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**

June 15, 2017

Overview

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 escpae 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.

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.

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.

Summary of Solution

Experimental Testing:

Unity Game

- 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.

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

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.

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.
- 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.
- 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.

Evasion Tactics

1. **If missile is fired head-on at BVR range:**
   - 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. **Climb (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. **Fourth tactic:**
   - place the missile at 3 oclock or 9 oclock 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.

Gudiance Laws

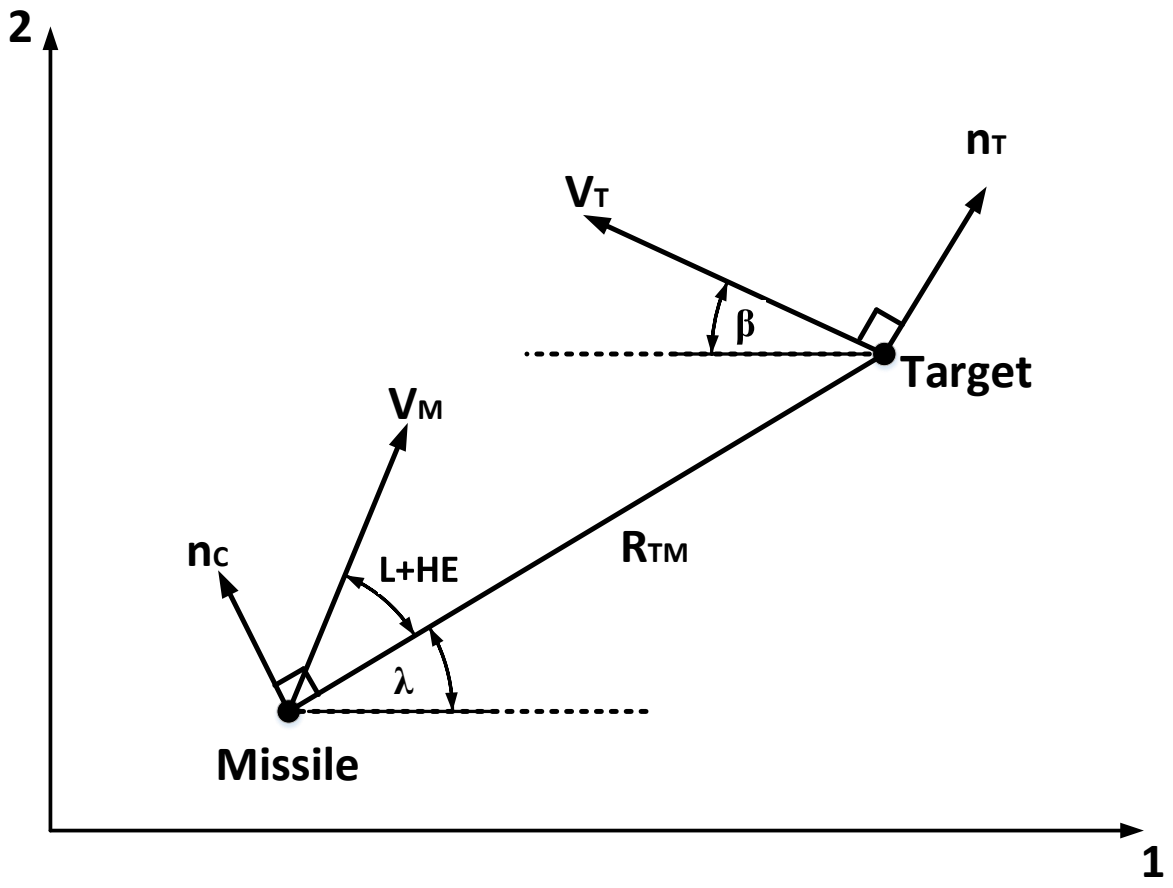
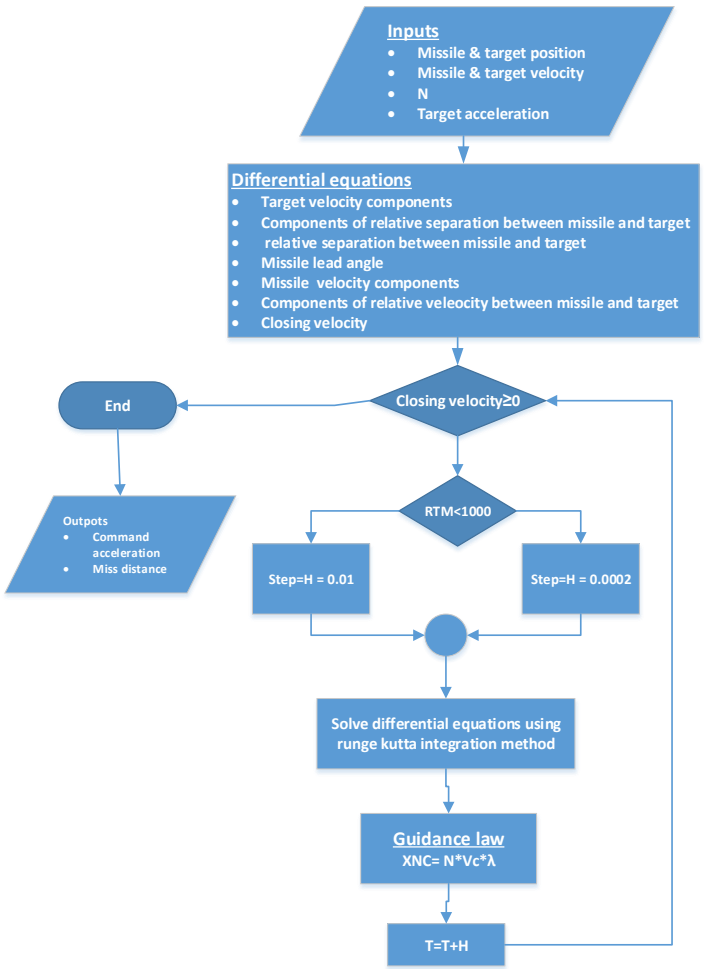


Figure: Two dimensional Missile-Target engagement geometry.

Maneuver Optimization



Target Maneuver Cases

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

Target Maneuver Cases

- Zero Target maneuver

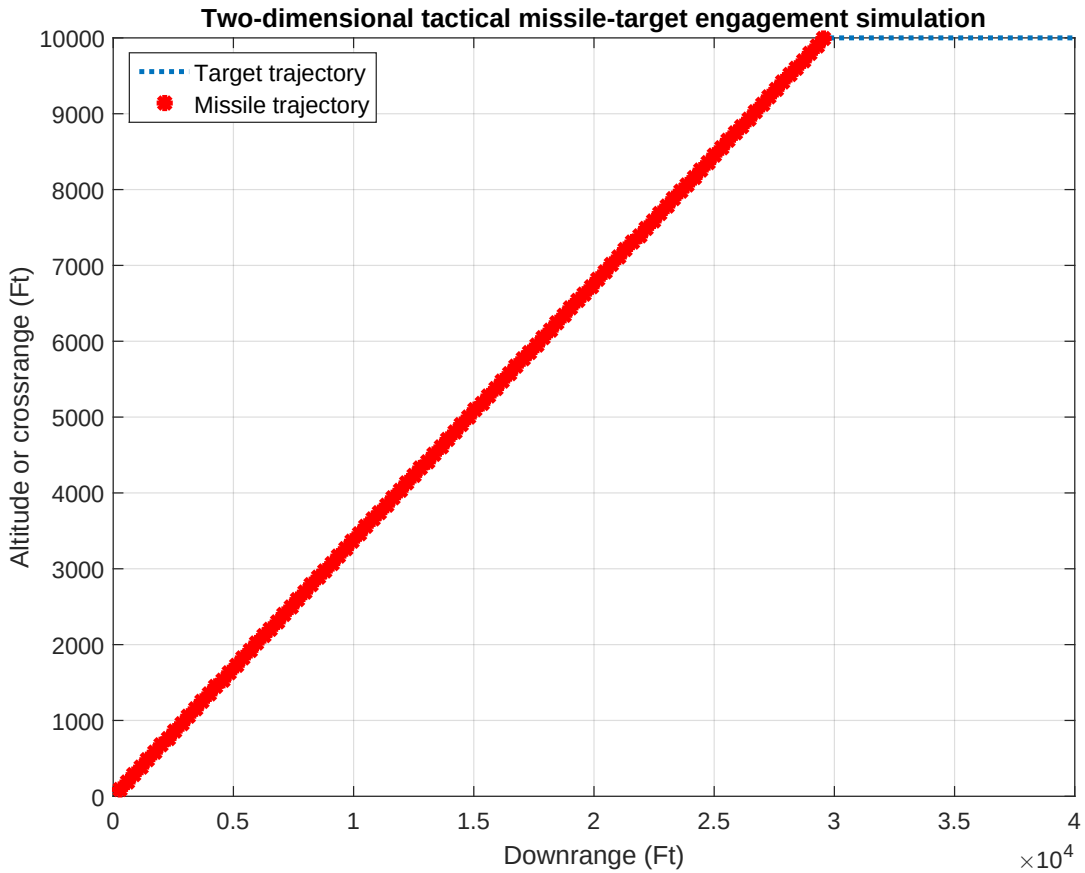


Figure: Trajectory of the target and attacker in case of zero target maneuver with zero heading error and $N' = 4$.

Target Maneuver Cases

- Constant Target maneuver

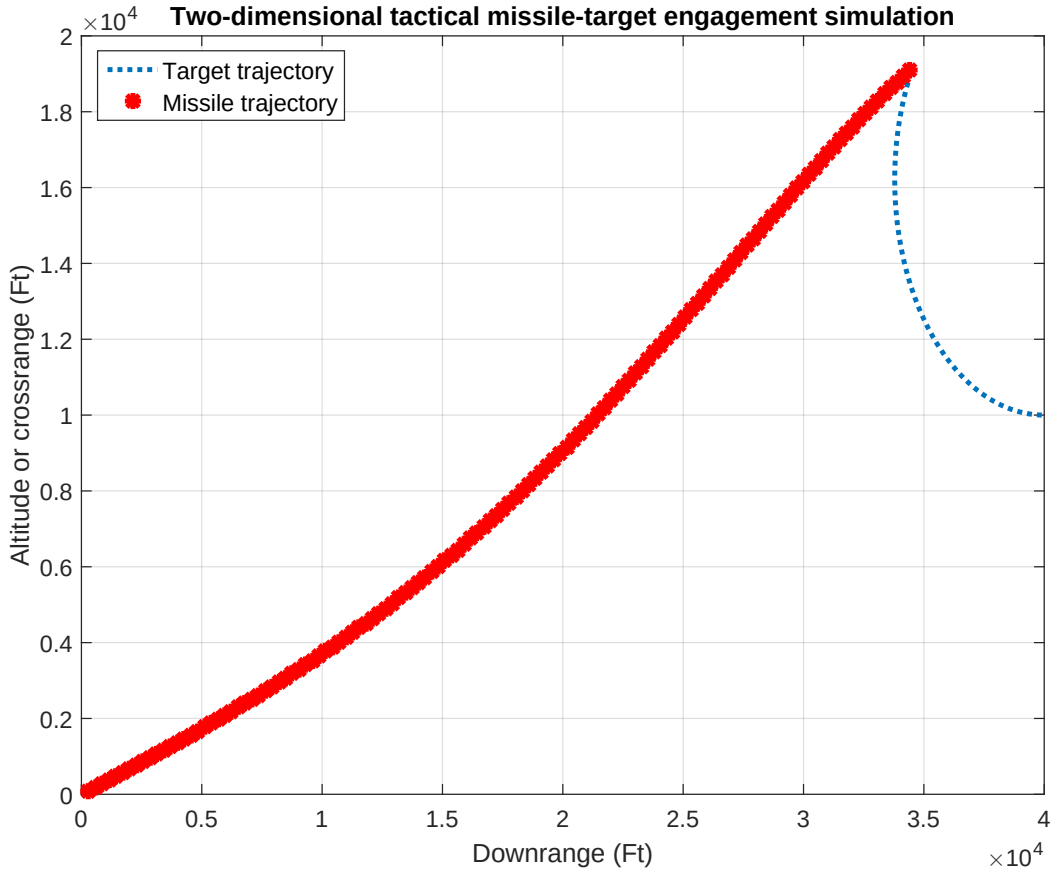


Figure: Trajectory of the target and attacker in case of constant target maneuver=5G with heading error=0 and $N' = 3$.

Target Maneuver Cases

- Polynomial Target maneuver

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

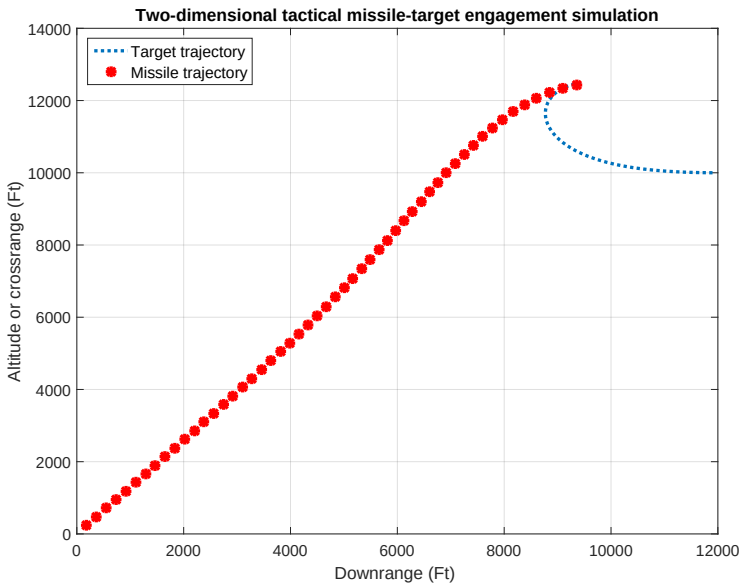


Figure: Trajectory of the target and attacker in case of polynomial of degree N=3 target maneuver with zero heading error and $N' = 3$.

Target Maneuver Cases

- Trapezoidal Target maneuver

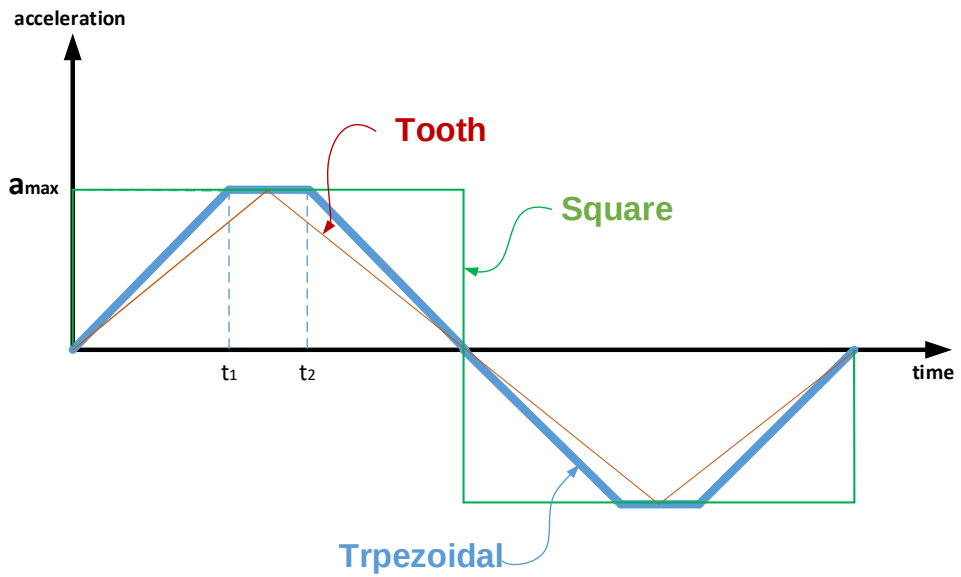


Figure: Trapezoidal Target maneuver.

Guidance Toolbox

A simple GUI (graphic user interface): the inputs are the locations and velocities for the Target (plane) and the Attacker (missile). Then the user have to choose the guidance law, which is the way that the Attacker tracking the Target, and there is 3 options:

- Proportional Navigation (PN)
- Augmented Proportional Navigation
- Optimal Guidance

Guidance Toolbox

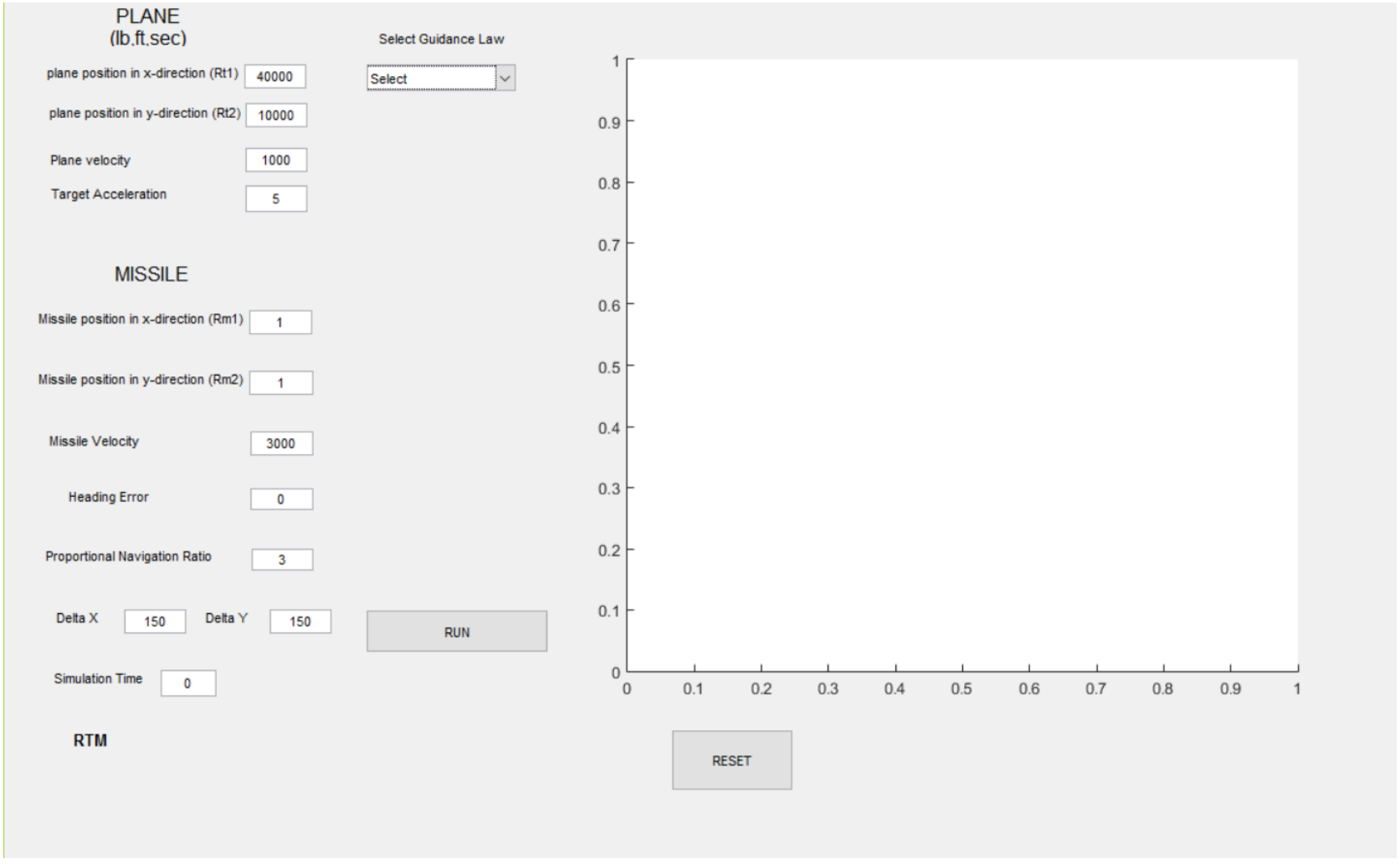


Figure: Guidance Toolbox

Guidance Toolbox

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

- Polynomial
- Trapezoidal
- Symmetric Trapezoidal

Guidance Toolbox

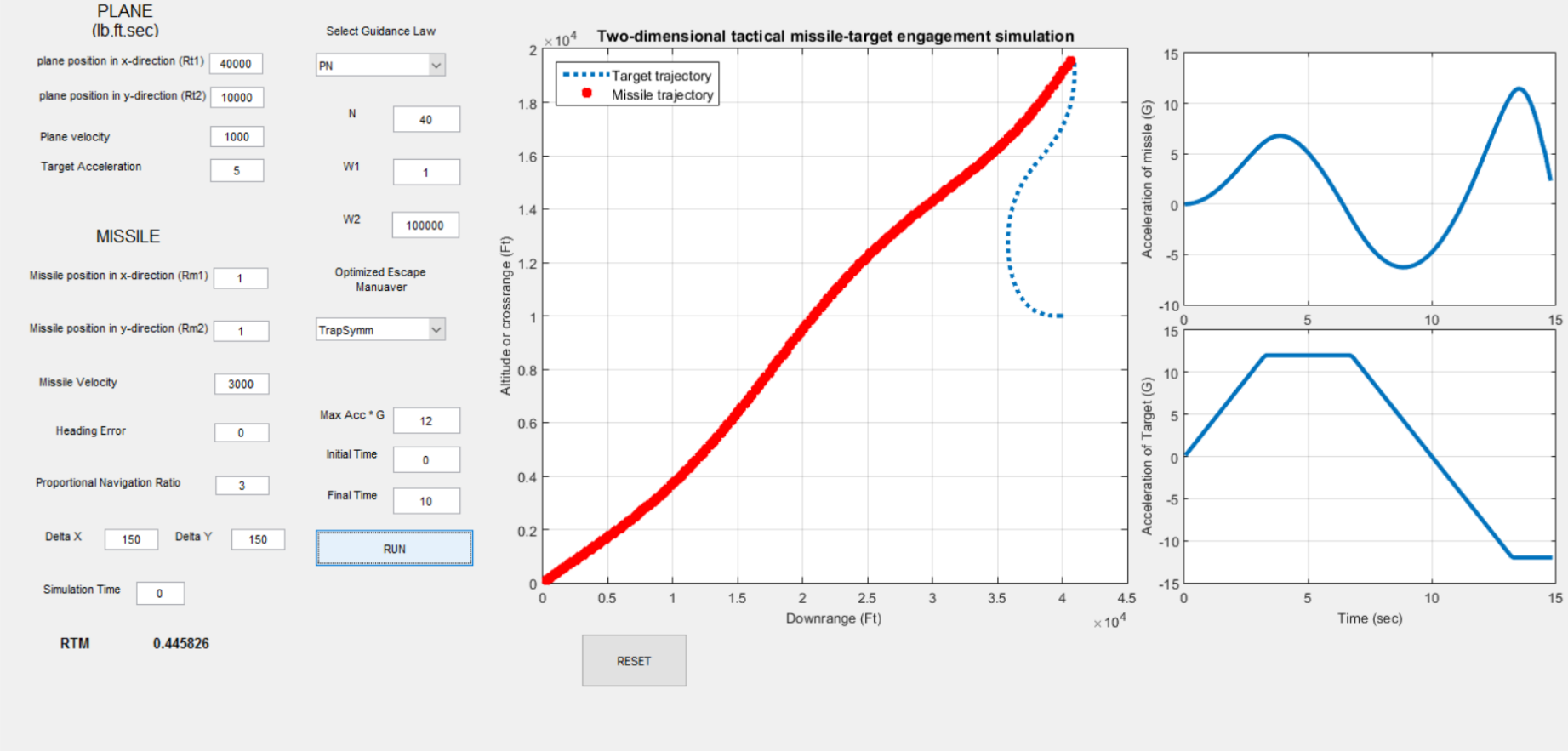


Figure: Guidance Toolbox options when PN selected

Genetic Algorithms

Most Important Parameters in GAs:

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

Genetic Algorithms

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.

Genetic Algorithms

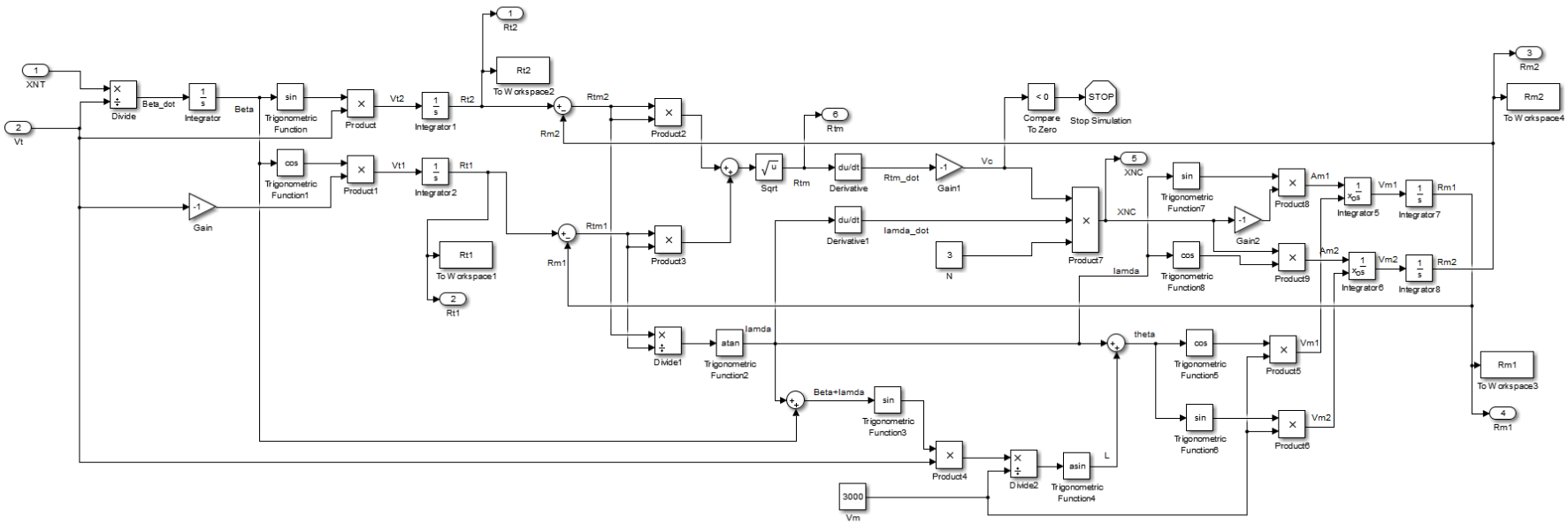


Figure: The Simulink model for proportional navigation equations.

Genetic Algorithms

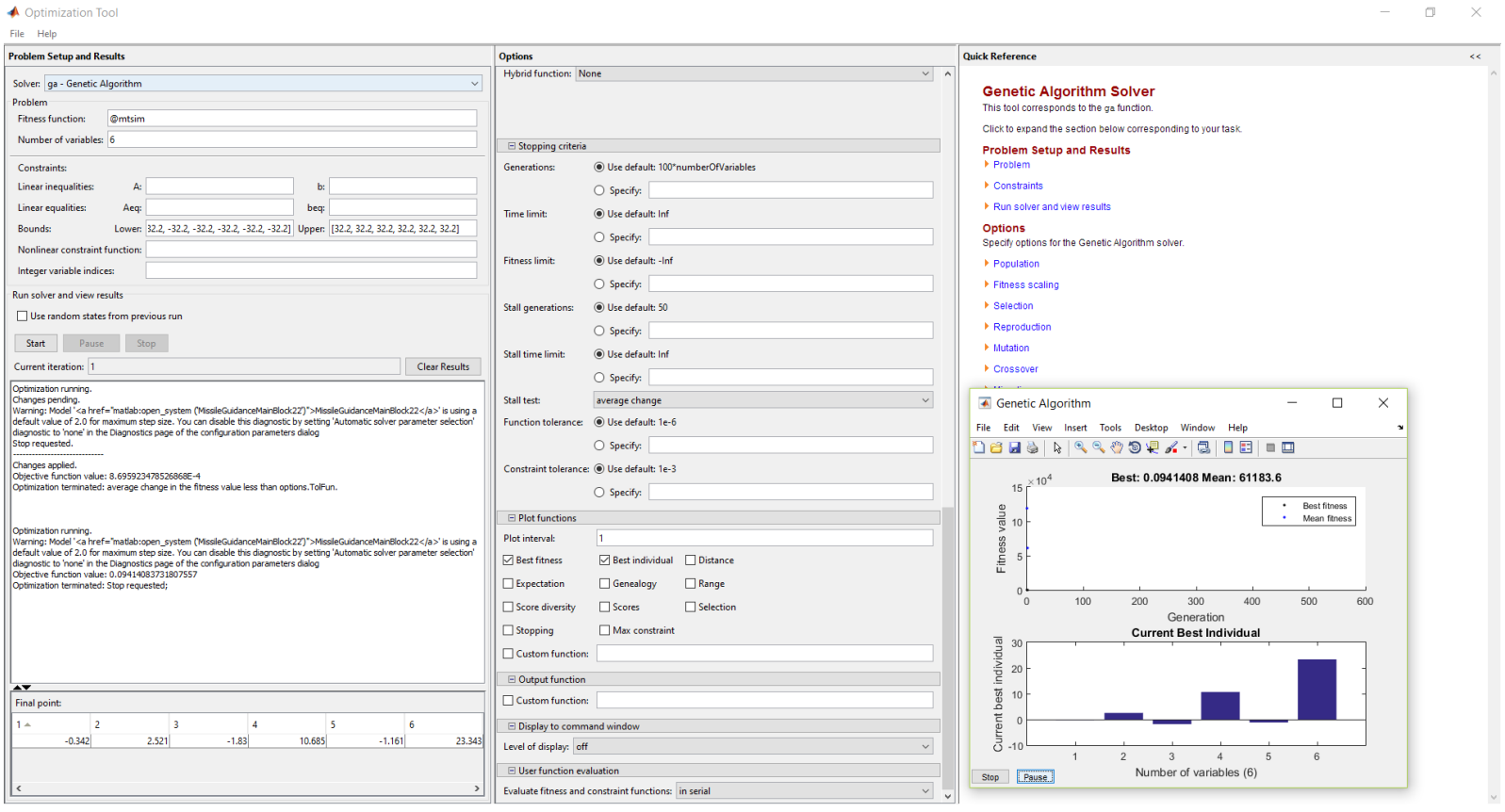


Figure: Genetic Algorithm Toolbox in Matlab.

Unity Game

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.

Unity Game

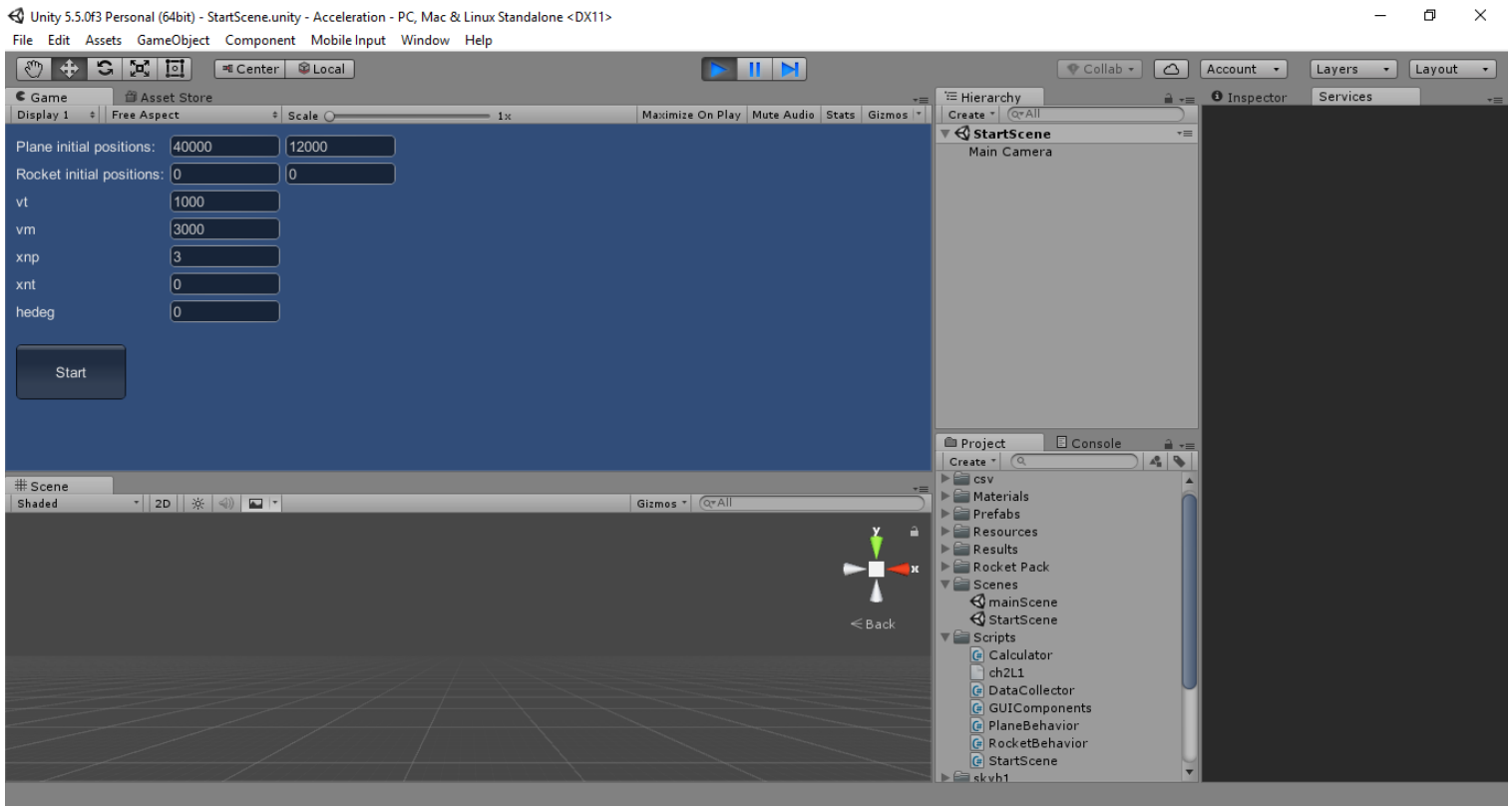


Figure: Unity game engine interface

Unity Game

Each game consists of two scenes:

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

Unity Game

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.

Unity Game

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.

Unity Game

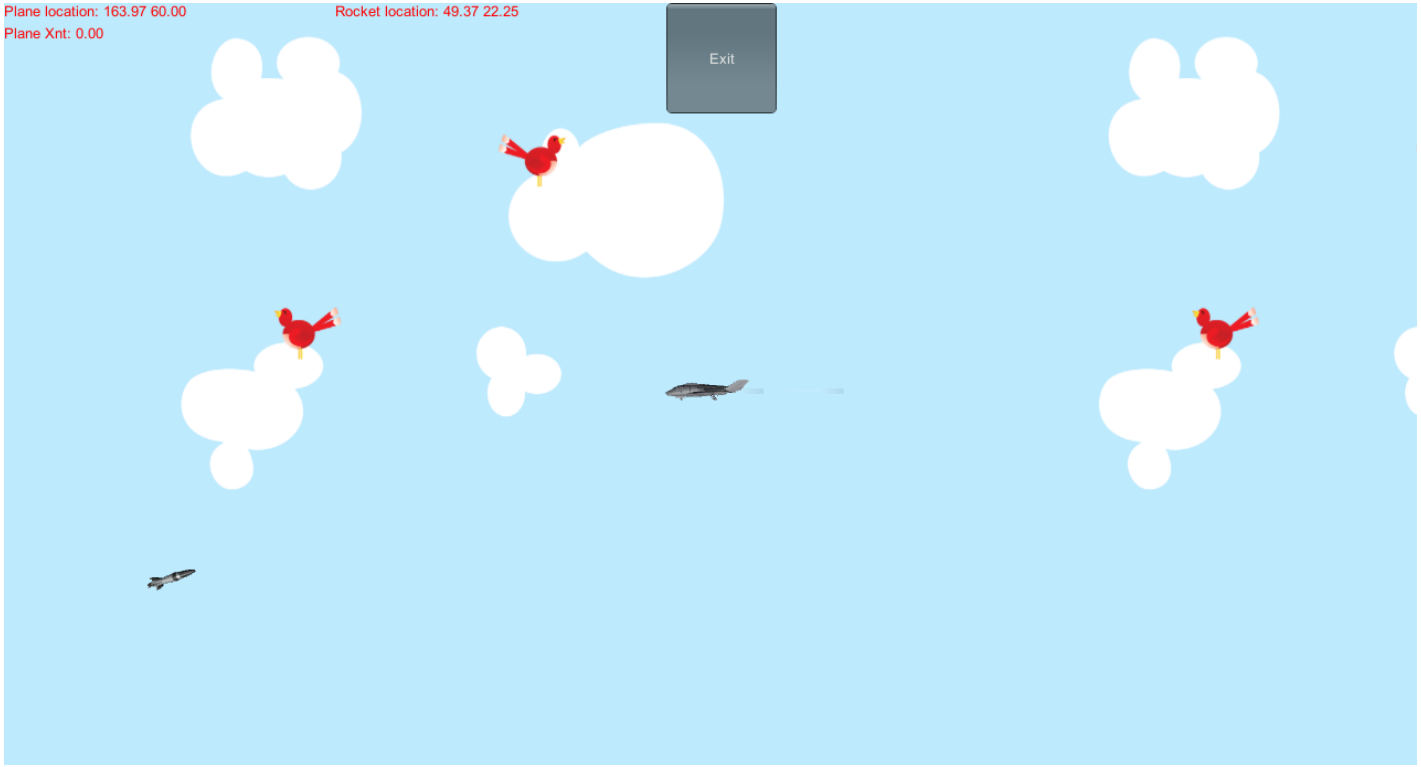


Figure: Unity game for simulating Target-Attacker engagement

Unity Game

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

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.

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 ($r \longrightarrow \infty$), 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.

$$\boxed{\frac{AI}{IB} = \frac{AE}{EB} = k,\ \{k \neq 1\}.} \tag{2}$$

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

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

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

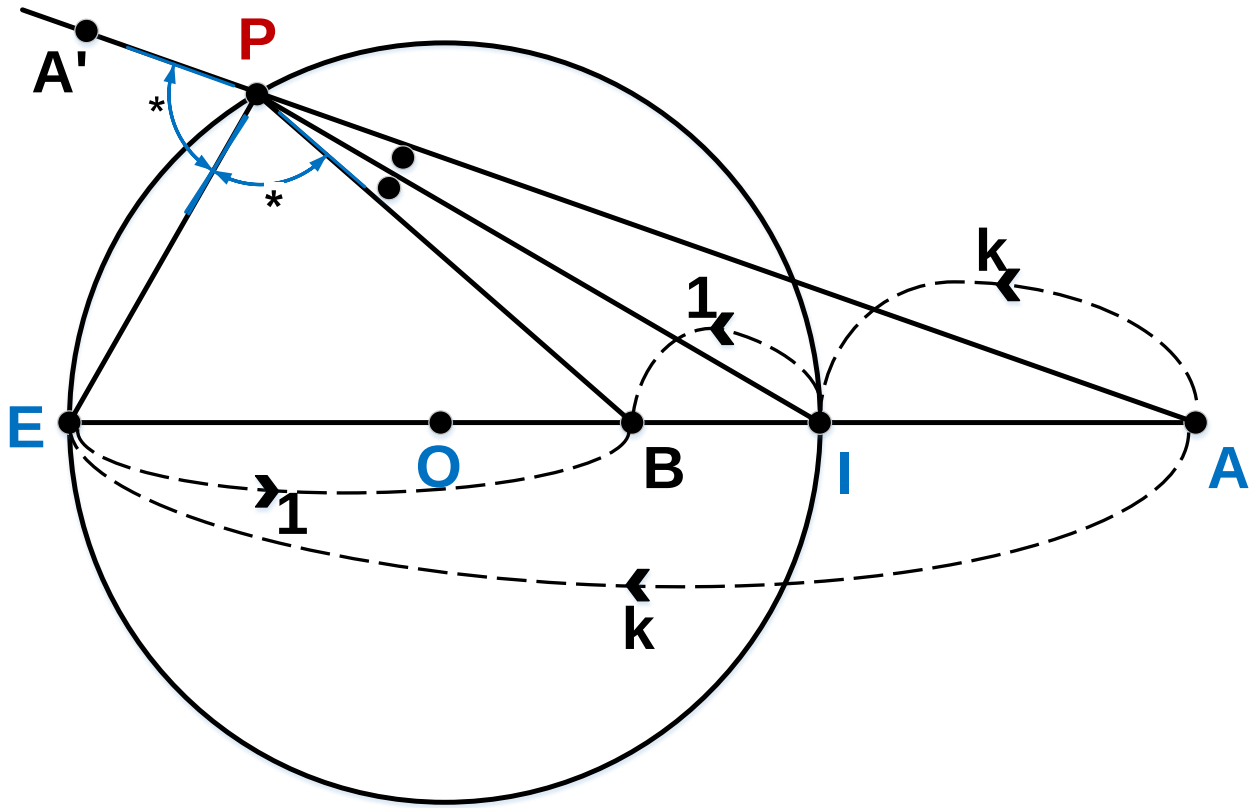


Figure: Apollonius circle for a moving point P such that $\frac{AP}{PB} = k > 1$. Here $m\angle API = m\angle BPI$ and $m\angle A'PE = m\angle BPE$.

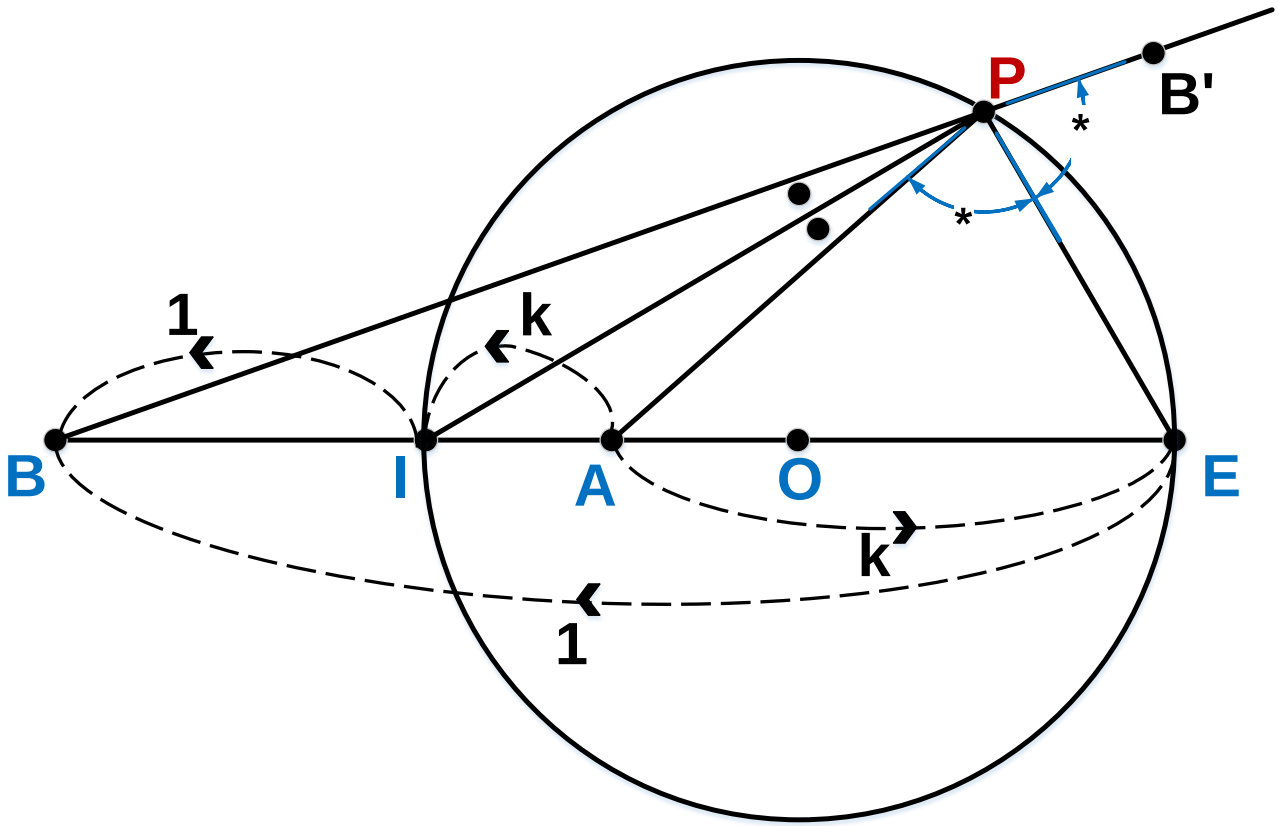


Figure: Apollonius circle for a moving point P such that $\frac{AP}{PB} = k > 1$.

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 $\boldsymbol{I}$ and $\boldsymbol{E}$

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

Radius of the Apollonius circle

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

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

case of $k = 1$

It is clear from (4) and (5) 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 $\overline{AB}$.

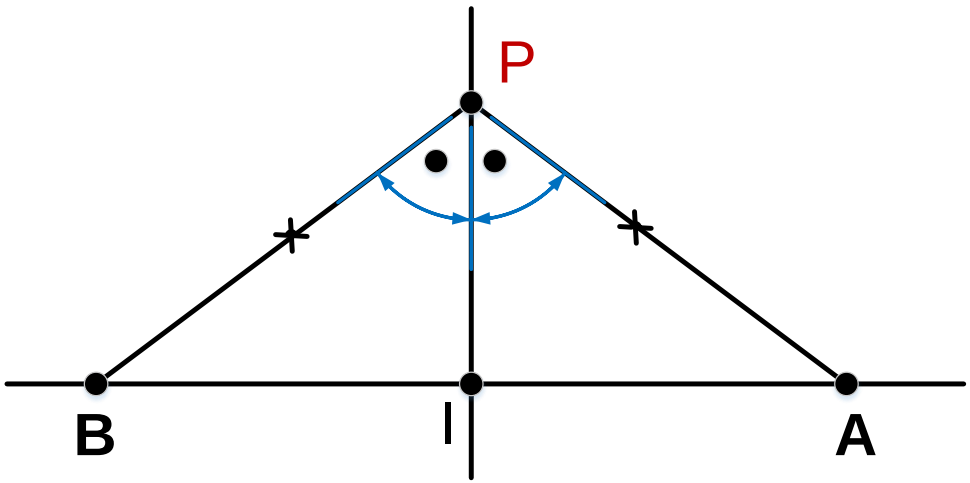


Figure: For $k = 1$, the Apollonius circle in the previous figures degenerates into the perpendicular bisector of the straight segment $\overline{AB}$. The point E disappears in this figure as it goes to ∞

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:

$$\gamma = \frac{V_A}{V_D}. \tag{7}$$

Substituting the values of $\boldsymbol{A}$ and $\boldsymbol{D}$ above for $\boldsymbol{A}$ and $\boldsymbol{B}$ in (2),(3),(4) and (5), respectively, and replacing k therein by γ,

$$\boldsymbol{I_1} = (\frac{1-\gamma}{1+\gamma}x_A, 0), \tag{8}$$

$$\boldsymbol{E_1} = (\frac{1+\gamma}{1-\gamma}x_A, 0), \tag{9}$$

$$\boldsymbol{O_1} = (\frac{1+\gamma^2}{1-\gamma^2}x_A, 0), \tag{10}$$

$$r_1 = \frac{2\gamma}{1-\gamma^2}x_A \tag{11}$$

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, k is replaced by:

$$\alpha = \frac{V_T}{V_A}. \tag{12}$$

Now, we substitute the values of $\boldsymbol{T}$ and $\boldsymbol{A}$ above for $\boldsymbol{A}$ and $\boldsymbol{B}$ in (2),(3),(4) and (5), respectively, and replace k therein by α

$$\boldsymbol{I_2} = \frac{1}{1+\alpha}(\boldsymbol{T} + \alpha\boldsymbol{A}) = \frac{1}{1+\alpha}(x_T + \alpha x_A, y_T), \tag{13}$$

$$\boldsymbol{E_2} = \frac{1}{\alpha-1}(-\boldsymbol{T} + \alpha\boldsymbol{A}) = \frac{1}{1-\alpha}(x_T - \alpha x_A, y_T), \tag{14}$$

$$\boldsymbol{O_2} = \frac{1}{1-\alpha^2}(\boldsymbol{T} - \alpha^2\boldsymbol{A}) = \frac{1}{1-\alpha^2}(x_T - \alpha^2 x_A, y_T), \tag{15}$$

$$\frac{\alpha}{\quad} \qquad \frac{\alpha d}{\quad} \tag{16}$$

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 $\boldsymbol{T} = (x_T, y_T)$ is outside R_r.

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

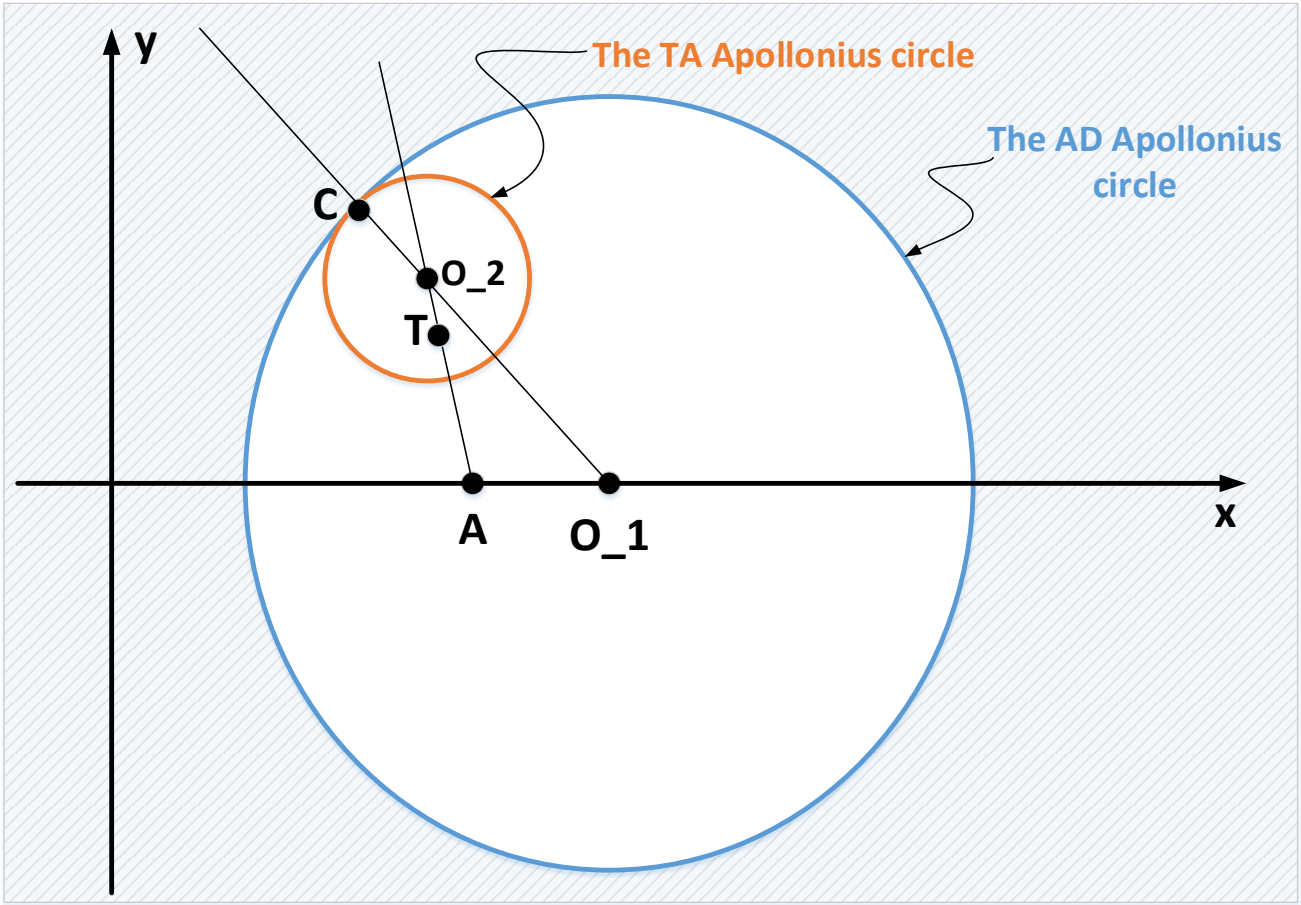


Figure: $\gamma < 1$

The critical speed ratio $\overline{\alpha}$, 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. This happens when

$$r_1 - r_2 = |\boldsymbol{O_1} - \boldsymbol{O_2}|. \tag{17}$$

Substituting for $r_1, r_2, \boldsymbol{O_1}$ and $\boldsymbol{O_2}$ from (10), (15), (10) and (14) respectively, and noting that $\gamma < 1$, one obtains

$$\frac{2\gamma}{1-\gamma^2}x_A - \frac{\alpha}{1-\alpha^2}d = |(\frac{1+\gamma^2}{1-\gamma^2}x_A, 0) - \frac{1}{1-\alpha^2}(x_T - \alpha^2 x_A, y_T)|$$

$$= [(\frac{1+\gamma^2}{1-\gamma^2}x_A - \frac{1}{1-\alpha^2}(x_T - \alpha^2 x_A))^2 + \frac{1}{(1-\alpha^2)^2}. \tag{18}$$

after several steps of simplification

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

Equation (18) is an improvement of equation (36) in [2]. while (18) is a quadratic equation in α, equation (36) in [2] is a quartic in α that has the same two roots of (18) in addition to the two irrelevant roots $\alpha \mp 1$. As a bonus, equation (18) is written in a self-verifying dimensionless form.

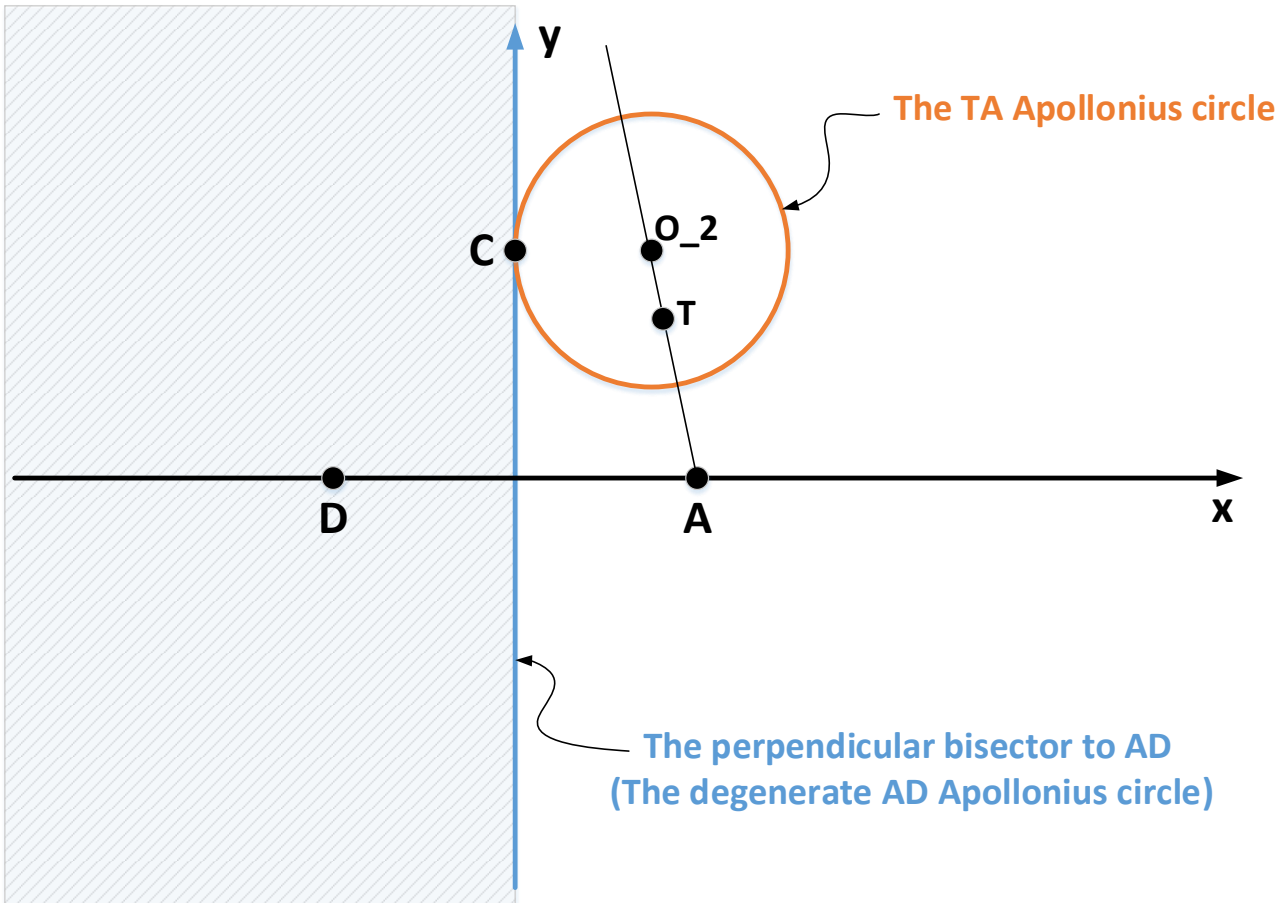


Figure: $\gamma = 1$

The critical speed ratio $\overline{\alpha}$ occurs when the *TA* Apollonius circle touches the *L.H.S.* of the $XY-$plane (Fig. 8), i.e., when

$$r_2 = \textit{Absicessa of } \boldsymbol{O_2}. \tag{20}$$

i.e., thanks to (14) and (15) when:

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

Multiplying (20) by $(1-\alpha^2)$ (thanks to the fact that $\alpha \neq 1$), and rearranging, one obtains the following quadratic equation in α

$$x_A\alpha^2 + d\alpha - x_T = 0, \tag{22}$$

which is equation (12) in [3]. It is interesting to note that equation (20) is the special case ($\gamma = 1$) of (18), though (18) was derived under the assumption $\gamma \neq 1$.

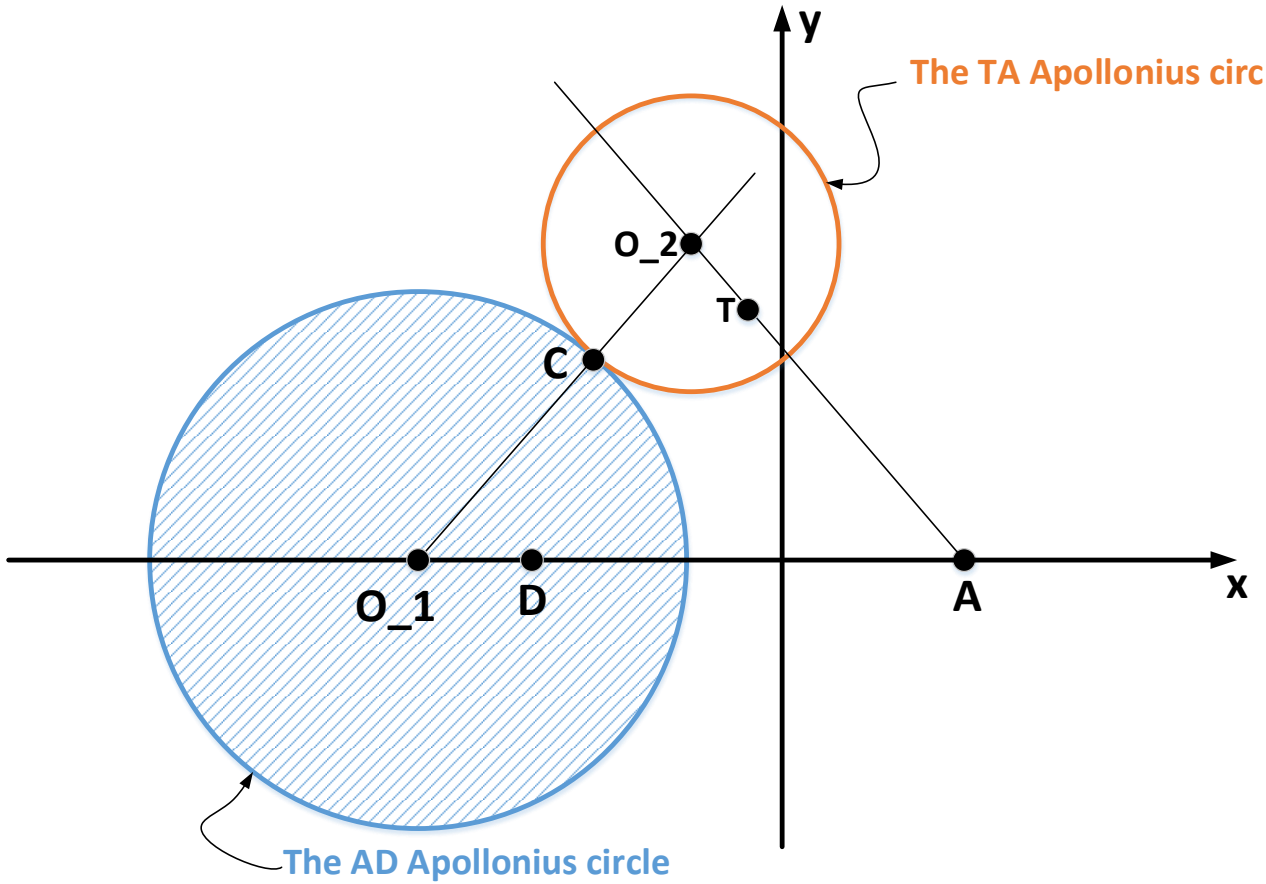


Figure: $\gamma > 1$

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. 9), i.e., when

$$r_1 + r_2 = |\boldsymbol{O_1} - \boldsymbol{O_2}| \tag{23}$$

Since $\gamma > 1$, we write

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

The above expression for $(r_1 + r_2)$ is exactly the negative of $(r_1 - r_2)$ in (17).

Equation (18) is thereby shown to hold for a slow Defender besides being true for a fast one. Therefore, we will use the quadratic formula (18) for all values of γ, and will solve it to obtain $\overline{\alpha}$ for any γ as

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

Equation (24) 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 γ, including $\gamma > 1$. Note that $\overline{\alpha}$ is definitely nonnegative and hence the minus sign in the solution for $\overline{\alpha}$ 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 (18) as a relation between y_T and x_T, we write

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

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 (26) 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 (26), 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 (26). 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 (26) is the general equation for the Voronoi diagram in our current *TAD* problem and it appears here for the first time.

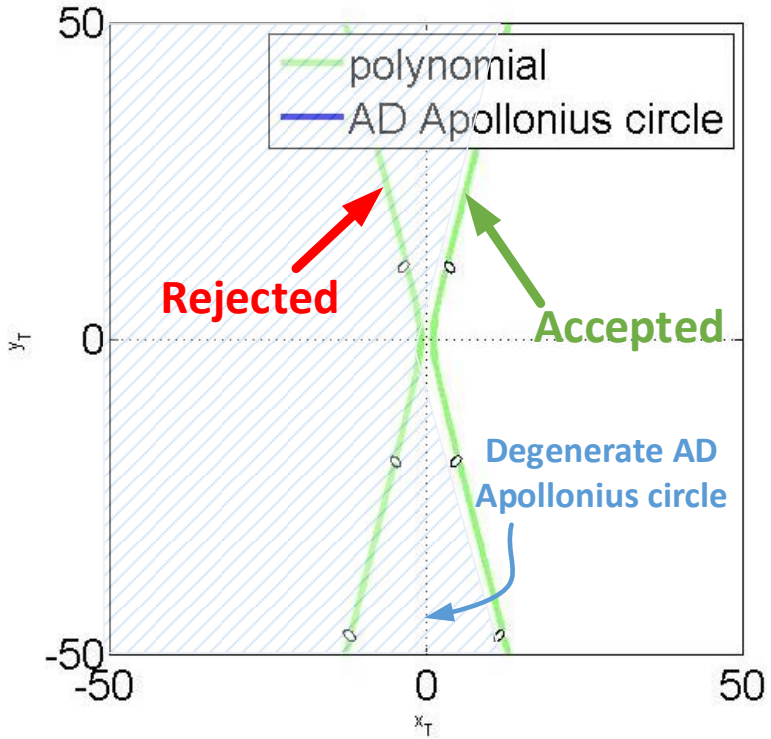


Figure: Generated computer output for the Voronoi diagram bordering the safe region for $x_A = 4,\ \alpha = 0.25,\ \gamma = 1$ (the safe region is the shaded area)

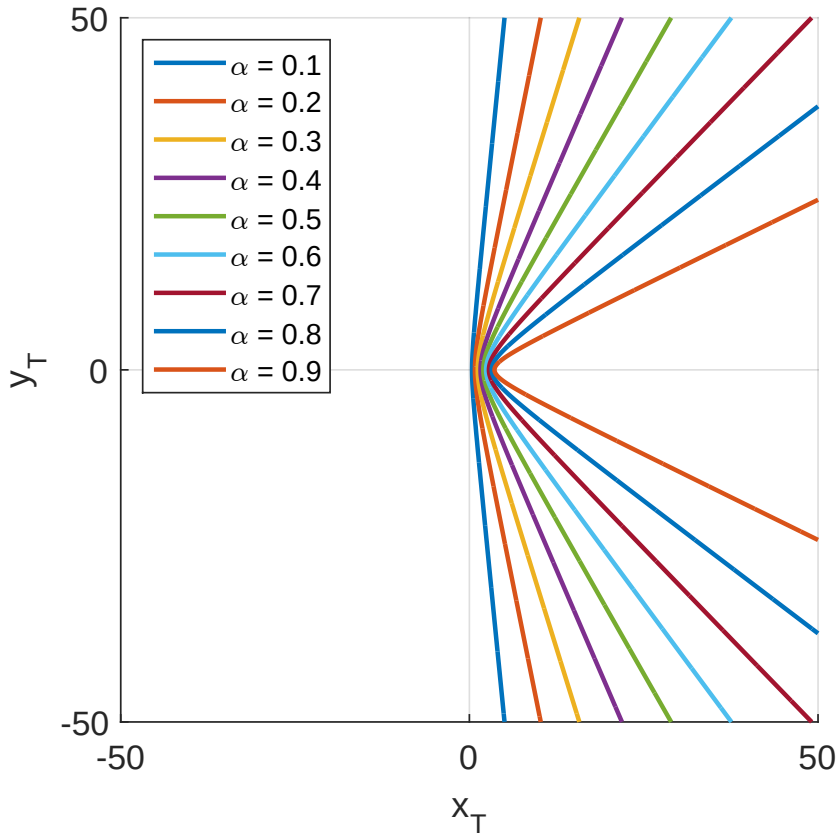


Figure: Various accepted branches of the voronoi diagram for $\gamma = 1$ and α as a parameter ranging from 0 to 1.

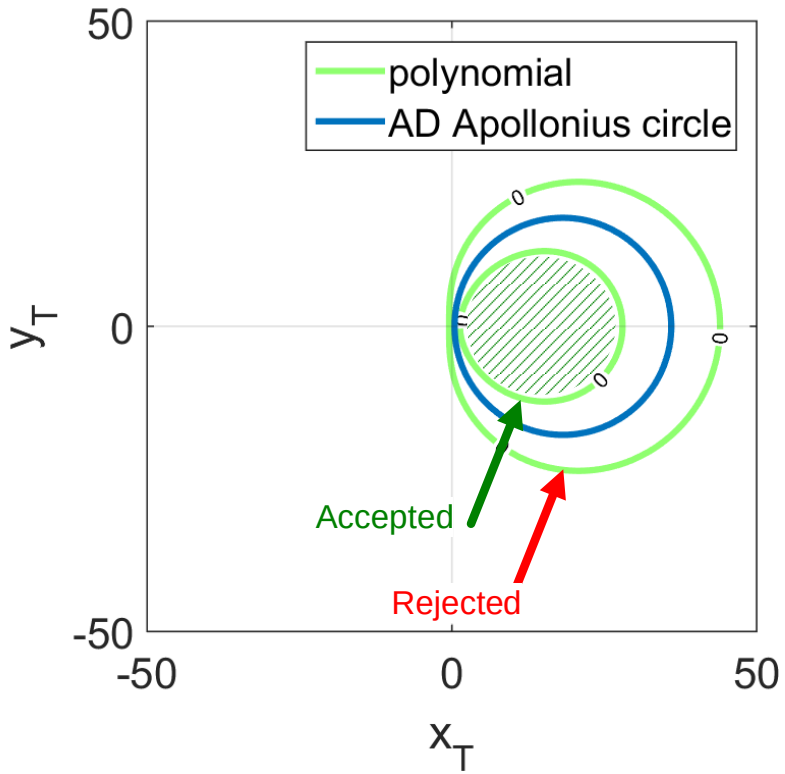


Figure: generated computer output for the Voronoi diagram bordering the safe region for $x_A = 4,\ \alpha = 0.25,\ \gamma = 0.8$ (the safe region is the unshaded area) the quartic in (26) produces two closed curves: one outside the *AD*-Apollonius circle (rejected) and the other inside the circle (accepted as the Voronoi diagram)

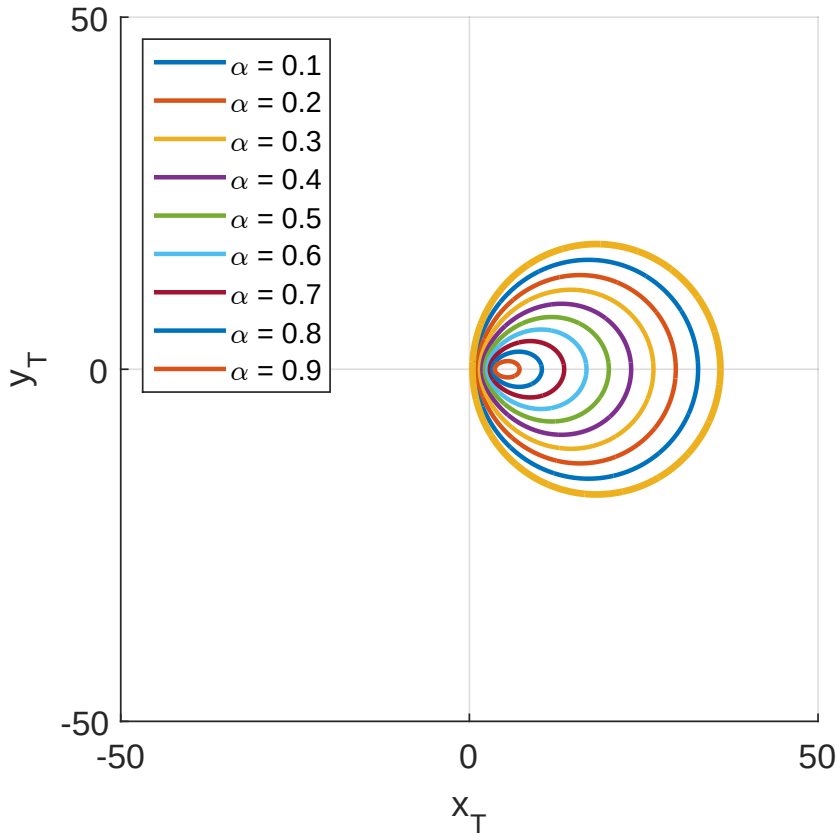


Figure: Various accepted branches of the voronoi diagram for $\gamma = 0.8$ and α as a parameter ranging from 0 to 1. These curves are computer generated from (26)

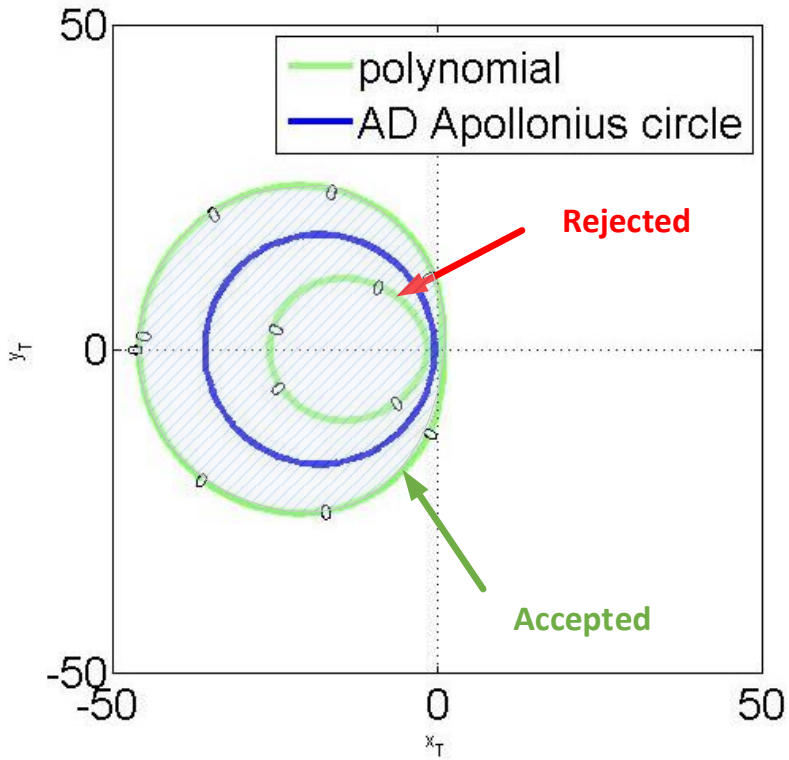


Figure: generated computer output for the Voronoi diagram bordering the safe region for $x_A = 4,\ \alpha = 0.25,\ \gamma = 1.25$ (the safe region is the shaded area) the quartic in (26) produces two closed curves: one inside the *AD*-Apollonius circle (rejected) and the other outside the circle (accepted as the Voronoi diagram)

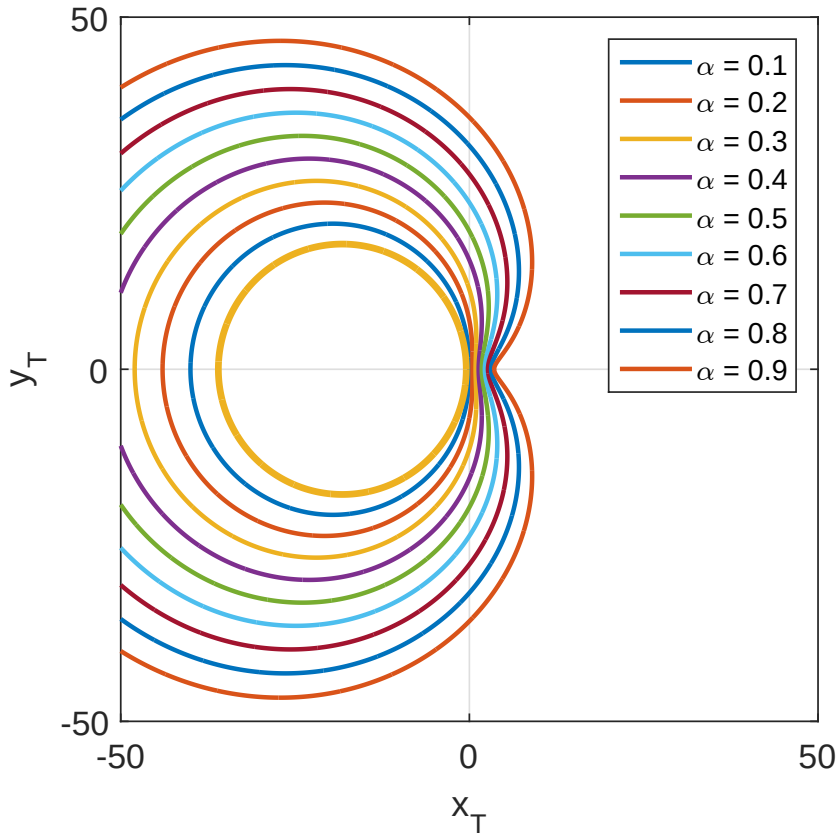


Figure: Various accepted branches of the voronoi diagram for $\gamma = 1.25$ and α as a parameter ranging from 0 to 1. These curves are computer generated from (26)

Concluding Remarks and Future Work

In this work, we presented:

-
-
-
-
-

Thank you!