Interceptor Models: Simulating Air & Missile Defense Scenarios

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Joshua Smith
Spring, 2022.

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature ____ Date 5/3/2022
Joshua Smith

Approved ____ Date __________
Daniel Graham, Department of Computer Science

Interceptor Models: Simulating Air & Missile Defense Scenarios

CS 4991 Capstone Report, 2022

Joshua Smith
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
Js5mv@virginia.edu

Abstract
EIMS, a software simulation package used in simulating global air and missile defense scenarios, needed to upgrade their interceptor models to realistically simulate how effective their defensive infrastructure is against potential threats. The interceptor models were upgraded to a modular architecture with more realistic flight physics, projection, triangulation and armament destruction. This was done by creating an object-oriented collection of classes for interceptors modeled more closely to real-life complex systems. The end result led to faster and more effective interceptor performance in defense scenarios with much more data for potential analysis. Looking forward, the interceptor models can continue to be expanded on with more complex trajectory projections and different types of interceptor models or techniques.

1 Introduction

As technology naturally progresses worldwide, an unfortunate byproduct of our progression allows for the development of instruments and machines that grow increasingly adept at neutralizing the life technology seeks to enhance. One of the most applied, and perhaps most threatening to civilians, are missile technologies, which have seen all sorts of deadly improvements in intercontinental, hypersonic, and even nuclear variations. This threat begs the question: how can one ensure their safety against this increasingly deadly threat? My project seeks to serve as an answer to this question. I introduce a toolkit of instruments: aircraft, surface and armament interceptors, to be used in simulations against a variety of enemy missile threats. This simulation toolkit allows for customers, generally national or regional militaries, to optimize their missile defense architecture to ensure that any inbound threats will have the highest probability of being eliminated before reaching their target. Through this optimization, my project hopes to enable nations and regions to minimize the threat to their assets and citizens.

2 Background

It is first important to understand the real life scenarios being emulated. A missile defense scenario, at least the one being portrayed in EIMS, involves a specified geographic area that can face enemy (red-team) threats from a variable number of directions. These directions depend on the region, but are fairly intuitive. For example, if the region we were simulating was Florida, then we would expect missile threats to come from Cuba. EIMS will then simulate these threats by creating inbound missiles from Cuba to Florida. It is then incumbent on the defending force (blue-team), Florida, to sense the missiles are inbound and triangulate their position and trajectory. Once the red-team missiles have been detected, the blue-team will send out interceptors that seek out the red-team missiles and attempt to eliminate them. It is these interceptors that my project focuses on, so their role of eliminating an inbound red-team missile threat based on blue-team sensor detection information is what's most important to understand to realize the applicability of my project.

3 Related Works

For further understanding of how real-life missile defense scenarios work, one can consult an explanation of how Israel's Iron Dome missile defense system works [1]. If one was curious about how countries optimize their defense architecture given various constraints, they can read the Congressional Budget Office's paper on national cruise missile defense [3]. And if one sought a deeper understanding of how (cruise) missiles are launched, delivered and utilized, they can consult the MDAA's cruise missile reading [2].

4 Process Design

Now, EIMS, and the contributions made by my project, will be further explained.

4.1 EIMS

EIMS is a simulation toolkit for a variety of missile defense scenarios developed by a team within the MITRE corporation which I interned at over the summer of 2021. For my project, I worked specifically on the aircraft and cruise missile defense module present within EIMS. This module simulates incoming red-team cruise missile threats against the blue-team's defense infrastructure, as described in the background section. This simulation can be intuitively split into three parts: the threat (incoming missiles), the defensive sensor architecture that detects incoming threats, and the interceptors that seek out and eliminate the threat. While my project focused only on the interceptors, it still relied heavily on the implementations of the threats and the sensor architecture, so these will be explained first.

The threats, or red-team missiles, within EIMS are randomly generated to start within some pre-defined geographical area. Once their starting location is determined, a target (defending city) is selected for the missile to move towards. They will then have a generated trajectory with random behavior towards this target. This generated trajectory is only used for calculations within the simulation; the defending blue-team will never know any missile's true location. The goal of these generated threats is to ensure that enough possibilities are generated to ensure every possible angle of attack is covered by the simulation.

The blue-team sensor architecture within teams is a pre-defined set of various sensor arrays stationed at different geographic points within the defending area. They will constantly scan and detect for enemy threats to try and relay the most accurate information possible for threat interception and elimination. The goal of the sensor architecture is to be able to try out different combinations and positions of sensors to see which ones are the most effective at preventing red-team threats successfully hitting their targets.

Lastly, there is the interceptor architecture. The interceptors receive sensor and detection data from the sensor architecture. Using this data, they then try and predict where the missile threats are heading to intercept and eliminate them before they reach their target. The original interceptor implementation in EIMS was fairly barebones and simple, only having functionality for aircraft interceptors that would just constantly move towards any detected threats and try to eliminate them. For my project, I completely overhauled and replaced the original implementation with my own interceptor models. These models expanded on the original by adding multiple different types of interceptors, modeling more complex behaviors, and providing a much better coding framework for future use.

4.2 Aircraft Interceptors

The first model I implemented within my project was essentially the functionality of the original interceptor code within EIMS. With object-oriented principles in mind, I designed a modular framework for the new interceptor code that would allow for different parts of EIMS to easily interact with the interceptor code and allow for extensibility of the interceptor functionality to more than just one model. This would also allow for models with different parameters (e.g. different types of aircraft with unique speeds, flight height) to be imported into EIMS so different customers can quickly utilize their unique aircraft.

The aircraft interceptors would first receive notification of a detection by the sensor architecture. Then, using the sensor's best detection information, the air interceptors would use the threat missiles current position and velocity to come up with a projection of the missile's future locations. With this projection, the interceptor would use its own parameters, such as its flight speed, to calculate the earliest possible point along the missiles projected path at which it could intercept, and immediately fly out to reach this point. After take-off, the interceptor would receive real-time updates from the sensor architecture to continually update the point of intercept with the inbound missile. Eventually, the interceptor will either get close enough to the missile to be able to launch an armament and attempt to eliminate it, or never have enough information to intercept the missile, leaving the missile to successfully hit its target.

4.3 Surface Interceptors

After successful implementation of the aircraft interceptors within my project, I utilized the object-oriented nature of the interceptor code to extend its functionality to another interceptor model, a surface-based interceptor model. Unlike an aircraft interceptor, the surface interceptor would be completely stationary, and instead attempt to eliminate an incoming missile via launching its armament at the optimal timing and trajectory. In order to do this, the surface interceptor would use the same threat projection from the sensor data as the aircraft interceptor. However, the surface interceptor would instead have a 'cone of influence' that signifies all the possible locations that its armament can reach. Using the inbound missiles projection, the surface interceptor would check if the projected path ever reaches its 'cone of influence', and if so, calculate the exact timing its armament would have to be launched in order to eliminate the inbound missile. Like the aircraft interceptor, this timing would continually be updated based off real-time sensor data.

4.4 Armament Interceptors

The last model implemented, which would be used by both the aircraft and surface interceptor, was the interceptor armament model. These armaments would only be launched once the inbound missile threat was calculated to be in range by the aircraft and surface interceptor. Once launched, similar to the other models, the armament would use the projected path of the inbound missile and approach the point at which it would intercept. If the armament successfully intercepts an inbound missile, colliding, a probability of kill calculation would then be used to determine whether or not the threat was eliminated, taking into account things like collision vectors, velocity, etc.

5 Results

The result of my project implementation was entirely successful. The key achievement and goal of my project was that I created an entirely new interceptor functionality within EIMS that was object-oriented for modularity while still performing all of the original functionality. On top of this, I expanded on the original functionality with more complex projection and interception behavior, which allows the simulations to more closely resemble real-life scenarios. I was then able to add two entirely new interceptor models, the surface and armament interceptors, which allow for various new types of simulations to be run. On top of the new functionality, my project greatly improved the simulation runtime, reported more complex data for analysis, and gave the framework for easier future extensibility.

6 Conclusions

The impacts of my project were easily seen by the rest of the EIMS development team. The objected-oriented interface of the new interceptor models allowed for easy integration into the EIMS codebase, and provide a modular, black-box framework which gives customers customizability and developers efficient ways to expand or work with the functionality. The added and improved upon models gave a variety of new features which allow EIMS simulations to be customized into a wider range of simulation and scenario types. Ultimately, all the changes allowed EIMS to model more complex data and behavior for customers to make more informed decisions and optimize their missile defense infrastructure better than ever before with EIMS.

7 Future Work

Thanks to the object-oriented framework the new interceptor models were implemented in, any future development is a simple as changing or tweaking a single method of functionality. In terms of improvements that can be made to the existing models, more complex projections of threats past just straight-line projections can be implemented and experimented with, and similarly so for the interception logic. In terms of entirely new models that can be explored, more advanced missile technologies such as ballistics, hypersonics or orbitals could be implemented for and against the interception models.

8 UVA Evaluation

Overall, I think my course of study prepared me decently well for this project and internship experience. Due to my quantitative background from my math and physics classes, I was easily able to come up with formulations for things like trajectory projection and interceptions of two separate trajectories. Similarly, thanks to my CS fundamentals from classes like CS 2110, CS 2150, I was able to produce readable, extendable code that was also performant in a relatively performance-dependent codebase. If I were to point out a weakness in my education, it would to be my inexperience with designing systems and general software development principles. While I did eventually learn these in CS 3240, I wish I had that knowledge before my project so I would have been more efficient in the initial design phases of the project.

References

[1] Donovan, J., 2021. How Israel's Iron Dome Defense System Works. HowStuffWorks. Available at: https://science.howstuffworks.com/iron-dome-news.htm

[2] Missiledefenseadvocacy.org. n.d. Cruise Missile Basics – Missile Defense Advocacy Alliance. Available at: https://missiledefenseadvocacy.org/missile-

threat-and-proliferation/missile-basics/cruise-missile-basics

[3] Swagel, P., 2021. National Cruise Missile Defense: Issues and Alternatives. Congressional Budget Office. Available at: https://www.cbo.gov/publication