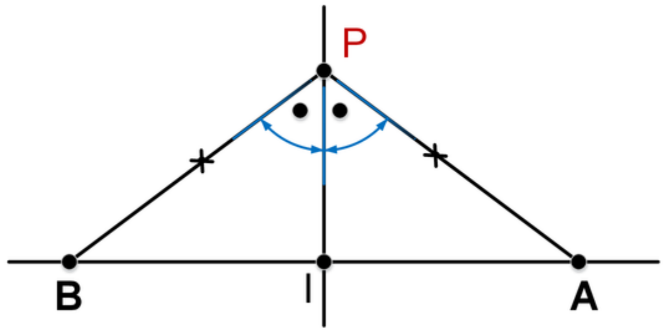


Figure 5.4: For $k = 1$, the Apollonius circle in figures 5.2 or 5.3 degenerates into the perpendicular bisector of the straight segment $\overline{AB}$. The point E disappears in this figure as it goes to ∞

2. $\gamma = 1$ (same-speed Defender) discussed in Garcia et al. [47, 49].

3. $\gamma > 1$ (slow Defender), which is a novel case.

Substituting the values of $\boldsymbol{A}$ and $\boldsymbol{D}$ above for $\boldsymbol{A}$ and $\boldsymbol{B}$ in (5.7),(5.8),(5.9) and (5.10), respectively, and replacing k therein by γ we obtain for $\gamma \neq 1$,

$$\boldsymbol{I_1} = (\frac{1-\gamma}{1+\gamma}x_A, 0), \tag{5.12}$$

$$\boldsymbol{E_1} = (\frac{1+\gamma}{1-\gamma}x_A, 0), \tag{5.13}$$

$$\boldsymbol{O_1} = (\frac{1+\gamma^2}{1-\gamma^2}x_A, 0), \tag{5.14}$$

$$r_1 = \frac{2\gamma}{|1-\gamma^2|}x_A. \tag{5.15}$$

Three interesting sets of limits are now considered. The first set of limits are those when $(\gamma \to 1)$, namely

$$\lim_{\gamma\to 1} \boldsymbol{I}_1 = (0,0), \tag{5.16}$$

$$\lim_{\gamma\to 1} \boldsymbol{E}_1 = (\mp\infty, 0), \tag{5.17}$$

$$\lim_{\gamma\to 1} \boldsymbol{O}_1 = (\mp\infty, 0), \tag{5.18}$$

$$\lim_{\gamma\to 1} r_1 = \infty, \tag{5.19}$$

which means that the AD Apollonius circle degenerates in the limit $(\gamma \to 1)$ to a circle of an infinite radius with center still on the $x-$axis but infinitely distant from the origin. More precisely, this limit is identified as a straight line, namely the perpendicular bisector of the straight-line segment $\overline{AD}$.

The second set of limits are those when ($\gamma \to 0$), namely

$$\lim_{\gamma\to 0} \boldsymbol{I}_1 = (x_A, 0), \tag{5.20}$$
$$\lim_{\gamma\to 0} \boldsymbol{E}_1 = (x_A, 0), \tag{5.21}$$
$$\lim_{\gamma\to 0} \boldsymbol{O}_1 = (x_A, 0), \tag{5.22}$$
$$\lim_{\gamma\to 0} r_1 = 0, \tag{5.23}$$

which means that the AD Apollonius circle collapses in the limit ($\gamma \to 0$) to a single point, namely the initial position of the Attacker $\boldsymbol{A} = (x_A, 0)$.

The third set of limits are those when ($\gamma \to \infty$), namely

$$\lim_{\gamma\to \infty} \boldsymbol{I}_1 = (-x_A, 0), \tag{5.24}$$
$$\lim_{\gamma\to \infty} \boldsymbol{E}_1 = (-x_A, 0), \tag{5.25}$$
$$\lim_{\gamma\to \infty} \boldsymbol{O}_1 = (-x_A, 0), \tag{5.26}$$
$$\lim_{\gamma\to \infty} r_1 = 0, \tag{5.27}$$

which means that the AD Apollonius circle collapses in the limit ($\gamma \to \infty$) to a single point, namely the initial position of the Defender $\boldsymbol{D} = (-x_A, 0)$.

5.3.2 The TA Apollonius circle

For the TA Apollonius circle, the two fixed points are the initial position of the Target $\boldsymbol{T} = (x_T, y_T)$ and the initial position of the Attacker $\boldsymbol{A} = (x_A, 0)$. Again, the fixed ratio of the circle k is replaced by a dimensionless quantity, namely the Target's speed normalized w.r.t. the Attacker's speed:

$$\alpha = \frac{V_T}{V_A}. \tag{5.28}$$

Here, we will consider only the case $\alpha < 1$, since for $\alpha \geqslant 1$, the Target always survives, even without any assistance from the Defender.

Now, we substitute the values of $\boldsymbol{T}$ and $\boldsymbol{A}$ above for $\boldsymbol{A}$ and $\boldsymbol{B}$ in (5.7),(5.8),(5.9) and (5.10), respectively, and replace k therein by α to obtain:

$$\boldsymbol{I_2} = \frac{1}{1+\alpha}(\boldsymbol{T} + \alpha \boldsymbol{A}) = \frac{1}{1+\alpha}(x_T + \alpha x_A, y_T), \tag{5.29}$$

$$\boldsymbol{E_2} = \frac{1}{\alpha - 1}(-\boldsymbol{T} + \alpha \boldsymbol{A}) = \frac{1}{1-\alpha}(x_T - \alpha x_A, y_T), \tag{5.30}$$

$$\boldsymbol{O_2} = \frac{1}{1-\alpha^2}(\boldsymbol{T} - \alpha^2 \boldsymbol{A}) = \frac{1}{1-\alpha^2}(x_T - \alpha^2 x_A, y_T), \tag{5.31}$$

$$r_2 = \frac{\alpha}{1-\alpha^2}(TA) = \frac{\alpha d}{1-\alpha^2}, \tag{5.32}$$

where $d = TA$ = the initial distance between the target and Attacker, namely

$$d = \sqrt{(x_T - x_A)^2 + y_T{}^2}. \tag{5.33}$$

Chapter 6: CRITICAL SPEED RATIO

Target survival is guaranteed if there is an overlapping of the region reachable by the Target before the Attacker (the interior of the TA Apollonius circle) and the region R_r reachable by the Defender before the Attacker (Fig. 5.1), since within this overlapping, the Defender can perform its intended role of intercepting the Attacker before the Attacker captures the Target. Target survival is critical when the aforementioned overlapping diminishes into a single point at which the aforementioned two regions barely touch, or are tangent to one another. For this situation, the normalized Target speed (the speed ratio) $\alpha = \dfrac{V_T}{V_A}$ attains its minimal or critical value $\overline{\alpha}$. Now, we consider three cases, in which the TA-Apollonius circle is tangent from outside to the boundary of the shaded region R_r. An implicit assumption throughout the forthcoming analysis is that $\boldsymbol{T} = (x_T, Y_T)$ is outside R_r.

6.1 The case $\gamma < 1$ (fast defender)

The critical speed ratio $\overline{\alpha}$, occurs when the TA Apollonius circle is *internally* tangent to the AD Apollonius circle, i.e., when the centers $\boldsymbol{O_2}$ and $\boldsymbol{O_1}$ of these two circles and their tangency point $\boldsymbol{C}$ are collinear (Fig. 6.1(a)). This happens when

$$r_1 - r_2 = |\boldsymbol{O_1} - \boldsymbol{O_2}|. \tag{6.1}$$

Substituting for r_1,r_2,$\boldsymbol{O_1}$and $\boldsymbol{O_2}$ from (5.15), (5.32), (5.15) and (5.31) respectively, and noting that $\gamma < 1$, one obtains

$$\begin{aligned}\frac{2\gamma}{1-\gamma^2}x_A - \frac{\alpha}{1-\alpha^2}d &= |(\frac{1+\gamma^2}{1-\gamma^2}x_A, 0) - \frac{1}{1-\alpha^2}(x_T - \alpha^2 x_A, y_T)| \\ &= [(\frac{1+\gamma^2}{1-\gamma^2}x_A - \frac{1}{1-\alpha^2}(x_T - \alpha^2 x_A))^2 + \frac{1}{(1-\alpha^2)^2}y_T^2]^{\frac{1}{2}}.\end{aligned} \tag{6.2}$$

In (6.2), we deliberately replaced $|1-\gamma^2|$ by $(1-\gamma^2)$ because $\gamma < 1$. We now square both sides of (6.2) to obtain:

$$\begin{aligned}\frac{4\gamma^2}{(1-\gamma^2)^2}x_A^2 + \frac{\alpha^2}{(1-\alpha^2)^2}d^2 - \frac{4\gamma\alpha d}{(1-\gamma^2)(1-\alpha^2)}x_A = \\ = \frac{(1+\gamma^2)^2}{(1-\gamma^2)^2}x_A^2 + \frac{x_T^2 - 2\alpha^2 x_T x_A + \alpha^4 x_A^2}{(1-\alpha^2)^2} - \frac{2(1+\gamma^2)(x_T - \alpha^2 x_A)x_A}{(1-\gamma^2)(1-\alpha^2)} + \frac{1}{(1-\alpha^2)^2}y_T^2.\end{aligned} \tag{6.3}$$

By squaring both sides of equation (6.2), the cardinality of the solution set for equation (6.2) is doubled. This means that when we solve the resulting equation, half of the resulting solutions will be extraneous or irrelevant and have to be rejected. Fortunately, we will have genuine reasons that enable us to identify such solutions, and compel us to reject them. The solutions retained after such a rejection are the correct solutions of (6.1) We now use (5.33) to replace y_T^2 by $(d^2 - x_A^2 - x_T^2 - 2x_A x_T)$ and rearrange terms in (6.3) to obtain

$$
\begin{aligned}
[\frac{4\gamma^2}{(1-\gamma^2)^2}-\frac{(1+\gamma^2)^2}{(1-\gamma^2)^2}-\frac{\alpha^4}{(1-\alpha^2)^2}-\frac{2\alpha^2(1+\gamma^2)}{(1-\gamma^2)(1-\alpha^2)}+\frac{1}{(1-\alpha^2)^2}]x_A^2 \\
+[\frac{\alpha^2}{(1-\alpha^2)^2}-\frac{1}{(1-\alpha^2)^2}]d^2 \\
-\frac{4\gamma\alpha x_A d}{(1-\gamma^2)(1-\alpha^2)}+[\frac{-1}{(1-\alpha^2)^2}+\frac{1}{(1-\alpha^2)^2}]x_T^2 \\
+[\frac{2\alpha^2}{(1-\alpha^2)^2}+\frac{2(1+\gamma^2)}{(1-\gamma^2)(1-\alpha^2)}-\frac{2}{(1-\alpha^2)^2}]x_T x_A = 0
\end{aligned}
\tag{6.4}
$$

The coefficient of x_A^2 in (6.4) is

$$
\begin{aligned}
&\frac{4\gamma^2}{(1-\gamma^2)^2}-\frac{(1+\gamma^2)^2}{(1-\gamma^2)^2}-\frac{\alpha^4}{(1-\alpha^2)^2}-\frac{2\alpha^2(1+\gamma^2)}{(1-\gamma^2)(1-\alpha^2)}+\frac{1}{(1-\alpha^2)^2} \\
&=\frac{4\gamma^2-1-\gamma^4-2\gamma^2}{(1-\gamma^2)^2}+\frac{(1-\alpha^4)}{(1-\alpha^2)^2}-\frac{2\alpha^2(1-\gamma^2)}{(1-\gamma^2)(1-\alpha^2)} \\
&=-1+\frac{1+\alpha^2}{(1-\alpha^2)}-\frac{2\alpha^2(1+\gamma^2)}{(1-\gamma^2)(1-\alpha^2)} \\
&=\frac{2\alpha^2}{(1-\alpha^2)}-\frac{2\alpha^2(1+\gamma^2)}{(1-\gamma^2)(1-\alpha^2)} \\
&=\frac{2\alpha^2}{(1-\alpha^2)}[1-\frac{(1+\gamma^2)}{(1-\gamma^2)}]=\frac{2\alpha^2}{(1-\alpha^2)}(\frac{-2\gamma^2}{(1-\gamma^2)}) \\
&=\frac{-4\alpha^2\gamma^2}{(1-\alpha^2)(1-\gamma^2)},
\end{aligned}
\tag{6.5}
$$

and the coefficient of d^2 in (6.4) is

$$
\frac{\alpha^2}{(1-\alpha^2)^2}-\frac{1}{(1-\alpha^2)^2}=\frac{-(1-\alpha^2)}{(1-\alpha^2)^2}=-\frac{1}{(1-\alpha^2)},
\tag{6.6}
$$

while the coefficient of $x_T x_A$ in (6.4) is

$$
\begin{aligned}
&\frac{2\alpha^2}{(1-\alpha^2)^2}+\frac{2(1+\gamma^2)}{(1-\gamma^2)(1-\alpha^2)}-\frac{2}{(1-\alpha^2)^2} \\
&=\frac{2(\alpha^2-1)}{(1-\alpha^2)^2}+\frac{2(1+\gamma^2)}{(1-\gamma^2)(1-\alpha^2)} \\
&=-\frac{2}{(1-\alpha^2)}+\frac{2(1+\gamma^2)}{(1-\gamma^2)(1-\alpha^2)} \\
&=\frac{2}{(1-\alpha^2)}[-1+(\frac{1+\gamma^2}{1-\gamma^2})]=\frac{2}{(1-\alpha^2)}(\frac{2\gamma^2}{1-\gamma^2}) \\
&=\frac{4\gamma^2}{(1-\alpha^2)(1-\gamma^2)}
\end{aligned}
\tag{6.7}
$$

This means that equation (6.4) can be simplified considerably to the equivalent form

$$\frac{-4\alpha^2\gamma^2}{(1-\alpha^2)(1-\gamma^2)}x_A^2 - \frac{1}{1-\alpha^2}d^2 - \frac{4\gamma\alpha x_A d}{(1-\gamma^2)(1-\alpha^2)} + \frac{4\gamma^2}{(1-\alpha^2)(1-\gamma^2)}x_T x_A \tag{6.8}$$

Now, we multiply (6.8) by $[-(1-\alpha^2)(1-\gamma^2)]$ noting that $\alpha \neq 1$ and $\gamma \neq 1$, to obtain

$$(4\gamma^2 x_A^2)\alpha^2 + (4\gamma x_A d)\alpha + (1-\gamma^2)d^2 - 4\gamma^2 x_T x_A = 0 \tag{6.9}$$

or equivalently

$$\boxed{\alpha^2 + \frac{d}{\gamma x_A}\alpha + [(\frac{1-\gamma^2}{4\gamma^2})(\frac{d}{x_A})^2 - \frac{x_T}{x_A}] = 0} \tag{6.10}$$

Equation (6.10) is an improvement of equation (36) in [49]. while (6.10) is a quadratic equation in α, equation (36) in [49] is a quartic in α that has the same two roots of (6.10) in addition to the two irrelevant roots $\alpha \mp 1$. As a bonus, equation (6.10) is written in a self-verifying dimensionless form.

6.2 The case $\gamma = 1$ (similar Defender)

The critical speed ratio $\overline{\alpha}$ occurs when the TA Apollonius circle touches the $L.H.S.$ of the $XY-$plane (Fig. 6.1(b)), i.e., when

$$r_2 = Absicessa\ of\ \boldsymbol{O_2}. \tag{6.11}$$

i.e., thanks to (5.31) and (5.32) when:

$$\frac{\alpha d}{1-\alpha^2} = \frac{x_T - \alpha^2 x_A}{1-\alpha^2}. \tag{6.12}$$

Multiplying (6.12) by $(1-\alpha^2)$ (thanks to the fact that $\alpha \neq 1$), and rearranging, one obtains the following quadratic equation in α

$$x_A\alpha^2 + d\alpha - x_T = 0, \tag{6.13}$$

which is equation (12) in [3]. It is interesting to note that equation (6.12) is the special case ($\gamma = 1$) of (6.10), though (6.10) was derived under the assumption $\gamma \neq 1$.

6.3 The case $\gamma > 1$ (Slow Defender)

The critical speed ratio $\overline{\alpha}$ occurs when the TA Apollonius circle is *externally* tangent to the AD Apollonius circle, i.e., when the centre $\boldsymbol{O_2}$, the tangency point $\boldsymbol{C}$ and the centre $\boldsymbol{O_1}$ are collinear(Fig. 6.1(c)), i.e., when

$$r_1 + r_2 = |\boldsymbol{O_1} - \boldsymbol{O_2}| \tag{6.14}$$

Since $\gamma > 1$, we write

$$r_1 + r_2 = \frac{2\gamma}{|1-\gamma^2|} + \frac{\alpha}{1-\alpha^2} d = \frac{-2\gamma}{1-\gamma^2} + \frac{\alpha}{1-\alpha^2} d. \tag{6.15}$$

The above expression for $(r_1 + r_2)$ is exactly the negative of $(r_1 - r_2)$ in (6.2). Hence, squaring of (6.14) results in (6.3), and consequently lead to (6.4) and (6.8), and finally to (6.9) and (6.10). Equation (6.10) is thereby shown to hold for a slow Defender besides being true for a fast one.

Therefore, we will use the quadratic formula (6.10) for all values of γ, and will solve it to obtain $\overline{\alpha}$ for any γ as

$$\begin{aligned} \overline{\alpha} &= \frac{1}{2}[-\frac{d}{\gamma x_A} \pm [(\frac{d}{\gamma x_A})^2 - 4(\frac{1-\gamma^2}{4\gamma^2})(\frac{d}{x_A})^2 + \frac{4x_T}{x_A}]^{1/2}] \\ &= \frac{1}{2\gamma x_A}[-d \mp \gamma \sqrt{d^2 + 4x_T x_A}] \\ &= \frac{1}{2\gamma x_A}[-\sqrt{(x_A - x_T)^2 + y^2} + \gamma \sqrt{(x_A + x_T)^2 + y_T^2}] \end{aligned} \tag{6.16}$$

Equation (6.16) was derived in [49] for $\gamma < 1$ and in [47] for $\gamma = 1$, provided $x_T > 0$. It is shown herein to hold for all values in γ, including $\gamma > 1$. Note that $\overline{\alpha}$ is definitely nonnegative and hence the minus sign in the solution for $\overline{\alpha}$ is rejected. The question remains whether (6.16) yields a non-negative value for $\overline{\alpha}$ when the positive sign is chosen in (6.16). It could be argued that (6.16) should be used (with the positive sign, of course) as long as it yields a nonnegative value for $\overline{\alpha}$, and otherwise $\overline{\alpha}$ should be set to zero. For example, if $\gamma = 1$ and $x_T < 0$, equation (6.16) yields a negative value for $\overline{\alpha}$ which need to be rejected and replaced by zero. Physically, the situation $\gamma = 1, x_T < 0$ is one in which the Target is initially in the region reachable by the Defender before the Attacker, i.e., it is within the effective protection of the Defender, even if it does not move itself at all. Therefor, we rewrite (6.16) in the following dimensionless form:

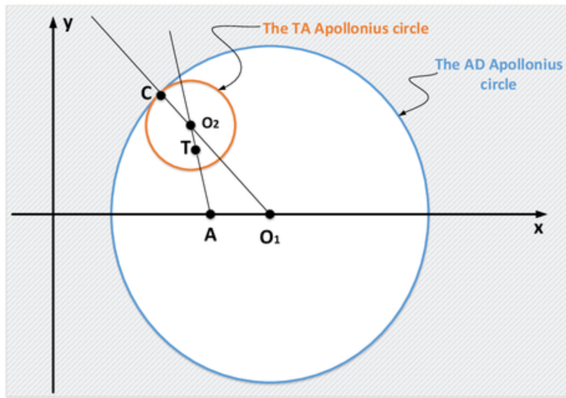
$$\overline{\alpha} = \frac{1}{2\gamma}[-\sqrt{(1-\frac{x_T}{x_A})^2 + (\frac{y_T}{x_A})^2} + \gamma \sqrt{(1+\frac{x_T}{x_A})^2 + (\frac{y_T}{x_A})^2}\ ] \tag{6.17}$$

provided that

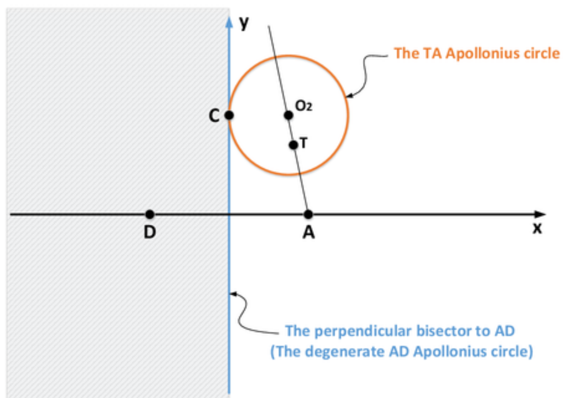
$$\gamma \sqrt{(1+\frac{x_T}{x_A})^2 + (\frac{y_T}{x_A})^2} \ > \ \sqrt{(1-\frac{x_T}{x_A})^2 + (\frac{y_T}{x_A})^2} \tag{6.18}$$

and whenever (6.18) is not valid, we simply set

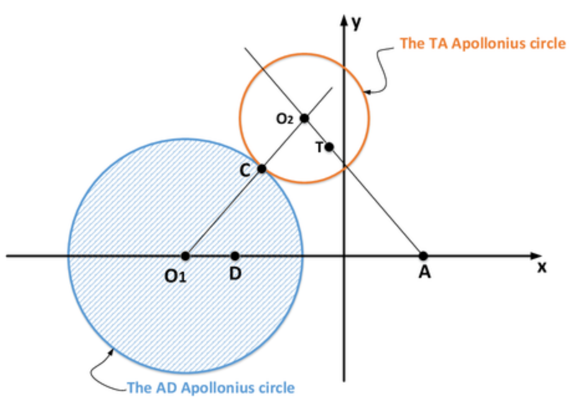
$$\overline{\alpha} = 0. \tag{6.19}$$



(a) $\gamma < 1$



(b) $\gamma = 1$



(c) $\gamma > 1$

Figure 6.1: The critical speed ratio $\overline{\alpha}$ is obtained when the TA Apollonius circle is tangent to the boundary of the shaded region R_r which is the region reachable by the Defender befor the Attacker.

Chapter 7: ESCAPE REGION AND VORONOI DIAGRAM

In this chapter, we develop a novel analytic expression for the pursuit-evasion Voronoi diagram [47] that marks or borders the "safe" or "escape" region R_e for the Target, i.e., the region defined by the set of all coordinate points (x, y) such that if the target's initial position (x_T, y_T) is inside this region, then the target is guaranteed to survive (to escape the Attacker) provided both the Target and Defender implement their optimal strategies, and regardless of the policy adopted by the Attacker, whether optimal or not. The Voronoi diagram obtaind divides the $XY-$plane, into two regions

- a safe or escape region R_e to which the Defender's initial position $\boldsymbol{D}$ belongs.
- a *potentially* unsafe region to which the Attacker's initial position $\boldsymbol{A}$ belongs.

Garcia et al. [47] treated this problem for $\gamma = 1$ only using some involved arguments with inequalities developed from scratch, and without due reference to their earlier analysis concerning the critical speed ratio.

A very important *insight* that markedly simplifies our work is to note that required Voronoi diagram is simply a rephrasing of the criticality equation (6.10) for any value of γ including the cases $\gamma > 1$ or $\gamma < 1$ and $\gamma = 1$, viewed as a relation between y_T and x_T rather than a polynomial in α. Working with equalities is definitely simpler than working with inequalities, and it specifies a Voronoi diagram that divides the whole plane into two distinct parts, namely: the safe region and the (potentially) unsafe region. The safe region is then identified simply as the region to which $\boldsymbol{D}$ belongs. Note that the set of points of the safe regions R_e is a superset of the points reachable by the Defender before the Attacker R_r, i.e., $R_e \supseteq R_r$. The two regions R_e and R_r are identical if the Target does not move at all ($\alpha = 0$), a situation in which the Target does not exert any own effort to secure its survival and relies solely on the Defender's helping efforts. The family of Voronoi diagrams for various values of the normalized speed α will be bounded by the one for $\alpha = 0$, which is the border of R_r (Fig. 5.1).
Now, we obtain the general Voronoi diagram by rewriting the critically equation (6.10) or (6.9) as a relation between y_T and x_T. This is achieved via (5.33) to substitute $(x_A^2 + x_T^2 - 2x_A x_T + y_T^2)$ for d^2. To avoid mistakes in the manipulations, one must make sure that the pertinent equation at every step is dimensionally homogeneous. From (6.9), we write

$$(4\gamma\alpha x_A)d = -(1-\gamma^2)d^2 + 4\gamma^2 x_A x_T - 4\gamma^2\alpha^2 x_A^2 \tag{7.1}$$

Now, we square both sides of (7.1) to obtain

$$\begin{aligned}16\gamma^2\alpha^2 x_A^2(x_A^2 + x_T^2 - 2x_A x_T + y_T^2) &= [-(1-\gamma^2)(x_A^2 + x_T^2 - 2x_A x_T + y_T^2) + 4\gamma^2 x_A x_T - 4\gamma^2\alpha^2 x_A^2]^2 \\ &= [-(1-\gamma^2)(x_A^2 + x_T^2 + y_T^2) + 2(1+\gamma^2)x_A x_T - 4\gamma^2\alpha^2 x_A^2]^2\end{aligned} \tag{7.2}$$

Now we rearrange (7.2) as a quartic equation (forth-degree polynomial) in x_T and y_T, namely

$$
\begin{aligned}
p(x_T, y_T) = {} & 16\gamma^2\alpha^2 x_A^4 \\
& + 16\gamma^2\alpha^2 x_A^2 x_T^2 \\
& - 32\gamma^2\alpha^2 x_A^3 x_T \\
& + 16\gamma^2\alpha^2 x_A^2 y_T^2 \\
& - (1-\gamma^2)^2[x_A^4 + x_T^4 + y_T^4 + 2x_A^2 x_T^2 + 2x_A^2 y_T^2 + 2x_T^2 y_T^2] \\
& - (1+\gamma^2)^2 * 4x_A^2 x_T^2 \\
& - 16\gamma^4\alpha^4 x_A^4 \\
& + 2(1-\gamma^4)(2x_A^3 x_T + 2x_A x_T^3 + 2x_A x_T y_T^2) \\
& - 8\gamma^2(1-\gamma^2)\alpha^2(x_A^4 + x_A^2 x_T^2 + x_A^2 y_T^2) \\
& + 16\gamma^2(1+\gamma^2)\alpha^2 x_A^3 x_T
\end{aligned}
\tag{7.3}
$$

Equation (7.3) is satisfied by the required Voronoi diagram. The quartic polynomial $p(x_t, y_T)$ satisfies

$$p(x_T, -y_T) = p(x_T, y_T), \tag{7.4}$$

since it has even powers only for y_T. This means $p(x_T, y_T)$ is an even function in y_T and its graph in the $x_T - y_T$ plane is *symmetric* w.r.t. the x_T-axis.

Though (7.3) is a quartic equation, the actual Voronoi diagram is expected to be a quadratic curve and not a quartic curve. Note that we performed a squaring operation in obtaining (6.9) and hence for obtaining (7.3), which doubled the pertinent degree from two to four. Hence, the Voronoi diagram is just a second-degree factor, part or branch of the fourth-degree curve depicted by (7.3). We will be able to identify the correct Voronoi diagram by rejecting any part of the curve that lies in R_r, i.e., the correct Voronoi diagram should define the border of R_e such that $R_e \supseteq R_r$.

Note that (7.3) is the general equation for the Voronoi diagram in our current TAD problem and it appears here for the first time. As a check on its correctness, we use it to produce the correct curve for $\gamma = 1$, earlier obtained by Garcia et al. [49], namely

$$16x_A^2[-(1-\alpha^2)x_T^2 + \alpha^2 y_T^2] = -16x_A^4\alpha^2(1-\alpha^2), \tag{7.5}$$

or equivalently

$$\frac{x_T^2}{\alpha^2 x_A^2} - \frac{y_T^2}{(1-\alpha^2)x_A^2} = 1. \tag{7.6}$$

Equation (7.5) represents a hyperbola which is an open curve of two branches. Interestingly enough the term hyperbola is believed to have been coined by our old friend Apollonius of Perga in his work on conic sections. This hyperbola is represented in Fig. 7.1 and has the following properties:

- The hyperbola has two focal points $\boldsymbol{F_1} = (c, 0)$ and $\boldsymbol{F_2} = (-c, 0)$, where

$$c^2 = \alpha^2 x_A^2 + (1-\alpha^2)x_A^2 = x_A^2, \tag{7.7}$$

i.e. the focal points coincide with the initial positions of the Attacker $\boldsymbol{A} = (x_A, 0)$ and the Defender $\boldsymbol{D} = (-x_A, 0)$. Note that these two focal points are shared by an infinite set of curves for various values of α $(0 < \alpha < 1)$.

- The line joining the focal points, i.e., the *transverse axis* coincides with the x_T-axis. The midpoint of the two focal points, i.e., the *center* of the hyperbola, is the origin (0,0). The conjugate axis (the line perpendicular to the hyperbola's transverse axis through its center) is the y_T-axis.

- The eccentricity of the hyperbola e is

$$e = \frac{c}{\alpha x_A} = \frac{1}{\alpha} > 1, \tag{7.8}$$

where the fact that $e > 1$ distinguishes the hyperbola from other conic sections, namely the ellipse (of $e < 1$) and the parabola (of $e = 1$). Note that the eccentricity e equals the ratio of the distance PF_i from a point P on the hyperbola to one focus F_i to the distance of the point to the corresponding directrix PQ_i, i.e.

$$PF_i = e\, PQ_i, \quad i = 1, 2, \tag{7.9}$$

- The vertices of the hyperbola (its intersections with the x_T-axis) are at $(\alpha x_A, 0)$ and $(-\alpha x_A, 0)$, and hence approach the two focal points as α tends to 1 (from below).

- Perpendicular to the transverse axis are the two directrices D_1 and D_2 given by

$$x_T = \mp\frac{\alpha x_A}{e} = \mp\alpha^2 x_A \tag{7.10}$$

These directrices also approach the focal points when $\alpha \to 1$.

- The hyperbola has two asymptotes that intersect at its center, and are given by

$$y_T = \mp\left(\frac{\sqrt{1-\alpha^2}}{\alpha}\right)x_T. \tag{7.11}$$

For $0 < \alpha < 1$, the magnitude of the slope for an asymptote increases as α decreases and vice verse. Unlike the vertices and directrices, the asymptotes are independent of x_A.

- As a special case of the polynomial $p(x_T, y_T)$, in (7.3) the hyperbola (7.6) is an even function of y_T and hence is symmetric about the x_T-axis, Incidently the hyperbola is also even in x_T and hence its two branches are symmetric about the y_T-axis but this observation is irrelevant since only one branch is retained in further study.

- In the limit $(\alpha \to 0)$, the hyperbola (7.6) degenerates into the y_T-axis.

- In the limit $(\alpha \to 1)$, the right branch or the hyperbole (7.6) degenerates into a ray emanating from $\boldsymbol{A}$ in the positive direction of the x_T-axis, while its left branch degenerates into a ray emanating from $\boldsymbol{D}$ in the opposite direction.

In passing, we note that though equation (7.6) was obtained herein as the special case $(\gamma = 1)$ of equation (7.3), we have first obtained it by a rephrasing of the criticality condition

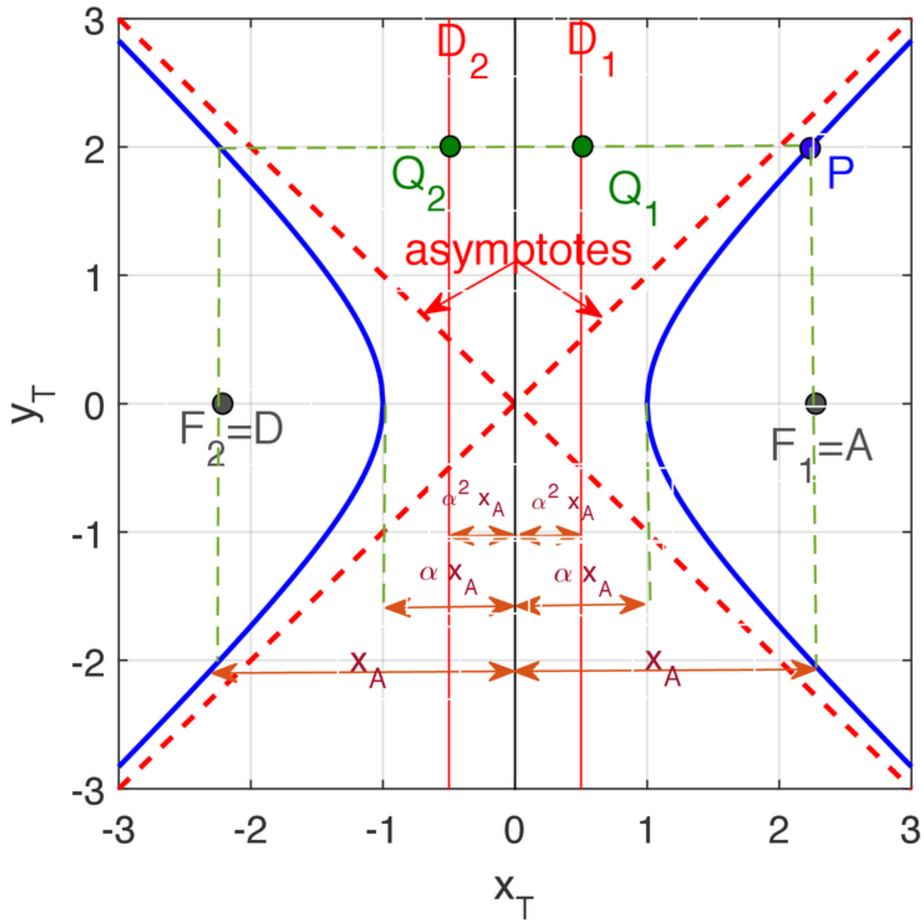


Figure 7.1: Graph and features of the hyperbola $\frac{x_T^2}{\alpha^2 x_A^2} - \frac{y_T^2}{(1-\alpha^2)x_A^2} = 1$

(6.13) which holds for $\gamma = 1$.

Now we note that the left branch of the hyperbola lies entirely in R_r and hence it is rejected. The right branch of the hyperbola is the required Voronoi diagram, and it is bordering the escape region R_e which extends to its left. Clearly, it is an enlargement of the reachability region R_r. Likewise, we expect the escape region to be a subset of the AD Apollonius circle in Fig. **??** where $\gamma < 1$, and to be a superset of this circle in Fig. **??** where $\gamma > 1$.

Figure 7.2 is a sketch of the hyperbola (7.6) for $\alpha = 0.25$. the figure stresses that the right branch of the hyperbola is accepted as a Voronoi diagram while the left branch is rejected as such. Figure 7.3 is computer generated Voronoi diagrams for $\gamma = 1$ and α varying from 0 (the y_T-axis) to slightly less than 1 (the straight-line ray emanating from $\boldsymbol{A}$ and coinciding with the x_T-axis).

Now, we come to a novel and really important contribution of this thesis, where we visualize the escape regions in the case of a fast Defender and a slow Defender. Figure 7.4

shows the AD-Apollonius circle for $\gamma = 0.8$ and $x_A = 4$ for which $\boldsymbol{I_1} = (0.444, 0)$, $\boldsymbol{E_1} = (36, 0)$, $\boldsymbol{O_1} = (18.22, 0)$, and $r_1 = 17.78$. Imposed on this circle is the quartic curve given by (7.3) which consists of two closed curves. There is one closed curve entirely outside the AD-Apollonius circle, i.e., entirely inside the Reachability region R_r and hence is rejected. The other closed curve is entirely inside the AD-Apollonius circle, i.e., the reachability region R_r and hence it is accepted as the Voronoi diagram. The escape region (safe region) is the *unshaded* area in this figure. Figure 7.5 generalizes Fig. 7.4 by demonstrating various accepted branches of the Voronoi diagram for $\gamma = 0.8$ and α as s parameter ranging from 0 (the AD-Apollonius circle) to a value approaching 1 from below (a closed curve collapsing to one barely encircling point A). This means that as α increases from 0 to 1, the safe or escape region expands from the exterior of the AD-Apollonius circle to almost the whole $x_T - y_T$ plane (excluding point A).

Now, we discuss the strikingly similar or dual case of a slow Defender. Figure 7.6 shows the AD-Apollonius circle for $\gamma = 1.25$ and $x_A = 4$ for which $\boldsymbol{I_1} = (-0.444, 0)$, $\boldsymbol{E_1} = (-36, 0)$, $\boldsymbol{O_1} = (-18.22, 0)$, $r_1 = 17.78$. Imposed on this circle is the quartic curve given by (7.3). Note that the whole graph in Fig. 7.6 is a *mirror image* of that in Fig. 7.4. The quartic curve again consists of two closed curves. There is one closed curve entirely inside the AD-Apollonius circle. i.e., entirely inside the Reachability region R_r and hence is rejected. The other closed curve is entirely outside the AD-Apollonius circle, i.e., entirely outside the Reachability region R_r, and hence is accepted as the Voronoi diagram. The escape region (safe region) is the *shaded* area in this figure. Figure 7.7 generalizes Fig. 7.6 by demonstrating various accepted branches of the Voronoi diagram for $\gamma = 1.25$ and α as a parameter ranging from 0 (the AD-Apollonius circle) to a value approaching 1 from below.

In passing, we discuss the situation of $\gamma = 0$ (when the Target does not move at all). In this case the TA circle collapse into a single point $\boldsymbol{T}$ and the critically condition (6.9) reduces to

$$(1 - \alpha^2)d^2 = 4\gamma^2 x_T x_A \tag{7.12}$$

which can readily be written as

$$(x_T - (\frac{1+\gamma^2}{1-\gamma^2})x_A)^2 + (y_T - 0)^2 = (\frac{2\gamma x_A}{1-\gamma^2})^2 \tag{7.13}$$

an equation that can be identified as the equation of the AD Apollonius circle in the $Y_T - X_T$ coordinates. This circle serves as the Voronoi diagram for $\alpha = 0$, a fact confirming our assertion that the escape region R_e reduces to the reachability region R_r when $\alpha = 0$. Likewise, when $\gamma = 1$, imposing the condition $\alpha = 0$ on the condition (6.13) results in

$$x_T = 0 \tag{7.14}$$

which means that the right branch of the hyperbolic Voronoi diagram (7.6) degenerates into the y_T-axis (the perpendicular bisector of $\overline{AD}$) and again the region R_e concides with the region R_r in this case.

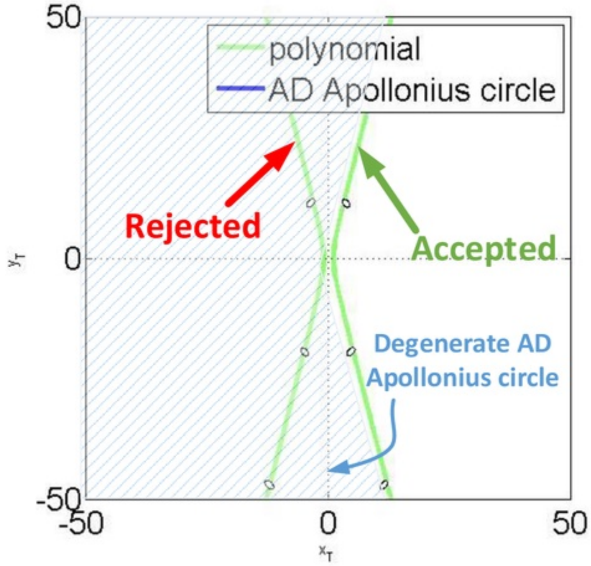


Figure 7.2: Generated computer output for the Voronoi diagram bordering the safe region for $x_A = 4,\ \alpha = 0.25,\ \gamma = 1$ (the safe region is the shaded area)

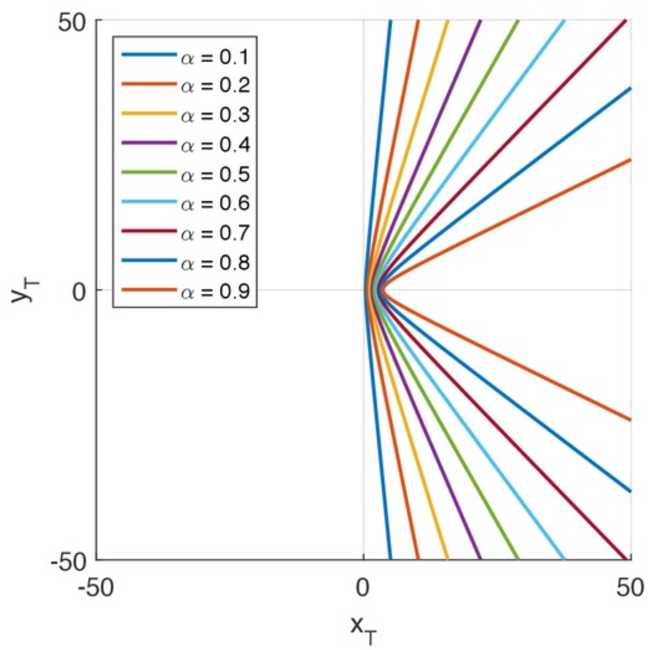


Figure 7.3: Various accepted branches of the voronoi diagram for $\gamma = 1$ and α as a parameter ranging from 0 to 1. These curves are computer generated from (7.3) and (7.6)

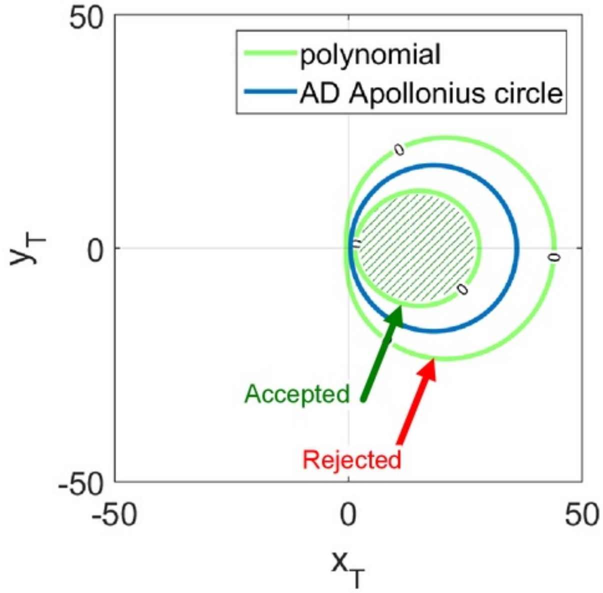


Figure 7.4: generated computer output for the Voronoi diagram bordering the safe region for $x_A = 4,\ \alpha = 0.25,\ \gamma = 0.8$ (the safe region is the unshaded area) the quartic in (7.2) or (7.3) produces two closed curves: one outside the AD-Apollonius circle (rejected) and the other inside the circle (accepted as the Voronoi diagram)

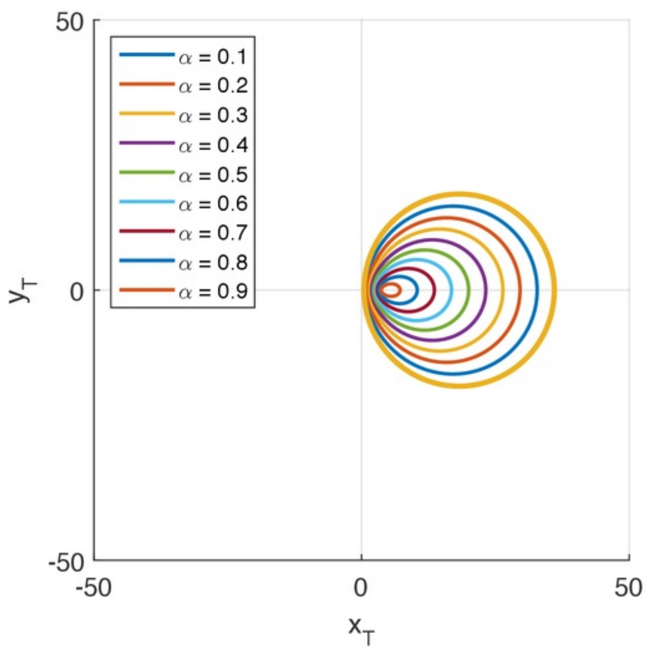


Figure 7.5: Various accepted branches of the voronoi diagram for $\gamma = 0.8$ and α as a parameter ranging from 0 to 1. These curves are computer generated from (7.3) and (7.6)

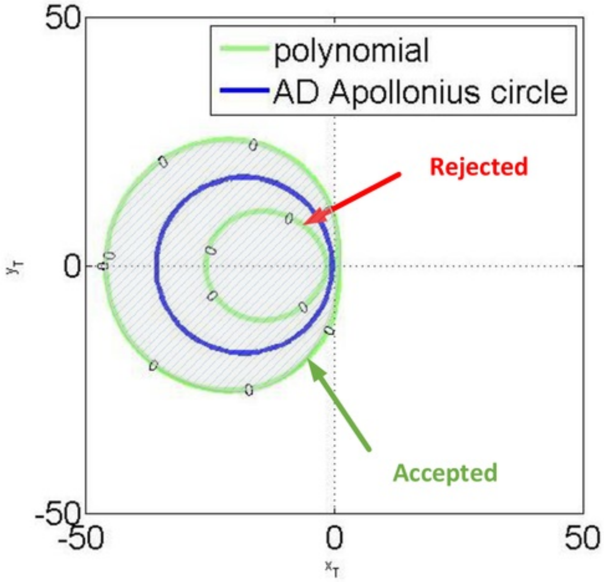


Figure 7.6: generated computer output for the Voronoi diagram bordering the safe region for $x_A = 4,\ \alpha = 0.25,\ \gamma = 1.25$ (the safe region is the shaded area) the quartic in (7.2) or (7.3) produces two closed curves: one inside the AD-Apollonius circle (rejected) and the other outside the circle (accepted as the Voronoi diagram)

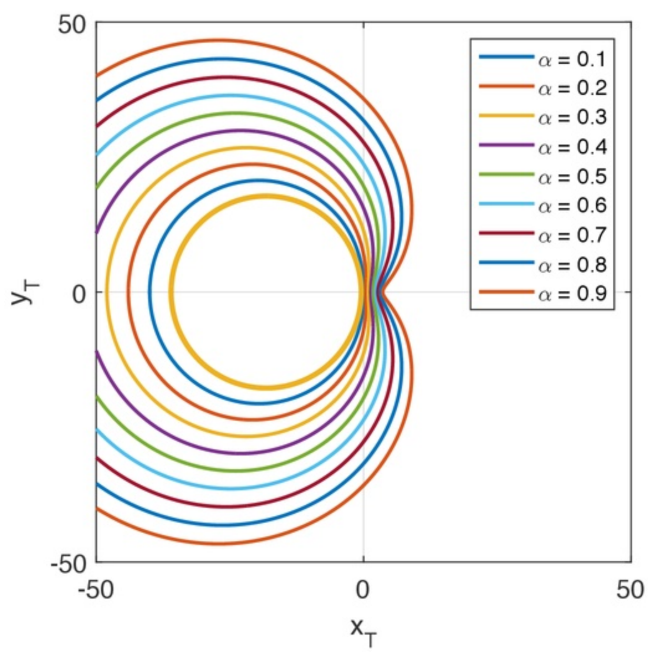


Figure 7.7: Various accepted branches of the voronoi diagram for $\gamma = 1.25$ and α as a parameter ranging from 0 to 1. These curves are computer generated from (7.3) and (7.6)

ANALYSIS AND SIMULATION OF THE TARGET-ATTACKER AND THE TARGET-ATTACKER-DEFENDER PROBLEMS_Literature Review

ORIGINALITY REPORT

30%

SIMILARITY INDEX

PRIMARY SOURCES

1 Turetsky, V.. "Missile guidance laws based on pursuit-evasion game formulations", Automatica, 200304 — 140 words — 10%
Crossref

2 www.nicolabasilico.net — 91 words — 7%
Internet

3 Ming-Hsiung Hsueh, Chin-I Huang, Li-Chen Fu. "A Differential Game Based Guidance Law for the Interceptor Missiles", IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society, 2007 — 69 words — 5%
Crossref

4 Timothy H. Chung. "Search and pursuit-evasion in mobile robotics : A survey", Autonomous Robots, 07/20/2011 — 49 words — 4%
Crossref

5 J.R. Cloutier. "Analytical guidance laws and integrated guidance/autopilot for homing missiles", Proceedings of IEEE International Conference on Control and Applications CCA-93, 1993 — 41 words — 3%
Crossref

6 Garcia, Eloy, David W. Casbeer, Khanh Pham, and Meir Pachter. "Cooperative aircraft defense from an attacking missile", 53rd IEEE Conference on Decision and Control, 2014. — 20 words — 1%
Crossref