Training Seminar Report

Efficacy Analysis of Ballistic Missiles for Runway Denial

At

Air Force Station, Jaipur

Submitted by

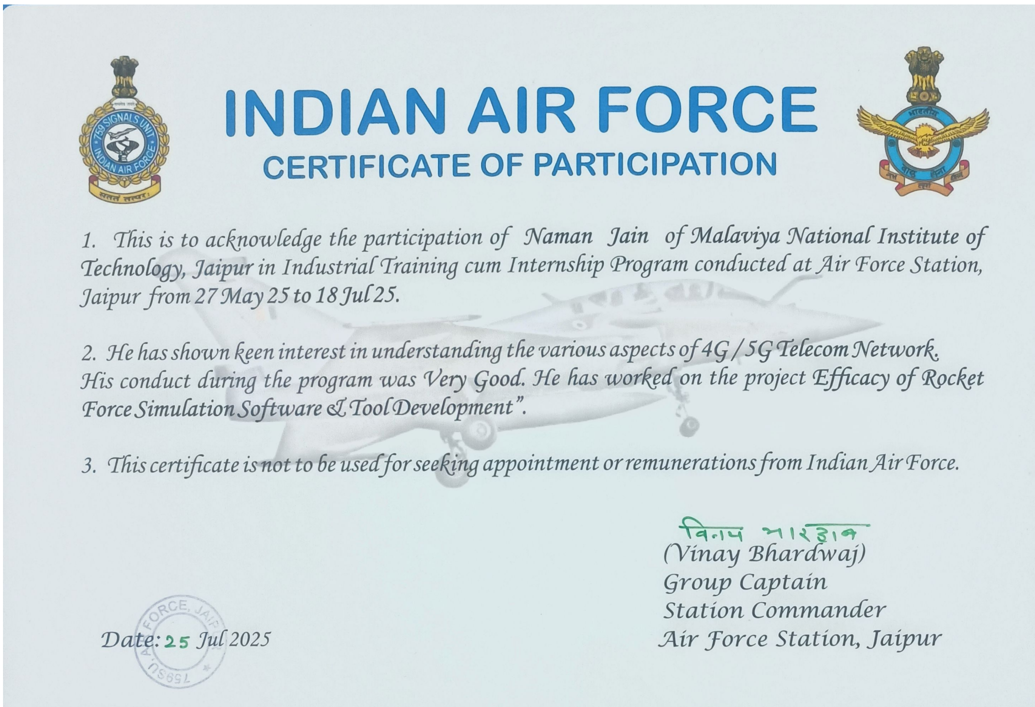
Naman Jain (2022UCP1450)

Supervised by

Group Captain Vinay Bharadwaj

Department of Computer Science and Engineering
Malaviya National Institute of Technology, Jaipur

Certificate



INDIAN AIR FORCE

CERTIFICATE OF PARTICIPATION

1. This is to acknowledge the participation of Naman Jain of Malaviya National Institute of Technology, Jaipur in Industrial Training cum Internship Program conducted at Air Force Station, Jaipur from 27 May 25 to 18 Jul 25.

2. He has shown keen interest in understanding the various aspects of 4G / 5G Telecom Network. His conduct during the program was Very Good. He has worked on the project Efficacy of Rocket Force Simulation Software & Tool Development".

3. This certificate is not to be used for seeking appointment or remunerations from Indian Air Force.

विनय भारद्वाज

(Vinay Bhardwaj)

Group Captain

Station Commander

Air Force Station, Jaipur

Date: 25 Jul 2025

Abstract

In the evolving landscape of modern security, military airfields and their runways, parking, and dispersal zones are prime, high-value targets for precision-guided missiles and submunition warheads. This project develops a Missile Trajectory Simulation and Runway Vulnerability Analysis Tool that couples a physics-based missile flight model with a probabilistic runway and parking-area destruction simulator. The trajectory module incorporates thrust, drag, gravity, variable mass, and realistic atmospheric modeling over the WGS-84 ellipsoid with great-circle geometry to generate accurate 3D flight paths. The airfield module models statistical impact errors via Circular Error Probable (CEP), submunition dispersal within a specified radius, and resulting damage to runway and parking areas.

Through an intuitive GUI, users can configure launch and target parameters, run simulations, and visualize results with 2D/3D plots and interactive web maps. Monte Carlo methods aggregate many stochastic trials to estimate probabilities of achieving single or multiple runway cuts and quantifying damaged parking-area percentages under varying CEP, submunition counts, and damage radii. Results show that improving missile accuracy (reducing CEP) and increasing effective submunition radius significantly reduce the number of missiles required to deny runway operations. The tool demonstrates educational and analytical value for understanding trade-offs between missile guidance, payload design, and airbase resilience in runway denial missions.

Declaration

I, Naman Jain, bearing Institute ID **2022UCP1450**, hereby declare that the internship work presented in this report, titled "Efficacy Analysis of Ballistic Missiles for Runway Denial", submitted to the Department of Computer Science and Engineering, Malaviya National Institute of Technology, Jaipur, is an authentic record of the work carried out by me during my internship at Air Force Station, Jaipur.

This work was completed under the mentorship of Group Captain Vinay Bharadwaj. The content of this report, in full or in parts, has not been copied from any existing work of any other individual or organization and has not been submitted elsewhere at any time for any academic or training requirement.

Naman Jain
2022UCP1450
Friday 21st November, 2025

Acknowledgement

I would like to express my deepest gratitude to Group Captain Vinay Bharadwaj Sir, under whose valuable guidance this project was successfully completed. I also sincerely acknowledge the Indian Air Force for providing support and inspiration throughout the duration of this internship. I extend my heartfelt thanks to my mentors, peers, and all those who contributed directly or indirectly to the success of this work. Finally, I am thankful to my institute, Malaviya National Institute of Technology Jaipur, and to my family and friends for their constant encouragement and support.

Contents

1 Introduction

1.1 Basic Overview

In the complex and evolving landscape of modern security, the accuracy, reliability, and resilience of critical infrastructure, particularly military airfields, are fundamental to a nation's rapid deployment and strategic defense capabilities. Airfield runways, parking, and dispersal zones represent high-value targets for adversaries employing precision-guided missiles and submunition warheads in contemporary conflict scenarios. Effective runway denial can decisively limit force projection and sortie generation.

This project introduces a comprehensive Missile Trajectory Simulation and Runway Vulnerability Analysis Tool that goes beyond basic flight visualization to deliver actionable, data-driven operational insights. The system integrates advanced trajectory physics, including thrust, drag, variable missile mass, and realistic atmospheric modeling, with precise geo-referencing of airfield runways and dispersal areas. This dual-layered simulation ensures physically accurate flight paths while also modeling the statistical and spatial effects of weapon impacts.

Unlike conventional simulation tools, the platform is designed for both depth and usability. Through an intuitive graphical user interface (GUI), users can define mission parameters, input missile characteristics, and interactively analyze outcomes. Detailed visualizations, including 2D/3D trajectory displays and geographic overlays, enable clear, mission-relevant feedback and support exploratory "what-if" analyses.

Key operational questions addressed include:

- How many missiles, and of what accuracy (CEP) and payload, are needed to achieve a physical cut on the runway or parking area under varying conditions?
- How does variation in submunition count, dispersal radius, or reliability affect the probability of denying airfield use?

- What is the relationship between missile system parameters and success rates for achieving single or multiple runway cuts?

By tightly coupling trajectory physics with high-fidelity damage and probability modeling, the project closes the gap between engineering simulation and real-world operational analysis, providing actionable insight into the relationship between missile employment and airbase resilience.

1.2 Problem Statement

Military airfields and runways are longstanding, high-value targets in both strategic and tactical conflicts due to their role in force projection and sortie generation. Traditional runway denial operations have evolved from mass unguided bombings to precision-guided munitions and advanced submunition warheads. With adversaries fielding hardened shelters, dispersal areas, and rapid repair capabilities, planners must answer:

- How many missiles, and of what characteristics, are needed to reliably deny an adversary runway usage?
- What is the likelihood of successful single or multiple cuts given real-world missile performance with uncertainties in accuracy and yield?
- How do changes in submunition count, dispersal radius, and CEP impact runway and parking-area damage?

Previous studies often assume perfect accuracy, ignore missile kinematics, or rely on static, non-interactive representations that limit exploration of alternative tactics or hardware investments. The challenge is to develop a simulation system that:

- Realistically models missile trajectories from launch to impact, including gravity, thrust, drag, and changing mass.
- Reflects the statistical nature of weapon accuracy (CEP) and submunition effectiveness, including random failures.

- Visibly demonstrates runway and parking-area destruction in a way that is physically grounded yet intuitively clear.
- Allows users to interactively explore missile employment options and iteratively refine strategies.

1.3 Goals and Scope

1.3.1 Project Goals

- **Physics-Based Trajectory Simulation:** Develop a simulation engine using real-world physical modeling for missile flight, considering thrust, gravity, drag, variable mass, and Earth curvature via WGS-84 and great-circle mathematics.
- **Comprehensive Runway Damage Modeling:** Accurately represent runway geometries and parking/dispersal areas, and model impact, submunition dispersal, damage radius, and random failures.
- **User-Friendly Graphical Interface:** Provide a smooth GUI so users can input parameters, visualize results, and run advanced scenario analyses without coding.
- **Statistical and Monte Carlo Analysis:** Include routines to simulate many shots with variable CEP and submunition parameters, summarizing results as probability curves and missile-count requirements.
- **Advanced Visualization and Mapping:** Present results as plots and geographic overlays using OpenStreetMap, Folium, and matplotlib, including real trajectory animations and runway cut visualizations.
- **Extensibility and Educational Value:** Ensure the engine and GUI are extensible to new munitions and targets, and usable as an educational tool in academic or training settings.

1.3.2 Scope and Boundaries

- The software targets fixed-wing missiles with ballistic or quasi-ballistic trajectories, not cruise or loitering munitions.
- Effects modeled focus on runway denial, not deep cratering physics, incendiary behavior, or long-term repair modeling.
- Atmospheric modeling uses an exponential density profile; wind and complex weather effects are not included.
- All calculations use unclassified, open-source parameterizations suitable for teaching and preliminary analysis.
- The implementation is in Python, using standard scientific, mapping, and GUI libraries for accessibility and modification.

2 System Design

2.1 Architecture

The system consists of two loosely coupled Python applications:

- **final.py**: Missile trajectory simulation.
- **runway_final.py**: Airfield destruction simulation.

The user begins with the trajectory GUI in `final.py`, sets launch and target parameters using runway coordinates, runs the simulation, and views interactive maps. After completion, `final.py` automatically launches the airfield destruction GUI `runway_final.py`, where runway and parking damage are analyzed at a tactical level.

2.2 High-Level Design

2.2.1 Missile Trajectory Module (final.py)

- **GUI:** Tkinter-based interface for inputting launch parameters, missile specifications, and target coordinates.

- **Optimization Engine:** Bayesian optimization (`gp_minimize`) is used to find launch parameters (elevation angle, initial velocity) that minimize mean impact error.
- **Physics Engine:** A custom simulator calculates a 3D trajectory using Euler integration, modeling gravity, thrust, drag, atmospheric density, and variable mass.
- **Visualization:** Folium generates interactive HTML maps with launch and target markers, path polylines, and a timestamped animated missile icon.

2.2.2 Airfield Destruction Module (runway_final.py)

- **GUI:** Tkinter-based interface with an OpenCV-backed canvas to visualize the airfield layout, runway, and parking areas.
- **Destruction Logic (Runway):** Missile impact points are sampled from a Gaussian distribution (CEP). Submunition dispersal is modeled via uniform sampling within a specified radius. Destroyed areas are tracked to determine whether a continuous Minimum Operating Strip (MOS) remains.
- **Destruction Logic (Parking):** Submunition damage circles are accumulated to form masks. The total damaged area is calculated using contour detection and compared to overall parking area.
- **Graphical Output:** Matplotlib plots show relationships such as number of missiles versus submunitions required for cuts or area denial.

2.3 Low-Level Design

2.3.1 Core Functions in final.py

- **objective_function:** Called by `gp_minimize`; takes launch parameters and returns mean impact error over simulated trajectories.

- **optimization_main_loop:** Driver that configures `gp_minimize`, handles iterations, logging, and intermediate visualization.

- **simulate_3d_trajectory_with_thrust_and_drag:** Iteratively updates position and velocity using Euler integration with gravity, thrust, drag, atmospheric density, and drag coefficients.

- **create_enhanced_osm_map_with_animation:** Builds Folium maps with tiles, markers, polylines, and `TimestampedGeoJson` animations.

2.3.2 Core Functions in runway_final.py

- **on_mouse_click / attack_from_button:** GUI event handlers that initiate attack simulations.

- **generate_strike_point:** Uses `np.random.normal` to sample missile impact points based on CEP.

- **generate_submunition_points:** Uses uniform sampling for radius and angle to disperse submunitions within the damage radius.

- **diff / cut_width:** Computes remaining undamaged runway strips and checks for MOS-width segments.

- **parking_rectangle:** Visualizes parking area and uses `cv2.bitwise_or` to accumulate destroyed regions; `cv2.findContours` measures damaged area.

2.4 Data Structures

- Python lists and NumPy arrays for coordinates, time steps, thrust history, and mass history.

- Tkinter widgets organized using `Frame` and `LabelFrame` for structured GUIs.

- Folium and GeoPandas objects for geospatial data and map layers.

- Heaps (`heapq`) for efficient tracking of destroyed runway segment boundaries.

2.5 Algorithms

2.5.1 Euler Integration

Euler integration is used in `simulate_3d_trajectory_with_thrust_and_drag` to iteratively update missile position and velocity over small time steps. At each step, forces from gravity, thrust, and drag are computed, and state vectors are advanced in time, enabling realistic 3D trajectory modeling.

2.5.2 Gaussian and Uniform Sampling

- **Gaussian Sampling:** Missile impact points around the target are drawn from a 2D Gaussian distribution governed by CEP, representing targeting uncertainty.
- **Uniform Sampling:** Submunition impact locations are sampled uniformly inside a circle of specified radius around each missile impact, modeling dispersal patterns.

2.5.3 Bayesian Optimization

Bayesian optimization via `gp_minimize` searches the continuous space of launch parameters to minimize mean impact error. It builds a probabilistic surrogate of the objective and selects new points using acquisition functions, making it more efficient than grid or random search.

2.5.4 Monte Carlo Simulation

The airfield destruction module relies on Monte Carlo simulation. For each trial, missiles may fail or succeed, impact locations are sampled with CEP, submunitions are dispersed, and runway/parking damage is recorded. Repeating this process yields empirical probabilities of single or multiple runway cuts and damage percentages.

2.5.5 Haversine Formula and Contour Detection

The Haversine formula computes great-circle distances between geographic coordinates for ground-range and mapping tasks. In parking-area analysis, `cv2.findContours` operates on accumulated damage masks to detect boundaries and measure total destroyed area.

2.6 Technology Stack

- **Language:** Python 3
- **GUI:** Tkinter, ttkbootstrap
- **Math/Physics:** NumPy, math, geopy
- **Optimization:** scikit-optimize (`skopt`)
- **Visualization:** matplotlib, Folium, OpenCV (`cv2`)
- **Geospatial:** GeoPandas, osmnx
- **Utilities:** heapq, Pillow (PIL) for images

3 Code Snippets

This section documents representative code fragments from the implementation. Key code segments include:

- Bayesian optimization setup and objective function.
- Main optimization loop invoking the trajectory simulator.
- The core 3D trajectory integration with thrust and drag.
- Optimization result visualization.
- Runway destruction logic including submunition impacts and MOS checks.

In the complete project, these are shown as screenshots illustrating the implementation details. Code snippets can be included using `\includegraphics` for exported images.

4 Testing and Evaluation

The system was validated through multiple layers of testing.

4.1 Unit Tests

Individual physics and geometry functions such as atmospheric density, drag coefficient, and Haversine distance were tested against known analytical or reference values. This ensured correctness of low-level computations used in the trajectory integrator and destruction model.

4.2 Integration Tests

Integration tests focused on ensuring correct data flow from `final.py` to `runway_final.py`. After trajectory optimization and simulation, the destruction GUI is launched with consistent parameters, and interactive visualization behaves as expected across modules.

4.3 Scenario-Based Testing

Several scenario tests were conducted:

- **Runway Attack:** Simulations with varying CEP and submunition counts verified that lower CEP and more submunitions reduce the missile count needed to achieve a runway cut.
- **Parking Area Attack:** Runs confirmed that the calculated percentage of destroyed parking area increases with each successful attack, with contour-based area calculations cross-checked on simple shapes.
- **GUI Usability:** The user interface was evaluated for clarity, labeling, flow, and ease of running simulations and interpreting outputs.

5 Results and Discussion

5.1 Missile Trajectory

The physics-based trajectory model produced realistic 3D flight paths, capturing the influence of thrust-to-weight ratio, loft angle, and aerodynamic drag. Interactive Folium maps displaying the path over real-world basemaps proved useful for visualizing launch and impact locations, as well as understanding ground-range behavior.

5.2 Airfield Destruction

The airfield destruction simulation effectively demonstrated the probabilistic nature of runway and parking attacks. Visual feedback in the GUI, including impact points, submunition damage circles, and cut indicators, made it easy to interpret whether a given missile salvo achieved a runway cut or significant parking damage.

5.3 Performance Graphs

Performance graphs showed clear inverse relationships between missile requirements and parameters such as submunition count and damage radius. For example, sensitivity plots indicated that reducing CEP from around 100 m to about 30 m can roughly halve the missile count required for a single runway cut in representative scenarios.

5.4 Role of Submunition Radius and CEP

5.4.1 Definition and Importance

Circular Error Probable (CEP) is a statistical measure defining the radius within which 50% of warheads are expected to land around a target point; smaller CEP values correspond to higher accuracy. **Submunition radius** represents the effective damage radius around each submunition impact point, within which runway or parking surfaces are rendered unusable.

5.4.2 Modeling Approach

In the simulator:

- Missile impact points are sampled from a 2D Gaussian distribution centered at the target, parameterized by CEP.
- Submunition impacts are sampled uniformly within a circle defined by the submunition radius around each missile impact.

Multiple Monte Carlo trials build a statistical picture of damage patterns for different CEP and radius combinations.

5.4.3 Impact on Runway Denial

Results highlight that:

- Lower CEP concentrates missile impacts, increasing the chance that enough submunitions fall on or near the runway to create a continuous operational gap.
- Larger submunition radius expands individual damage footprints, leading to more overlap and reducing the number of missiles required for both runway cuts and parking denial.

Example curves show that improved accuracy and larger damage radii significantly shift probability curves in favor of successful denial with fewer missiles.

5.4.4 Practical Implications

The ability to model CEP and submunition radius enables:

- Trade-off analysis between investments in guidance systems versus payload improvements.
- Quantitative assessment of required missile inventories to meet specified denial probabilities.
- Identification of critical threshold values for CEP and damage radius in planning.

6 Conclusion

This project successfully developed an integrated simulation platform that combines a realistic, physics-based missile trajectory model with a probabilistic airfield destruction engine. The tool enables users to explore how missile accuracy, submunition characteristics, and attack configurations influence runway and parking-area denial outcomes. Graphs and visualizations validate intuitive relationships, such as decreasing missile requirements with higher accuracy and larger submunition radii.

The dual-GUI design separating trajectory and destruction analysis proved effective for distinguishing strategic and tactical perspectives. Overall, the system provides a powerful and intuitive environment for defense-oriented analysis and education, bridging the gap between abstract missile performance parameters and concrete impacts on airbase resilience.

7 Future Work Recommendations

Potential extensions include:

- Incorporating environmental effects such as wind, Coriolis forces, and more detailed atmospheric models into the trajectory simulator.
- Modeling simultaneous multi-missile attacks from multiple launch locations.
- Adding defensive measures such as surface-to-air missiles or electronic countermeasures and their impact on CEP and missile survival.
- Extending the destruction model to other high-value targets such as command centers, fuel depots, and aircraft shelters.
- Implementing fully interactive 3D visualization using libraries like PyVista or VTK to unify trajectory and airfield views.
- Supporting data persistence for saving and loading scenarios and results for later analysis.

8 Limitations

Despite its capabilities, the current system has several limitations:

- The physics model uses simplified drag and gravity assumptions and does not fully capture high-fidelity missile aerodynamics.
- Destruction modeling is two-dimensional and assumes ground-level impacts, omitting aerial bursts and structural complexities.
- Environmental factors such as wind, weather, and detailed atmospheric variability beyond an exponential density profile are not modeled.
- Damage is assumed homogeneous within the submunition radius, ignoring material differences and complex damage patterns.
- Secondary effects (e.g., debris fields, blast overpressure beyond the modeled radius) and repair timelines are not included.
- The definition of a runway cut relies solely on MOS-based gaps, which may oversimplify operational usability.
- Geospatial data from osmnx may be incomplete for certain military locations, and the two modules depend on manual GUI interaction without robust inter-process communication or automatic result persistence.

References

References

[1] Wan-Li Wei, Ye-Qing Chen, Zhen-Qing Wang, Jian-Hui Wang, Xin-Jian Ren, and Shang Ma. Research on Runway Blockage Probability through Simulation. *Shock and Vibration*, Vol. 2022, Article ID 5631800, 13 pages, 2022.

[2] C. Zhai and X. Chen. Damage assessment of the target area of the island reef under the attack of missile warhead. *Defence Technology*, Vol. 16, No. 1, pp. 18-28, 2020.

[3] J. Morio, R. Pastel, and F. Le Gland. Missile target accuracy estimation with importance splitting. *Aerospace Science and Technology*, Vol. 25, No. 1, pp. 40-44, 2013.

[4] Y. H. Sun, C. C. Shi, Z. Liu, and D. S. Wen. Theoretical research progress in high-velocity hypervelocity impact on semi-infinite targets. *Shock and Vibration*, Vol. 2015, Article ID 265321, 15 pages, 2015.

[5] M. Y. Wang, Z. Liu, Y. Y. Qiu, and C. C. Shi. Study on the similarity laws for local damage effects in a concrete target under the impact of projectiles. *Shock and Vibration*, Vol. 2015, Article ID 585230, 16 pages, 2015.

[6] J. Wang, H. Wang, K. Jiang, and S. Jiang. Analysis of the aerodynamic impact of the dispenser stance for the separation of cluster munitions. *Engineering Applications of Computational Fluid Mechanics*, Vol. 10, No. 1, pp. 427-440, 2016.

[7] S. Jiang, F. Tian, S. Sun, and W. Liang. Integrated guidance and control of guided projectile with multiple constraints based on fuzzy adaptive and dynamic surface. *Defence Technology*, Vol. 16, No. 6, pp. 1130-1141, 2020.

[8] W. Ding and J. Wu. An airport knowledge-based method for accurate change analysis of airport runways in VHR remote sensing images. *Remote Sensing*, Vol. 12, No. 19, p. 3163, 2020.

[9] M. D. Shields. Adaptive Monte Carlo analysis for strongly nonlinear stochastic systems. *Reliability Engineering System Safety*, Vol. 175, pp. 207-224, 2018.

[10] C. Zhai and X. Chen. Probability damage calculation of building targets under the missile warhead strike. *Reliability Engineering System Safety*, Vol. 202, 2020.

[11] R. Singh Dhari and A. Sekar. Split warhead simultaneous impact. *Defence Technology*, Vol. 13, No. 6, pp. 434-438, 2017.

[12] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino. Matrix completion for weakly-supervised multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 1, pp. 121-135, 2015.

[13] O. Aytekin, U. Zongur, and U. Halici. Texture-based airport runway detection. *IEEE Geoscience and Remote Sensing Letters*, Vol. 10, No. 3, pp. 471-475, 2013.

[14] W. Kim, H. Yoon, and G. Lee. A new calibration metric that considers statistical correlation, marginal probability and correlation residuals. *Reliability Engineering System Safety*, Vol. 195, 2020.

[15] P. Lv, G. H. Wei, and W. Cui. Vector motion parameter estimation for an approaching missile based on an extended-target model. *IEEE Transactions on Antennas and Propagation*, Vol. 66, No. 10, pp. 5464-5474, 2018.