

GRAPHICAL SIMULATION FOR UNMANNED AERIAL VEHICLE MISSION

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Alemayehu Kasahun Gebremariam

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

July 2014

Fargo, North Dakota

North Dakota State University
Graduate School

Title

**GRAPHICAL SIMULATION FOR UNMANNED AERIAL VEHICLE
MISSION**

By

Alemayehu Kasahun Gebremariam

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Simone Ludwig

Chair

Dr. Saeed Salem

Dr. Limin Zhang

Approved:

7/22/2014

Date

Dr. Brian Slator

Department Chair

ABSTRACT

Unmanned Aerial Vehicles (UAVs) are automated flying planes currently focused on military and civilian applications. This work simulates the motion of UAV in the battlefield environment; making use of the theoretical conditions for the design of real systems. The design of the algorithm is scalable to specify the number of UAVs involving in the mission. The coordinated flights of the UAVs with each other enable them to arrive at the enemy target at the same time. The simulation is based on real flight conditions with some assumption. The snapshot of the results show the different scenarios when only one or more than one UAV(s) participate in the mission arriving from different angles spaced equally to hit the target. The design of the simulation also allows to specify an enemy zone, which is a circular area from the target in which the UAVs enter the zone at the same time.

ACKNOWLEDGMENTS

I would like to thank to my advisor Dr. Simone Ludwig for continues support help and direction. Also I would like to express my sincere thanks to Dr. Kendall Nygard for his idea and support on this research. I would like to extend my gratitude to Dr. Saeed Salem and Dr. Limin Zhang for your being on my graduate committee.

This entire undertaking would not have been possible without the never ending support and love from my family. Lastly but not least, I would like to thank my friends who encouraged me to complete this work.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGMENTS	iv
LIST OF FIGURES	vii
1. INTRODUCTION	1
2. CURRENT RESEARCH AND RELATED WORK.....	6
3. PROBLEM DESCRIPTION AND ASSUMPTIONS	9
3.1. Problem Description.....	9
3.2. Assumptions	9
3.3. High Level Design and Architecture.....	10
3.4. Static Model	11
3.4.1. Class Diagram.....	11
3.5. Dynamic Diagram	12
3.5.1. Flow Chart	12
3.5.2. Sequence Diagram	13
4. METHODOLOGY AND ALGORITHM	15
4.1. Methodology	15
4.2. Scaling and Coordinate Systems	15
4.2.1. Pseudo Code.....	16
4.2.2. Pseudo Code.....	16
4.2.3. Pseudo Code.....	20

4.2.4. Pseudo Code:	20
5. RESULTS AND DISCUSSION.....	23
5.1. Scenario 1	24
5.2. Scenario 2	26
5.3. Scenario 3	27
5.4. Scenario 4	29
5.5. Scenario 5	30
5.6. Scenario 6	31
5.7. Scenario 7	32
5.8. Scenario 8	33
5.9. Scenario 9	34
5.10. Scenario 10	35
5.11. Scenario 11	36
5.12. Scenario 12	38
6. DISCUSSION.....	40
7. CONCLUSION AND FUTURE WORK	41
8. REFERENCES	42

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1-1. MQ-9 Reaper in flight (2007). USAF photographic archives (image permalink).....	2
1-2. Charles Perley unmanned aerial bomber (1863) [6]	2
3-1. Architectural diagram	11
3-2. Class diagram	12
3-3. Flow chart	13
3-4. Sequence diagram	14
4-1. Construction of a cubic Bezier curve	17
4-2. A cubic Bezier curve that passes through per determined points P_1 and P_2	18
4-3. The movement of UAV from point A to B	21
5-1. Flight controlling buttons, parameters, and area to display the flight.....	24
5-2. A single UAV projected path heading towards the target	25
5-3. A single UAV full travelled path towards the target	25
5-4. Projected course for two UAVs	26
5-5. Path travelled by two UAVs.	27
5-6. Projected path of the three UVAs	28
5-7. Path travelled by three UAVs towards enemy target.	28
5-8. Projected path for four UAVs.	29
5-9. Path followed by four UAVs.	30
5-10. The path followed by five UAVs on their mission.	31
5-11. Six UAVs path in the way to destroy enemy target.	32
5-12. Flight path for seven UAVs.	33

5-13. Eight UAVs flight path.	34
5-14. Nine UAVs flight path towards the enemy target.	35
5-15. Ten UAVs course of flight.	36
5-16. Projected path for three UAVs on mission crossing enemy zone of 200 units of radius.	37
5-17. Flight path followed by three UAVs on mission with 200 units' enemy zone radius.	37
5-18. Projected path for three UAVs on mission with 50 units' enemy zone radius.	38
5-19. Flight path followed by three UAVs on mission with 50 units' enemy zone radius.	39

1. INTRODUCTION

An UAV, also known as drone, is an aircraft with no human pilot on board. UAVs' flight can be controlled either autonomously based on preprogrammed flight plans with onboard computers, or by the remote control of a pilot on the ground or in another vehicle [1]. UAVs are defined as "remotely piloted or self-piloted aircraft that can carry cameras, sensors, communications equipment or other payloads" [2].

UAVs are currently used for a number of missions, usually deployed for military and special operation but also for reconnaissance of military and civil applications [2]. To name a few, UAVs domestic applications includes search and rescue, weather forecasting, law enforcement, border patrol, firefighting, disaster response, precision farming, commercial fisheries, scientific research, aerial photography, mail delivery, communications relay, infrastructure monitoring and emergency management [3]. When compared to manned aerial vehicles, UAVs are believed to provide two important benefits - they are cost effective and reduce the risk to a pilot's life. However, accident rates in today's UAVs are over 100 times than that of manned aircraft. Therefore, improved safety and reliability are still required [4]. In Figure 1-1, the General Atomics MQ-9 Reaper (formerly named Predator B) is UAV capable of remote controlled or autonomous flight operations, developed by General Atomics Aeronautical Systems (GA-ASI) primarily for the United States Air Force. UAVs are also referred to as drones [5].



Figure 1-1: MQ-9 Reaper in flight (2007). USAF photographic archives ([image permalink](#))

Unmanned technology has progressed a great deal exploiting rapid technological revolution and advancement that continues today. UAV technologies are often centered on military conflicts reaching back as far as 1863, two years after the start of the Civil War, an inventor from New York City named Charles Perley registered a patent for an unmanned aerial bomber (see Figure 1-2) [6].

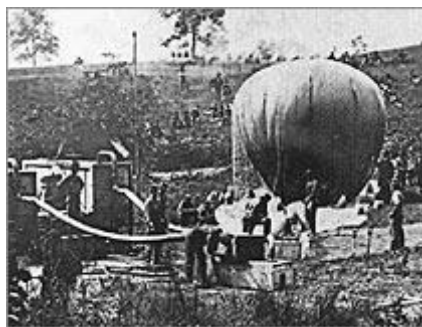


Figure 1-2: Charles Perley unmanned aerial bomber (1863) [6]

During WWI the U.S. developed a pilotless aircraft known as the “Kettering Bug”, which is small biplane equipped to carry a bomb load equal to its own weight—300 pounds and it can fly for a predetermined time before releasing its wings and plunging to the earth. Several were built but none flew in combat. It was not until 1920 that the U.S. military made further progress by demonstrating the first truly remote controlled aircraft Sperry Messenger, that was built via remote radio control [6]. But peacetime interest diverted funding away from this and other remotely operated aircraft. Few people, at the time, saw any practical civil use for the technology.

Then in the 1930’s and 1940’s, with war once again looming, interest in UAVs saw a resurgence, but this time UAVs were primarily produced as target drones in training scenarios for anti-aircraft gunners and applied in various other training environments. During this time the British developed and produced more than 400 called “Queen Bees”. Even though the U.S. and U.K. were producing small unmanned systems in volumes, during World War II, Nazi Germany's innovative V-1 demonstrated greatest advances to unmanned aviation introducing formidable threat a UAV could pose in combat [6].

During the Vietnam War, US developed technologically advanced UAVs known as Firebee. This vehicle was unique in defining a new role for UAVs photo-reconnaissance aircraft. Surveillance was found to be the ideal mission for UAVs, Firebee can sustain longer flights at higher altitudes suitable for surveillance task [7]. Following Vietnam U.S. investment in UAVs decreased, other countries began to develop UAV programs, but none more successfully than Israel. Impressed by America's AQM-34 Ryan Firebee UAV, Israel secretly purchased 12 Firebees from the U.S. in 1970, modified them, and designated those Firebee 1241 UAVs. These

Firebee 1241s played an important role in the 1973 Yom Kippur War between Israel, Egypt, and Syria, both as reconnaissance vehicles and as new kinds of UAVs, so called decoys [8, 9].

Since then, UAVs have been playing a critical role in military operations and continue to expand, covering a wide range of mission capabilities [8]. They are capable of carrying a lethal or nonlethal payload and come in different sizes, from the size of an insect to that of a commercial airliner. These devices have proven their effectiveness in recent warzones, such as Kosovo, Afghanistan, and Iraq, giving them mass media exposure [10]. Continued UAVs advancement heavily depends on technological innovation such as Global Positioning System for navigational and targeting functions. As new standards and technologies emerge, communications integration and interoperability are critical to ensuring the advancement of UAV capabilities. Continued improvement in computer processing technologies and networking plays a critical role in the autonomous development of UAV platforms. The network-centric operations between UAVs potentially reduce human involvement and ground support and that enables UAVs capable of making autonomous decisions without human intervention. Currently, computer processing technologies lack the sophistication needed to mimic the human brain but it is forecasted to be plausible.

The current technological development of UAV systems are fast, however, it is difficult specially during testing due to potentially dangerous operations, site availability, considerable resource requirement, and weather conditions. Therefore, computer simulation is a best tool used to model and study complex systems. Simulation can be performed prior to the development of the actual system or without changing the existing system. A simulation receives a set of input values from the analyst, runs for a specified amount of time or number of replications, and outputs the result in terms of a performance measure. UVA computer simulation can allow you

to see how a system might respond before you design or modify it. This avoids mistakes and one can try different ideas before the real product is produced, making it cheaper as there is no need to make different prototypes every time and testing them out. It is an advantage to find this out in a model rather than testing the real thing. Designing an item as a model on a computer before the real item is built saves time. Simulations can be slowed down to study behavior more closely. Simulation models allow analysts and researchers to produce complex scenarios without changing the actual systems being modeled. In addition, a simulation runs for a specified amount of time or number of replications and outputs the result.

This document is organized in the following manner: Chapter 1 is the introduction to unmanned aerial vehicle, and Chapter 2 contains related work current research. Chapter 3 elaborates the objective the problem focuses on as well as the assumption made. In Chapter 4, the high level and architectural design is described. Methods and approach to do the simulations are described in Chapter 5. Results from the simulation and discussion are included in Chapter 6. Chapter 7 suggests future improvement. In Chapter 8, is the reference section, and Chapter 9 source code of the simulation is included.

2. CURRENT RESEARCH AND RELATED WORK

Due to the improvements in embedded computing, communications, and sensing technologies, UAVs have become increasingly capable of carrying out sophisticated tasks. Most notable tasks that are ultra-long-endurance and high-risk mission acceptance, which cannot be reasonably performed by manned aircrafts, however, can be done with UAVs. In addition, quick growth of UAVs' onboard systems and great advantages in operations allow them being increasingly used in military and civilian domains. This technological advancement and capability helps UAVs to work as a team [11]. Multiple UAVs' cooperation as a team will enable new operational standards and better team performance. Controlling autonomous multiple UAVs that work as a team to accomplish missions has been an emerging issue since the 21st century [12].

Many researches have shown the benefits of cooperation among unmanned systems [13-15]. The demand with cooperation of multiple UAVs has brought into focus several challenges associated with it. Many studies have been done on cooperative operations of multiple UAVs as well as formation flight strategies [16-18]. However, many UAVs are remotely controlled by many skilled human pilots since there is no much interoperability among the swarm members, and these pilots must be able to deconflict airspace demands, mission requirements, and situational changes in near real time [19]. For instance, Predators, a type UAV, currently requires the full attention of two human operators. With this kind of men to machine ratio to control UAVs, in the future there will not be enough trained pilots for all the Predator missions. On the top of this, each Predator team of men and machine acts independently at a time. However, in order to carry out complex coordinated missions, these teams will have to share information and cooperate among each other to perform in an optimal manner. In order to reduce UAV

dependence on limited numbers of human pilots', autonomous unmanned aerial vehicle are the best options. Hence, autonomous unmanned aerial vehicles allow human pilots' personnel to focus on more important issues like interpreting the gathered information rather than to find out how to acquire the information [20]. There is clear benefit building intelligent unmanned vehicles that can plan and adapt autonomously to the environment they perceive, to make mission critical decisions, while collaborating with the human as appropriate [21]. Even though, human pilots judgment is much better than that of an autonomous vehicle, they can help solve navigation and targeting problems, which are complicated for humans to tackle in real time [22].

In practice UAVs act as distinct entities and share a little if any information between systems because they are largely disconnected. This disconnection to share information impacts their cooperation and coordination that leads to a highly inefficient and possibly dangerous environment. Control and coordination among multiple UAVs can create safe and optimal flight paths, and working as a group to solve large complex task [23]. Command control and coordination among multiple aerial vehicles is a difficult task. Inherent nature for the choice of network architecture for decision-making among aircraft is complicated. Centralized architectures can resolve conflicting situational awareness data collected by independent sensors at different battle space vantage points via a single fusion protocol. However, they are often not operationally feasible due to significant communication overheads and over-reliance on a central decision-maker susceptible to failure. In contrast, decentralized decision-making architectures typically offer more robustness, but they are sensitive to information discrepancies across the UAV team [23].

There has recently been much research done in the area of autonomous vehicle control for surveillance type missions. Almost all of the research has dealt with centralized cooperative

control, with little research concerned with the more realistic decentralized problem [24]. Vidal, implemented vision base navigation system for autonomous airships, some researchers used potential fields methods for multiple flights [25-27], Consensus Based Bundle Algorithm (CBBA) has been proposed to solve account for inconsistent information among the UAVs [28, 29].

Building a simulation environment helps to implement and evaluate cooperative control strategies for UAVs on a mission. This paper presents a simulation environment of several UAV that are on a mission to the enemy target using the C# language. The designed system has one centralize station that manages the motion and cooperation of each UAV in the mission.

3. PROBLEM DESCRIPTION AND ASSUMPTIONS

3.1. Problem Description

This paper addresses the development of a computer simulation for multiple UAVs coordinated flight where they launched from their airbase towards the enemy target to hit and destroy it. The simulation allows the user to select up to ten numbers of UAVs that are participating in the mission. Each UAVs are set to leave from their airbase, which is not predetermined at the beginning rather the random number generator determine the position in terms of X-Y coordinates, while the target is set to be in the middle. Since the UAVs' airbase is not predetermined a lot of times the airbases distances from the target are not the same for each UAVs, i.e., for some UAVs the distance to get to the target is shorter compared to the others. However, at the end all UAV need to get into the enemy zone at the same time from different angles separated equally, that is, if we consider two UAVs, the angular separation between the planes when they get to the enemy zone should be 180 degree, if they are three the separation among them should be 120, and the same way for four planes the angular separation between the planes is 90 degree and so forth. Therefore, to accomplish their mission the UAVs need to coordinate with each other to find at what angle and time each should enter the enemy zone, which requires calculating the position, speed, and direction of each UAVs.

3.2. Assumptions

To create a computer simulation of a real world problem of a UAV flight, it is essential to make some assumptions to reduce the problem to a more manageable size.

- Only unmanned aerial vehicles are treated.

- The UAVs do not change altitude. This is a reasonable assumption since generally it is not necessary for autonomous vehicles to change altitude often while performing a mission.
- The simulation assumed to be in two dimensions (X-Y plane). This assumption reduces the complexity of the algorithm as the UAVs get to certain altitude and they tend to fly at that height.
- The mission target is in the middle. This assumption is based on the limitation to view the target on the screen in order to keep it visible at all time.
- Enemy zone is a circular area from the target where the enemy assumed to exist and used as a boundary reference at which the planes enter the zone at the same time from a different angle.
- The UAVs are capable of flying at different speeds for instance 400mil/hr to 900mil/hr.

3.3. High Level Design and Architecture

This part focuses on the description of the subsystem decomposition, hardware and software mapping, and data management. Also, the section describes the design goals that must be met in order to satisfy the requirements.

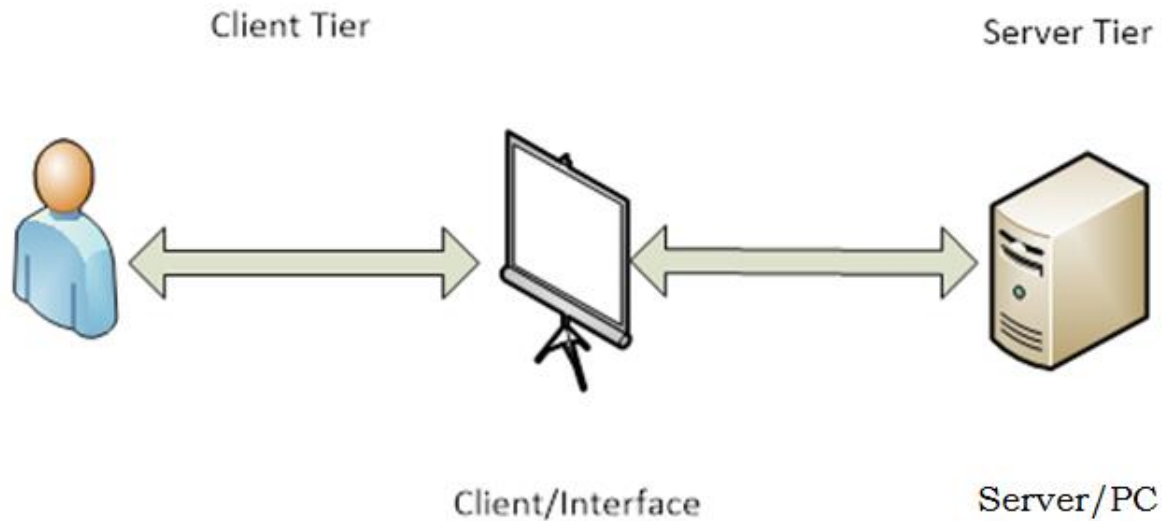


Figure 3-1: Architectural diagram

3.4. Static Model

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram which forms the main structure and therefore stable. Class Diagram is static UML diagram that has set of classes and their relationships.

3.4.1. Class Diagram

Today, object oriented programming techniques are the most popular programming techniques among the industrial level projects, and thus I also used the object oriented programming technique.

First, the requirements are examined thoroughly and identified the classes and its related methods and attributes. These classes are implemented using C# code in a .NET Environment, that enable the necessary object oriented concepts such as code reusable, inheritance, and encapsulation are implemented (see Figure 3 2 for class diagram for this simulation).

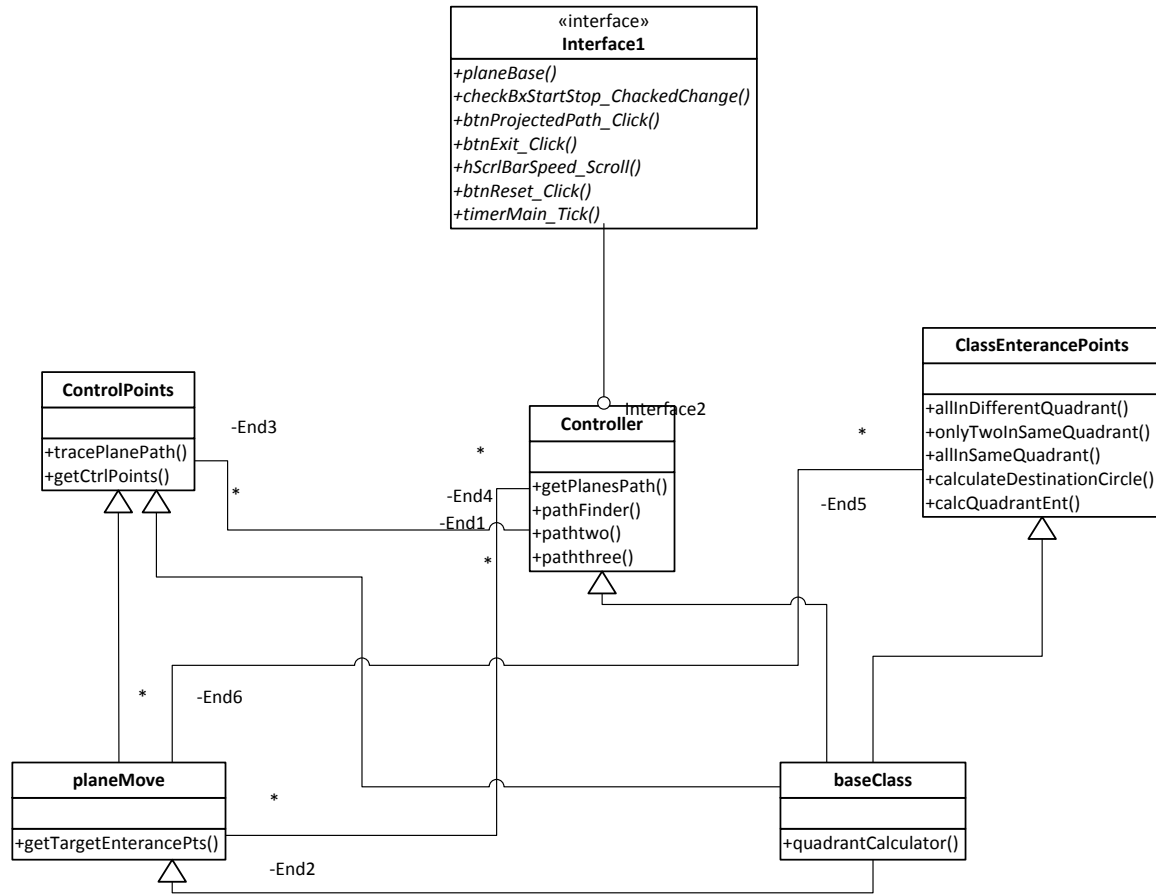


Figure 3-2: Class diagram

3.5. Dynamic Diagram

Dynamic diagrams, also called behavioral diagrams, capture the dynamic aspect of a system. Since behavior diagrams illustrate the behavior of a system, they are used extensively to describe the functionality of software systems.

3.5.1. Flow Chart

The simulation for a UAV system achieves the following functions as illustrated in the flow chart diagram in Figure 3-3:

- UAV simulation system initialization includes the target, and number and location of UAVs

- Flight operation course, which offers the function of pause, continue, project flight path
- And the stopping condition

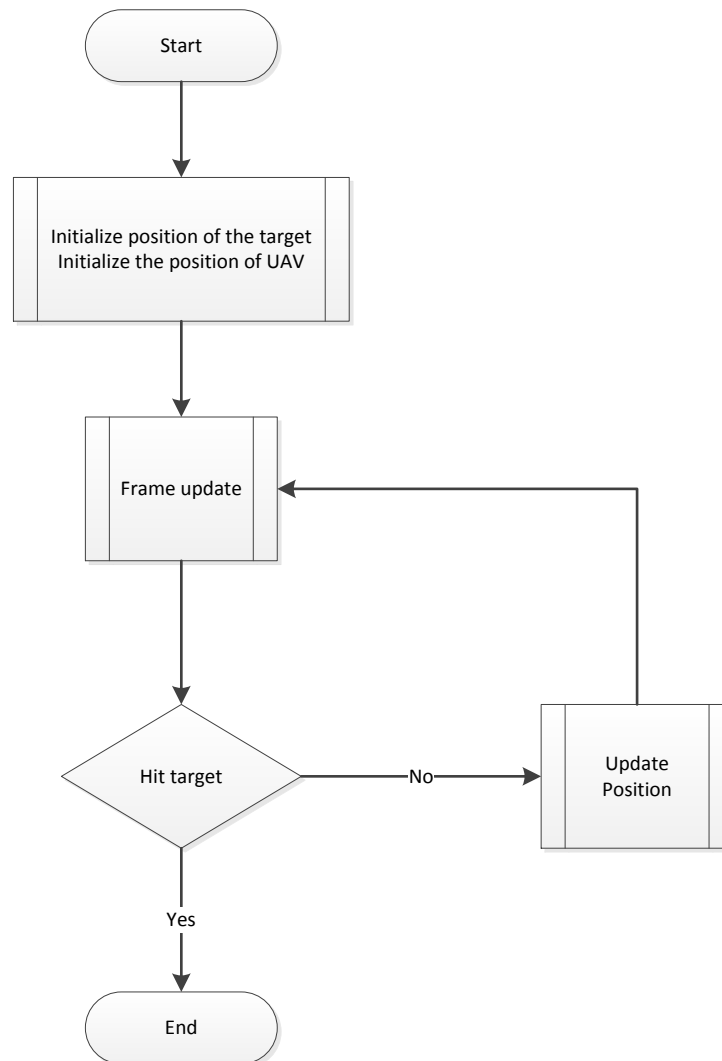


Figure 3-3: Flow chart

3.5.2. Sequence Diagram

The sequence diagram describes a sequential event that occurs from initialization until when the UAVs hit their target (see Figure 3-4).

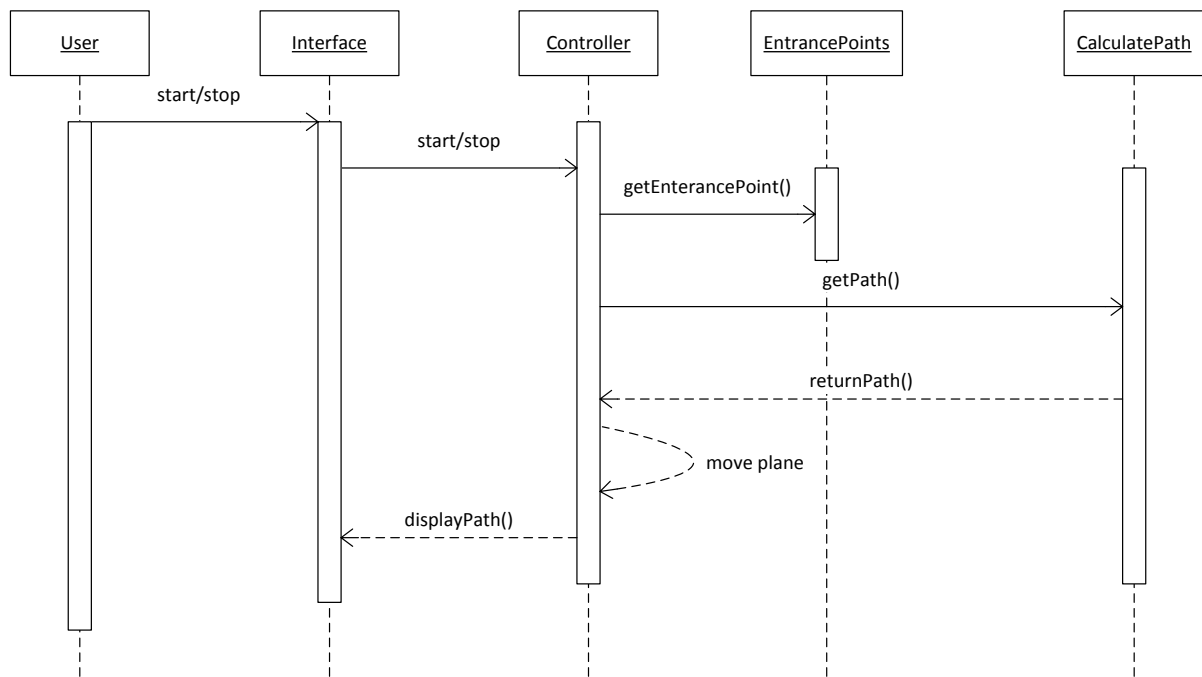


Figure 3-4: Sequence diagram

4. METHODOLOGY AND ALGORITHM

4.1. Methodology

This section describes the scaling and the approach used to address the real system in the simulation.

4.2. Scaling and Coordinate Systems

Simulation is a representation of the real-world phenomenon and in this UAV simulation it is necessary to design the simulation environment to mimic the reality. Therefore, the position of UAVs is defined in the Cartesian coordinate system to locate and track the UAVs at any given moment as well as a scale factor that allows the simulated calculations to be mapped to a real-world. The Cartesian coordinate system used in this simulation in order to map directly to a computer screen graphics environment origin starts at (0, 0) at the top left corner of the screen. And the x-coordinate is along the horizontal increase from the origin to the right, while the y-coordinate starts from the origin and increases downward. All locations in this simulation are specified using the floating point precision number representation chosen that allows us to represent real world locations within a meter precision.

Simulation time is the time it takes the UAVs to travel from the airbase to the destination, and is also another factor that needs scaling. The movement of the UAV in this simulation is not smooth rather their positions are updated in a discrete time interval or time step. This time step represents a discrete amount of time that has passed since the last sequence of events occurred. This simulation implements constant time intervals for each discrete event in the simulation.

The requirement for these UAVs, is that a UAV can start from any distance outside of the enemy zone from the target. However, at the end all UAVs need to get into the enemy zone at the same time from different angles separated equally. There are two methods that make sure all

the planes arrive in the enemy zone at the same time. The first method is to make the plane move slowly, and if that is not enough, then the second method is to add an additional path for the plane to go around so that the furthest plane(s) can reach the enemy zone at the same time. Generally, the farthest UAV to the target will be allowed to enter through the nearest enemy zone boarder to the plane, and the nearest UAV will enter through farthest entry point of the enemy zone (the pseudo code below describes this process).

4.2.1. Pseudo Code

- Calculate entrance points for UAVs to the enemy zone:
- Input enemy zone entrance point for the first plane
- Find the line connecting entrance point and the target
- Find the angle θ_1 of the line relative to the horizontal axis
- Then add on $\theta_1 \cdot 360/(\text{number of UAVs})$ to find entrance for 2nd plane, then 3rd, etc

4.2.2. Pseudo Code

- Add additional path when the plane is very near the enemy target: method `tracePlanePath()`
- Find the relative position of the UAV and entrance point to the enemy zone
- Based on the relative position set points that the UAV needs to pass through

When UAVs turn they do not take a sharp turn rather they follow a curved path, to represent this curved turn. In this simulation a Cubic Bezier curve is used. A Bézier curve is a parametric curve frequently used in computer graphics and related fields Bézier curves are used to model smooth curves that can be scaled indefinitely [30].

A cubic Bezier curve is defined by four points $P_0(x_0, y_0)$, $C_1(x_{11}, y_{11})$, $C_2(x_{22}, y_{22})$ and $P_3(x_3, y_3)$ in the plane or in a higher-dimensional space. The curve starts at P_0 going toward C_1 and arrives at P_3 coming from the direction of C_2 . This means the control points may not lie on the curve; these points are control points that are only there to provide directional information. The distance between P_0 and C_1 determines "how long" the curve moves into direction C_2 before turning towards P_3 .

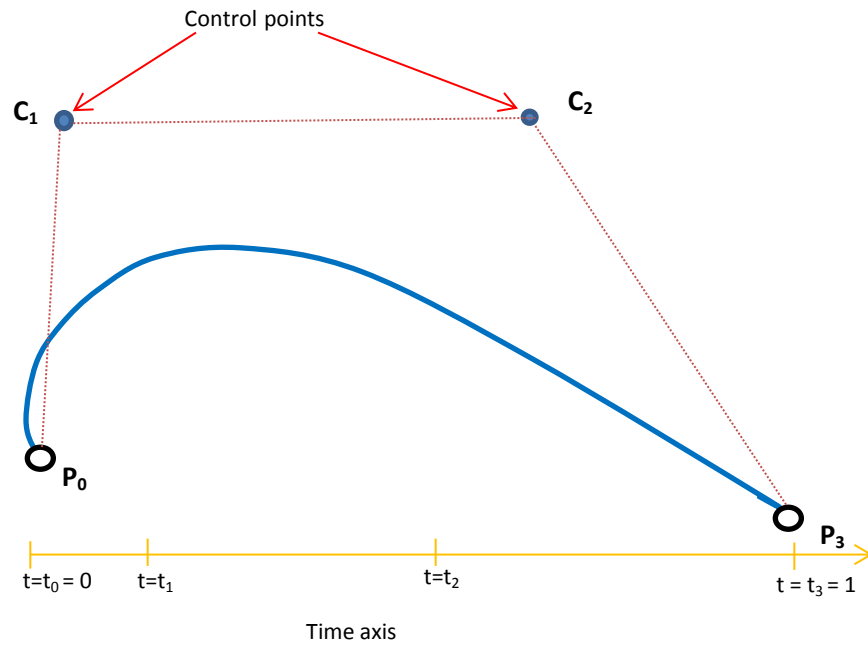


Figure 4-1: Construction of a cubic Bezier curve

The explicit form of the curve is:

$$P(x, y, t) = (1 - t)^3 P_0 + 3(1 - t)^2 t C_1 + 3(1 - t) t^2 C_2 + t^3 P_3, \quad t \in [0, 1] \quad (1)$$

In the simulation, we now have the starting point and destination point but we have to determine what would be the value of the control points. To determine the value of the control point, I used reverse engineering since I know at what coordinate points the planes should pass. To do this we have four defined points namely P_0 , P_1 , P_2 and P_3 where the Bezier curve passes through all four points and we want to determine C_1 and C_2 .

As illustrated in Figure 4-2, suppose we want the curve to pass through P_0 , when $t = 0$, through P_1 when $t = t_1$, P_2 when $t = t_2$, and P_3 when $t = 1$ where $t_1 < t_2$ to preserve the condition where the curve passes through P_1 first before P_2 . Next, we substitute the desired points in Equation 1, two points on the curve, i.e., P_1 and P_2 :

$$P_1 = (1 - t_1)^3 P_0 + 3(1 - t_1)^2 t_1 C_1 + 3(1 - t_1) t_1^2 C_2 + t_1^3 P_3 \quad (2)$$

and

$$P_2 = (1 - t_2)^3 P_0 + 3(1 - t_2)^2 t_2 C_1 + 3(1 - t_2) t_2^2 C_2 + t_2^3 P_3 \quad (3)$$

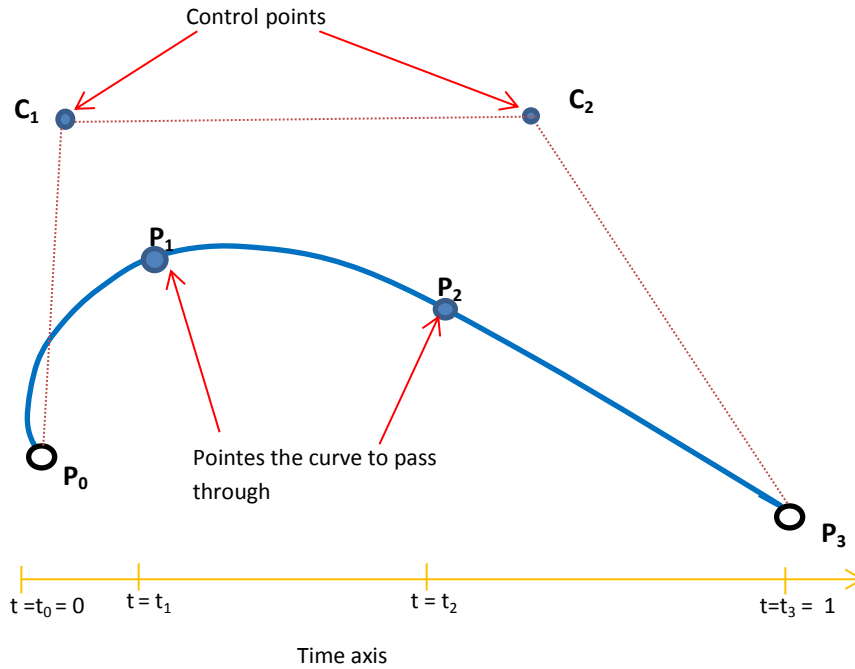


Figure 4-2: A cubic Bezier curve that passes through per determined points P_1 and P_2

So going back to Equation 2 and 3, when t is zero, the equation effectively collapses into just P_0 . When t is one, the equation gives P_3 . When t is between zero and one, the resulting point

lies on the curve itself, so iterating t from zero to one will give the Bezier curve. Since we know the curve will pass through P_0 and P_3 , we need to find P_1 and P_2 .

Collecting the constants, the above equation can be rewritten as:

$$D = 3(1 - t_1)^2 t_1 C_1 + 3(1 - t_1)t_1^2 C_2 \quad (4)$$

$$\text{Where } D = P_1 - (1 - t_1)^3 P_0 - t_1^3 P_3$$

$$E = 3(1 - t_2)^2 t_2 C_1 + 3(1 - t_2)t_2^2 C_2 \quad (5)$$

$$\text{Where } E = P_2 - (1 - t_2)^3 P_0 - t_2^3 P_3$$

We can rewrite the above equation in matrix form:

$$\begin{pmatrix} D \\ E \end{pmatrix} = \begin{pmatrix} 3(1 - t_1)^2 t_1 & 3(1 - t_1)t_1^2 \\ 3(1 - t_2)^2 t_2 & 3(1 - t_2)t_2^2 \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \quad (6)$$

The determinant of the above matrix is different from zero as long as t_1 and t_2 are different from 0 or 1 and $t_1 < t_2$, hence, we can find a unique solution for the above equations.

Next, we multiply the inverse of the 2-by-2 matrix on the left of both sides of the equation and we get:

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \frac{1}{\det} \begin{pmatrix} 3(1 - t_2)t_2^2 & -3(1 - t_1)t_1^2 \\ -3(1 - t_2)^2 t_2 & 3(1 - t_1)^2 t_1 \end{pmatrix} \begin{pmatrix} D \\ E \end{pmatrix} \quad (7)$$

$$\text{Where } \det = \begin{vmatrix} 3(1 - t_1)^2 t_1 & 3(1 - t_1)t_1^2 \\ 3(1 - t_2)^2 t_2 & 3(1 - t_2)t_2^2 \end{vmatrix} = 9t_1 t_2 (1 - t_1)(1 - t_2)[t_2 - t_1]$$

Therefore,

$$C_1 = \frac{1}{\det} (3(1 - t_2)t_2^2 D + 3(1 - t_1)t_1^2 E) \quad (8)$$

$$C_2 = \frac{1}{\det} (-3(1 - t_2)^2 t_2 D + 3(1 - t_1)^2 t_1 E) \quad (9)$$

$$\text{Where } C_1 = C_1(x, y)$$

4.2.3. Pseudo Code

- Find Control Point: Input values for initial, end and mid points on the curve: p_0, p_3, p_1, p_2
- Initialize : $t_1=0.1$, and $t_2=0.9$
- Find the determinant (Equation 7):
- Determinant = $9t_1t_2(1 - t_1)(1 - t_2)[t_2 - t_1]$
- Calculate constants (Equation 4 and 5):

$$D = P_1 - (1 - t_1)^3P_0 - t_1^3P_3$$

$$E = P_2 - (1 - t_2)^3P_0 - t_2^3P_3$$

- Calculate control points (Equation 8 and 9):

$$C_1 = \frac{1}{\det} (3(1 - t_2)t_2^2 D + 3(1 - t_1)t_1^2 E)$$

$$C_2 = \frac{1}{\det} (-3(1 - t_2)^2 t_2 D + 3(1 - t_1)^2 t_1 E)$$

Now we can calculate the controlling point based on the two points (P_1 and P_2) on the curve, that allow us to use Equation 1 to find the curved path. The implemented source code for that calculates the path followed by the UAV once the initial position of the UVA is given.

4.2.4. Pseudo Code:

- Find a path: Inputs values for initial, end and mid points on the curve: p_0, p_3, p_1, p_2

For time $t \in [0, 1]$ in interval of dt (using Equation 1)

$$P(x, y, t) = (1 - t)^3P_0 + 3(1 - t)^2 t C_1 + 3(1 - t)t^2 C_2 + t^3P_3$$

End for loop

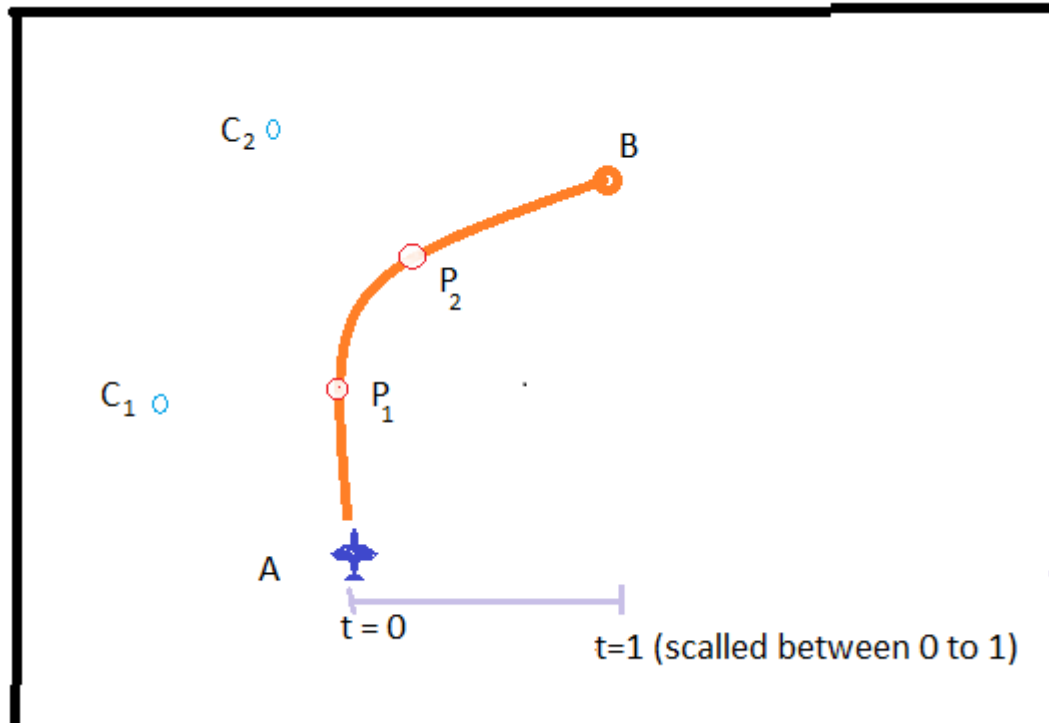


Figure 4-3: The movement of UAV from point A to B

Suppose a UAV starts from point A at time $t = 0$ heading to point B passing through P_1 and P_2 . In order to find all intermediate points along the path from point A to B, first we use Equations 8 and 9 to determine the controlling points C_1 and C_2 as shown in the Figure 4-3. Once C_1 and C_2 are determined using Equation 1 we can find any point on the diagram along the path from Point A to B at any given time t . In doing so we can divide equally the time interval between 0 and 1, and then calculate the positions at each time interval, hence, at each time step the position of a UAV will be updated on the screen.

The simulation is written using C# in .Net visual studio, and executed on Gateway laptop with Intel Core i5 CPU 2.4 GH. GUI rendered by Windows Graphics Device Interface (GDI) called GDI+ that consists of the set of .NET base classes that are available to control custom

drawing on the screen. GDI+ can create graphics, draw text, and manipulate graphical images as on Windows Forms and controls [31].

5. RESULTS AND DISCUSSION

The simulation has an interface that allows the user to choose certain parameters such as the number of unmanned aerial vehicles involved in the mission (varying from one to ten), enemy zone radius, the speed of the simulation from slow to fast, project the path of each UAVs, start and stop of the simulation, and reset after the simulation is started. Some of the parameters are set by default, these are the number of UAVs selected are 3, enemy radius is set to be 10, and the speed of the simulation selected to be fast.

Figure 5-1 shows the graphical interface containing the flight Controlling buttons, parameter settings, and display area. To start the simulation, by default three aerial vehicles are selected with 100 unit radius of enemy zone - a circular area centered at the target controlled by the enemy, and fast moving frames. These parameters can be changed before starting the simulation. The start button starts/pauses the simulation, projected path displays the projected path of the planes' track, reset button resets the current flight simulation to start over, this can be done at any time during the simulation, and also this button can be used to start a fresh simulation. To do that the current simulation needs to be paused and double clicked on the reset button. The exit button closes the simulation.

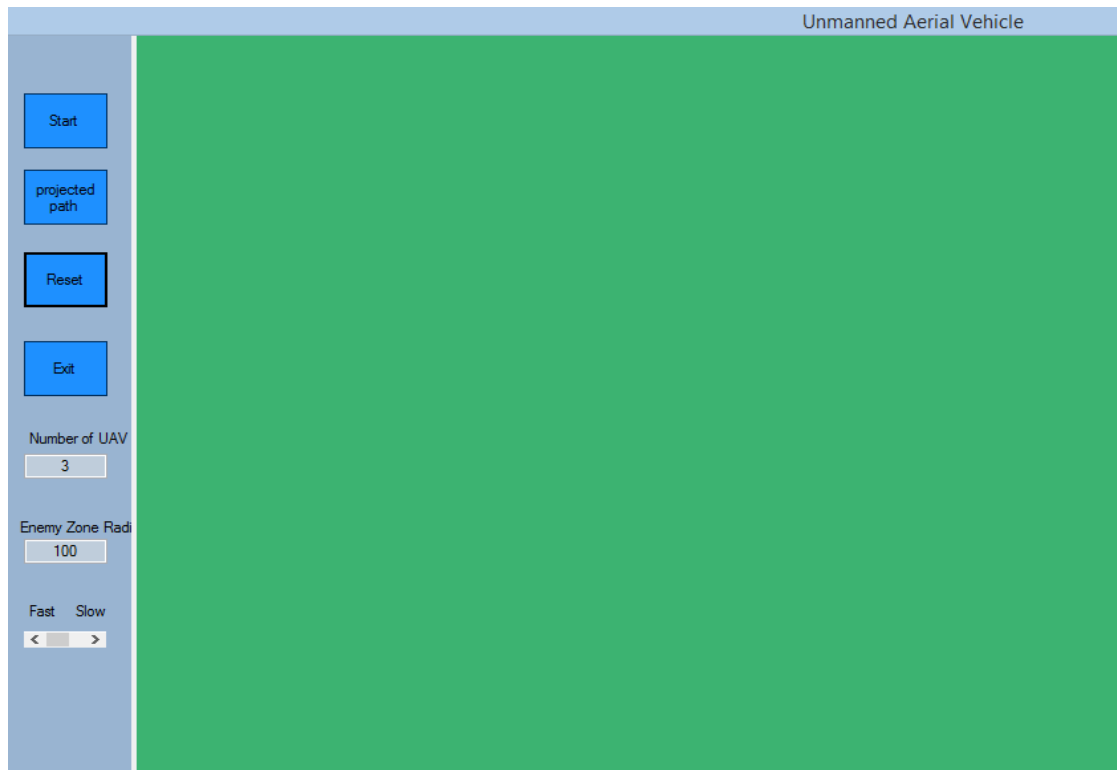


Figure 5-1: Flight controlling buttons, parameters, and area to display the flight

5.1. Scenario 1

Figure 5-2 and 5-3 show a mission by a single UAV to hit the target. Figure 5-2 is the screen snapshot taken after a while since the UAV flight started and the black solid line is the path travelled by the UAV, and the dotted red line shows the projected path that the UAV will go through. On the other hand, Figure 5-3 shows the full track the UAV travelled towards the target. The radius of the enemy zone in this case is 100 units as marked by the red circle.

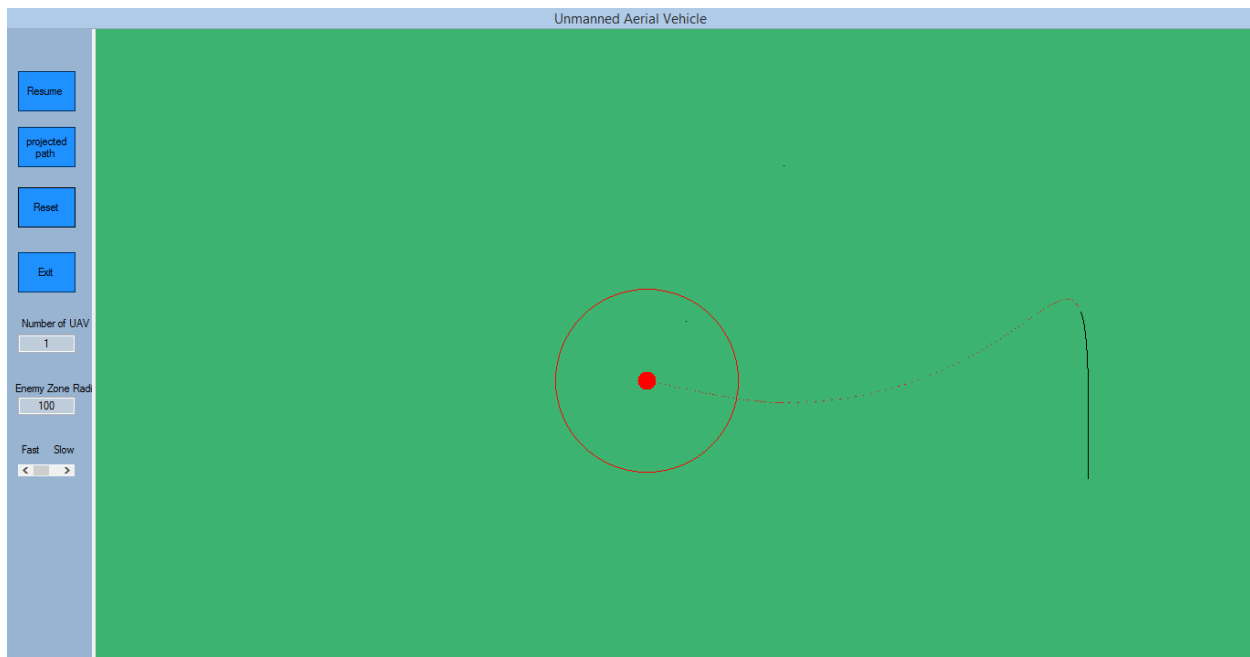


Figure 5-2: A single UAV projected path heading towards the target

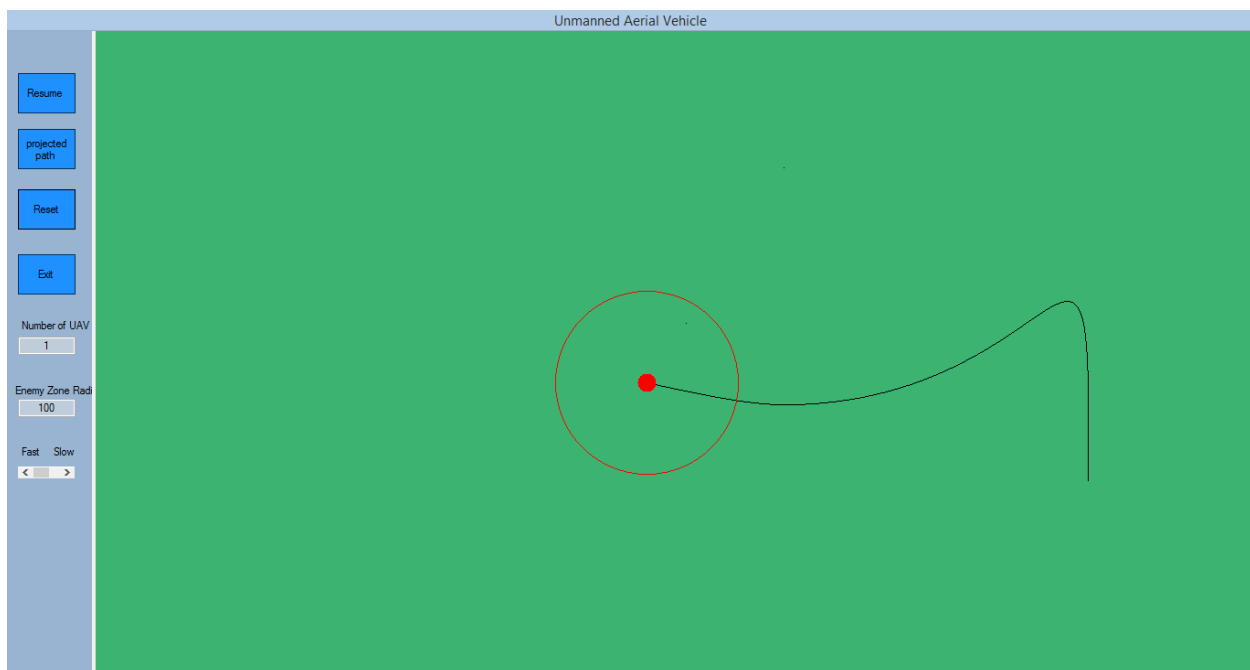


Figure 5-3: A single UAV full travelled path towards the target

5.2. Scenario 2

These snapshots Figure 5-4 and 5-5 are for the case of two UAVs as they travel towards the target. In this scenario, the planes started from their airbase outside the enemy zone that is determined at the beginning of the each simulation and heads towards the enemy target and enter the enemy zone at different angles separated by 180° from each other, which helps to reduce enemy detection. Figure 5-4 is the projected path after the planes launched from their airbase, and Figure 5-5 shows the full travelled path to hit the enemy target. The radius of the enemy zone for this simulation is 100 units as shown by the red circle.

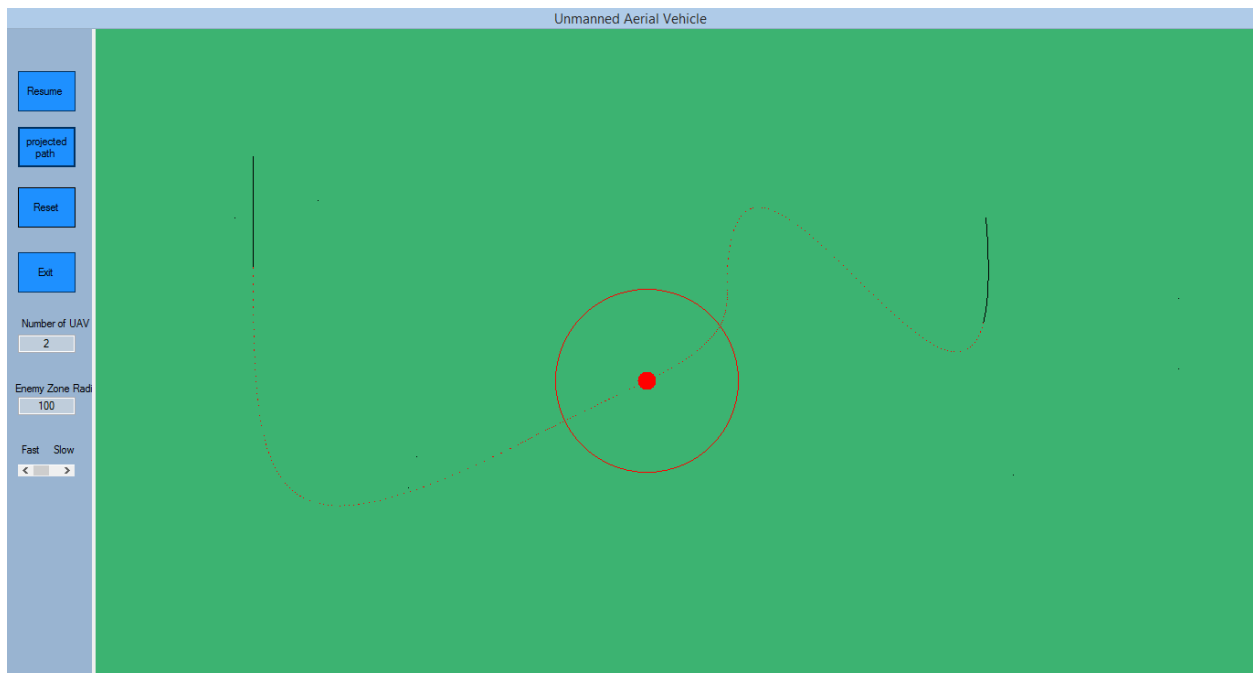


Figure 5-4: Projected course for two UAVs

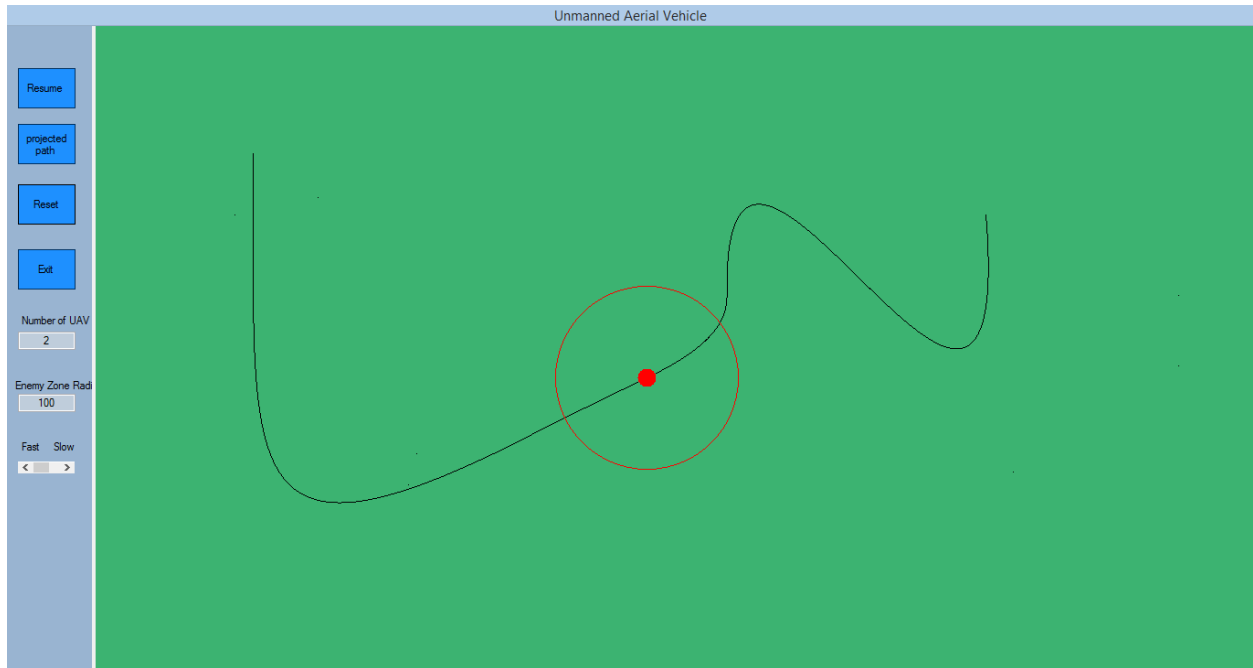


Figure 5-5: Path travelled by two UAVs

5.3. Scenario 3

This scenario shows three UAVs as they enter the enemy zone at 120° from each other. Figure 5-6 shows the projected path for three UAVs after they left their airbase and Figure 5-7 shows the full path travelled by the UAVs towards the target. The starting position is determined at the beginning of each simulation, which is outside of the enemy zone. The enemy zone in this simulation is a 100 unit's radius.

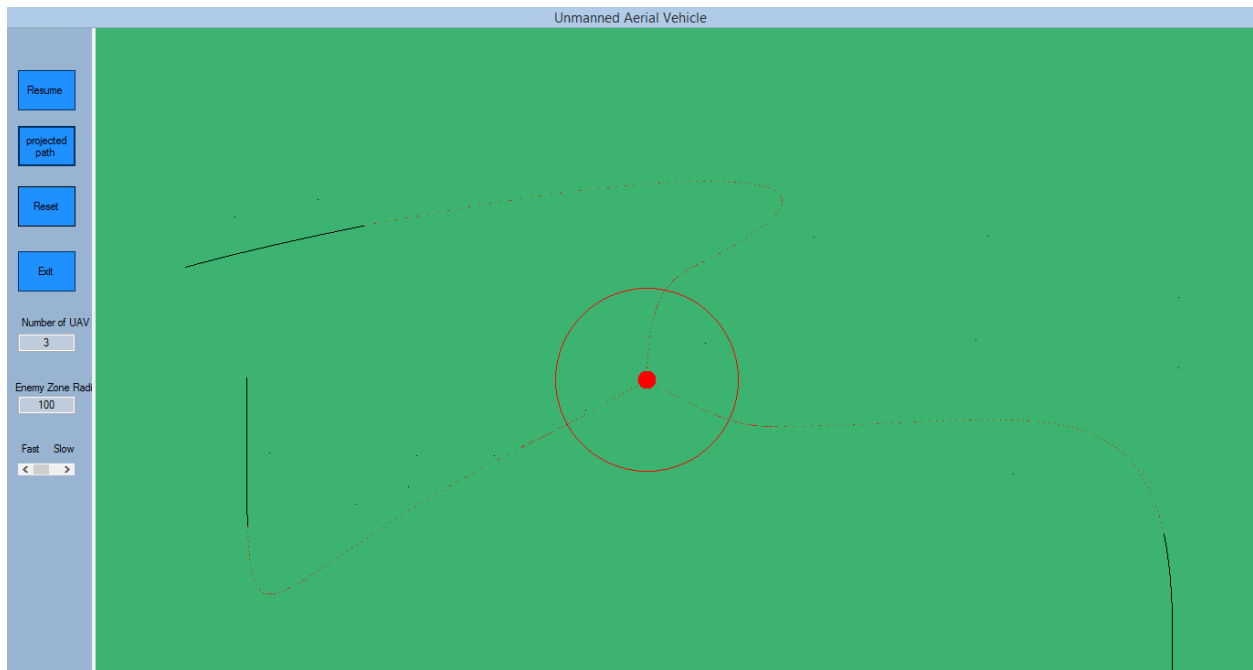


Figure 5-6: Projected path of the three UAVs

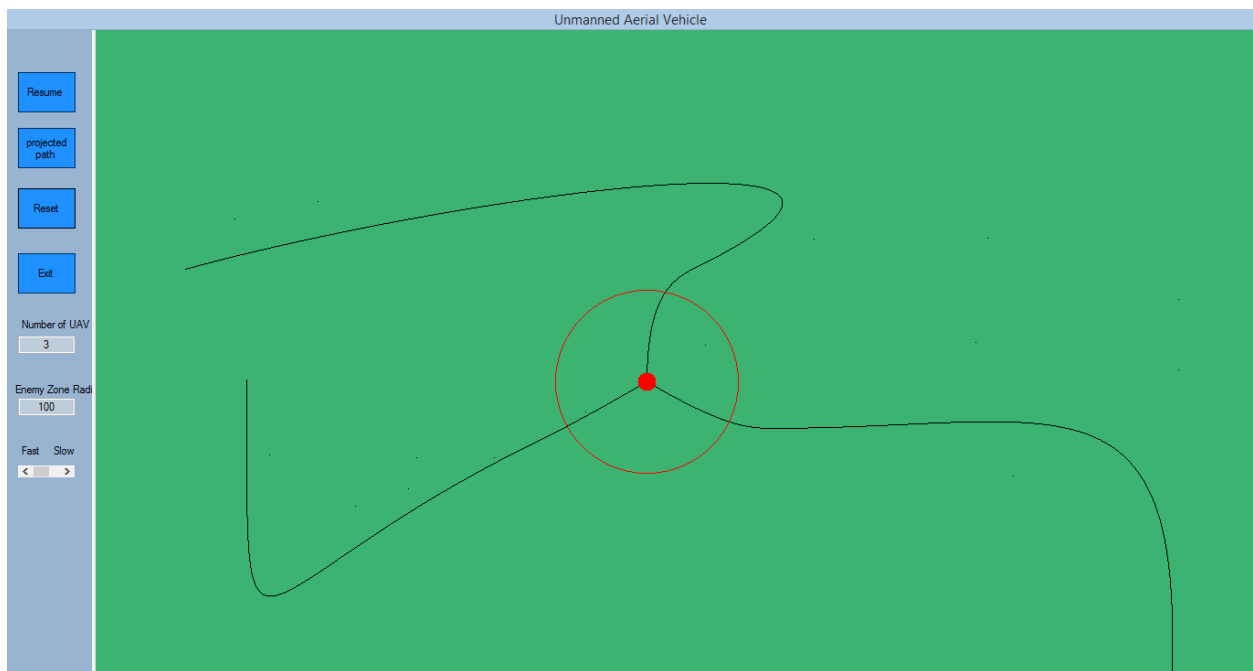


Figure 5-7: Path travelled by three UAVs towards enemy target

5.4. Scenario 4

The following snapshots are showing four UAVs on a mission to enter the enemy zone and destroy the enemy target coming from different angle separated by 90° from each other. Figure 5-8 shows the projected path for four UAVs after they left their airbase, and Figure 5-9 shows the full path travelled by the UAVs towards the target. The enemy zone has a 100 units radius.

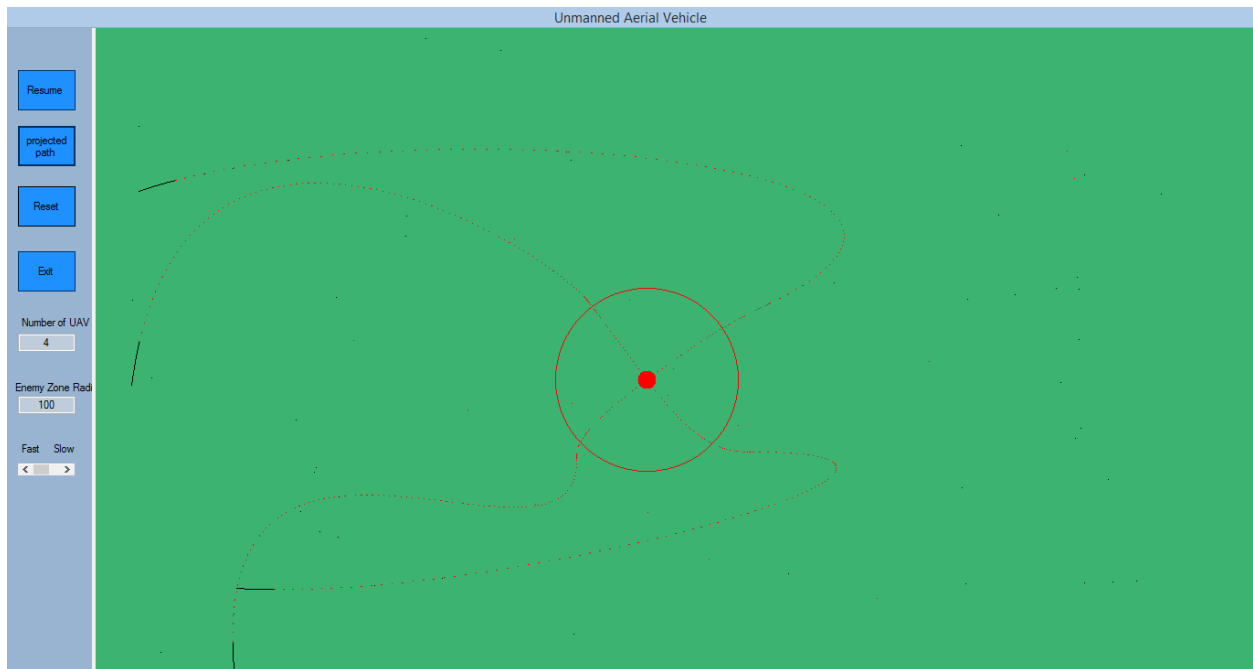


Figure 5-8: Projected path for four UAVs

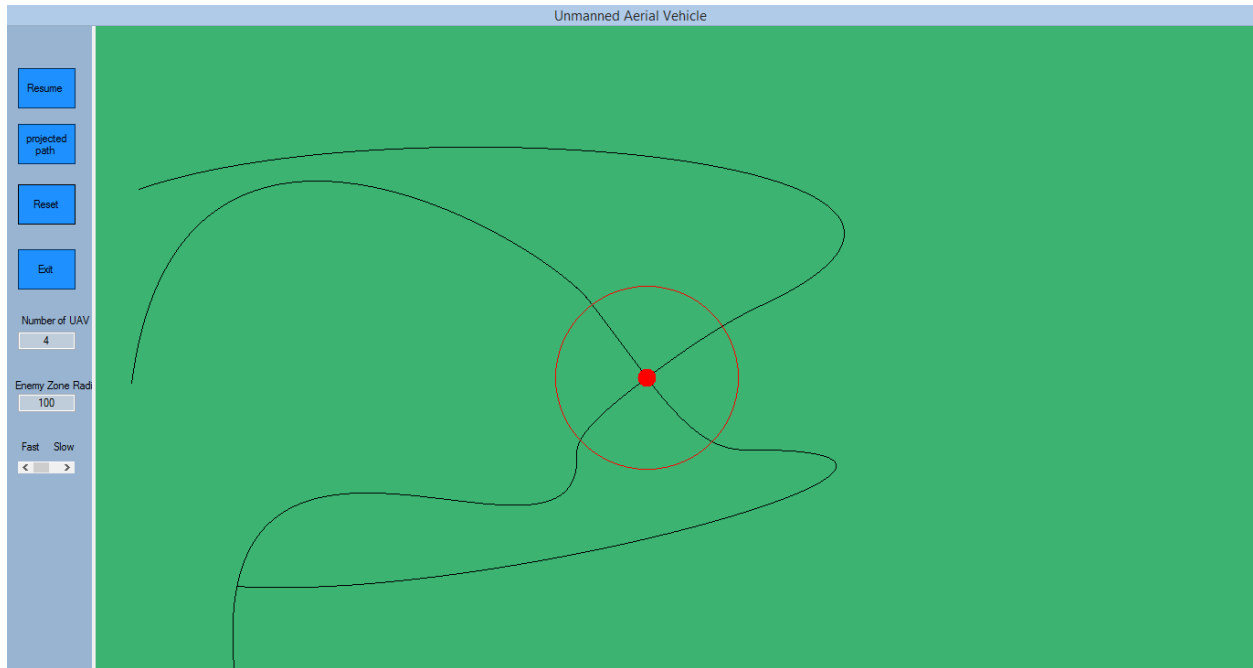


Figure 5-9: Path followed by four UAVs

5.5. Scenario 5

Figure 5-10 is a snapshot for five UAVs on a mission entering the enemy zone coming at different angle with 72° apart from each other. The enemy zone radius for this simulation is equal to 100 units.

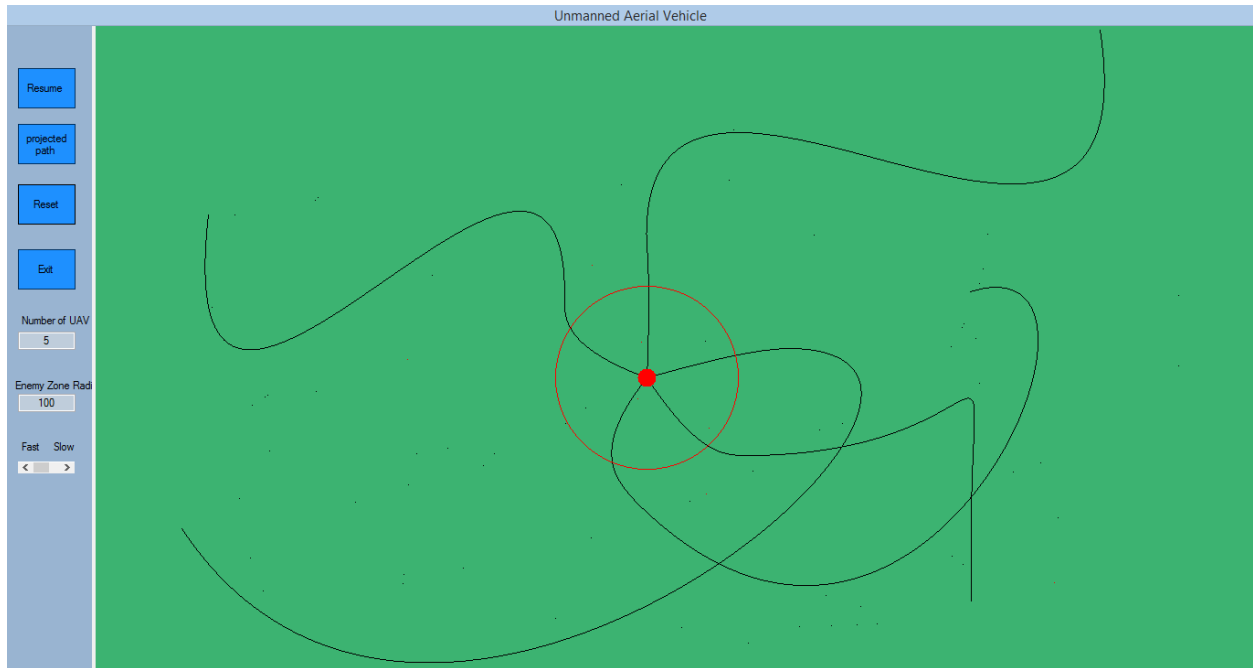


Figure 5-10: The path followed by five UAVs on their mission

5.6. Scenario 6

In this scenario, there are six UAVs on the mission and Figure 5-11 shows the course of planes launching from their airbase and crossing the enemy zone, radius of 100 units, to the enemy target. These UAVs entered the enemy zone separated at angle of 60° from each other.

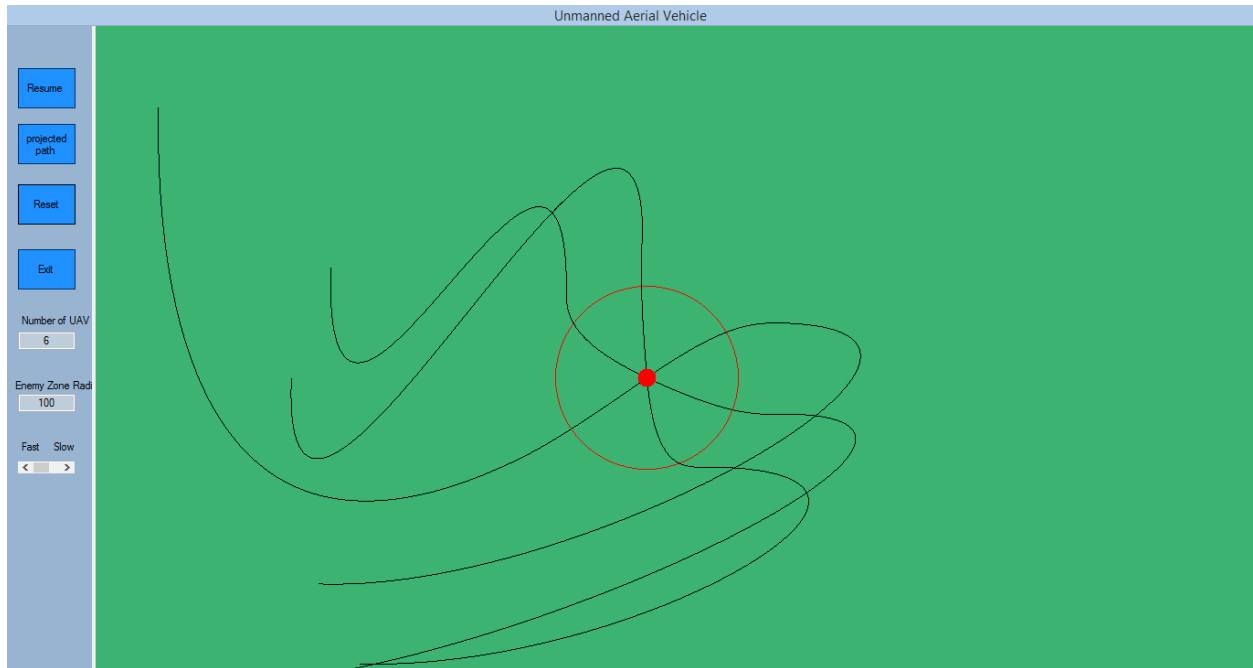


Figure 5-11: Six UAVs path in the way to destroy enemy target

5.7. Scenario 7

Seven UAVs on the mission entering the enemy zone at 51.4° angle from each other. Figure 5-12 shows the path travelled by the UAVs from their airbase to hit and destroy their target. The enemy zone for this simulation is 100 units of radius.

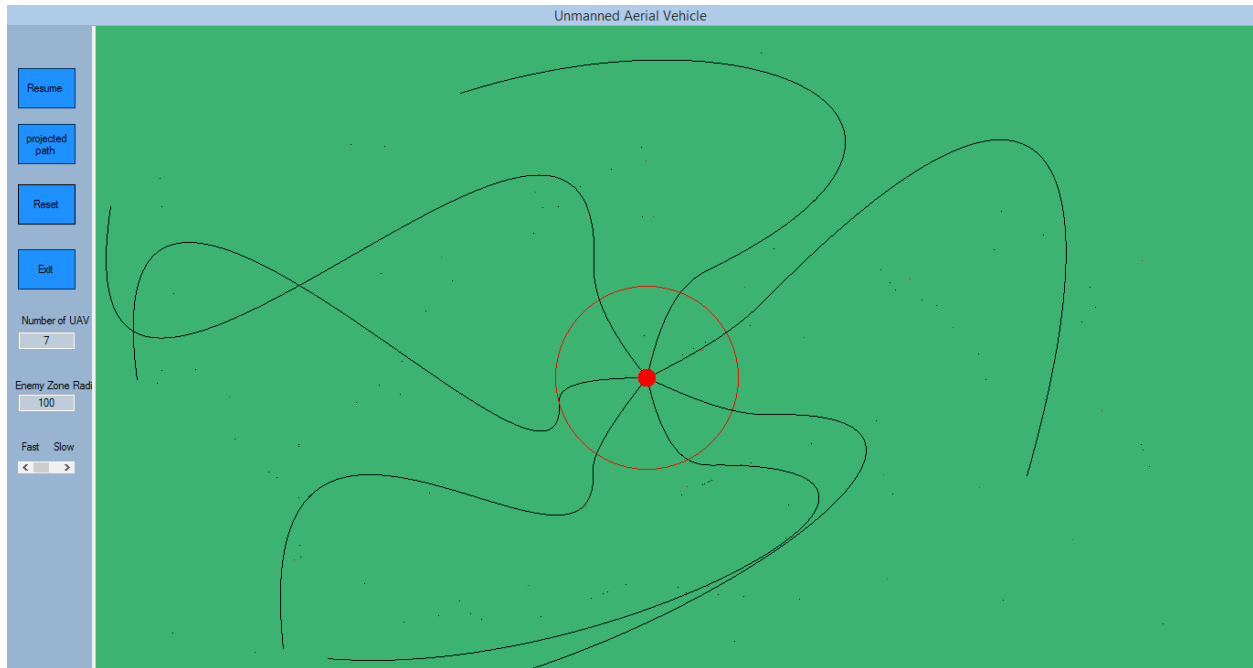


Figure 5-12: Flight path for seven UAVs

5.8. Scenario 8

Figure 5-13 shows eight UAVs on the mission. The planes started from their airbase where the position is determined at the beginning of each simulation crossing the enemy zone radius of 100 units to destroy the enemy target. The UAV entered the enemy zone at 45° apart from each other to each other.

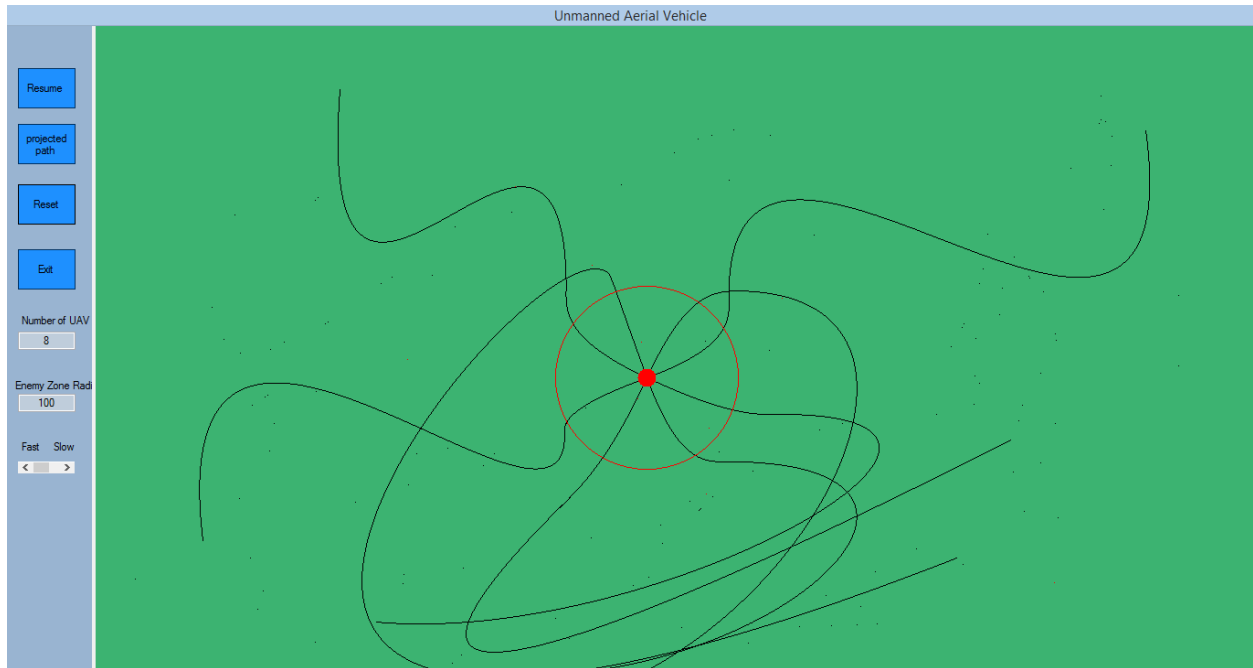


Figure 5-13: Eight UAVs flight path

5.9. Scenario 9

Figure 5-14 is the course followed by nine UAVs on the mission to destroy the enemy target. The planes entered the enemy zone at 40° apart from each other in the enemy zone with radius 100 units shown in the red circle.

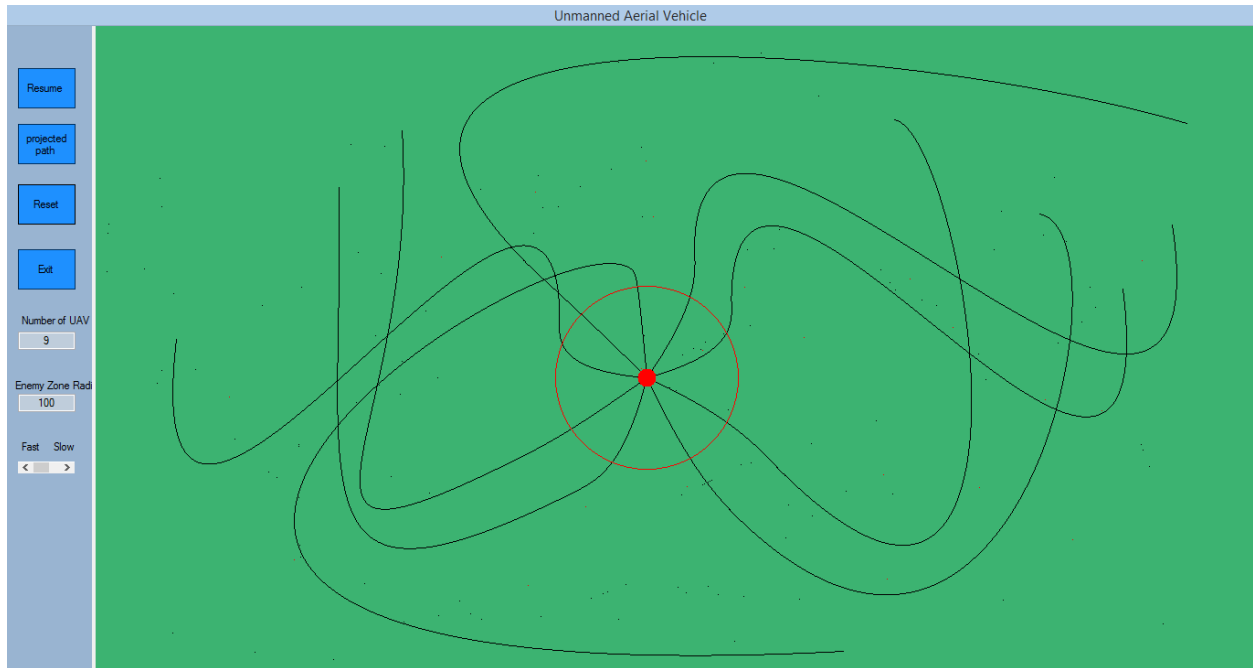


Figure 5-14: Nine UAVs flight path towards the enemy target

5.10. Scenario 10

In this scenario, ten UAVs are sent out on the mission to destroy the enemy target entering the enemy zone of radius 100 units coming from at an angle of 36° from each other. The snapshot is taken after a while since the UAV left the airbase. The airbase positions are determined randomly at the beginning of each simulation.

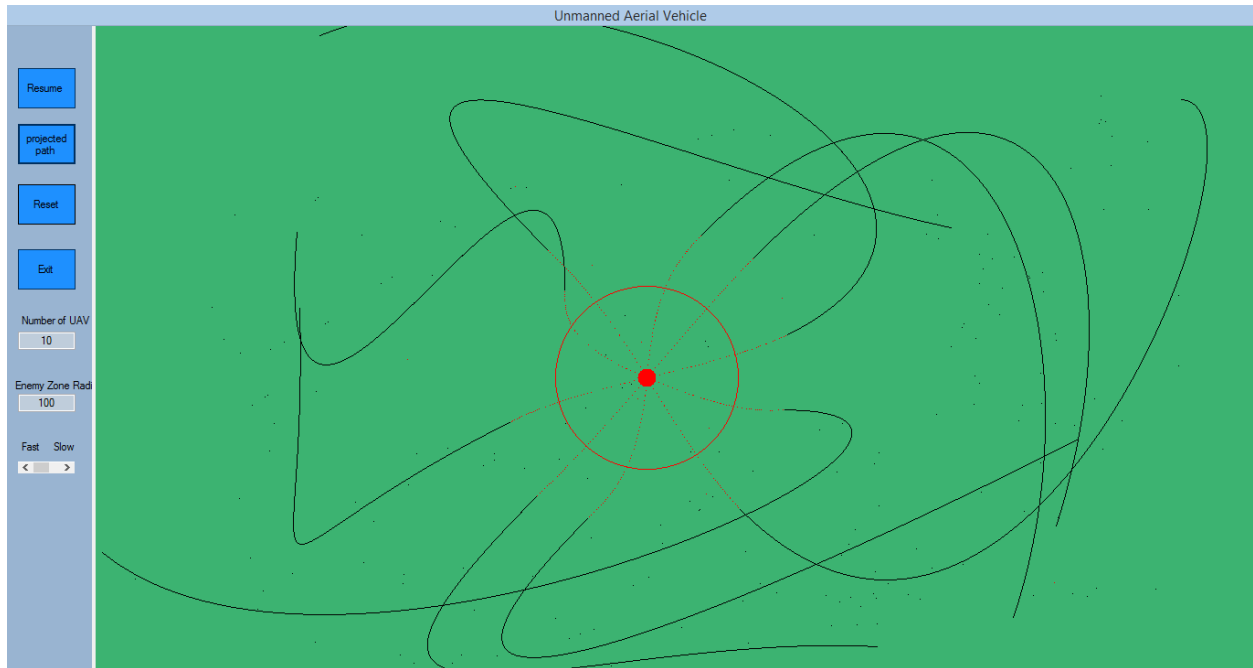


Figure 5-15: Ten UAVs course of flight

5.11. Scenario 11

In this scenario, revisiting three UAVs on mission to destroy enemy target, but the enemy zone is twice as large as in the previous scenarios, enlarged from one to ten. The enemy areal zone radius in this case is 200 units as shown by the red circle. Figure 5-16 shows the projected path with the red dotted line as UAVs enter the enemy zone at the same time coming from different angles separated by 120° from each other's. Figure 5-17 shows the scenario after the planes reached the enemy target.

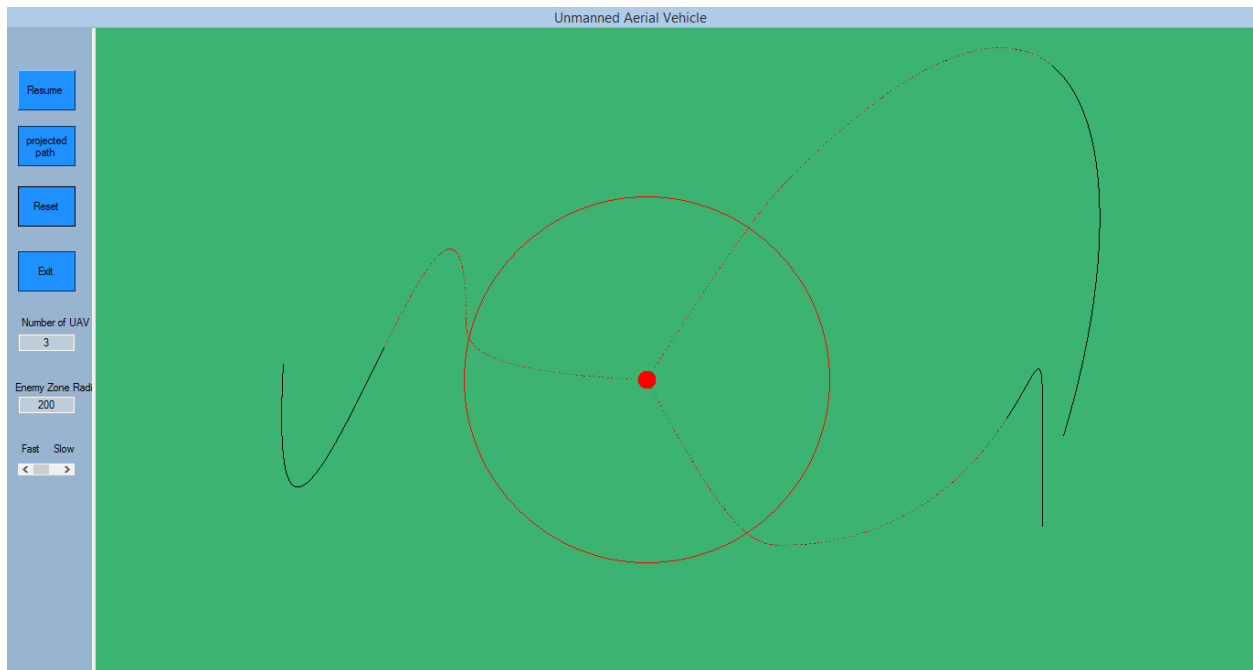


Figure 5-16: Projected path for three UAVs on mission crossing enemy zone of 200 units of radius

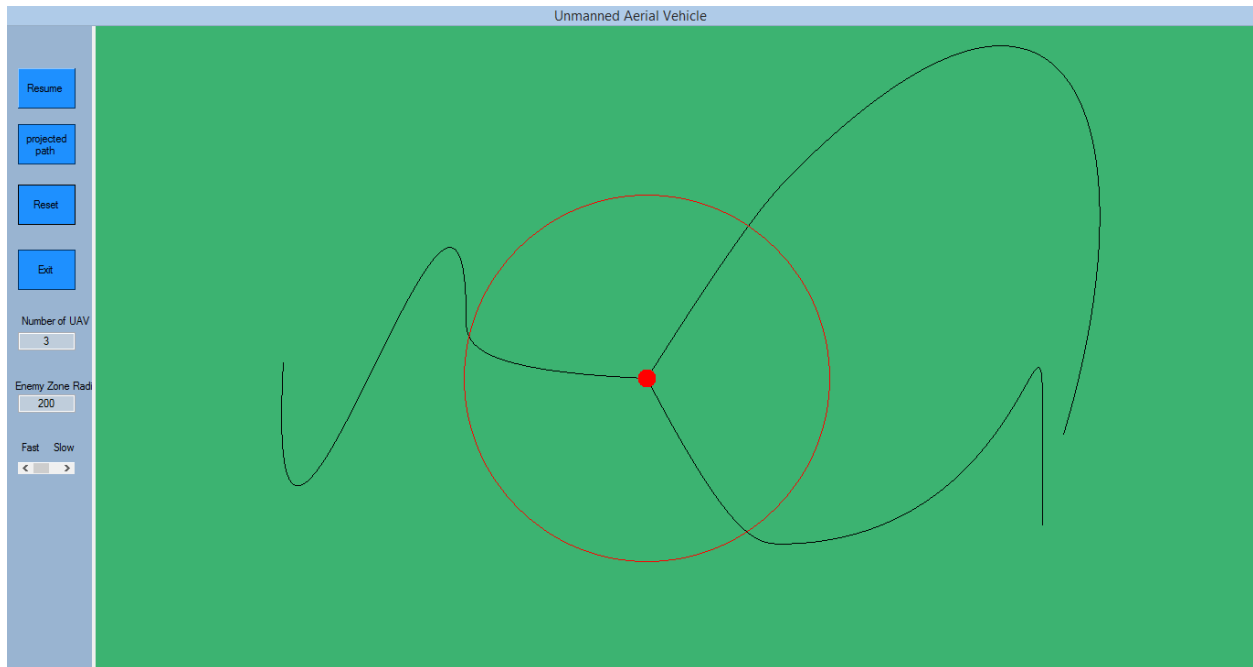


Figure 5-17: Flight path followed by three UAVs on mission with 200 units' enemy zone radius

5.12. Scenario 12

This scenario is also when three UAVs sent out on the mission to eliminate enemy target but the enemy zone areal radius is smaller as compared to all previous scenarios. Here the radius of the enemy zone is 50 units, and Figure 5-18 shows the UAVs travelled for a while, whereas Figure 5-19 is after the UAVs reached to the enemy target.



Figure 5-18: Projected path for three UAVs on mission with 50 units' enemy zone radius

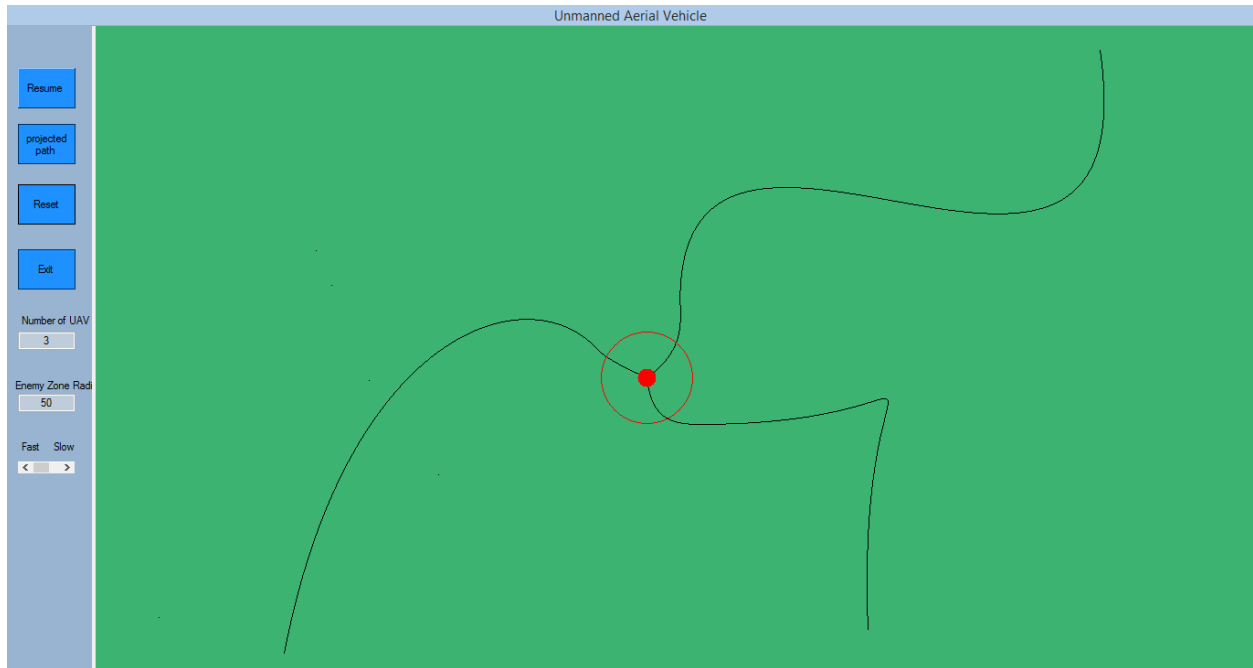


Figure 5-19: Flight path followed by three UAVs on mission with 50 units' enemy zone radius

6. DISCUSSION

The above results show different scenarios for the UAVs launched to destroy their enemy zone coming from different angle to hit the target at the same time. The scenarios 1-10 show that one to ten UAVs are in mission to hit their target in this case the enemy zone radius is 100 units, where the plane enters at the same time the target area. The scenarios 11-12 show when three UAVs are in the mission for different size of enemy zone namely 200 units and 50 units. In all cases of different number of UAVs and enemy zone sizes, the planes get to the enemy zone at the same time to hit the target.

7. CONCLUSION AND FUTURE WORK

The demand for UAVs in military is increasing along with the technology particularly autonomous aerial vehicles as they reduce human errors and shortage of trained pilots. This work addresses the simulation for unmanned aerial vehicles to launch from different airbases that are not predetermined to hit their target at the same time coming from different angle. The results illustrated the flight path followed by autonomous UAVs for a number of cases where different numbers of UAVs participate in the battlefield to destroy the enemy targets. Although, the maximum numbers of UAVs involved in the simulation are up to 10, the algorithm is scalable to more than 10 UAVs to be launched at the same time to accomplish their mission. In this simulation the enemy zone, a strong enemy hold area, is scalable from a small size circular area measured from the target to a large area depending on the strength of the enemy defensive capability. But in all cases the UAVs launched from outside the enemy zone and hit the target entering the enemy zone at the same time to reduce visibility and retaliation of the enemy. There are also some assumptions considered to simplify the design and implementation with less effect on the real world case. This work can be extended by including additional constraints such as terrain and manmade structures, weather conditions, GPS system on each plane to interact with each other. These additional parameters make the simulation more realistic but it comes with a higher computational time.

8. REFERENCES

- [1] “Air Force officials announce remotely piloted aircraft pilot training pipeline,”
www.af.mil. (July 2014).
- [2] A.Watts , V.Ambrosia and E. Hinkley, “Unmanned Aircraft
Systems in Remote Sensing and Scientific Research: Classification and Considerations of
Use”, Remote Sens. 2012, 4, 1671-1692.
- [3] H. Eisenbeiss, “A Mini Unmanned Aerial Vehicle (Uav): System Overview And Image
Acquisition”, International Workshop on "Processing And Visualization Using High-
Resolution Imagery" November 18-20 2004, Pitsanulok, Thailand.
- [4] Unmanned Aerial Vehicles (UAVs) – Overview, History and Market,
<http://www.azom.com/article.aspx?ArticleID=10156>. (July 2014).
- [5] Wikipedia, “General Atomics MQ-9 Reaper,”
http://en.wikipedia.org/wiki/General_Atomics_MQ-9_Reaper. (July 2014).
- [6] “Time Line for UAV,” <http://www.pbs.org/wgbh/nova/wartech/uavs.htm>. (July 2014).
- [7] C. Jones, “Unmanned Aerial Vehicles (UAVs) an Assessment of
Historical Operations and Future Possibilities” (Mar 97).
- [8] J. Pike, “Unmanned Aerial Vehicles”, <http://www.fas.org/irp/program/collect/uav.htm>,
(Aug 05).
- [9] K. Cook, “The Silent Force Multiplier: The History and Role of UAVs in Warfare,”
Aerospace Conference, 2007 IEEE.
- [10] <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA469584>.
(July 2014).
- [11] S. Banda, J. Doyle, R. Murray, J. Paduano, J. Speyer, G. Stein, “Research needs in

- dynamics and control for uninhabited aerial vehicles (UAVs),” Panel Rep 1997.
- [12] Q. Deng, J. Yu, N. Wang, “Cooperative task assignment of multiple heterogeneous unmanned aerial vehicles using a modified genetic algorithm with multi-type genes,” Chinese Journal of Aeronautics Volume 26, Issue 5, October 2013, Pages 1238–1250.
- [13] Y. Yoon, S. Gruber, L. Krakow, and D. Pack, “Autonomous target detection and localization using cooperative UAVs,” Optimization and Cooperative Control Strategies, Lecture Notes in Control and Information Sciences, 2008, 381-1.
- [14] K. Mullens, B. Troyer, R. Wade, B. Skibba, and M. Dunn, “Collaborative engagement Experiment,” In SPIE Proc of Unmanned Systems Technology VIII, Defense Security Symposium, April 2006.
- [15] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, “Cooperative air and ground surveillance,” IEEE Robotics & Automation Magazine, 1070-9932, 2006, 16–26.
- [16] X. Wang, V. Yadav, S.N. Balakrishnan, “Cooperative UAV Formation Flying with Stochastic Obstacle Avoidance,” AIAA Guidance, Navigation and Control Conference and Exhibit, San Francisco, California, August 2005, 15-18.
- [17] S. Bayraktar, G.E. Fainekos, and G.J. Pappas, “Experimental Cooperative Control of Fixed-Wing Unmanned Aerial Vehicles,” In Proceedings of the 43rd IEEE Conference on Decision and Control, The Bahamas, December 2004.
- [18] D. Nelson, T. McLain, R. Christiansen, R. Beard, David Johansen, “Initial Experiments in Cooperative Control of Unmanned Air Vehicles,” In Proceedings AI 3rd Unmanned Unlimited Systems Conference and Workshop, Chicago, September, 2004.
- [19] R. Bamberger, D. Watson, D. Scheidt, and K. Moore, “Flight demonstrations

- of unmanned aerial vehicle swarming concepts,” Johns Hopkins APL Technical Digest, Vol. 27, No. 1, 2006, pp. 41–55.
- [20] M. Hirsch, H. Ortiz-Pena, H., N. Sapankevych, N., and R. Neese, “Efficient Flight Formation for Tracking of a Ground Target,” Proceedings of the National Fire Control Symposium, 2007, pp. 1–16
- [21] Defense Science Board Task Force Report, “The role of autonomy in DoD systems,” 2012.
- [22] L. Bertuccelli y, Han-Lim Choi z, Peter Cho x, and Jonathan P. “How, Real-time Multi-UAV Task Assignment in Dynamic and Uncertain Environments”
- [23] C. Cook, “Perspectives on Acquisition, Test, and Early Fielding of UAV Systems,” Department of Operational Test and Evaluation (2008).
- [24] T. Shima and S. Rasmussen, “UAV Cooperative Decision and Control,” SIAM, 2009.
- [25] R. Garcia and L. Barnes, “Multi-UAV Simulator Utilizing X-Plane,” J. Intell. Robotics Syst., vol. 57, no. 1-4, pp. 393–406, Jan. 2010.
- [26] L. Barnes, M. Fields, and K. Valavanis, “Unmanned ground vehicle swarm formation control using potential fields,” in 15th Mediterranean Conference on Control & Automation, July 27 - 29 2007.
- [27] T. Paul, T. Krogstad, and J. Gravdahl, “Modelling of UAV formation flight using 3rd potential field,” Simulation Modelling Practice and Theory, vol. 16, no. 9, pp. 1453 – 1462, 2008.
- [28] H. Choi, L. Brunet, and J. How, “Consensus-based decentralized auctions for robust task allocation,” IEEE Trans. on Robotics, Vol. submitted, 2008.
- [29] L. Brunet, H. Choi, and J. How, “Consensus-based auction approaches for decentralized task assignment,” AIAA Guidance, Navigation and Control Conference, 2008.

- [30] http://en.wikipedia.org/wiki/B%C3%A9zier_curve. (July 2014).
- [31] Graphics and Drawing in Windows Forms, <http://msdn.microsoft.com/en-us/library/a36fascx%28v=vs.110%29.aspx>. (July 2014).