

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
ooooooo  
o

# Analysis and Simulation of the Target-Attacker and the Target-Attacker-Defender Problems

**Mostafa Ali Rushdi**

Aerospace Engineering, Cairo University

**Prof. Ayman Hamdy Kasem and Prof. Gamal AlBayoumy**

September 19, 2017



Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooooooo  
o

# Overview

## Introduction

Problem Statement

Summary of Solution

## Target-Attacker

## ToolBox

## GA

## Unity Game

Game Design

Experimental Results

## Target-Attacker-Defender

## Optimal Strategies

**Introduction**

oo

Target-Attacker

oooooooooooooo  
oooooooooooooo

ToolBox

oooo

GA

oooo

Unity Game

oooooooooooo  
ooooooo

Target-Attacker-Defender

oooooooooooooo  
oooooooooooooooooooo

Optimal Strategies

oooooooooooo  
o

# Introduction

## Problem Statement

We address the problem of a target **aircraft** that tries to evade an **attacker** missile and avoid being hit by it.



# Thesis Overview

This thesis deals with the following problems:

## TA: Target-Attacker

- 2-agent pursuit-evasion
- Target (aircraft) and attacker (missile)
- Seeking optimal escape maneuver

## TAD: Target-Attacker-Defender

- 3-agent pursuit-evasion
- Target (aircraft), defender (missile), and attacker (missile)
- Seeking safe region, and optimal heading angles

## Unity Game

An experimental game to find the best escape maneuver of a TA problem by collecting and analyzing data from human players.

Introduction  
oo

**Target-Attacker**  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
ooooooo  
o

# Target-Attacker

# Summary of Solution

## Scenario 1:

### TA: Target-Attacker

- We present several methodologies to find the **optimal escape maneuver** for a target against an attacking missile.
- We simulate 2D proportional-navigation using MATLAB and Simulink.
- Optimization is achieved via the techniques of Monte-Carlo simulation and genetic algorithms.

# Main References



Zarchan, Paul (2012)

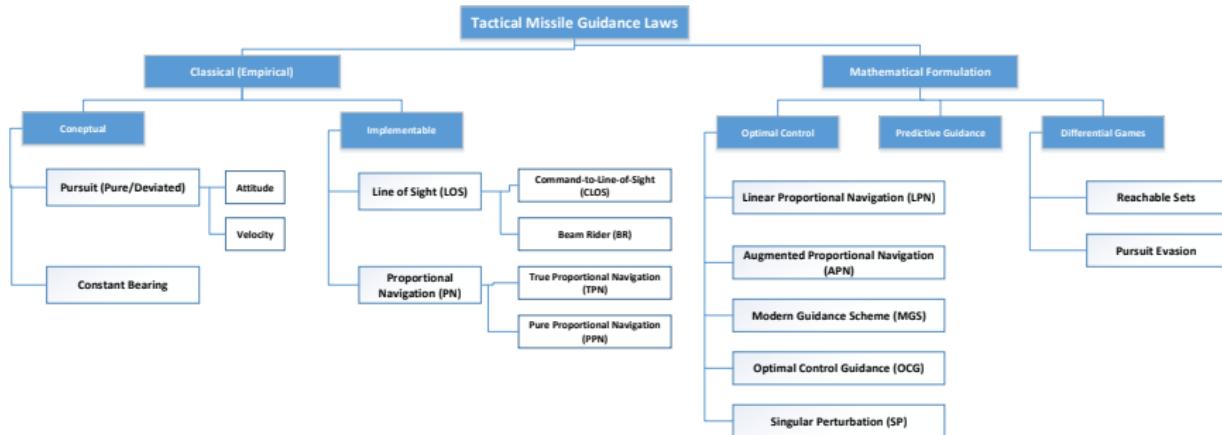
Tactical and strategic missile guidance

*American Institute of Aeronautics and Astronautics.*

## Target-Attacker Assumptions

- The missile is on the ground at (0,0). The plane is in the air at (10000,40000).
- The missile is faster than the aircraft, but cannot turn tighter than the aircraft, so it takes a longer path.
- Missile always attempts to trace the target. Thus if the target changes heading, it will necessary for the Missile to change heading similarly, but this is too difficult for him to achieve.
- The main problem with evading missiles is their speed, which makes timing somewhat difficult.

# Guidance Laws



## 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

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

where  $n_c$  is the acceleration command (for the missile) in ( $m/s^2$ ),  $N'$  is the effective navigation ratio, a unit-less designer-chosen gain (usually in the range of  $3 \rightarrow 5$ ),  $V_c$  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 ( $rad/s$ ).

# Proportional Navigation Equations - Geometry

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

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

$$R_{TM1} = RT1 - RM1$$

$$R_{TM2} = RT2 - RM2$$

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

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

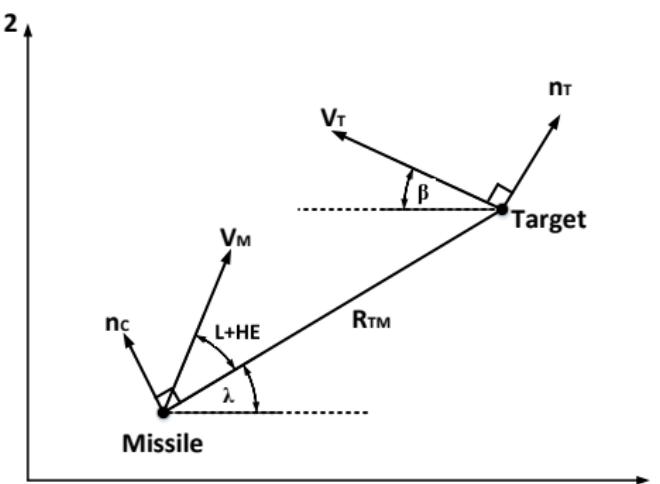
$$L = \sin^{-1}\left(\frac{V_T \sin(\beta + \lambda)}{V_M}\right)$$

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

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

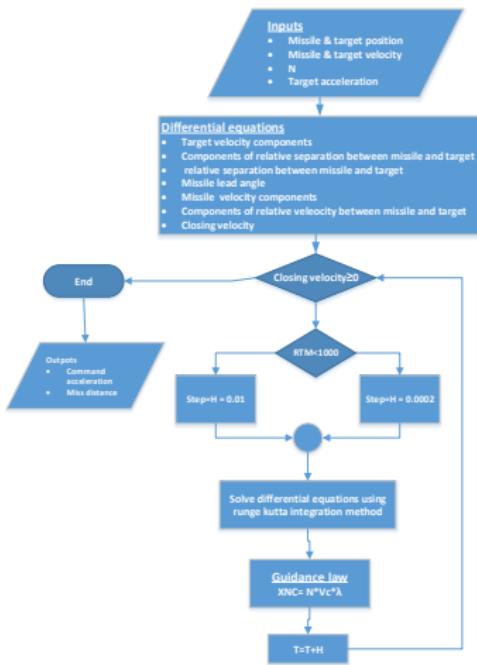
$$V_{TM1} = V_{T1} - V_{M1}$$

$$V_{TM2} = V_{T2} - V_{M2}$$

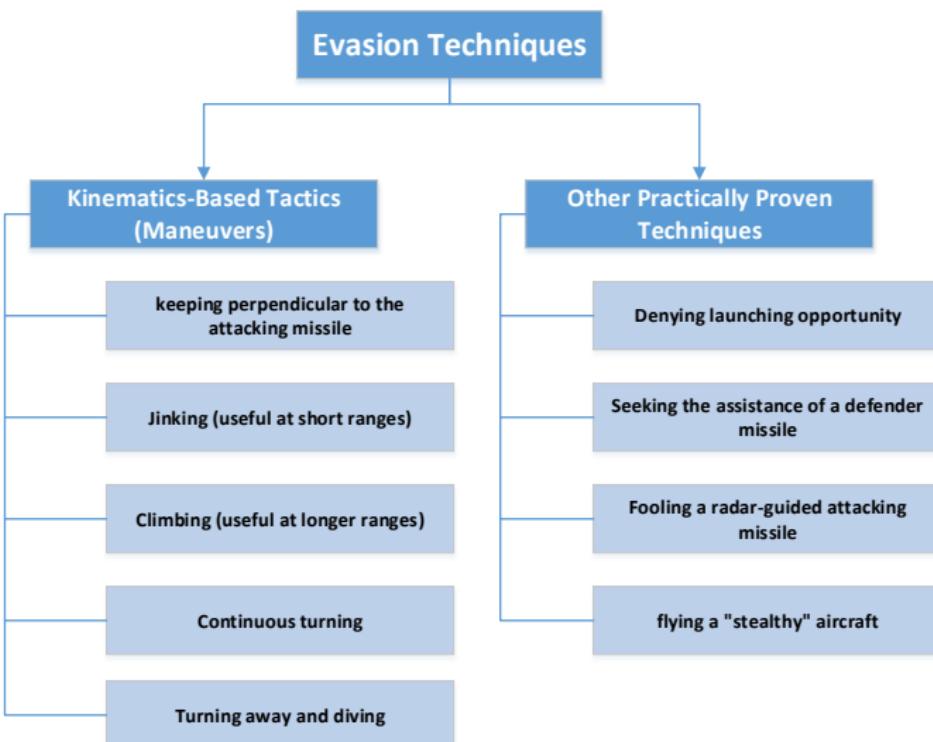


2d geometry of TA engagement.

# Proportional Navigation Flowchart



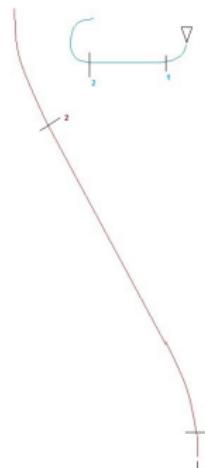
# Evasion Tactics Taxonomy



# Evasion Tactics

## 1. keeping perpendicular to the attacking missile:

- Turn hard to either left or right so as to fly at roughly 90 degrees angle to attacking aircraft (This forces missile to bleed off the energy and to lead the target).
- Once target aircraft makes a hard turn to reverse a direction, missile with its far larger turn circle will be unable to compensate.



## Evasion Tactics

### 2. Jinking (useful at short ranges):

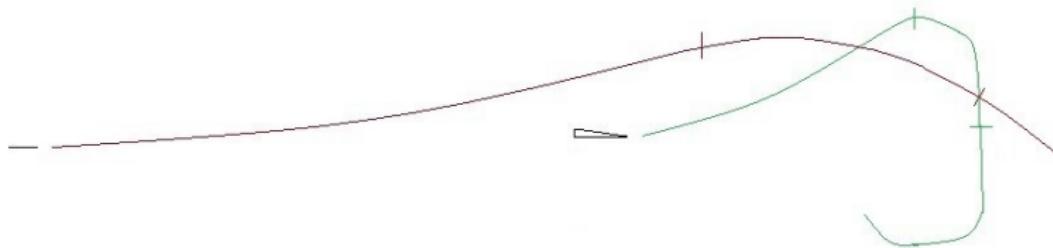
- Aircraft must be positioned so that it is at angle (30-60 degrees is optimum) relative to missiles flight path.
- Once missile gets closer, aircraft will make a hard turn in opposite direction.
- As there is a lag between aircraft changing the direction and missile following (for several reasons, most important of which is missiles inertia), this will cause missile to head in wrong direction until it manages to correct, and also to bleed off the energy.
- Missile will fly past the aircraft and miss.



## Evasion Tactics

### 3. Climbing (useful at longer ranges):

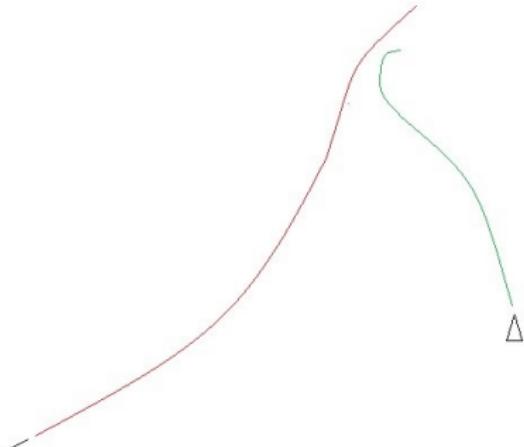
- Since at long range missile will have burned out its engine, it will rely on inertia to keep it flying, and climbing will mean that it will bleed off energy rapidly.
- Once missile reaches a close range (maybe around 1,500 meters), dive for the ground, then pull up (This will allow pilot to gain energy and using it to evade the missile).



# Evasion Tactics

## 4. Continuous turning:

- place the missile at 3 o'clock or 9 o'clock position
- maintain sufficient turn to keep the missile
- This tactic forces the missile to execute a continuous turn, bleeding the energy entire time, making it easier to outturn the missile once it comes close.

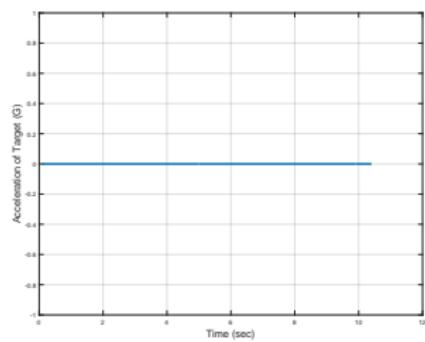
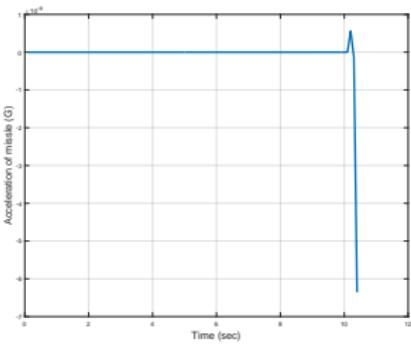
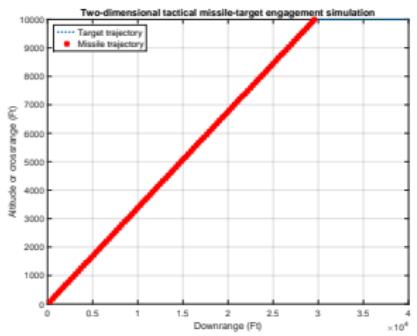


# Target Maneuver Cases

- Zero Target maneuver
- Constant Target maneuver
- Polynomial Target maneuver
- Trapezoidal Target maneuver

# Target Maneuver Cases

**Zero Target maneuver.** Heading error = 0 and  $N' = 4$ .



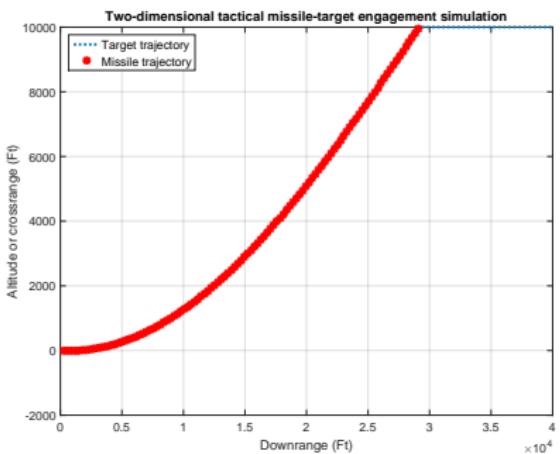
Target and attacker trajectories.

Attacker (missile) acceleration.

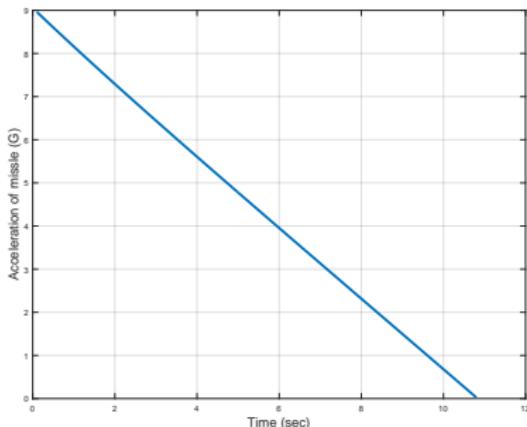
Target acceleration.

# Target Maneuver Cases

**Zero Target maneuver.** Heading error =  $-20$  and  $N' = 3$ .



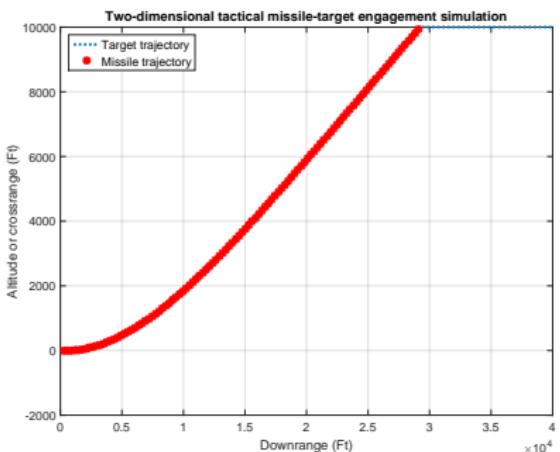
Target and attacker trajectories.



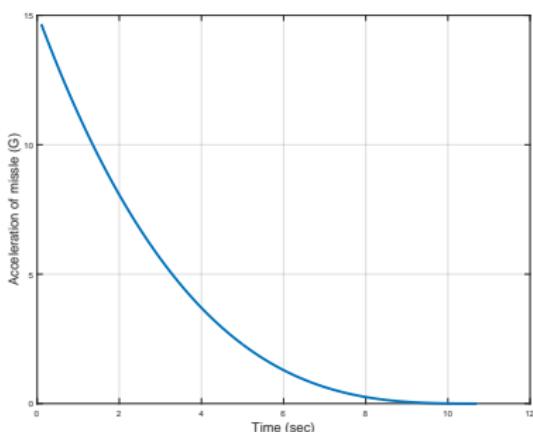
Attacker (missile) acceleration.

# Target Maneuver Cases

**Zero Target maneuver.** Heading error =  $-20$  and  $N' = 5$ .



Target and attacker trajectories.

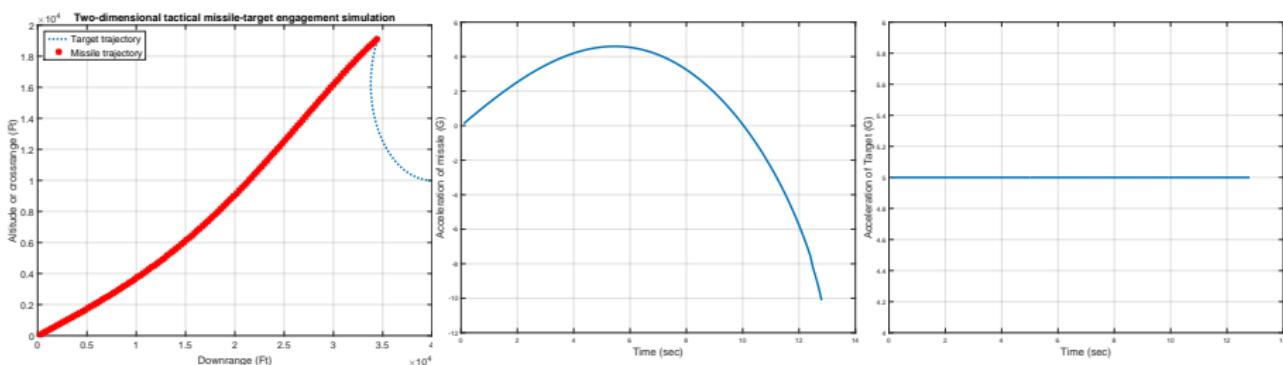


Attacker (missile) acceleration.

- Increasing the effective navigation ratio causes heading error to be removed rapidly.
- The effective navigation ratio has an effect on the acceleration of the missile; the way that the missile will bleed energy.
- The total acceleration (area under the curve) is increasing inversely proportional with the effective navigation ratio  $N'$ .

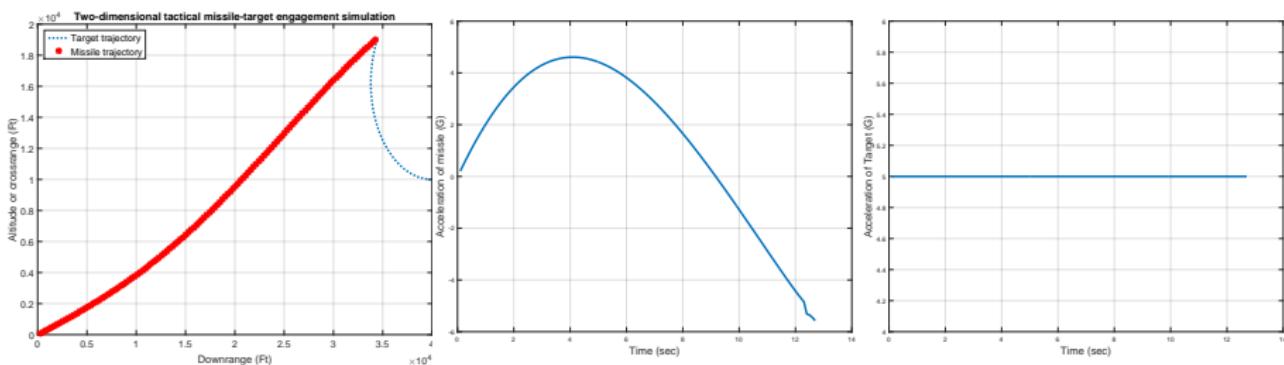
# Target Maneuver Cases

**Constant Target maneuver ( $XNT = 5G$ ,  $HE = 0$ ,  $N' = 3$ )**



# Target Maneuver Cases

**Constant Target maneuver ( $XNT = 5G$ ,  $HE = 0$ ,  $N' = 5$ )**



- Higher effective navigation ratio yields less acceleration to hit maneuvering target.
- This causes the missile to lead the target slightly more than a lower effective navigation ratio does.

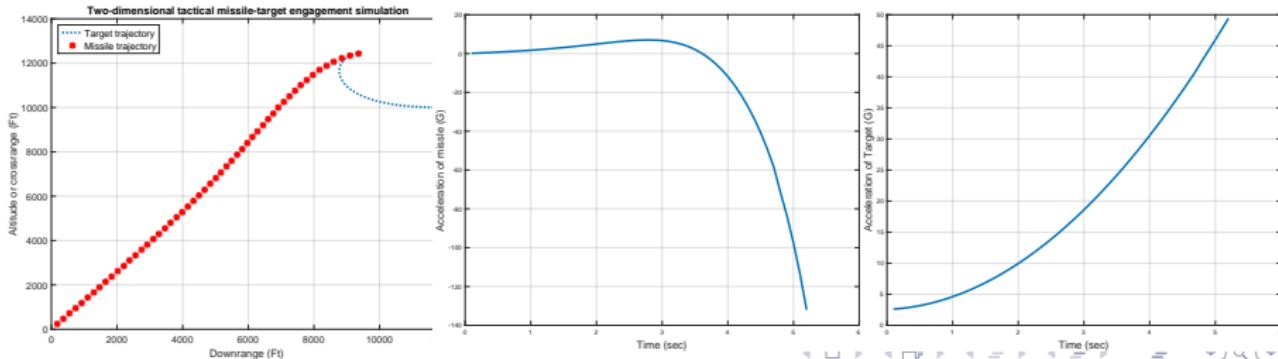
# Target Maneuver Cases (Optimization)

## Cost Function

- Missile acceleration
- Miss distance

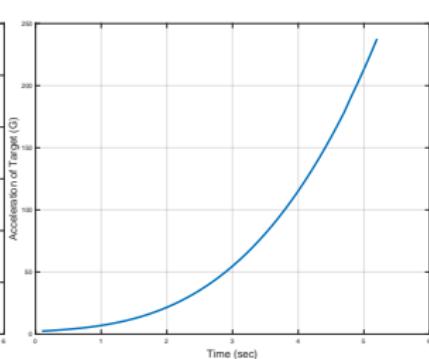
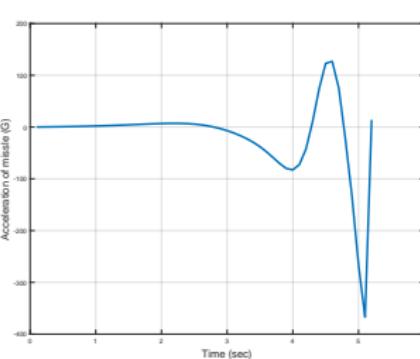
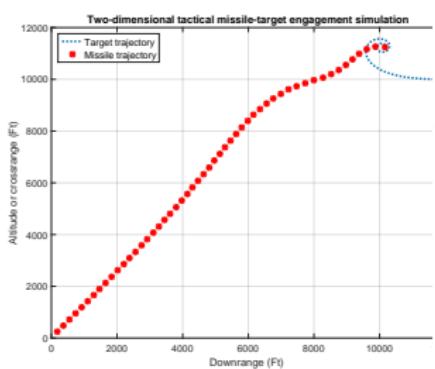
**Polynomial Target maneuver.** (polynomial degree=3 ,  $HE = 0$ ,  $N' = 3$ )

$$f(t) = c_0 + c_1 T + c_2 T^2 + c_3 T^3 + \dots + c_N T^N \quad (1)$$



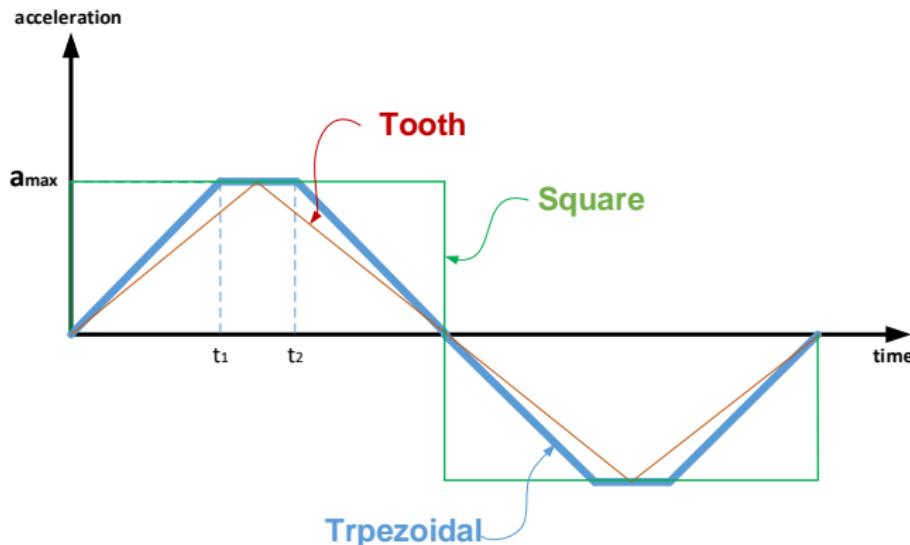
# Target Maneuver Cases (Optimization)

**Polynomial Target maneuver.** (polynomial degree=4 ,  
 $HE = 0$ ,  $N' = 3$ )



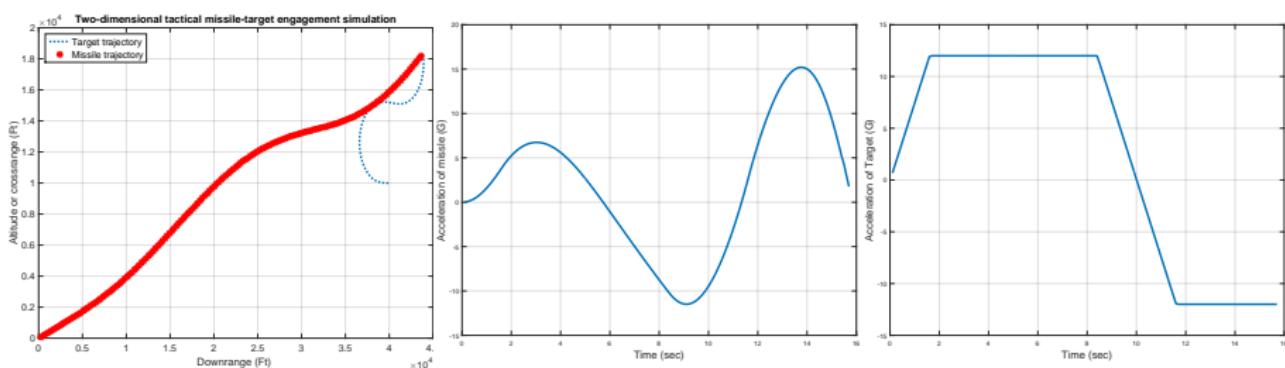
# Target Maneuver Cases

## Trapezoidal Target maneuver



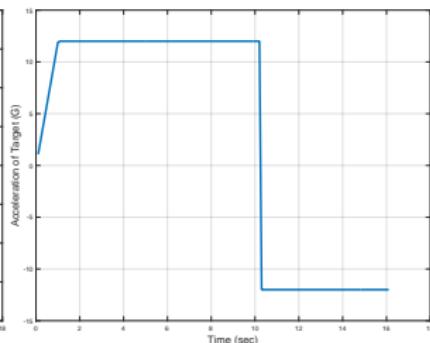
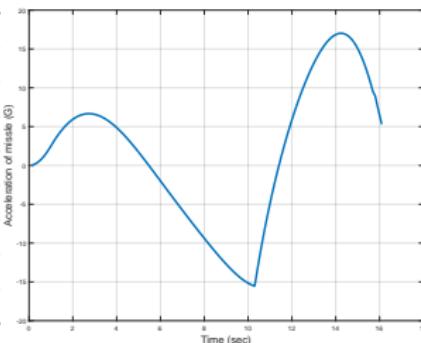
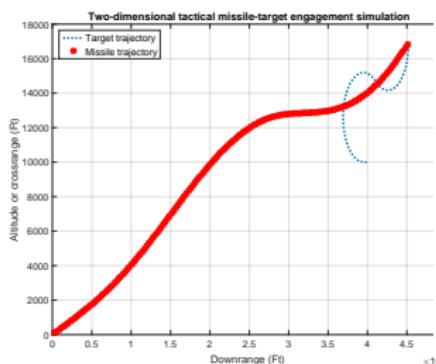
# Target Maneuver Cases

**Trapezoidal Target maneuver (Symmetric => Square, tooth, trapezoidal)**



# Target Maneuver Cases

## Trapezoidal Target maneuver (not Symmetric)



Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

**ToolBox**  
oooo

GA  
oooo

Unity Game  
oooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
ooooooo  
o

## ToolBox

# Guidance Toolbox

We establish a simple GUI (graphic user interface): the inputs are the locations and velocities for the Target (plane) and the Attacker (missile). The user could choose a guidance law (future work), which is the way that the Attacker tracking the Target, and the escape maneuver technique which is the way that Target evade the target.

Introduction



Target-Attacker



ToolBox



GA



Unity Game



Target-Attacker-Defender



Optimal Strategies



# Guidance Toolbox

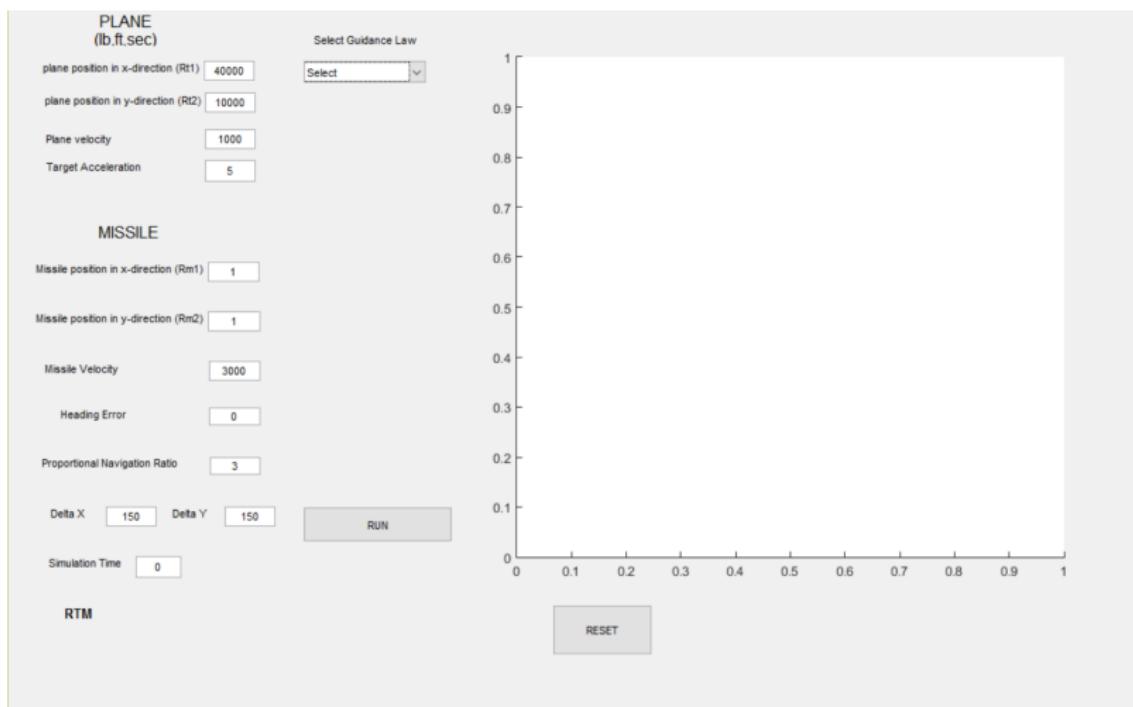


Figure: Guidance Toolbox

## Guidance Toolbox

Till now, if the user chooses the PN guidance law, there will be more options available, Now he could select the type of escape maneuver:

- Polynomial
- Trapezoidal
- Symmetric Trapezoidal

# Guidance Toolbox

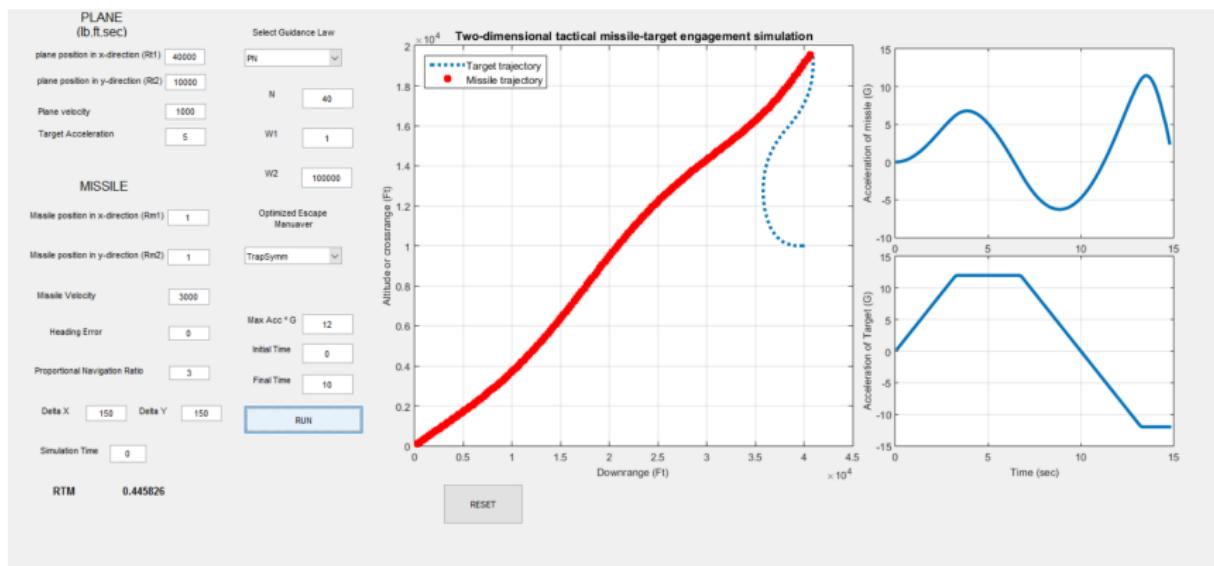


Figure: Guidance Toolbox options when PN selected

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooooooo  
o

# GA

# Genetic Algorithm Solution

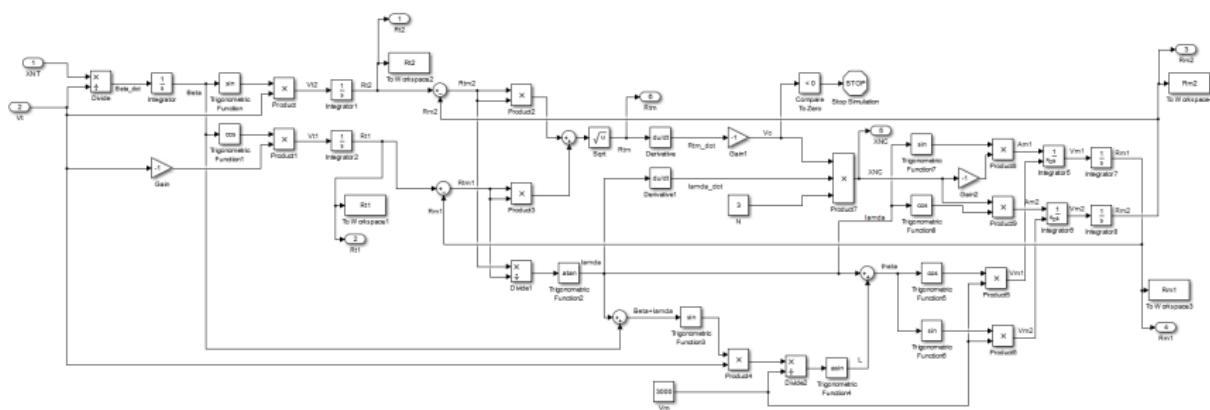


Figure: The Simulink model for proportional navigation equations.

# Genetic Algorithm Solution

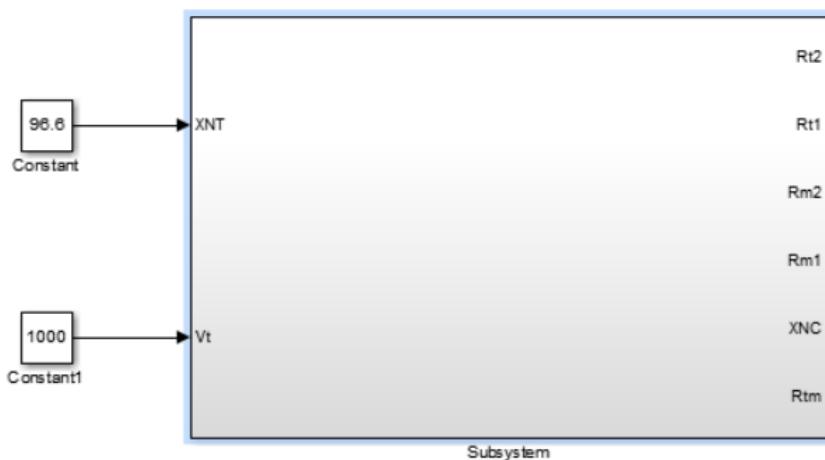


Figure: The Simulink main block for solving proportional navigation equations

Introduction



Target-Attacker



ToolBox



GA



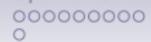
Unity Game



Target-Attacker-Defender



Optimal Strategies



# Genetic Algorithm Solution

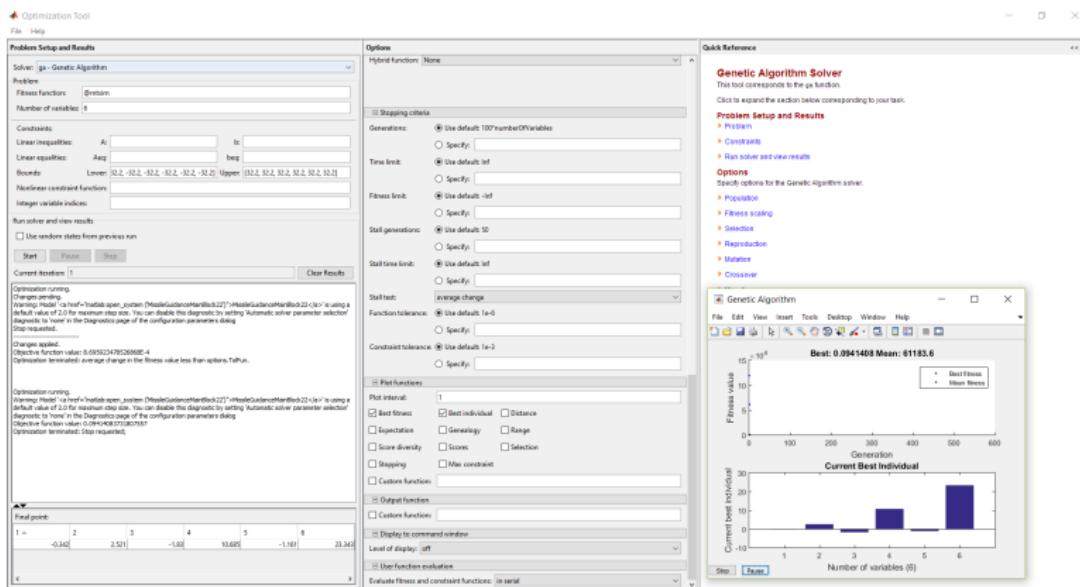


Figure: Genetic Algorithm Toolbox in Matlab.

# Genetic Algorithm Results

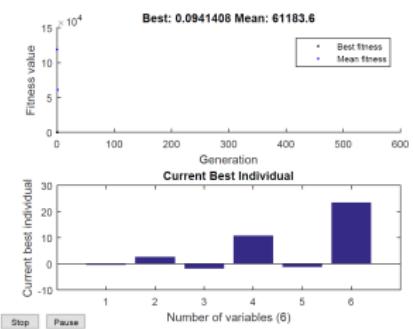


Figure: Best fitness values

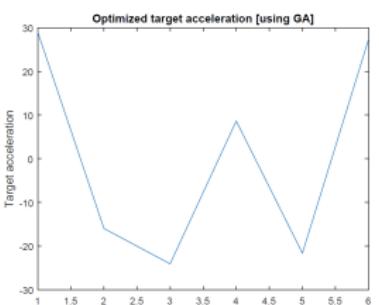


Figure: Optimized target acceleration using GA.

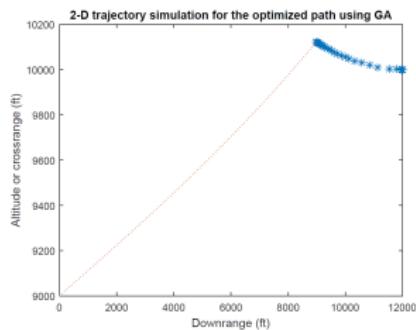


Figure: Optimized trajectory using GA.

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooooooo  
ooooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooooooo  
o

## Unity Game

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
●oooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooo  
o

## Summary of Solution

### Experimental Testing:

#### Unity Game

- We 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.

## Some types of games

**Competitions** between humans and computers in games of increasing difficulty, including the games of Chess , Arimaa , Jeopardy! , and Go. 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.

## Some types of games

The **gamification** phenomenon, 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.

## Some types of games

The paradigm of **games with a purpose**, 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

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooo●oooo  
oooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooo  
o

## Some types of games

**Human Computation**, which is the idea of using human 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.

## Some types of games

**Crowdsourcing**, 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.

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooo●ooo  
oooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooooooo  
o

## Unity Platform

Unity is a cross-platform game engine developed by Unity Technologies and used to develop video games for PC, consoles, mobile devices and websites. It is a free software (<https://unity3d.com/>), working with C# programming language.

Introduction



Target-Attacker



ToolBox



GA



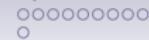
Unity Game



Target-Attacker-Defender



Optimal Strategies



# Unity Platform

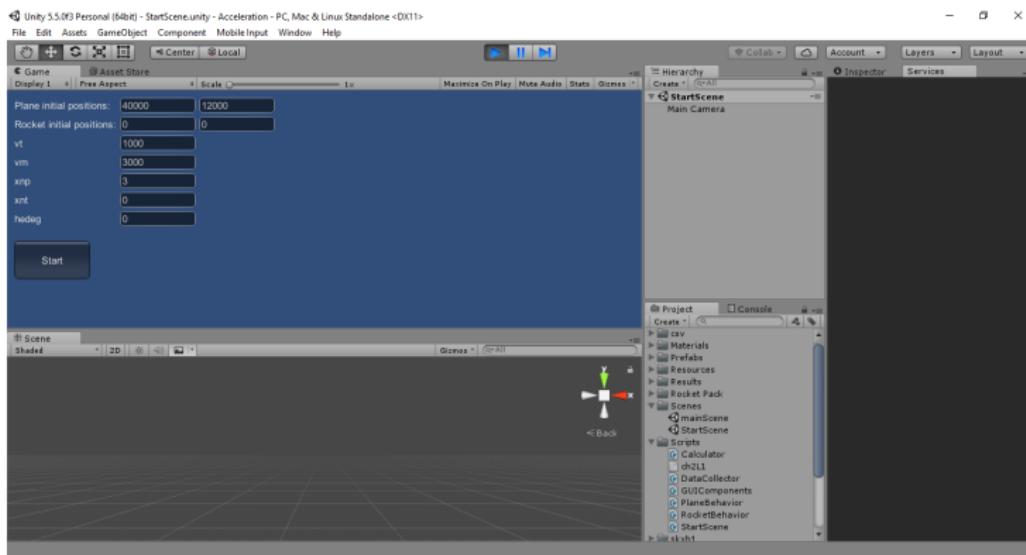


Figure: Unity game engine interface

# Unity Platform

Each game consists of two scenes:

1. Starting scene.
2. Game scene, which updated each frame per second.

In the starting scene there is data fields (position, velocity, ...).

After this scene ends, all the data are destroyed, so we save the information we need in file called "player prefs".

The game scene consists of some objects, each object has his own script, this script must contain 2 points:

- Start: initialization.
- Updated : every frame.

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooooooo  
●ooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooooooo  
o

# Unity Game Design

The game consists of two objects: plane (Target) and missile (Attacker). A human player controls the increasing and decreasing of the target acceleration (XNT) by two arrows on the keyboard. The missile object is moving on according to the proportional navigation guidance law.

Introduction

oo

Target-Attacker

oooooooooooo  
oooooooooooo

ToolBox

oooo

GA

oooo

Unity Game

oooooooooooo  
o●oooo

Target-Attacker-Defender

oooooooooooooooo  
oooooooooooooooooooo

Optimal Strategies

oooooooooooo  
o

# Unity Game Design

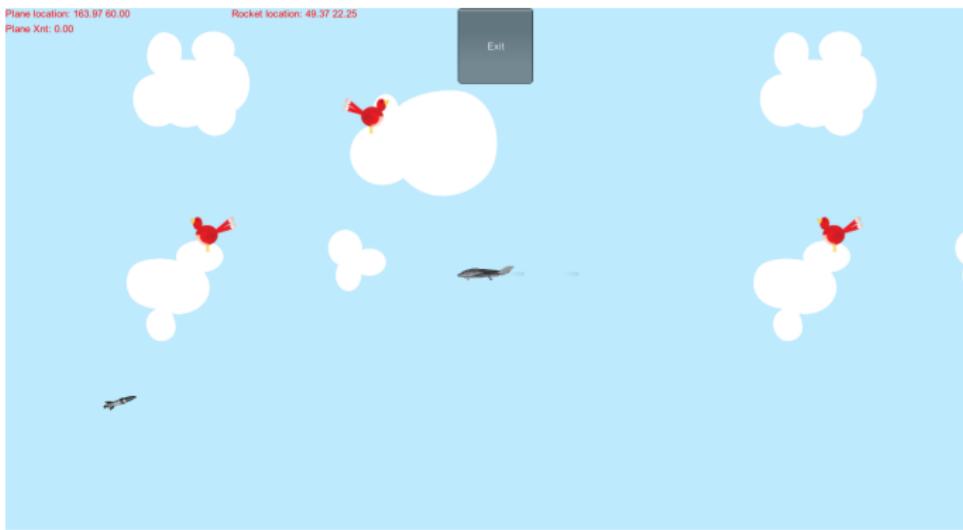


Figure: Unity game for simulating Target-Attacker engagement

# Unity Game Design

In our game we have 3 objects:

1. **Plane:** its script contain some commands to control the plane with two arrows in the keyboard, which increase and decrease the target acceleration by upward arrow and downward arrow respectively.
2. **Missile:** it does not contain a script, the equations controlling its behavior is in the script with the "Data collector" object.
3. **Object "Data collector"**

# Unity Game Design

## Object "Data collector":

- Start
  1. initialization of the variables in the equations.
  2. set the location of all objects (plane and missile).
- Update
  1. get plane location.
  2. execute your equations.
  3. update rocket location (according to PN equations).
  4. update informations to be printed to excel.
  5. check the breaking condition, if true, load starting scene.

# Results

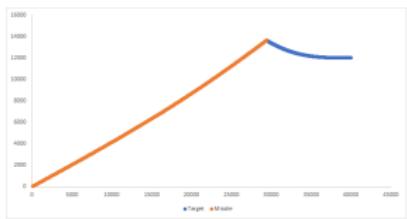


Figure:  
Target-Attacker 2D  
trajectory.

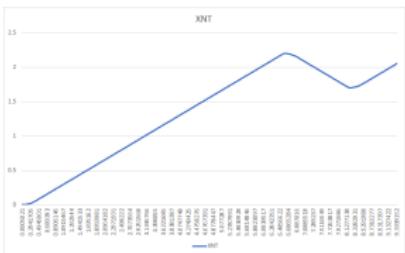


Figure: Optimized  
target acceleration  
versus time.

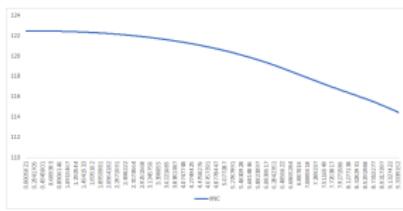


Figure: Missile  
acceleration versus  
time.

## Results

Our experience with Evasion-1 demonstrates that the game is working correctly. Figures 12-14 illustrate some of our preliminary results, obtained by the player with the best score so far. The target acceleration curve in Fig. 13 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.

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooooooo  
oooooo

**Target-Attacker-Defender**  
oooooooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooooooo  
o

## Target-Attacker-Defender

# Summary of Solution

## Scenario 2:

### TAD: Target-Attacker-Defender

- A unified analysis is presented via the construction of two Apollonius circles, considering all possibilities of the ratio between the speeds of the attacker and defender.
- We obtain the critical target speed and the Voronoi diagram bordering the safe or escape region for the target optimal strategies. Numerical results and plots allow useful and insightful qualitative interpretations.
- Optimal heading angles to be followed by the target to stay in the safe region. We use Hamiltonian equations to formulate an exact two-point boundary value problem that is solved numerically, verifying our earlier results.

# Main References



Pachter Meir, and Garcia, Eloy and Casbeer, David W (2014)

Active target defense differential game

*IEEE 52nd Annual Allerton Conference on Communication, Control, and Computing*, 46 – 53.



Garcia, Eloy and Casbeer, David W and Pachter, Meir (2015)

Active target defense differential game with a fast defender

*arXiv preprint arXiv:1502.02747*.



Garcia, Eloy and Casbeer, David W and Pachter, Meir (2015)

Escape Regions of the Active Target Defense Differential Game

*arXiv preprint arXiv:1504.07900*.

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooooooo  
oooooo

Target-Attacker-Defender  
oo●oooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooooooo  
o

# Applications

Area	Agent 1	Agent 2	Agent 3
Aerospace	Target	Attacker(missile)	Defender(missiles)
Biology	Prey	Predator	Protector
Society	Lady	Bandits	Bodyguards
Criminology	Robber	Policemen/Cops	Gangsters

# Apollonius Circles

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 ( $r \rightarrow \infty$ ), the circle degenerates into a straight line.

## Apollonius Circle

The locus of a point moving such that the ratio of its distances from two fixed points A and B is a constant  $k$ .

$$\frac{AI}{IB} = \frac{AE}{EB} = k, \quad \{k \neq 1\}. \quad (2)$$

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  $k$  ( $k \neq 1$ ).

Express  $I$  and  $E$  in terms of  $\mathbf{A}$  and  $\mathbf{B}$  as

$$I = \frac{1}{k+1}(\mathbf{A} + k\mathbf{B}), \quad (3)$$

$$E = \frac{1}{k-1}(-\mathbf{A} + k\mathbf{B}). \quad (4)$$

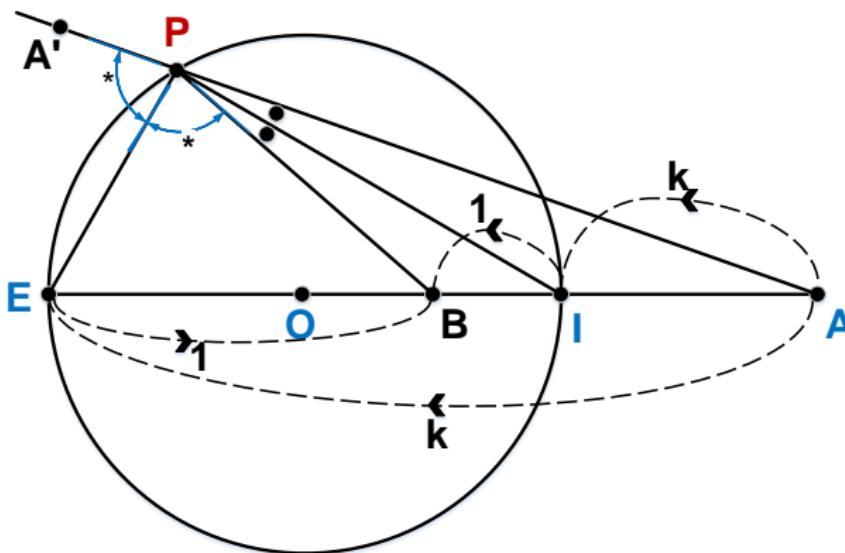


Figure: Apollonius circle for a moving point  $P$  such that  $\frac{AP}{PB} = k > 1$ .  
 Here  $m\angle API = m\angle BPI$  and  $m\angle A'PE = m\angle BPE$ .

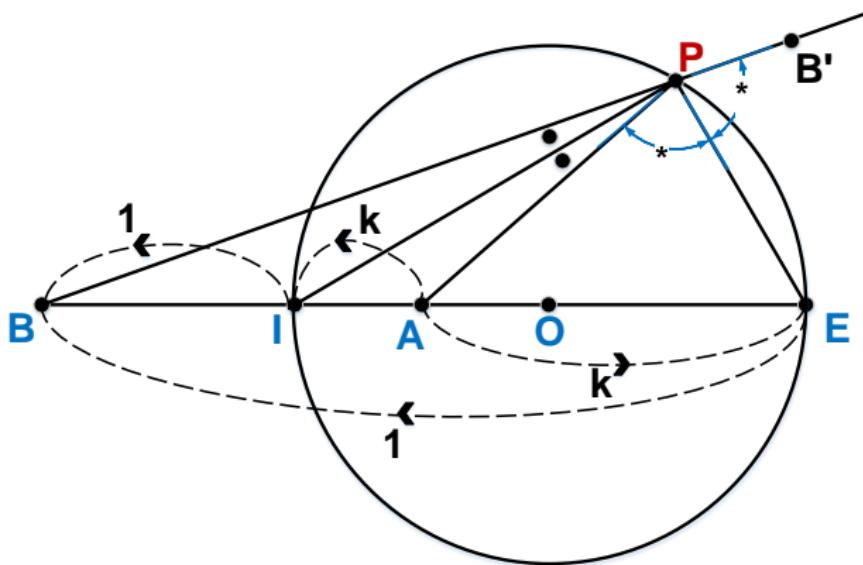


Figure: Apollonius circle for a moving point  $P$  such that  $\frac{AP}{PB} = k > 1$ .  
 Here  $m\angle API = m\angle BPI$  and  $m\angle A'PE = m\angle BPE$ .

## Center of the Apollonius circle

center of the circle is the midpoint of points  $I$  and  $E$

$$\begin{aligned} \mathbf{O} &= \frac{1}{2}(\mathbf{I} + \mathbf{E}) \\ &= \frac{1}{2}\left[\left(\frac{1}{k+1} - \frac{1}{k-1}\right)\mathbf{A} + k\left(\frac{1}{k+1} + \frac{1}{k-1}\right)\mathbf{B}\right] \quad (5) \\ &= -\frac{1}{k^2-1}\mathbf{A} + \frac{k^2}{k^2-1}\mathbf{B}, \end{aligned}$$

## Radius of the Apollonius circle

the radius of the circle is half the length of the displacement from  $I$  to  $E$ ,

$$\begin{aligned} r &= \frac{1}{2}|I - E| \\ &= \frac{1}{2}\left|\left(\frac{1}{k+1} + \frac{1}{k-1}\right)\mathbf{A} + k\left(\frac{1}{k+1} - \frac{1}{k-1}\right)\mathbf{B}\right| \\ &= \left|\frac{k}{k^2-1}\mathbf{A} - \frac{k}{k^2-1}\mathbf{B}\right| \\ &= \left|\frac{k}{k^2-1}\right|\left|\mathbf{A} - \mathbf{B}\right| = \frac{k}{k^2-1}(AB). \end{aligned} \tag{6}$$

## case of $k = 1$

It is clear from (5) and (6) that  $\lim_{k \rightarrow 1} |O| \rightarrow \infty$  and  $\lim_{k \rightarrow 1} r \rightarrow \infty$ , and hence for  $k = 1$ , the Apollonius circle degenerates into a straight line, namely the perpendicular bisector of the straight segment  $\overline{AB}$ .

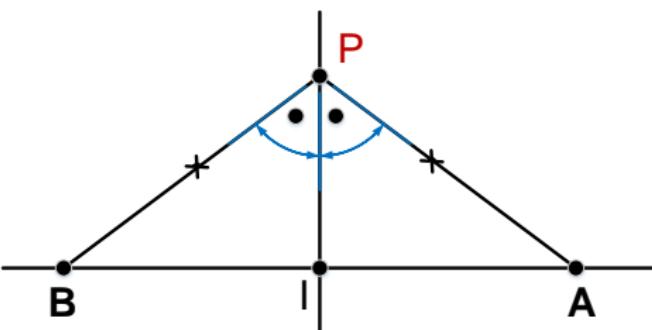


Figure: For  $k = 1$ , the Apollonius circle in the previous figures degenerates into the perpendicular bisector of the straight segment  $\overline{AB}$ . The point  $E$  disappears in this figure as it goes to  $\infty$

## The *AD* Apollonius circle

For the *AD* Apollonius circle, the two fixed points are the initial positions of the Attacker  $\mathbf{A} = (x_A, 0)$  and the initial position of the defender  $\mathbf{D} = (-x_A, 0)$ .  $k$  is replaced by:

$$\gamma = \frac{V_A}{V_D}. \quad (7)$$

Substituting the values of  $\mathbf{A}$  and  $\mathbf{D}$  above for  $\mathbf{A}$  and  $\mathbf{B}$  in (3),(4),(5) and (6), respectively, and replacing  $k$  therein by  $\gamma$ ,

$$\mathbf{I}_1 = \left( \frac{1 - \gamma}{1 + \gamma} x_A, 0 \right), \quad (8)$$

$$\mathbf{E}_1 = \left( \frac{1 + \gamma}{1 - \gamma} x_A, 0 \right), \quad (9)$$

$$\mathbf{O}_1 = \left( \frac{1 + \gamma^2}{1 - \gamma^2} x_A, 0 \right), \quad (10)$$

$$r_1 = \frac{2\gamma}{|1 - \gamma^2|} x_A. \quad (11)$$

## The *TA* Apollonius circle

here, the two fixed points are the initial positions of the Target and Attacker  $\mathbf{T} = (x_T, y_T)$  and  $\mathbf{A} = (x_A, 0)$ . Again,  $k$  is replaced by:

$$\alpha = \frac{V_T}{V_A}. \quad (12)$$

Now, we substitute the values of  $\mathbf{T}$  and  $\mathbf{A}$  above for  $\mathbf{A}$  and  $\mathbf{B}$  in (3),(4),(5) and (6), respectively, and replace  $k$  therein by  $\alpha$

$$\mathbf{I}_2 = \frac{1}{1+\alpha}(\mathbf{T} + \alpha\mathbf{A}) = \frac{1}{1+\alpha}(x_T + \alpha x_A, y_T), \quad (13)$$

$$\mathbf{E}_2 = \frac{1}{\alpha-1}(-\mathbf{T} + \alpha\mathbf{A}) = \frac{1}{1-\alpha}(x_T - \alpha x_A, y_T), \quad (14)$$

$$\mathbf{O}_2 = \frac{1}{1-\alpha^2}(\mathbf{T} - \alpha^2\mathbf{A}) = \frac{1}{1-\alpha^2}(x_T - \alpha^2 x_A, y_T), \quad (15)$$

$$r_2 = \frac{\alpha}{1-\alpha^2}(TA) = \frac{\alpha d}{1-\alpha^2} \quad (16)$$

## Target survival and 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, 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.

An implicit assumption throughout the forthcoming analysis is that  $\mathbf{T} = (x_T, y_T)$  is outside  $R_r$ .

# Target survival and Critical Speed Ratio

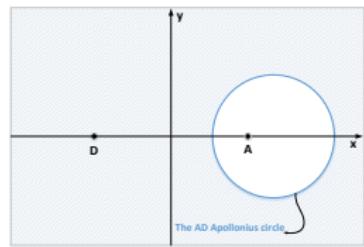


Figure:  $\gamma < 1$

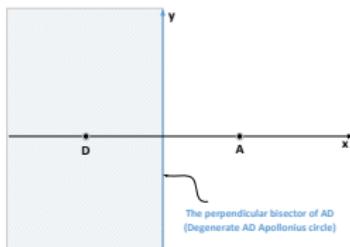


Figure:  $\gamma = 1$

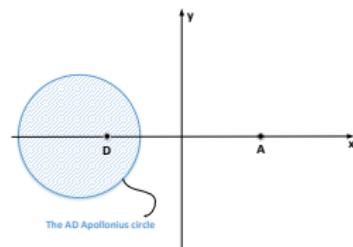


Figure:  $\gamma > 1$

$\bar{\alpha}$  is obtained when the TA Apollonius circle is tangent to the boundary of the shaded region  $R_r$

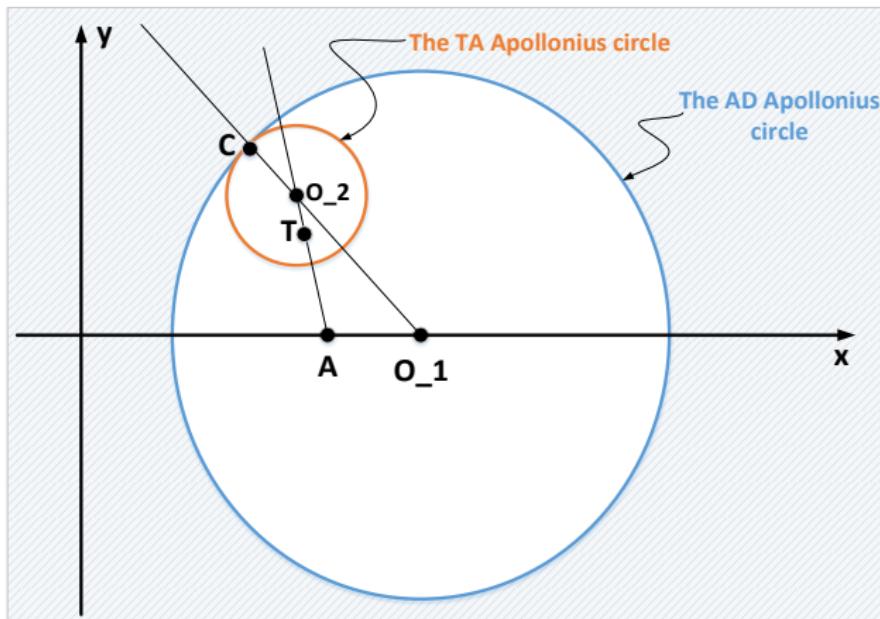


Figure:  $\gamma < 1$

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

$$r_1 - r_2 = |O_1 - O_2|. \quad (17)$$

Substituting for  $r_1, r_2, O_1$  and  $O_2$  from (11), (16), (11) and (15) respectively, and noting that  $\gamma < 1$ , one obtains

$$\begin{aligned} \frac{2\gamma}{1-\gamma^2}x_A - \frac{\alpha}{1-\alpha^2}d &= \left| \left( \frac{1+\gamma^2}{1-\gamma^2}x_A, 0 \right) - \frac{1}{1-\alpha^2}(x_T - \alpha^2x_A, y_T) \right| \\ &= \left[ \left( \frac{1+\gamma^2}{1-\gamma^2}x_A - \frac{1}{1-\alpha^2}(x_T - \alpha^2x_A) \right)^2 + \frac{1}{(1-\alpha^2)^2}y_T^2 \right]^{1/2}. \end{aligned} \quad (18)$$

after several steps of simplification

$$\alpha^2 + \frac{d}{\gamma x_A} \alpha + \left[ \left( \frac{1 - \gamma^2}{4\gamma^2} \right) \left( \frac{d}{x_A} \right)^2 - \frac{x_T}{x_A} \right] = 0 \quad (19)$$

Equation (19) is an improvement of equation (36) in [2]. while (19) is a quadratic equation in  $\alpha$ , equation (36) in [2] is a quartic in  $\alpha$  that has the same two roots of (19) in addition to the two irrelevant roots  $\alpha \mp 1$ . As a bonus, equation (19) is written in a self-verifying dimensionless form.

Introduction



Target-Attacker



ToolBox



GA



Unity Game



Target-Attacker-Defender



Optimal Strategies

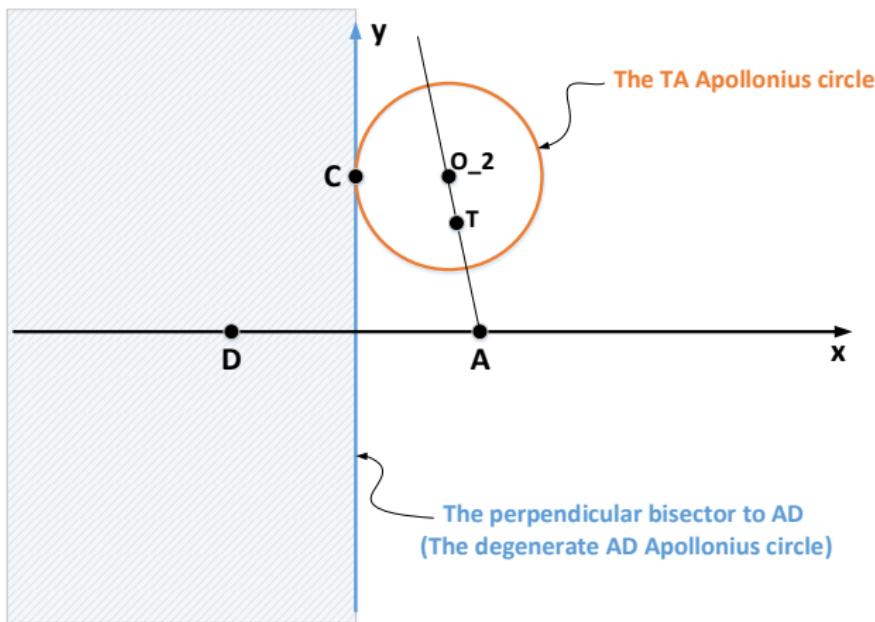
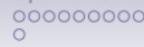


Figure:  $\gamma = 1$

The critical speed ratio  $\bar{\alpha}$  occurs when the *TA* Apollonius circle touches the *L.H.S.* of the *XY*-plane (Fig. 22), i.e., when

$$r_2 = \text{Absicessa of } O_2. \quad (20)$$

i.e., thanks to (15) and (16) when:

$$\frac{\alpha d}{1 - \alpha^2} = \frac{x_T - \alpha^2 x_A}{1 - \alpha^2}. \quad (21)$$

Multiplying (21) by  $(1 - \alpha^2)$  (thanks to the fact that  $\alpha \neq 1$ ), and rearranging, one obtains the following quadratic equation in  $\alpha$

$$x_A \alpha^2 + d\alpha - x_T = 0, \quad (22)$$

which is equation (12) in [3]. It is interesting to note that equation (21) is the special case ( $\gamma = 1$ ) of (19), though (19) was derived under the assumption  $\gamma \neq 1$ .

Introduction

oo

Target-Attacker

oooooooooooo  
oooooooooooo

ToolBox

oooo

GA

oooo

Unity Game

oooooooo  
oooooo

Target-Attacker-Defender

oooooooooooo  
oooooooo●oooooooooooo

Optimal Strategies

oooooooo  
o

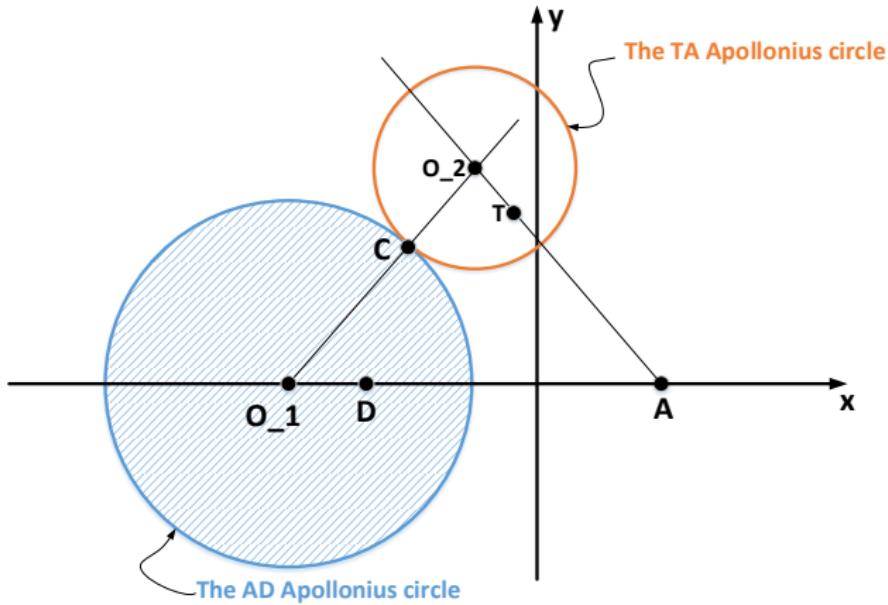


Figure:  $\gamma > 1$

Introduction  
ooTarget-Attacker  
oooooooooooo  
ooooooooooooToolBox  
ooooGA  
ooooUnity Game  
oooooooo  
ooooooTarget-Attacker-Defender  
oooooooooooo  
oooooooo●ooooooooooooOptimal Strategies  
oooooooo  
o

The critical speed ratio  $\bar{\alpha}$  occurs when the *TA* Apollonius circle is *externally tangent* to the *AD* Apollonius circle, i.e., when the centre  $O_2$ , the tangency point  $C$  and the centre  $O_1$  are collinear(Fig. 23), i.e., when

$$r_1 + r_2 = |O_1 - O_2| \quad (23)$$

Since  $\gamma > 1$ , we write

$$r_1 + r_2 = \frac{2\gamma}{|1 - \gamma^2|} + \frac{\alpha}{1 - \alpha^2} d = \frac{-2\gamma}{1 - \gamma^2} + \frac{\alpha}{1 - \alpha^2} d. \quad (24)$$

The above expression for  $(r_1 + r_2)$  is exactly the negative of  $(r_1 - r_2)$  in (18).

Introduction  
ooTarget-Attacker  
oooooooooooo  
ooooooooooooToolBox  
ooooGA  
ooooUnity Game  
oooooooo  
ooooooTarget-Attacker-Defender  
oooooooooooo  
oooooooo●ooooooooOptimal Strategies  
oooooooo  
o

Equation (19) is thereby shown to hold for a slow Defender besides being true for a fast one. Therefore, we will use the quadratic formula (19) for all values of  $\gamma$ , and will solve it to obtain  $\bar{\alpha}$  for any  $\gamma$  as

$$\begin{aligned}\bar{\alpha} &= \frac{1}{2} \left[ -\frac{d}{\gamma x_A} \pm \left[ \left( \frac{d}{\gamma x_A} \right)^2 - 4 \left( \frac{1-\gamma^2}{4\gamma^2} \right) \left( \frac{d}{x_A} \right)^2 + \frac{4x_T}{x_A} \right]^{1/2} \right] \\ &= \frac{1}{2\gamma x_A} [-d \mp \gamma \sqrt{d^2 + 4x_T x_A}] \\ &= \frac{1}{2\gamma x_A} [-\sqrt{(x_A - x_T)^2 + y^2} + \gamma \sqrt{(x_A + x_T)^2 + y_T^2}]\end{aligned}\tag{25}$$

Equation (25) was derived in [3] for  $\gamma < 1$  and in [1] for  $\gamma = 1$ , provided  $x_T > 0$ . It is shown herein to hold for all values in  $\gamma$ , including  $\gamma > 1$ . Note that  $\bar{\alpha}$  is definitely nonnegative and hence the minus sign in the solution for  $\bar{\alpha}$  is rejected.

## Pursuit-Evasion Voronoi Diagram

We develop a novel analytic expression for the pursuit-evasion Voronoi diagram 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.

we obtain the general Voronoi diagram by rewriting the critically equation (19) as a relation between  $y_T$  and  $x_T$ , we write

$$(4\gamma\alpha x_A)d = -(1 - \gamma^2)d^2 + 4\gamma^2 x_A x_T - 4\gamma^2 \alpha^2 x_A^2 \quad (26)$$

Introduction  
ooTarget-Attacker  
oooooooooooo  
ooooooooooooToolBox  
ooooGA  
ooooUnity Game  
oooooooo  
ooooooTarget-Attacker-Defender  
oooooooooooo  
oooooooooooo●ooooooooOptimal Strategies  
oooooooo  
o

finally we get,

$$\begin{aligned} p(x_T, y_T) = & 16\gamma^2 \alpha^2 x_A^4 \\ & + 16\gamma^2 \alpha^2 x_A^2 x_T^2 \\ & - 32\gamma^2 \alpha^2 x_A^3 x_T \\ & + 16\gamma^2 \alpha^2 x_A^2 y_T^2 \\ & - (1 - \gamma^2)^2 [x_A^4 + x_T^4 + y_T^4 + 2x_A^2 x_T^2 + 2x_A^2 y_T^2 + 2x_T^2 y_T^2] \\ & - (1 + \gamma^2)^2 * 4x_A^2 x_T^2 \\ & - 16\gamma^4 \alpha^4 x_A^4 \\ & + 2(1 - \gamma^4)(2x_A^3 x_T + 2x_A x_T^3 + 2x_A x_T y_T^2) \\ & - 8\gamma^2(1 - \gamma^2)\alpha^2(x_A^4 + x_A^2 x_T^2 + x_A^2 y_T^2) \\ & + 16\gamma^2(1 + \gamma^2)\alpha^2 x_A^3 x_T \end{aligned} \tag{27}$$

Though (27) 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 (27), 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 (27). 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 (27) is the general equation for the Voronoi diagram in our current *TAD* problem and it appears here for the first time.

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooo●ooooo

Optimal Strategies  
oooooooo  
o

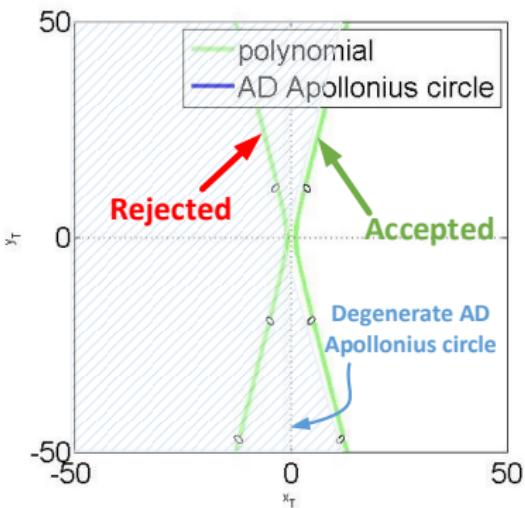


Figure: Generated computer output for the Voronoi diagram bordering the safe region for  $x_A = 4$ ,  $\alpha = 0.25$ ,  $\gamma = 1$  (the safe region is the shaded area)

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooo●oooo

Optimal Strategies  
oooooooo  
o

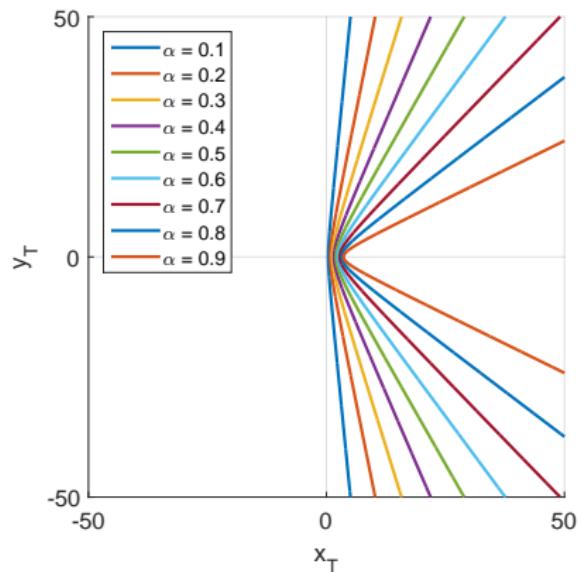


Figure: Various accepted branches of the voronoi diagram for  $\gamma = 1$  and  $\alpha$  as a parameter ranging from 0 to 1.

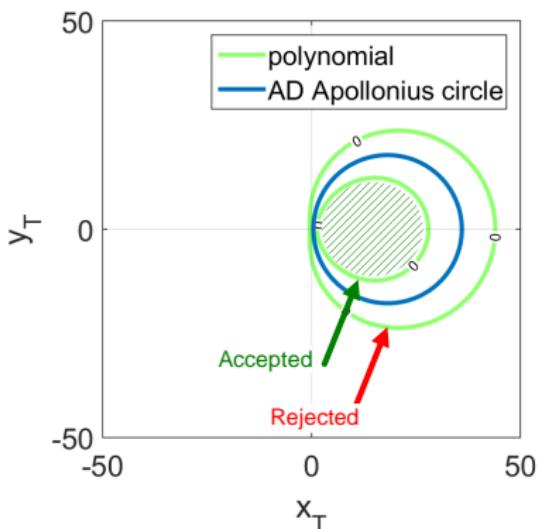


Figure: generated computer output for the Voronoi diagram bordering the safe region for  $x_A = 4$ ,  $\alpha = 0.25$ ,  $\gamma = 0.8$  (the safe region is the unshaded area) the quartic in (27) produces two closed curves: one outside the AD-Apollonius circle (rejected) and the other inside the circle (accepted as the Voronoi diagram)

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooo●oo

Optimal Strategies  
oooooooo  
o

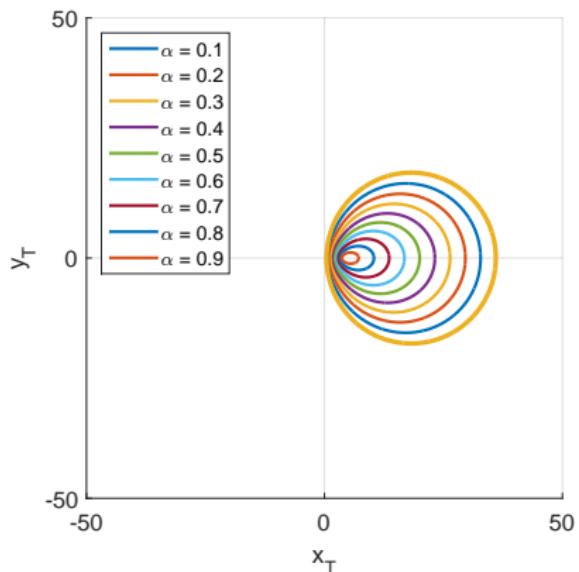


Figure: Various accepted branches of the voronoi diagram for  $\gamma = 0.8$  and  $\alpha$  as a parameter ranging from 0 to 1. These curves are computer generated from (27)

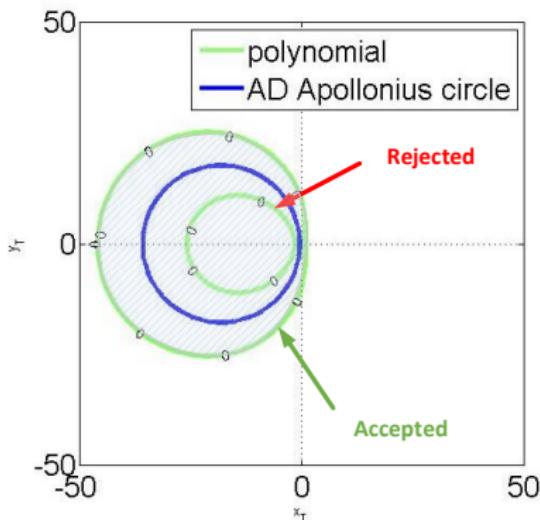


Figure: generated computer output for the Voronoi diagram bordering the safe region for  $x_A = 4$ ,  $\alpha = 0.25$ ,  $\gamma = 1.25$  (the safe region is the shaded area) the quartic in (27) produces two closed curves: one inside the AD-Apollonius circle (rejected) and the other outside the circle (accepted as the Voronoi diagram)

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooo  
oooooooooooo●

Optimal Strategies  
oooooooo  
o

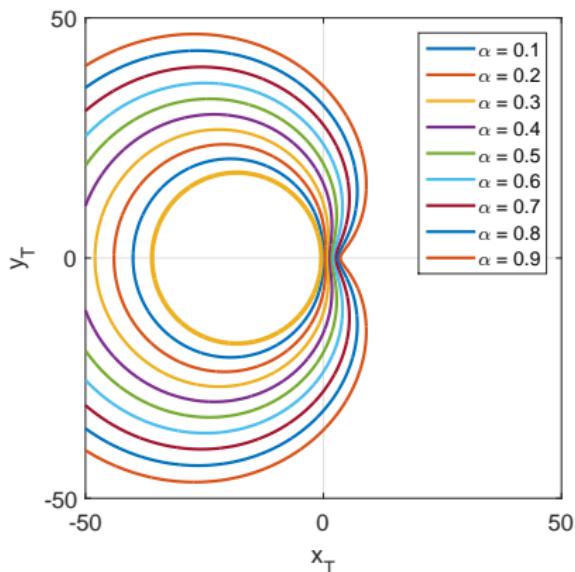


Figure: Various accepted branches of the voronoi diagram for  $\gamma = 1.25$  and  $\alpha$  as a parameter ranging from 0 to 1. These curves are computer generated from (27)

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooooooo  
oooooo

Target-Attacker-Defender  
oooooooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooooooo  
o

## Optimal Strategies

# Optimal Heading Angels

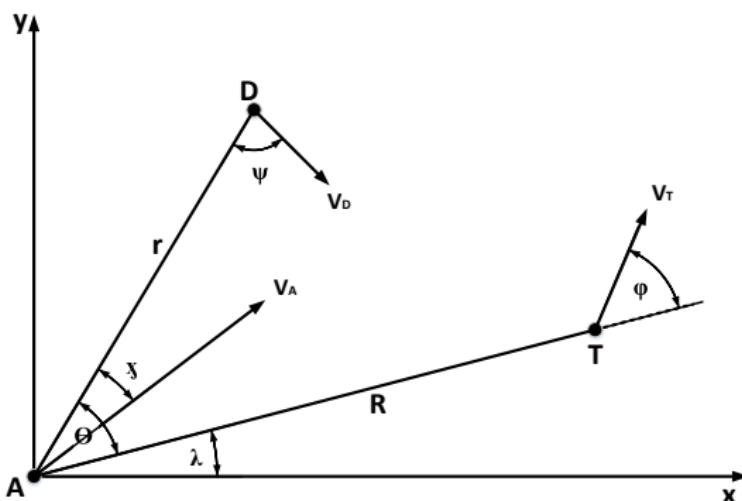


Figure: Scenario for the Attacker (A), Target (T), and Defender (D). The origin is arbitrary situated at the Attacker position (A).

# Optimal Heading Angels

State equations

$$\dot{R} = V_A(\alpha \cos \phi - \cos(\theta - \chi)) \quad (28)$$

$$\dot{r} = V_A(-\cos \chi - \gamma \cos \psi) \quad (29)$$

$$\dot{\theta} = V_A[-\frac{\alpha}{R} \sin \phi + \frac{1}{R} \sin(\theta - \chi) - \frac{\gamma}{r} \sin \psi + \frac{1}{r} \sin \chi] \quad (30)$$

initial conditions

$$R(t_0) = R_0,$$

$$r(t_0) = r_0,$$

$$\theta(t_0) = \theta_0,$$

## Optimal Heading Angels

The objective of the Target-Defender team is to maximize the separation between the Target and the Attacker at the interception time  $R(t_f)$ , where the terminal time  $t_f$  is free, such that  $r(t_f) = r_c$ . The objective of the Attacker is to minimize the same distance  $R(t_f)$ . This can be expressed as

$$\max_{\phi, \psi} \min_{\chi} J = \int_{t_0}^{t_f} \dot{R} dt$$

# Optimal Heading Angels

The Hamiltonian is

$$\begin{aligned} H = & \cos(\theta - \chi) - \alpha \cos \phi \\ & + [\alpha \cos \phi - \cos(\theta - \chi)] \lambda_R \\ & - [\cos \chi + \cos \psi] \lambda_r \\ & + [-\frac{\alpha}{R} \sin \phi + \frac{1}{R} \sin(\theta - \chi) - \frac{1}{r} \sin \psi + \frac{1}{r} \sin \chi] \lambda_\theta \end{aligned} \tag{31}$$

# Optimal Heading Angels

the co-state equations

$$-\frac{\partial H}{\partial R} = \dot{\lambda}_R = \frac{\lambda_\theta}{R^2}(\sin(\theta - \chi) - \alpha \sin(\phi)) \quad (32)$$

$$-\frac{\partial H}{\partial r} = \dot{\lambda}_r = \frac{\lambda_\theta}{r^2}(\sin(\chi) - \alpha \sin(\psi)) \quad (33)$$

$$-\frac{\partial H}{\partial \theta} = \dot{\lambda}_\theta = (1 - \lambda_R) \sin(\theta - \chi) - \frac{\lambda_\theta}{R} \cos(\theta - \chi) \quad (34)$$

the terminal conditions

$$\begin{aligned} r(t_f) &= r_c \\ \lambda_R(t_f) &= 0 \\ \lambda_\theta(t_f) &= 0 \end{aligned} \quad (35)$$

$$\alpha^2 + 2(\alpha + \cos \theta(t_f))\lambda_R(t_f) - 1 = 0$$

# Optimal Heading Angels

optimal control headings

$$\tan \psi^* = \frac{\lambda_\theta / r}{\lambda_r} \quad (36)$$

$$\tan \phi^* = \frac{(\lambda_\theta / R)}{1 - \lambda_R} \quad (37)$$

$$\tan \chi^* = \frac{\chi_s}{\chi_c} \quad (38)$$

where

$$\chi_s = (1 - \lambda_R) \sin \theta - \frac{\lambda_\theta}{R} \cos \theta + \frac{\lambda_\theta}{r}$$

$$\chi_c = (1 - \lambda_R) \cos \theta + \frac{\lambda_\theta}{R} \sin \theta - \lambda_r$$

## Optimal Heading Angels

The expressions for the optimal heading angles (36), (37), and (38) are to be used in the numerical solution of the Two-Point Boundary Value Problem (TPBVP). That solution is found by substituting the optimal headings into the state equations (28, 29, 30), and the co-state equations (32, 33, 34) with the terminal conditions (35 ).

# Optimal Heading Angels

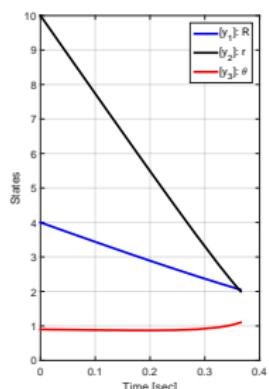


Figure: States

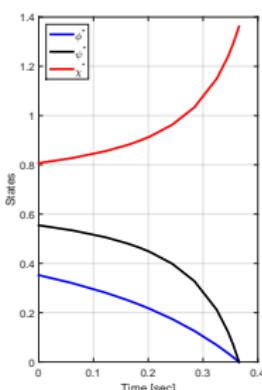


Figure: Optimal heading angles

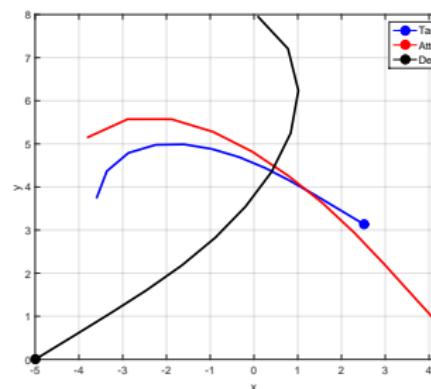


Figure: Optimal trajectories

## Concluding Remarks

In this work, we presented:

- 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 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. Also it is open source for development.
- We offer a unified analytic treatment of the TAD problem via the construction of two Apollonius circles, considering all possibilities of the ratio between the speeds of the attacker and defender (the case of slow defender is novel).
- We define the escape region for the 3 cases.

Introduction  
oo

Target-Attacker  
oooooooooooo  
oooooooooooo

ToolBox  
oooo

GA  
oooo

Unity Game  
oooooooooooo  
ooooooo

Target-Attacker-Defender  
oooooooooooooooo  
oooooooooooooooooooo

Optimal Strategies  
oooooooooooo  
●

# Thank you!