Membrane Technology
ISSN (online): 1873-4049

Computer Simulation Framework for Naval Artillery: An Integrable Solution for Naval Simulation

Cao Xuan Canh[1,*,✉], **Nguyen Danh Nam**[1], **Nguyen Hai Duong**[1], **Nguyen Hai Anh**[1]

[1]*Viettel High Technology Industries Corporation, Hanoi, Vietnam*
[*,✉] *canhcx@viettel.com.vn (PhD)*

Abstract:
This paper presents a novel simulation framework for naval gun systems. The framework is designed with a modular architecture comprising four main components: (1) Scenario Unit: Environment Conditions & Computer-Generated Force, (2) Sensors Unit: detection and tracking, (3) Computing Unit: focused on ballistics and trajectory calculations, interception, and stability control, and (4) I/O Unit: Data Interface for User Interface and Experience. The framework establishes a comprehensive simulation system for diverse naval gun types, encompassing multi-target tracking and engagement capabilities. It is customizable for anti-surface and anti-air warfare based on predictive firing computation methods and control techniques for gun systems and guided projectiles. Furthermore, the framework provides standardized interface that allow integration with other naval simulation systems. The primary objective is to create a flexible, reusable, and extensible platform for various anti-air and anti-surface gun simulation systems. This framework supports development, evaluation, and training in naval ship simulators while facilitating easy integration with hardware platforms such as Degrees of Freedom (DOF) for gun simulation. This research improves the accuracy and operational efficiency of modern naval gun simulation systems.
Keywords: Naval artillery, Simulation framework, Simulation-based optimization, Modular architecture, Intercept firing, Guided projectiles, Environmental compensation, Radar simulation.

1 INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction

Naval artillery has a significant contribution to the offensive and defensive capacities of a modern warship. The rise in threat complexity has increased the need for more advanced simulation tools for training, evaluation, and tactical try-on. Simulation platforms offer a cost-effective and safe environment for these purposes, enabling extensive testing and training without the risks and expenses associated with live-fire exercises.

The development of a complete computer simulation framework for naval artillery systems leads to significant challenges. This entails the integration of complex environmental data into naval simulation systems, the emulation of sensor behavior for detecting, visualizing, and tracking targets, ballistic calculations, gun control mechanisms, and transparent interaction between human operators and machine systems. Current solutions tend to concentrate on narrow aspects of naval artillery simulations, falling short of the integration and flexibility required for a complete simulation experience and adaptability to wider applications.

This research intends to design a flexible, consolidated, and extendable simulation environment for the gun systems of naval ship. The specific objectives include: formulating a distinct modular framework for naval gun simulations; developing a modular architecture with four core components which are capable of reusable and extensible; creating a flexible framework that supports different scenarios and applications for various naval gun types, by adapt technical parameters; and providing a deployable framework with robust connectivity and seamless integration between software (simulating naval guns for existing ship simulators) and hardware (simulating gun control with hardware platforms such as degrees-of-freedom models) through standardized interfaces.

The paper is organized as follows: the Abstract provides a summary of the study. The Introduction and Literature Review define the problem's scope and aims, as well as the previous work done in the field. The Framework Architecture section describes the architecture and main components of the proposed system. The Methodology explains the research approach for our framework. The Implementation section provides a use case and discusses the result of the use. Finally, the Discussion, Conclusion, and Future Works section outlines the findings, concludes the study and suggests directions for future research.

1.2 Literature Review

From the late 19th century through the early 20th century, game designers attempted to portray naval wargames (see [21]; [15]). Notably, Fred Jane (between 1898 and 1918) and Fletcher Pratt (1930s) developed two prominent methods that aimed to replicate naval gun aiming and damage assessment in naval battles using scale models of warships. Although these games had to use a large play area and high amounts of time for preparation, measurements, and analysis of results, they established the groundwork for contemporary naval wargames and training simulators.

Subsequently, in order to improve the simulation of artillery training, [32] presented some novel means of training, which included Dotter (a method invented by Percy Scott for British Navy gunnery training) that used a moving target with a bull’s-eye and a pencil that marked errors during firing. Then, similarly, the Morris tube, employed by American officers, uses a smaller caliber barrel set into a larger gun barrel. While these methods improved targeting for long-range and moving targets beyond the scope of traditional optical sights, some certain inaccuracies occurred.

The simulation of naval weapon systems, particularly naval gun systems, has been an active research area for decades. Early efforts by [39] developed SEAROADS (Simulation, Evaluation, Analysis, and Research on Air Defense Systems) to evaluate air defense systems on naval destroyers. This model allows the integrating multiple weapon systems, simulating sensors, and providing insights into system performance. Earlier, [27] introduced the Navy's Fleet Air Defense Simulation (FADS) to analyze naval fleet air defense, modeling radars, missiles, aircraft, and interactions among them in the battlefield of engagements between maritime defense forces and attacks from enemy air forces. More recently, studies such as [12], [13], and [33] describing simulation systems frameworks like the AEGIS CIC Air Defense Simulation, Naval Combat Systems Simulator (NCSS), ADL (Air Defense Laboratory), and developing AEGIS Cruise Air-Defense Simulation (ADC Simulation), which simulate naval weapons, gun control, and radar systems, in addition to mentioning the Bayesian Simulation expert model. [44] also proposed a multi-objective optimization model for weapon-target assignments (WPA) problems by using genetic algorithms (GA). [26] developed simulation models to analyze air defense for warships under different operating conditions. These studies have highlighted the value of simulations in tactical analysis and system performance evaluation.

Despite these successes, most of these models are limited to air defense scenarios, focusing on combat operations, target detection and evaluation, and air defense command missions in combat information centers (CIC). They have not only focused on simulation models of ship artillery systems applied to warships. Furthermore, the simulation has simplified related factors such as weather conditions and terrain.

In the domain of combat simulations, [40] developed a visual simulation system for naval gunnery training. The system aims to improve the anti-ship combat command capabilities of naval artillery commanders. This study immersed models such as relative target motion, environmental effects, and gunnery dynamics by using firing parameters and viewing angle parameters. Similarly, [22] proposed a simulation model for testing naval gun fire control systems (NGFCS) in the sea surface combat system. The research proposed a system integrating Combat System Data Bus (CSDB) to transmit data of the combat system such as Sensor Log (real log data excluding environmental conditions), BCU Log of the actual fire control block BCU (Ballistic Computing Unit), and BCU Log of the simulated BCU. Meanwhile, the simulator BCU (using the same source code as the real system) modeled target motion, target trajectory, and sensor characteristics to generate virtual simulation scenarios. The simulation results were compared with actual firing results to demonstrate the accuracy and verify the actual system development process. This They are same paragraph is a comprehensive testing process for naval gunfire control in different stages from design to shore testing, and sea trials, using both real data and simulated data. However, the modeling system is designed specifically for testing purposes, not for training simulation, and it was limited to specific ship and gun types. Then, [23] advanced this further by developing an Adaptive Naval Gun Fire Simulator on a Naval Combat System, that is capable of recording data, analyzing recorded data, and evaluating gun-firing performance. It includes core components such as DataRecorder, LogRecorder, BCUAnalysis, FIAS, and FireBalls, enabling comparisons between simulations and real data. However, the research is still for testing, comparisons, and basic training under favorable conditions, not fully simulating all functionalities of a complete naval gun system (which includes target tracking, interception, and pursuit firing).

Projectile modeling has also received significant attention, many studies have focused on developing accurate ballistic models. NATO's [31], a NATO standardization introduced ballistic modeling through Modified Point Mass trajectory models for spin-stabilized projectiles and Five Degrees of Freedom for fin-stabilized guided projectiles. However, these models have not counted the effects of weather conditions and ship movements. Then, Baranowski ([4], [5], [6], [7]), [8], [45], and [37] proposed studies on projectile trajectory stability analysis and the influence of environmental factors. [42] also developed a method for simulating, calculating, and analyzing the dynamic characteristics of naval artillery systems based on the Adams model. [25] also simulates field artillery, although taking into account factors such as terrain, weather, and corrections based on user experience, not for specific factors of naval artillery such as ship motion, ship oscillation, etc. Although recent studies have highlighted the effects of the environment on the trajectory, however, modern guided projectiles pose new challenges in simulation due to internal factors that lead to trajectory modifications, which remain underexplored.

In addition to creating the foundation for simulating individual components of naval artillery systems as in previous studies, recent researches have focused on developing integrated simulation platforms. [30] proposed NATO Virtual Ships standards, using High-Level Architecture (HLA) to develop distributed simulations by using the High-Level Architecture (HLA). [20] developed an air defense simulation framework integrating radar detection systems and synthetic environments to reproduce the influence of environmental conditions on radar signals. This framework is designed and developed to be compatible with HLA (High-Level Architecture) distributed architecture to allow synchronization and data exchange between simulation components at the engagement level, 3 federates: simulation, radar detection, and environment. However, this research still has some limitations such as: only focusing on the radar system to detect airborne targets, and lacking ballistic and firing interception simulations for naval guns.

Despite significant contributions and advances, these solutions exhibit existing limitations as follows: focusing on separated subsystems or specific scenarios, lacking flexibility and comprehensiveness, and limited interactivity and integration with other simulation systems. The incomplete adoption of standards such as HLA and DIS within the field of naval artillery simulation leads to further compounding of the problem, that hinders the development of distributed and federated simulation environments.

This study seeks to resolve these gaps by integrating environmental models, sensors, computation, and programming interfaces into a unified framework. This research also aims to provide a flexible and robust solution for a wide range of naval artillery simulations by proposing a comprehensive, modular, and interoperable simulation framework for artillery simulation on warship simulators.

2 NAVAL GUN SIMULATION FRAMEWORK ARCHITECTURE

The computer simulation framework for naval artillery is designed to provide an integrated solution comprehensively for gun simulation systems into warship training simulation platforms and other naval artillery training systems. The primary objective of this framework is to offer a flexible environment that allows customization of various gun simulations, therefore satisfy many combat scenario training requirements.

The modular architecture enables the platform to acquire, process, compute, and provide customizable sensor and radar information from environmental data for the ship simulator. This optimization enhances training functionality and improves combat effectiveness, mirroring real-world systems. Simultaneously, it provides an extensible platform for research, performance evaluation in design, development, and deployment of naval gun systems.

The proposed system comprises four fundamental component units, Fig. 1:

- Scenario Unit: This unit interacts implicitly with the ship simulation system and database sources. It is responsible for collecting, synthesizing, organizing, and providing data on the host ship, environment, and targets for the computational units (Sensor Unit and Computing Unit).
- Sensor Unit: This unit processes data to generate inputs for the computational blocks and the output block, serving to display processed sensor information (for Computing Unit and I/O Unit).
- Computing Unit: This important unit is responsible perform the basic functions for artillery, by modelling the gun operation and its projectile.
- I/O Unit: This unit interfaces with users and the ship simulation system directly via distributed data distribution standards, receiving control inputs and providing simulation results produced by the computational units.

This modular architecture allows for flexibility in modifying or upgrading individual components. The system is designed with high integration capability, utilizing standard protocols to ensure compatibility with various simulation systems. These features enable the framework to be easily integrated into existing warship simulations while facilitating future expansions and upgrades.

2.1 Scenario Unit: Environment and CGF

The Scenario Unit is responsible for collecting real-time data from the simulation system (federation) via distributed data distribution standard (HLA, DIS), as well as implicit data retrieval from the database. The collected information includes:

- Ship motion status (velocity, roll and pitch angles, and rates under environmental effects).

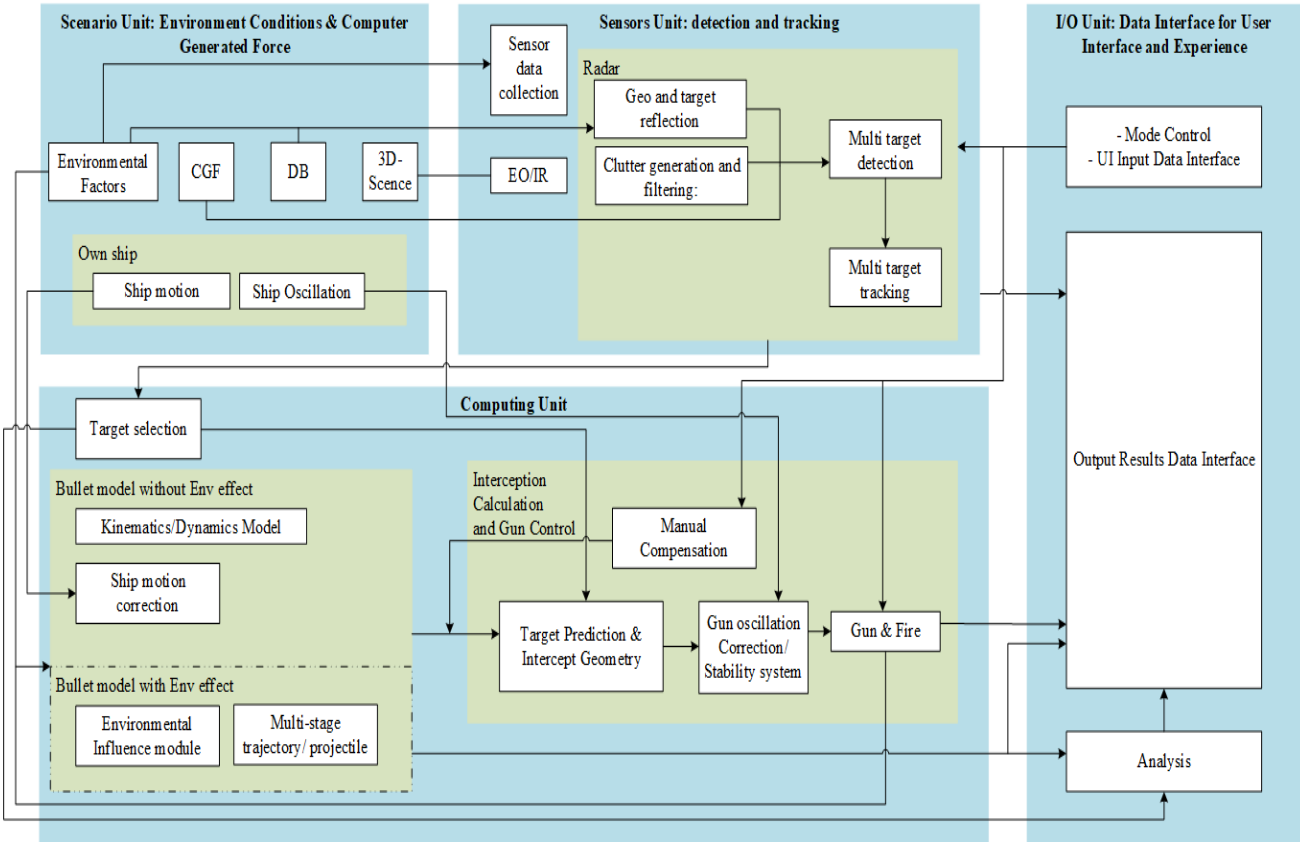


Figure 1: Framework system's component architecture and data relationships.

- Simulated environmental conditions, including atmospheric (temperature, precipitation, cloud cover) and maritime (surrounding the ship and in the extended space from ship to target).
- CGF information, including allied forces, opposing forces, and electromagnetic interference or electronic countermeasures (if applicable).
- Scene information: 3D scene and corresponding terrain elevation map.
- User input for environmental condition settings and target information for specific training scenarios.

This unit transmits information to the sensor unit for calculation, recognition, and display. For information not provided by the simulation system, this module offers fields for manual input to ensure comprehensive data for processing units and to satisfy user disease.

To achieve these objectives, the Scenario unit comprises several sub-modules:

- Environment Factors: collects and synthesizes environmental information
- 3D-Scene: gathers and compiles operational scene information
- DB (Database): provides terrain elevation map data
- CGF: collects and synthesizes information about virtual objects and their interactions
- Own ship: collects and provides information about the host ship's operational status

2.2 Scenario Unit: Environment and CGF

a) Sensor Modules for Data Collection and pre-Processing

Naval gun systems require specialized observation and control systems, thereby the Sensor Unit incorporates two key modules:

- Sensor data collection Module: collect or simulate environmental conditions (e.g., wind, temperature, humidity) near the ship by using aggregated data which get from the Scenario Unit.
- EO/IR Image Generation Module: generates electro-optical/infrared (EO/IR) imagery based on 3D scene data, observation direction, and sensor specifications, in order to ensure realistic visual outputs.

b) Radar Block

Radar systems are very important for target detection, tracking, and observation. The radar unit includes:

- Geo and target reflection Module: simulate the terrain and target reflections.
- Clutter generation and filtering Module: generate environmental noise (e.g., rain and sea clutter), and implement radar functionalities such as gain, rain, and sea clutter filtering control.

The results from these modules are combined with the characteristics and motion states of CGFs to calculate, classify, and detect targets through the Multi-target detection module. Finally, based on this information, decisions are made regarding tracking, trajectory estimation, and position prediction of targets in the Multi-target tracking module.

2.3 Computing Unit

a) Projectile Modelling Block

This block uses a projectile model for simulating the projectile trajectory, determines and evaluates the accuracy of target hits. The projectile model is described under various environmental influences and is divided into two sub-models for specific use cases:

- Bullet Model without Environmental Effects: this is a theoretical ballistic calculation, which is used to determine firing angles and interception points, disregarding environmental influences. It includes:

- Kinematics/Dynamics Module: models projectile motion, such as parabolic trajectories.
- Ship motion (translational motion) correction module: adjusts projectile trajectory calculation for ship movement. Accounts for the ship's movement during firing time, considering the influence of the ship moving on the projectile trajectory.

- Bullet model with environmental effects: to generate the actual projectile trajectory for framework output. This dynamic model incorporates environmental factors (wind, air density, etc.) and guidance (fin control) effects on the projectile's real trajectory, specifically including two sub-modules:

- Environmental influence module: calculates the impact of environmental parameters on the projectile during flight (distinct from ship-measured parameters).
- Multi-stage trajectory: computes trajectory changes during flight phases (for guided projectiles/ fin control).

b) Interception Calculation and Gun Control Block

This unit ensures accurate firing and includes the following modules:

- Manual Compensation Module (or Environmental influence correction): allows user adjustments for environmental effects before firing angle calculations. Receives input from the I/O unit to adjust firing angle calculations based on predicted environmental effects. The amount of compensation simulates environmental influences, if inaccurate compensation leads to the wrong calculation for interception point, this depends on operator experience.
- Target Prediction & Intercept Geometry (IG): calculates optimal interception point and barrel elevation angle based on predicted target trajectory and theoretical ballistic model, using interception equation solving methods and objective function optimization.

- Gun oscillation correction/Stability system: to reduce ship motion influence (rotational motion/ oscillation) by adjusting the barrel angles or stabilizing the gun mount. In cases of active gun mount stabilization, the barrel maintains balance regardless of ship oscillation.
- Gun & Fire module: controls barrel speed and angles, and manages the loading system. Gun barrel and mount angles are updated for display output (Output Results Data Interface) and used for actual trajectory calculations (Computing unit).

2.4 I/O Unit: Data Interface for User Interface and Experience

The I/O Data Unit Interface for User Interface and Experience is a critical component that supports interaction between users and the naval ship simulation interface, and transmitting simulation results. Unlike data collection implicitly by the Scenario Unit, the I/O Unit explicitly handles:

- Mode Control and UI Input Data Interface Module: manages control signals for gun and sensor configurations across various modes.
- Output Results Data Interface Module: receives target tracking data, EO/IR imagery, sensor information, gun angles, and real projectile trajectories. These outputs conform to standards such as HLA or DIS for integration into tactical displays and ship simulators.
- Analysis Module: Evaluates firing results to determine accuracy and performance, providing insights for scoring and feedback by using fuzzy algorithm, or artificial intelligence techniques. Outputs are transmitted via the Output Results Data Interface for integration with ship simulators.

2.5 System Integration

a) Framework Components Integration

The Data Distribution Service (DDS) model is employed for communication among units, entities, and modules within the naval artillery simulation framework, see Fig. 2. DDS offers several advantages, including high performance, simplicity, user-friendliness, and excellent scalability without being constrained by rigid rules. Specifically, it supports QoS (Quality of Service) for ensuring information is delivered to other components while eliminating the need for brokers (central servers) and relying only on DDS Middleware on the hardware.

b) Integration with Naval and Other Simulation Systems

A prerequisite for integrating this framework into naval ship simulators and other simulation systems is compatibility. To ensure seamless interaction across various systems, the framework needs a standardized protocol, which must support real-time data processing with low latency to maintain simulation accuracy, Fig. 2.

Key integration requirements include:

- Scalability: the framework must allow for the addition of new modules in the future.
- Time Synchronization: strict time synchronization is crucial in distributed simulations to ensure coherence.

Widely used standards such as High-Level Architecture (HLA) or Distributed Interactive Simulation (DIS) are well-suited for military simulations and war-gaming applications. These standards are excellent in managing time synchronization through features such as time-stepped execution, event-driven coordination, federation-wide synchronization, Time Stamp Order (TSO), and time regulation or constraint mechanisms ([17]).

To enhance compatibility and adaptability with various systems, an intermediate layer capable of translating between different protocols such as HLA and DIS can be implemented. This middleware layer enables interoperability across different simulations while maintaining flexibility and scalability.

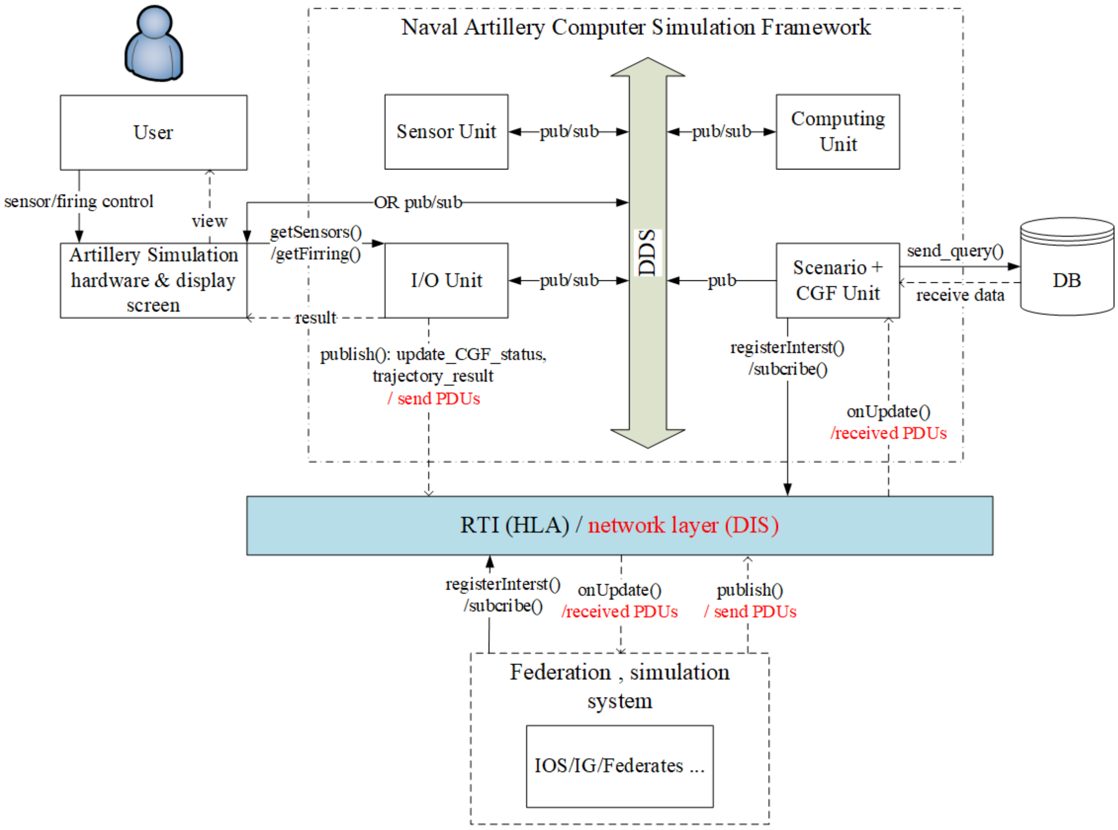


Figure 2: Framework system's integration

3 METHODOLOGY

This framework describes a modern naval artillery simulation system, which considers environmental conditions, sensors, signal processing, target tracking, ballistic calculations, and fire control. It also counted system integration to form a comprehensive platform for the development and evaluation of naval artillery systems. To achieve these objectives, four components of the framework are developed and implemented through specific methodologies detailed below.

3.1 Scenario Unit: Environment and Computer-Generated Forces (CGF)

Within the simulation workflow, the Scenario Unit actively retrieves environmental simulation data by subscribing to (register the interested) object and attribute classes defined in the Federation Object Model (FOM) when the naval simulation platform conforms to the High-Level Architecture (HLA) standard. Using Runtime Infrastructure (RTI), the unit employs callback functions to receive updates whenever new data is distributed. These updates include 3D environmental parameters, meteorological conditions, ship states, and target object attributes.

If the simulation platform instead utilizes the Distributed Interactive Simulation (DIS) standard, which employs a broadcast mechanism, the Scenario Unit directly receives Protocol Data Units (PDUs) without the need for explicit subscription as in HLA. However, additional processing and filtering are required to manage the increased volume of unfiltered data received via broadcast. After that, the data is categorized into specific groups to meet the requirements of each unit or module within the system.

For information that is not provided by the ship simulator, the Scenario Unit offers fields for either manual input. These fields ensure the completeness of information for computational units and allow users to define configurations directly. Manual entered data will overwrite corresponding data received from the simulation system. It provides flexibility in scenarios for use.

3.2 Sensor Unit: Detecting and Tracking

For the simulation of imaging functions for electro-optical/infrared (EO/IR) systems, the framework integrates Image Generation (IG) technologies to create optical and infrared images by processing the fields of view on a 3D scene, influenced by the observation direction and sensor specifications.

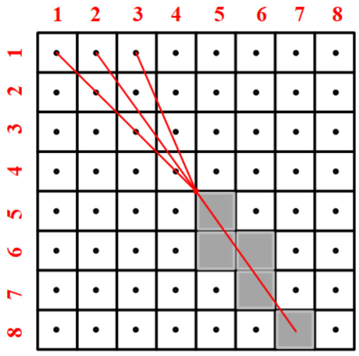


Figure 3: Circular search on Matrix Pixel DEM and a description of Bresenham Algorithm.

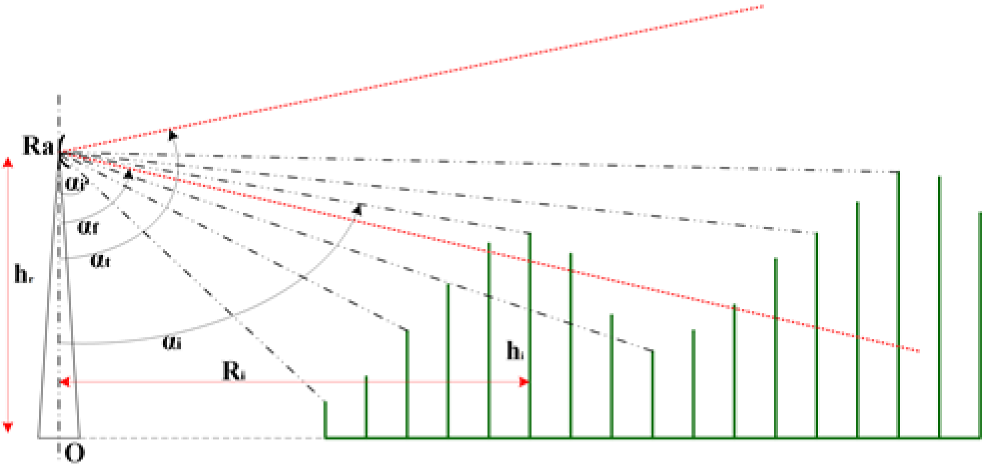


Figure 4: Visible detection by using angle comparison techniques.

Naval artillery radar typically operates in standby, observation, target acquisition, and target tracking modes, necessitating technologies to generate images of reflected wave intensities on the Plan Position Indicator (PPI) screen from terrain, coastlines, targets, and environmental noise factors such as sea and rain clutter.

In practical radar systems, during the scanning process, the radar's view is updated by identifying objects along the line of propagation of electromagnetic waves at a given moment. To simulate this scanning effect and display environmental updates, a circular search centered on the radar's field of view is employed, analogous to the scanline approach ([43]). This method evaluates pixels lying along a beam projection. The Bresenham line-drawing algorithm ([16]; [11]) is applied to the circular search on a matrix of pixels, approximating a straight line between two endpoints with pixels it traverses. A square grid and scanning beams are used to analyze the terrain's Digital Elevation Model (DEM). An equidistant method calculates scan lines to prevent sand-hole effects, ensuring unequal angles between adjacent scan line pairs, (see Fig. 3). For each scan line, from the found pixels, elevation comparison methods ([35]) or angle comparison techniques ([43]) can be used to determine visible and obscured pixels.

Algorithm 1 Detect Visible Scan Cells in Array $A[n]$

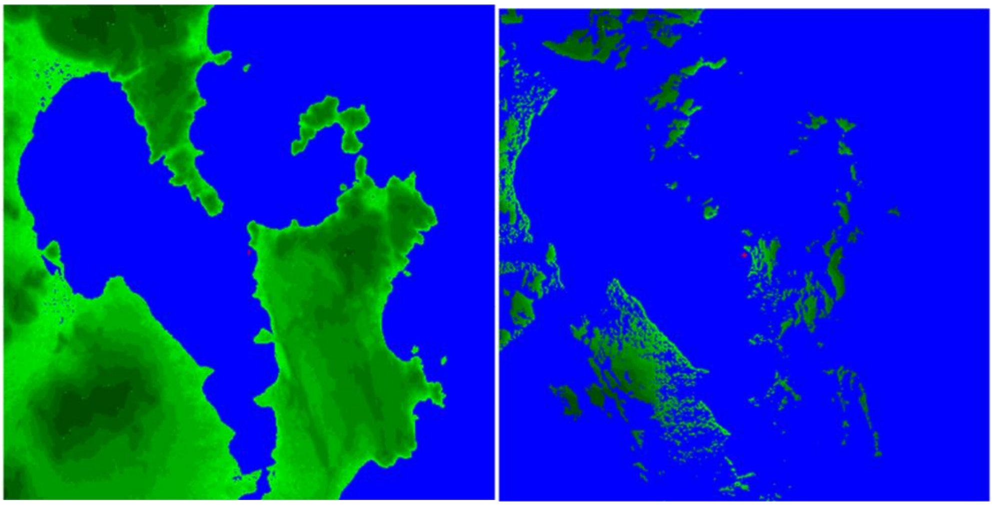
Input: An array $A[n]$, initialized with all values set to $false$
Initialization:
Set $\alpha_{max} = \alpha_f$ ➢ Initialize maximum angle
$i = 1$ ➢ Start index
while $i \leq n$ **do**
Calculate $\alpha_i = f(R, h_r, h_i)$ ➢ Compute current angle
if $\alpha_f \leq \alpha_i \leq \alpha_t$
if $\alpha_i \geq \alpha_{max}$
$\alpha_{max} \leftarrow \alpha_i$ ➢ Update maximum angle
$A[i] \leftarrow true$ ➢ Mark visible cell
end if
end if
$\boldsymbol{i} \leftarrow \boldsymbol{i} + \mathbf{1}$ ➢ Move to the next index
end while

15: **Output** The resulting array $A[n]$, where elements corresponding to visible scan points are marked $true$.

Naval artillery radars typically operate under scanning modes constrained by vertical beamwidths, resulting in varying radar displays. An algorithm for processing radar scan data can be implemented as Algorithm 1, where, (α_f), (α_t) represent the lower (from) and upper (to) bounds for the vertical angle, Fig. 4. The array ($A[n]$) represents the pixels along the scan path, and (α_i) is the elevation angle at terrain vertices.

$$\alpha_i = arctan\left(\frac{R_i}{h_r - h_i}\right) \quad (1)$$

By combining the results off multiple scan lines, a visual area of terrain that can be observed by radar is obtained, Fig. 5.



(a) (b)

Figure 5: (a) Original terrain DEM map (left). (b) result by using scan lines(right).

The integration of target elevation maps into DEM allows the calculation of radar reflection. The reflected signal intensity of an object (terrain or target) illuminated by radar can be calculated as:

$$P_r = \frac{P_t G^2 \lambda^2 \sigma}{64\pi^3 R^4} \quad (2)$$

where, P_r is the received power [W], P_t is the transmitted power [W], G is antenna gain, λ is the electromagnetic wavelength [m], σ is radar cross section (RCS) of radar target [m^2], R is the radar-target distance [m].

Initial data preparation for terrain material incurs high costs. Simplified modeling approaches, such as reflection angle calculation of each scan line or lookup tables, are used. Noise models, including random distributions for rain and snow ([29]) and coherent noise such as Perlin Noise for ocean surfaces, can be applied. For intensity distributions, methods for distributed targets, such as rain and snow ([36]), are empirical sea clutter models ([19]) which calculate RCS based on factors like grazing angle (incidence angle) θ, radar wavelength λ, and sea-state (e.g., wave height h_w): $\sigma_{SEA} = \Gamma_{SEA}(\theta, \lambda, h_w)$, [$m^2$] ([41]), or statistical distribution models, such as K-distribution or Rayleigh models ([2]; [3]; [38]; [19]; [1]; [34]; [18]). Filtered signal intensities are then used for target detection and evaluation.

3.3 Computing Unit

a) Modelling Naval Artillery for Interception Tasks

Projectile Modelling

The projectile model can be implemented using a 3 Degrees of Freedom (3DOF) or 6 Degrees of Freedom (6DOF) framework ([5], [7]). It consists of the following components (Fig. 6).

Data Block:

- Projectile Specifications: Includes geometric properties (e.g., diameter), mass (affecting gravity), moments of inertia, and initial velocity characteristics.
- Initial Conditions: States prior to firing, such as position, barrel angle, and muzzle velocity.
- Environmental Data: User-inputted environmental conditions, which reflect simulation-world values, affecting projectile trajectory.
- Aerodynamic Properties: Characteristics specific to projectile types, determined experimentally or via simulation, influencing aerodynamic forces such as drag, lift, and spin. These parameters are critical for determining the trajectory and external forces acting on the projectile during motion.

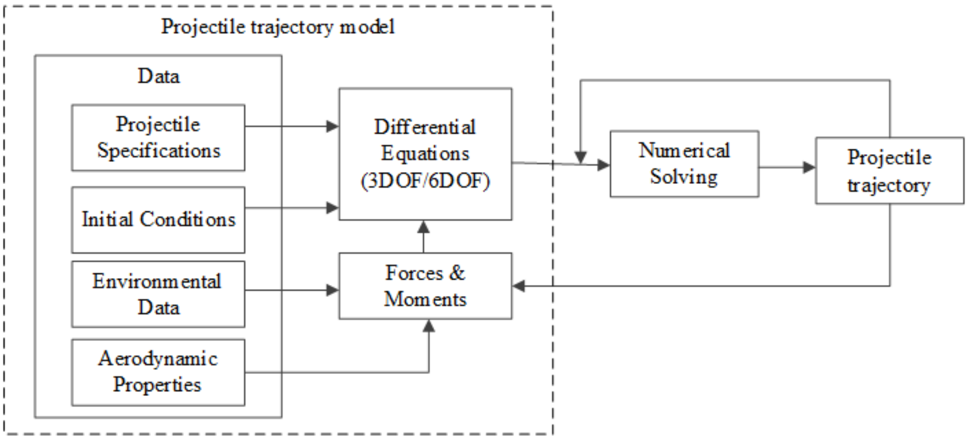


Figure 6: Projectile trajectory model.

Differential Equations Block: The initial approach employs a 3DOF kinematic model representing the projectile as a point mass. The position vector of the projectile at any time t, denoted as $\boldsymbol{p}_B(t)$, is computed as:

$$\boldsymbol{p}_B(t) = \boldsymbol{p}_{B0} + \boldsymbol{u}_{B0}t + \frac{\boldsymbol{a}t^2}{2} \quad (3)$$

where $\boldsymbol{p}_B(t)$ is the position vector after time t, $\boldsymbol{p}_{B0}$, $\boldsymbol{u}_{B0}$ are the initial position and velocity vectors of the projectile at launch, and $a = [0, 0, -g]$ is the acceleration vector due to gravity. This equation provides the theoretical trajectory under ideal conditions, assuming no environmental disturbances.

Under realistic environmental conditions, wind and other factors are incorporated to refine trajectory calculations. These conditions cause deviations from theoretical trajectories, and generates the actual simulation of trajectories. To account for these effects, the kinematic equation is extended as:

$$\boldsymbol{p}_B(t) = \boldsymbol{p}_{B0} + \boldsymbol{u}_{B0}t + \frac{\boldsymbol{a}t^2}{2} + \boldsymbol{\Delta}(\boldsymbol{\beta}, t) \quad (4)$$

where: $\boldsymbol{\Delta}(\boldsymbol{\beta}, t)$- Accumulative impact of environmental factors on the trajectory over time. It is formulated as: $\boldsymbol{\Delta}(\boldsymbol{\beta}, t) = \sum_{i=1}^{n} \frac{\beta_i}{c_i} t^{\alpha_i}$, here c_i, α_i are dependent coefficients and exponents that characterize the influence of each factor via time.$\boldsymbol{\beta} = \{\boldsymbol{\beta}_i\}_{i=1}^{n}$: A vector set representing the influence of individual environmental i on the trajectory in different directions. This approach effectively substitutes the influence of environmental forces in the dynamic model to extend the kinematics model, enabling the prediction of a realistic projectile trajectory.

The second approach, which is a 6DOF (six degrees of freedom) dynamic model. This model provides a more comprehensive representation of projectile motion by accounting for both translational and rotational dynamics under the influence of aerodynamic forces and moments. [4], [5], [6], [7] offering a detailed and robust methodology for simulating projectile behavior in complex operational environments. The equations governing the 6DOF dynamics of the projectile are expressed as:

$$\begin{cases} m\left(\frac{\delta \boldsymbol{u}}{\delta t} + \boldsymbol{\Omega}_r \times \boldsymbol{u}\right) = \boldsymbol{F} + m\boldsymbol{g} + m\boldsymbol{\Lambda} \\ \frac{\delta \boldsymbol{H}}{\delta t} + \boldsymbol{\Omega}_r \times \boldsymbol{H} = \boldsymbol{M} \end{cases} \quad (5)$$

where $\boldsymbol{\Omega}_r$is the vector of angular velocity of the system moving together with the projectile, $\boldsymbol{M} = [M_x, M_y, M_z]$ is the vector of aerodynamic moments acting on the projectile, $m$is the projectile mass, $\boldsymbol{u}$ is the velocity vector of the projectile, $\boldsymbol{g}$ is the gravitational acceleration vector, and $\boldsymbol{\Lambda}$ represents additional accelerations (e.g., Coriolis or external perturbations). $\boldsymbol{H}$ is the total angular momentum of the projectile, and $\boldsymbol{F} = [F_x, F_y, F_z]$ is the vector of aerodynamic forces acting on the projectile.

The forces and moments block acting on the projectile are computed using aerodynamic coefficients and bullet properties: $\boldsymbol{F} = \frac{1}{2} C_f \rho \boldsymbol{u}_B^2 S$, $\boldsymbol{M} = \frac{1}{2} C_m \rho \boldsymbol{u}_B^2 S d$, where ρ is the air density, $\boldsymbol{u}_B(t) = \boldsymbol{u}_{B0} + \boldsymbol{a}t$ is the relative velocity, S is the reference area of the projectile, d is the projectile reference diameter, C_f is the aerodynamic force coefficient, and C_m is the moment coefficient. Adjustments must account for the ship's velocity and inclination during firing, influencing the initial projectile states.

Numerical Solution and Simulation: The Runge-Kutta 4th order method (RK4) is employed to solve differential equations. This method is well-suited for integrating differential equations, allowing precise computation of the projectile's instantaneous states, including position, angular velocity, and orientation under the influence of forces and moments. Trajectory data from prior steps feed into subsequent calculations for continuous simulation.

Target Interception Problem

For solving target interception problem, the artillery simulation framework uses target prediction and intercept geometry process. It consists of the following components (Fig 7):

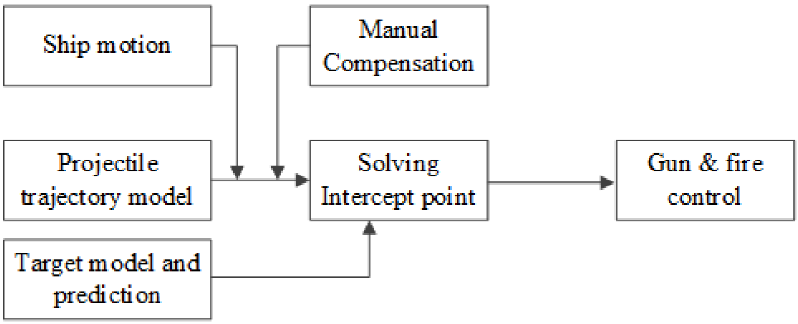


Figure 7: Target Prediction & Intercept Geometry process.

• Projectile trajectory model: In order to solving interception problem, the framework employs a trajectory computation model that excludes real environmental conditions, however it can be combined with manual compensation for environmental influences to fine-tune the projectile trajectory.
• Manual Compensation Module: A manual compensation module is incorporated, where parameters are provided by operators through the user interface (UI). These parameters predict environmental effects on the projectile (wind, air density, etc.), leading to deviations from the theoretical trajectory.
• Target model and prediction: To calculate target interception, it is essential to predict the target's motion. The target's movement model is defined with the following assumptions over the interception time interval:

$$\boldsymbol{p}_T(t) = \boldsymbol{p}_{T0} + \boldsymbol{u}_T t \qquad (6)$$

where $\boldsymbol{p}_{T0}$ represents the initial position vector of the target, and $\boldsymbol{u}_T t$ is the predicted displacement of the target after a time t from the initial position.
• Solving interception point: Depending on whether the projectile is modeled using a 3 Degrees of Freedom (3DOF) or 6 Degrees of Freedom (6DOF) approach, different methodologies can be applied to solve the target interception problem, Fig 8.

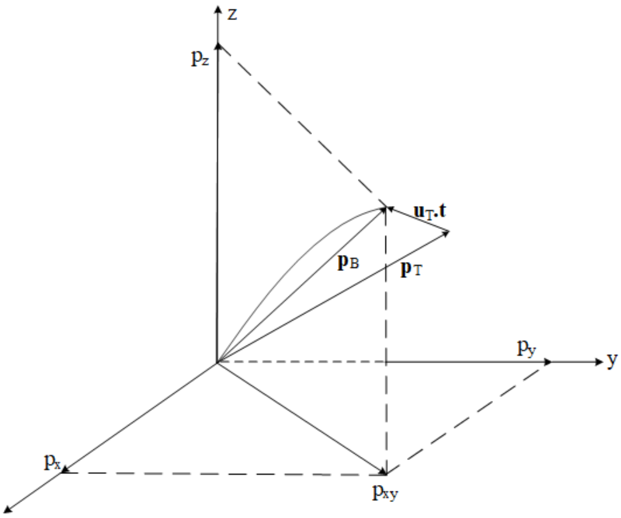


Figure 8: Interception Problem.

For a projectile modeled with 3 Degrees of Freedom (3DOF), the solution to the target interception problem involves determining the time t at which the projectile and target intersect. This is achieved through vector addition based on the interception triangle. The condition for interception is defined as the interception equation:

$$\boldsymbol{p}_B(t) = \boldsymbol{p}_T(t) \qquad (7)$$

Expanding and solving this equation iteratively with an initial assumption for time t. At each iteration, the azimuth angle for projectile guidance can be calculated as $H_{angle} = arctan\left(\frac{(Y_T - Y0_B)}{(X_T - X0_B)}\right)$. Using the substitution method, reduce the problem to a quadratic equation with the unknown being the angle of elevation T_{angle}.Solve for the angle of elevation, and compute the time t according to the formula $t_B = \frac{(X_T - X0_B)}{(\cos(T_{angle}).\cos(H_{angle}).|\mathbf{u}_{B0}|)}$. Compare t_B and the time t: $If\ |t - t_B| > \varepsilon$, update the value of t using the formula $t = t + \Delta t$, where $\Delta t = \frac{(t_B - t)}{10}$. The error margin ε and the update time-step Δt can be adjusted based on the requirements of the problem. The iteration stops when $|t - t_B| < \varepsilon$ (with ε being sufficiently small).

In addition to the substitution approach, based on the interception equation, the vector equations can be expanded into component to obtain the system of equations are:

$$\begin{cases} X0_B + u_{xB}.t = X0_T + u_{xT}.t \\ Y0_B + u_{yB}.t = Y0_T + u_{yT}.t \\ Z0_B + u_{zB}.t - 0.5.g.t^2 = Z0_T + u_{zT}.t \end{cases} \quad (8)$$

Squaring both sides, and summing them results to transform the problem into a scalar equation $f(t) = 0$, as follows

$$\begin{aligned} f(t) = {} & (X0_T - X0_B - u_{xT}.t)^2 \\ & +(Y0_T - Y0_B - u_{yT}.t)^2 \\ & +(Z0_T - 0.5.g.t^2 - Z0_B - u_{zT}.t)^2 \\ & -|\boldsymbol{u}_{B0}|^2.t^2 \end{aligned} \quad (9)$$

The Newton Raphson method is applied directly to solve for the value of t. The initial condition for t is estimated estimated as: $t_0 = \frac{Range}{(|\boldsymbol{u}_{B0}| - |\boldsymbol{u}_T|)}$.

where: u_T is the velocity of the target; $Range$ is the distance between the projectile and the target: $Range = \sqrt{(X0_T - X0_B)^2 + (Y0_T - Y0_B)^2 + (Z0_T - Z0_B)^2}$

The iterative Newton-Raphson formula is: $t_{n+1} = t_n - \frac{f(t)}{f'(t)}$. The iteration continues until $t_{n+1} - t_n \le \varepsilon$, where ε is the computational tolerance (threshold) specified by the problem's requirements. After determining the time t, the gun azimuth and elevation angles are calculated using the following formula: $\tan(H_{angle}) = \frac{Y_T - Y0_B}{X_T - X0_B}$, $\cos(T_{angle}) = \frac{Y_T - Y0_B}{sin(H_{angle}).|\boldsymbol{u}_{B0}|.t}$. The same procedure applies to the model incorporating manual compensation for the environmental effects on the projectile.

Note that, for scenarios considering the influence of environmental conditions, operators rely on lookup tables or experience-based estimations derived from these environmental parameters to determine a set of manual compensation values $\boldsymbol{\delta} = \{\boldsymbol{\delta}_i, i = 1, n\}$. These compensation values are applied to ensure the projectile reaches its target, even as its trajectory deviates due to environmental factors. Consequently, the trajectory of the projectile under predicted environmental influences can be expressed as:

$$\boldsymbol{p}_B(t) = \boldsymbol{p}_{B0} + \boldsymbol{u}_{B0}.t + \frac{\boldsymbol{a}.t^2}{2} + \Delta(\boldsymbol{\delta}, t) \quad (10)$$

here, $\Delta(\boldsymbol{\delta}, t) = \sum_{i=1}^{n} \frac{\delta_i}{c_i} t^{\alpha_i}$ represents the predicted impact of environmental conditions.

This equation serves as the basis for target prediction and firing computations, yielding the firing parameters (e.g., firing angles) for the naval artillery system through the methods discussed earlier. To ensure accurate engagement under the influence of environmental conditions, the compensation set $\boldsymbol{\delta}$ must closely approximate the actual environmental influence set $\boldsymbol{\beta}$. The selection of $\boldsymbol{\delta}$ relies entirely on the operator's evaluation of the environment, typically based on their experience. If the chosen compensation values are accurate, the calculated firing angles will produce a trajectory that successfully engages the target. Otherwise, discrepancies between the predicted and actual trajectories will result in missed shots.

In the 6DOF projectile model, it is difficult to apply substitution (analytical) or Newton-Raphson methods due to the complexity of the motion equations, which take the form of differential equations. Numerical methods are required to solve these equations. For the 6DOF model, an objective function is employed to directly compute the elevation and azimuth angles. The time t (sufficiently large) is assumed for the calculation of the projectile's and the target's trajectory, considering any firing angles (elevation and azimuth angles), obtaining the trajectory in discrete numbers. From this, the distance between the bullet trajectory and the target predicted position over time can also be computed using the following equation:

$$d = \sqrt{(X_T - X_B)^2 + (Y_T - Y_B)^2 + (Z_T - Z_B)^2} \quad (11)$$

From here, the minimum distance between the bullet trajectory and the target predicted position is determined for any firing angle, denoted as:

$$\min\_dist = min(d) \quad (12)$$

The method continuously recalculates and update within the predefined limits of the gun's azimuth and elevation angles to find corresponding values. The objective function is constructed based on these results, which are then optimized to find the firing angles corresponding to the minimum value. The initial gun angles used for optimization can be predicted based on operator experience or by utilizing 3DOF trajectory model to prevent large deviations that might affect the optimization results.

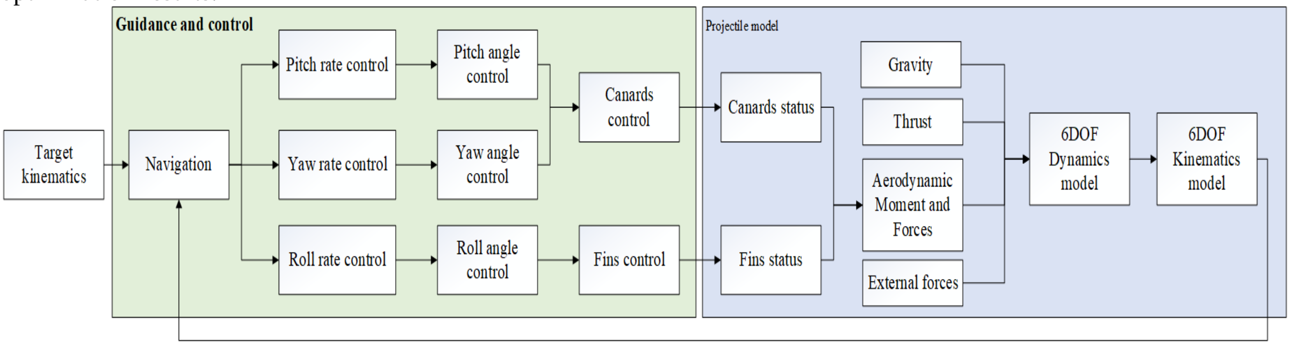


Figure 9: Guided projectile model.

b) Guided Projectile Model for Pursuit Tasks

For guided projectiles, the artillery system employs reconnaissance UAVs or observation stations to detect and track targets. They use a laser designator to mark the target area, and then transmit information (the target's location) back to the artillery system ([9]). The projectile utilizes onboard sensors to detect and track reflected laser energy from the target, therefore it requires continuous laser designation. These sensors guide the projectile by adjusting its flight trajectory via control fins, ensuring high precision.

During its flight, the guided projectile spend over several dynamic phases beside the traditional ballistic trajectory ([31]; [14]). To simulate these phases, this proposed framework integrates a 6DOF artillery projectile model with additional aerodynamic, propulsion, and control components. Fig 9 illustrates this guided projectile model, with key stages as follows:

Launch Phase (Shot/Initial Climb)

This is the initial stage of the firing process when the projectile is launched (exits) from the barrel. During this stage, the projectile is not yet able to self-control, but entering a preliminary trajectory towards the target. After firing, stability projectile is maintained by tail fins, which reduces undesirable roll motions ([10]; [24]). The phase begins at t_0, ending at t_1, when the projectile's onboard propulsion system activates. During this phase, the impulse from the gun acting on the bullet is the main source of energy.

Boost Phase

During this phase (t_1 to t_2), fuel combustion in the onboard projectile propulsion system generates thrust, pushes the projectile to continue the initial trajectory, and combines with the tail fins to reach a stable state in the direction. The thrust from the fuel combustion acts as an energy source for the projectile to accelerate, but the propulsion duration is limited by fuel capacity.

Climb Phase

After the fuel runs out, due to inertia, the guided projectile continues flying up with a decreasing velocity under the effects of resistance and gravity. At the time reaching maximum altitude (t_3), the guided projectile opens its canards and ends the phase. The climb time is determined from t_2 to t_3.

Gliding Phase

Entering the gliding phase (t_3 to t_4), the canards are deployed to generate aerodynamic forces to control the direction of the projectile to the target position. The guidance system is also activated, steering the projectile toward an intermediate position. The intermediate position with Oxy coordinates close to the target position but the Oz coordinates as high as possible position to take advantage of the potential energy from the high altitude to convert into kinetic energy to provide the projectile with acceleration towards the target, which helps to increase accuracy and damage.

Similar to the tail-fins, the canards also have 4 blades but with a different function than the tail-fins with the purpose of controlling the yaw angle (rotation angle around the Oz axis in the axis system attached to the guided projectile body - yaw) and pitch angle (rotation angle around the Oy axis in the axis system attached to the guided projectile body - pitch) for the projectile. The gliding time is determined from the moment the four canard control blades are opened t_3 until the projectile reaches the intermediate position t_4. During this period, the aerodynamic forces generated from the canard's control blades play the main role in controlling the yaw angle and pitch angle for the projectile.

The control system for the three barrel angles is a state feedback controller based on the Backstepping technique and Lyapunov stability criterion. Backstepping is a nonlinear control approach that separates the state variables of the system and creates a series of control steps starting from the control state variable and ending at the control actuator. At each step, the Lyapunov stability criterion is applied to ensure system stability in the control step, and the overall stability of the system is evaluated after the final step. Since the equations describing the three angles of the projectile have the same form, the following steps are applied uniformly to all three angles:

Step 1: Deriving the desired Angular Velocity q_c from the Set Angle θ_c

Upon receiving the desired angle $\boldsymbol{\theta_c}$from the guidance subsystem, in order for the projectile angle $\boldsymbol{\theta}$ to reach this value, the controller determines the required angular velocity $\boldsymbol{q_c}$ (rotation around the Oy axis) based on the kinematic model. It is necessary to minimize the error $e_\theta = \theta_c - \theta \rightarrow 0$. Take the differential of e_θ and applying the Lyapunov stability criterion, the angular velocity around the Oy axis $\boldsymbol{q_c}$ is designed as:

$$q_c = \frac{k_\theta e_\theta + \dot{\theta}_c - f_\theta(p)}{g_\theta(p)} \qquad (13)$$

where k_θ is a convergence rate coefficient for elevation angle, and $f_\theta(p)$, $g_\theta(p)$ are system-specific functions derived from the artillery's state parameters.

Step 2: Determining the Desired Aerodynamic Moment L_a from q_c

Based on the projectile dynamics model, to achieve the desired angular velocity $\boldsymbol{q_c}$, the aerodynamic moment $\boldsymbol{L_a}$, generated by the bullet's control wings, is calculated. Minimizing $e_q = q_c - q \rightarrow 0$, the differential equation for e_q is formed and Lyapunov stability is applied to design aerodynamic moment $\boldsymbol{L_a}$ as:

$$L_a = \frac{k_q e_q + \dot{q}_c - f_q(p)}{g_q(p)} \qquad (14)$$

where k_q controls the convergence rate coefficient for the angular velocity, and $f_q(p)$, $g_q(p)$ depend on the artillery's dynamics.

Step 3: Calculating the Fin Deflection Angles from L_a

• For Roll Angle Control via Fins: The relationship between fin deflection angles and the aerodynamic moment is governed by:

$$\begin{cases} L_a = \frac{1}{2}\rho V^2 S l C_{L\delta}\delta_{rf} \\ \delta_{rf} = \frac{1}{4}(\delta_1 + \delta_2 + \delta_3 + \delta_4) \end{cases} \qquad (15)$$

where , δ_1, δ_2, δ_3, δ_4 are the fin deflection angles [rad], ρ is air density, S is the control fin area, l is the length of the control fin, V is projectile velocity, and $C_{L\delta}$ is aerodynamic coefficient characterizing acting on the moment around the Ox axis (coefficient due to roll rate) of the guided projectile. Therefore, the deflection angle of the 4 fins is determined by the formula:

$$\delta_1 + \delta_2 + \delta_3 + \delta_4 = \frac{8L_a}{\rho V^2 S l C_{L\delta}} \qquad (16)$$

• For Pitch and Yaw Control via Canards: The pitch (M_a) and yaw (N_a) aerodynamic moments are linked to the canard deflections as:

$$\begin{cases} M_a = \frac{1}{4}\rho V^2 S l C_{M\delta}(\delta_2 + \delta_4) \\ N_a = \frac{1}{4}\rho V^2 S l C_{N\delta}(\delta_1 + \delta_3) \end{cases} \qquad (17)$$

where , δ_1, δ_2, δ_3, δ_4 are the canards deflection angles, ρ is air density, S is the control canards area, l is the length of the control canards, V is projectile velocity, and $C_{M\delta}, C_{N\delta}$are, respectively, aerodynamic coefficient characterizing acting on the moment around the Oy axis (coefficient due to pitch rate) and Oz axis (coefficient due to yaw rate) of the guided projectile. Therefore, the deflection angle of the 4 canards is determined by the formula:

$$\begin{cases} \delta_2 + \delta_4 = \frac{4M_a}{\rho V^2 S l C_{M\delta}} \\ \delta_1 + \delta_3 = \frac{4N_a}{\rho V^2 S l C_{N\delta}} \end{cases} \qquad (18)$$

Terminal Phase Control

The final phase is defined from the moment the projectile reaches the intermediate position t_4 until it reaches the target and explodes t_5. From the intermediate position at height h, the projectile's canards adjust its trajectory to strike the target conveniently and accurately. Depending on the target's motion, there are two guidance methods are employed: pursuit and interception. For interception, the system calculates the future collision position by using the positions and velocities of the projectile and target, a trajectory is generated to guide the projectile toward this point. At this phase, the control system is the same as in stage 4 (Gliding Phase).

This structured stage approach enables the simulation framework to model the complex dynamics of guided artillery projectiles, especially for multi-phase trajectories.

3.4 I/O Unit: Data Interface for User Interface and Experience

For data transmission between the naval artillery simulator hardware and the framework (including mode control, configuration parameters, and related results), we propose two approaches (see Fig. 2) depending on the number of

hardware components and the need for scalability. The first approach is suitable for artillery simulators with multiple hardware components or intends to add additional hardware elements that able to support DDS. In this case, we could employ a pub/sub mechanism connected to the DDS middleware of the framework. The second approach is designed for artillery simulators with fixed hardware components or for hardware that does not support DDS. In this case, APIs, WebSocket, or MQTT can be used if the hardware supports network protocols such as HTTP/HTTPS or TCP/IP. For hardware that does not directly support such protocols, data can be retrieved via drivers for embedded protocols (e.g., CAN BUS, UART, I2C. The data is then formatted into DDS on the I/O unit and sent to the DDS middleware.
Integrating within distributed and federated simulation includes updating CGF status, projectile trajectory, and shooting results organized into distributed data distribution interfaces that adhere to integration standards such as High-Level Architecture (HLA), and Distributed Interactive Simulation (DIS). Where data are converted into PDUs and then sent to DIS or objects/interactions and then published to HLA for sharing in a distributed environment. This structure allows the I/O unit could communicate with other modules of the naval simulation platforms or systems, supporting both real-time interactions and offline analysis, Fig. 2.

Table 1: Parameters for intercept firing problems.

Parameters		Case 1	Case 2
Target initial position	Position [m]	[3000,4000,200]	[3000,4000,200]
Target velocity magnitude	Speed [m/s]	200	200
Target initial moving Direction	A zimuth angle relative to Ox [degree]	-20 (clockwise)	-20 (clockwise)
	Elevation angle relative to Ox [degree]	0	0
Initial projectile velocity magnitude	Speed [m/s]	880	880
Environmental conditions: Wind	Velocity magnitude [m/s]	20	20
	Azimuth angle relative to Ox [degree]	-45(clockwise)	-45(clockwise)
	Elevation angle relative to Ox [degree]	0	0
Compensation input values	For wind	no compensation $\Delta(\boldsymbol{\delta}, t) = \{0,0,0\}$	accurate compensation $\Delta(\boldsymbol{\delta}, t) \approx \Delta(\boldsymbol{\beta}_{wind}, t)$ $c_{wind} = 1, \alpha_{wind} = 1$

a) Mode Control and UI Input data interface
In order to manage the input received from the user interface (UI) and transmitting it to the naval artillery framework system for processing, it handles the following:
• Device Configuration Settings: The system receives and manage various configuration setting from user. These settings include the operational modes of onboard systems (such as sensors, radar, and artillery) as well as specific parameters for the artillery system, including fire control adjustments (manual compensation) based on the user's experience or predefined rules.
• User Interaction for Targeting and Fire Control: Through the UI, the users can interact with the simulation to input parameter for firing solutions, including target location, movement, environmental conditions (such as wind speed and atmospheric pressure), and other mission-specific data, or even control the gun directly. The system then uses this information to modify the simulation accordingly, adjusting artillery aiming and fire control to match the real-time inputs.
b) Output Results Data Interface and Analysis Modules
These modules work by compiling the simulation and converting its results into a form usable by the various naval simulation systems. Moreover, these assist the system in visualizing and analyzing the simulations. The outputs include the following:
• Firing Output: Export data of the firing such as the angle of the barrel, projectile trajectory, etc to present them on naval simulation.
• Sensor Data Output: The data is simulated by various onboard sensors (radar, cameras, EO/IR), which are synthetic images or video. These data are then sent out to the naval simulation system to provide a clear representation of the simulation in real-time.

4 PRACTICAL APPLICATION, RESULT AND DISCUSSION

To demonstrate the framework's capability in simulating different types of artillery, we have conducted simulations of two types of artillery projectile for implementation of the framework: a regular type for intercept firing problem under environmental conditions, and a guided projectile type for target pursuit.

4.1 Intercept Firing Problem

This computer simulation framework for naval artillery provides a multifunctional platform for analyzing complex firing scenarios, particularly in the context of target prediction and intercept firing geometry. For the first implementation, focus on a specific experiment scenario to evaluate the effectiveness of manual compensation in artillery firing against moving aerial targets under varying environmental conditions.

Investigating two instances of intercept firing against an aerial target that maintains constant velocity and direction. In both cases, the projectile trajectory is influenced by weather conditions, specifically wind velocity, as shown in Table 1. The key distinction between the two cases lies in the accuracy of the manual compensation input:

- Case 1: Inaccurate environmental correction input
- Case 2: Accurate environmental correction input

In the first scenario, without manual compensation applied, the simulation results reveal that the projectile's actual trajectory deviates from the desired trajectory points, leading to substantial positional error relative to the target at the intended intercept point, as shown in Fig 10(a). This emphasizes the critical influence of environmental factors, particularly wind, on the accuracy of solving for intercept firing problem. In contrast, the second scenario, which used precise manual wind compensation, produces improved results, the projectile's actual trajectory matched with the desired trajectory points altogether, significantly reduced miss distance and the intercept of the moving target successfully, as illustrated in Fig 10(b).

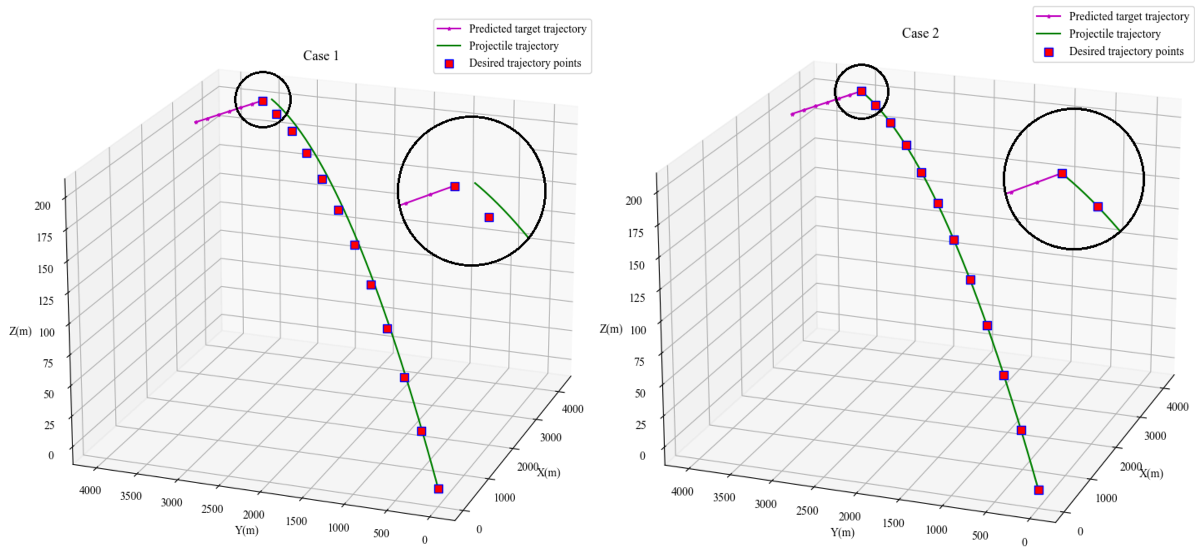
Table 2 demonstrates that, for case 1, where the experiment does not account for manual compensation, the computed firing parameters to control the artillery barrel are as follows: an azimuth angle of 40.54° and an elevation angle of 4.081° relative to OX axis. The results indicate that when the wind direction is -45° (clockwise), the entire projectile trajectory is shifted forward and results in a miss on the target. In case 2, where the experiment incorporates manual compensation based on wind direction, the adjusted parameters are 41.86° for the azimuth angle counterclockwise to ensure that the projectile successfully hits the target. The adjustments enabled successful target engagement, emphasizing the role of accurate environmental data and effective compensation techniques in trajectory calculating for naval artillery systems.

4.2 Guided Projectile Problem

This study provides an integrable solution for naval simulation, particularly focusing on guided projectile pursuit firing scenarios. For the second implementation, this section discusses the results of two key simulation cases: firing at a stationary target and a moving target on the sea surface, at the same initial conditions, see Table 3.

Table 2: Results for intercept firing problems.

Parameter		Case 1	Case 2
Computed firing parameters	Azimuth angle [degree]	40.54	41.86
	Elevation angle [degree]	6.261	4.0808
Predicted time to hit the target	Calculated from firing moment, [s]	6.261	6.25
Predicted intersection point	between the projection and the target	[4176.6,3571.7,200]	[4174.6,3572.4,200]
Actual position of the projectile	at predicted time	[4265.2,3483.1,200]	[4174.6,3572.4,200]



a) b)

Figure 10: Interception Problem, a) case 1 no compensation, b) case 2 correct compensation.

Table 3: Parameters for guided projectile problems.

Parameter		Case 1	Case 2
Initial position and direction of gun barrel	Position [m]	[0; 0; 5]	[0; 0; 5]
	Direction [degree]	[0;35˚;0]	[0;35˚;0]
Initial projectile velocity	Translational velocity [m/s]	[200;0;0]	[200;0;0]
	Angular velocity [rad/s]	[0;0;0]	[0;0;0]
Initial position and velocity direction of target	Position [m]	[6000;8000;4]	[6000;8000;4]
	Direction [degree]	[0;1;120˚]	[0;1;120˚]
Target translational and angular velocity	Translational velocity [m/s]	[0;0;0]	[20;0;0]
	Angular velocity [rad/s]	[0;0;0]	[0;0;0]

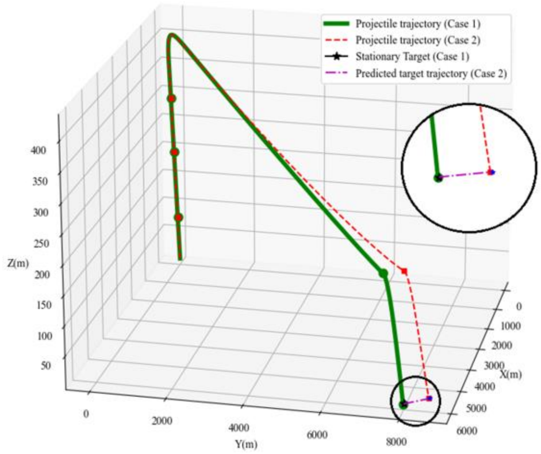


Figure 11: Results for both cases: stationary target and a moving target.

Fig 11 shows the trajectories of guided projectiles for stationary and moving targets engagements. The solid line illustrates the projectile trajectory through the stationary target, while the dashed line presents the trajectory towards the moving target. The significant distinction between the two curves indicates how the system can account for target motion, a critical feature in in authentic naval combat situations where targets are generally considered not motionless, and moving unpredictable. In both cases, it can be clearly seen the projectiles successfully converge on their respective targets, which shows the effectiveness of the guidance system in ensuring high accuracy (Terminal Accuracy). Thus, both trajectories demonstrate the guided projectile's ability to track the target motion and to maneuver itself towards the target, regardless of the target's motion state, it reassesses and adjusts trajectory continuously. This adaptability is important for modern naval artillery simulation systems for improving their effectiveness in various engagement scenarios (Adaptive Trajectory). The marked points along each trajectory indicate the end of one flight phase and as such the beginning point of other.

Fig 12 demonstrates the practical application of the computer simulation framework for naval artillery within a warship simulator, in a case against targets on the sea surface. Thus, the evaluation of this application indicates that the framework can be seamlessly integrated into a larger naval simulation system. This highlights its potential as a versatile tool for naval artillery development and training.

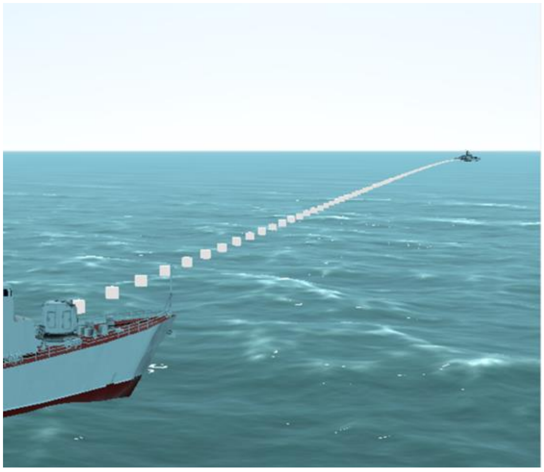


Figure 12: Simulation of naval artillery gunfire sea surface target.

5 CONCLUSION, AND FUTURE WORK

From this study, novel contributions to the modeling and simulation of naval artillery include the proposed robust naval artillery simulation framework. The flexibility of the framework is highlighted in the research through its different uses. This simulation framework presents considerable potential for training naval personnel in various scenarios without the requirement for live-fire exercises. Implementation of the framework has shown its ability to integrate into broader and more comprehensive naval simulation systems. This work presented a flexible framework that can easily be extended or modified based on user-specific requirements by using a modular architecture, which is one of its major advantages. The example for the adaptive, framework can support integration with advanced targeting algorithms, flexible trajectory, and various environmental modeling. Acquiring and developing these capabilities ensures that it sustains relevance in current and future navy simulation requirements.

According to the evaluation made in the previous sections, the described framework includes the sufficient and necessary elements for meeting the qualities of suitable naval artillery simulation framework as well as providing the fundamental starting point for the integrative naval combat systems. Its ability to simulate different situations effectively and adapt to system changes due to user requirements makes it useful for both research and operational planning.

Future development efforts will focus on the following:

- Enhanced Integration: Using artificial intelligence (AI) and machine learning when making decisions in uncertain conditions.
- Multi-Domain Simulation: Expanding the framework to cover interoperation of air, land, and sea systems for supporting the tactical planning.

By attending to these directions, the framework has the potential to expand into a comprehensive solution that effectively fills the gap between theoretical models of simulation and actual naval use.

Acknowledgement:

We would like to express our sincere appreciation to the Modeling and Simulation Center, part of Viettel High Technology Industries Corporation, for their support and valuable resources during this study. Our heartfelt thanks also go to the leadership team at the Modeling & Simulation Institute for their continuous encouragement and guidance throughout this work. We also deeply appreciate the team at the Modeling and Simulation Center for their hands-on support, thoughtful insights, and crucial contributions during the testing and improvement of the algorithms explored in this project.

References

[1] Angelliaume, S., Rosenberg, L., & Ritchie, M. Modeling the Amplitude Distribution of Radar Sea Clutter. *Remote Sensing* 11, 3 (2019), 319.
[2] Antipov, I. Analysis of Sea Clutter Data. *DSTO Electronic and Surveillance Research Laboratory*, Department Of Defence Defence Science and Technology Organisation (1998a).
[3] Antipov, I. Simulation of Sea Clutter Returns. *DSTO Electronic and Surveillance Research Laboratory*, Department of Defence Defence Science and Technology Organisation (1998b).
[4] Baranowski, L. Effect of the mathematical model and integration step on the accuracy of the results of computation of artillery projectile flight parameters. Bulletin of the Polish Academy of Sciences. *Technical Sciences* 61, 2 (2013a), 475-484.
[5] Baranowski, L. Equations of motion of a spin-stabilized projectile for flight stability testing. *Journal of Theoretical and Applied Mechanics* 51, 1 (2013b), 235-246.
[6] Baranowski, L. Feasibility analysis of the modified point mass trajectory model for the need of ground artillery fire control systems. *Journal of Theoretical and Applied Mechanics* 51, 3 (2013c), 511-522.
[7] Baranowski, L. Numerical testing of flight stability of spin-stabilized artillery projectiles. *Journal of Theoretical and Applied Mechanics* 51, 2 (2013d), 375-385.
[8] Baranowski, L., Gadomski, B., Szymonik, J., & Majewski, P. Comparison of explicit and implicit forms of the modified point mass trajectory model. *Journal of Theoretical and Applied Mechanics* 54, 4 (2016), 1183-1195.
[9] Barth, J., Fendt, A., Florian, R., & Kieslich, W. Dual-Mode Seeker with Imaging Sensor and Semi-Active Laser Detector. *Proceedings of SPIE* 6542 (2007), 65423B-1.
[10] Blair, A. B., Allen, J. M., & Hernandez, G. Effect of Tail-Fin Span on Stability and Control Characteristics of a Canard-Controlled Missile at Supersonic Mach Numbers. NASA Technical Paper 2157. *National Aeronautics and Space Administration, Scientific and Technical Information Branch* (1983).
[11] Bresenham, J. E. Algorithm for computer control of a digital plotter. *IBM Systems Journal* 4, 1 (1965), 25-30.
[12] Calfee, S. H. Autonomous Agent-Based Simulation of An Aegis Cruiser Combat Information Center performing Battle Group Air-Defense Commander Operations. Master's thesis, *Naval Postgraduate School, Monterey, California* (2003).
[13] Calfee, S. H., & Rowe, N. C. Multi-Agent Simulation of Human Behavior in Naval Air Defense, AEGIS Cruiser Air-Defense Commander (ADC) Simulation. *Naval Engineers' Journal* 116, 4 (2004), 53-64.
[14] Chen, Q., Wang, Z., Chang, S., & Fu, J. Multiphase trajectory optimization for gun-launched glide guided projectiles. Proceedings of the Institution of Mechanical Engineers, Part G: *Journal of Aerospace Engineering* 0, 0 (2015), 1-16.

[15] Curry, J., & Pratt, F. Pratt's Naval Wargame: Wargaming with Model ships 1900-1945. *History of Wargaming Project* (2012).
[16] Foley, J. Computer graphics: principles and practice. *Addison-Wesley Professional* (1996).
[17] Fujimoto, R. M. Time Management in the High Level Architecture. *Simulation* 71, 6 (1998), 388-400.
[18] Gomes, N. R., Dammert, P., Pettersson, M., Vu, V. T., & Hellsten, H. Comparison of the Rayleigh and K-Distributions for Application in Incoherent Change Detection. *IEEE Geoscience and Remote Sensing Letters* 16 (2019), 756-760.
[19] Gregers-Hansen, V., & Mital, R. An Improved Empirical Model for Radar Sea Clutter Reflectivity. *Naval Research Laboratory, Washington, DC* (2012).
[20] Ham, W. K., Kwon, Y., & Park, S. C. Combat simulation framework including continuous detection system. *International Journal of Simulation Modelling* 13, 4 (2014), 395-408.
[21] Jane, F., Castle, W., Colomb, P., Chamberlain, H., & Curry, J. Over open sights early naval wargaming rules 1873-1904 early wargame (Vol. 6). *History of Wargaming Project* (2014).
[22] Kim, E. J. Naval Gun Fire Control System Simulation for Verification Depending on Development Phase. *Journal of Korea Society for Simulation* (2011).
[23] Kim, K., Jo, H., & Kim, D. Design and Implementation of Adaptive Naval Gun Fire Simulator on a Naval Combat System. *Journal of the KIMST* 21, 5 (2018), 630-639.
[24] Kim, S. H., Lee, Y. I., & Tahk, M. J. New Structure for an Aerodynamic Fin Control System for Tail Fin-Controlled STT Missiles. *Journal of Aerospace Engineering* 24, 4 (2011), 505-510.
[25] King, F. Artillery Engagement Simulation. *Simulation Systems Technical Area, U.S. Army, Research Institute for the Behavioral and Social Sciences* (1980).
[26] Kumek, M. Ship Self Air Defense Analysis for Different Operation Conditions via Simulation. Master's thesis, *Sabancı University* (2007).
[27] Leonhardt, R. J., & Schelegelmilch, C. R. A computer Simulation for the Evaluation of Fleet Air Defense Employing Monostatic and Bistatic search radar. Master's thesis, *Naval Postgraduate School Monterey, California* (1975).
[28] Li, Y. Design and realization of a novel radar simulator. *35th Chinese Control Conference (CCC)* (2016).
[29] Li, Y., & Ren, H. A Modified Algorithm of Radar Simulator Echo Images Generation. *12th World Congress on Intelligent Control and Automation (WCICA)* (2016).
[30] McTaggart, K., Henry, G., Oakey, S., & Van Spengen, *J. Naval Platform Simulation Using the NATO Virtual Ships Standard*. STO-MP-MSG-159 (2019).
[31] STANAG 4355 JAIS. The modified point mass and five degrees of freedom trajectory models. *NATO Standardization Agency* (Edition 3) (2009).
[32] O'Gara, G. C. Theodore Roosevelt and the Rise of the Modern Navy. *Princeton University Press, Praeger* (1943, new Ed - 1970).
[33] Ozkan, B., Rowe, N. C., Calfee, S. H., & Hiles, J. E. Three Simulation Models of Naval Air Defense. *The Space and Naval Warfare Systems Center, San Diego, California, 10th ICCRTS*. Paper: I-194 (2005).
[34] Parthiban, A., Madhavan, J., Radhabishna, P., Savitha, D., & Kumar, L. S. Modeling and Simulation of Radar Sea Clutter Using K-Distribution. *International Conference on Signal Processing & Communications (SPCOM)* (2004).
[35] Ranaweera, W. A. Lightweight Approach to Simulate a 2D Radar Coverage for Virtual Maritime Environments. *8th International Conference on Computer Science & Education (ICCSE)* (2013).
[36] Rauber, R. M., & Nesbitt, S. W. Radar Meteorology: A First Course. *John Wiley & Sons Ltd* (2018).
[37] Skande, M. Numerical Solution to a Nonlinear External Ballistics Model for a Direct Fire Control System. *KTH, School of Industrial Engineering and Management (ITM), Machine Design (Dept.)* (2014).
[38] Trunk, G. V. Radar Properties of Non-Rayleigh Sea Clutter. *IEEE Transactions on Aerospace and Electronic Systems AES-8*, 2 (1972).
[39] van Dongen, M. P. F. M., & Kos, J. The Analysis of Ship Air Defense: *The Simulation Model SEAROADS. Naval Research Logistics* 42 (1995), 291-309.
[40] Wang, H., You, D., Ma, Y., & Wang, D. Research on the Visualization for the Naval-Gun Fire-Training Scene. *International Conference on Computer Application and System Modeling (ICCASM)* (2010).
[41] Ward, K., Tough, R., & Watts, S. Sea Clutter Scattering, the K Distribution and Radar Performance (2nd Ed.). *The Institution of Engineering and Technology* (2013).
[42] Yang, M., & Wang, G. The Building of Naval Gun Dynamics Simulation Model Based on Adams. *International Conference on Computer Science and Service System (CSSS)* (2014).
[43] Zhang, J., Chen, Y., Fu, X., & Chen, F. Research of Echo Image Simulation in Maritime Radar Simulator. *Conference on Industrial Electronics and Applications (ICIEA), IEEE* (2017).
[44] Zhi, P., Liang, Z., Zhu, W., & Zhu, Z. Weapon target assignment strategy for shipboard power system considering high power pulse loads integration. *3rd International Conference on Power Engineering (ICPE2022)* (2023).
[45] Żuluk, A. Numerical simulation of the effect of wind on the missile motion. *Journal of Theoretical and Applied Mechanics* 52, 2 (2014), 335-344.