

UNIVERSITY OF SÃO PAULO
INSTITUTE OF MATHEMATICS AND STATISTICS
BACHELOR'S DEGREE IN APPLIED AND COMPUTATIONAL MATHEMATICS

**Study of the expected cost of neutralizing
naval surface drones**

*A 6-DOF ballistic simulation applied to
5"/38 naval artillery*

Luiz Guilherme de Padua Sanches

FINAL ESSAY

MAP 2429 – CAPSTONE PROJECT

Advisor: Prof. Dr. Clodoaldo Grotta Ragazzo

São Paulo
2025

*The content of this work is published under the CC BY 4.0 license
(Creative Commons Attribution 4.0 International License)*

Abstract

Luiz Guilherme de Padua Sanches. **Study of the expected cost of neutralizing naval surface drones: A 6-DOF ballistic simulation applied to 5"/38 naval artillery.**

Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2025.

In recent years, small unmanned vessels loaded with explosives have begun to be employed as low-cost weapons in littoral environments. This work analyses the expected cost of neutralizing a naval surface drone by means of 5"/38 naval artillery, explicitly taking into account ballistic uncertainty and the cost associated with a possible failure to stop the target. The projectile trajectory is modelled by a six-degrees-of-freedom system of differential equations, based on classical external ballistics theory, with aerodynamic coefficients obtained from United States government documents for 5"/38 ammunition. The numerical solution provides, for each pair of azimuth and elevation angles, the impact position relative to the target. Uncertainty in the firing parameters is represented by Gaussian perturbations in elevation and azimuth, and hit probabilities are estimated by Monte Carlo simulations. A cost function is then defined that combines ammunition expenditure over a sequence of shots and the cost of losing the ship in the event of non-neutralization, resulting in the expected neutralization cost for each firing strategy. For the reference scenario studied, in which the ship employs only the main battery against a single surface drone, the expected neutralization cost is on the order of 77 million dollars, of which about 192.7 thousand dollars correspond to the average ammunition cost. The dominant term in the cost is therefore the risk of losing the naval platform, whereas ammunition expenditure remains below the estimated value of the drone itself. It is concluded that, under the assumptions adopted, the isolated use of 5"/38 artillery is not sufficient, from a purely ballistic standpoint, to restore a cost asymmetry favorable to the defending ship.

Keywords: 6-DOF. naval drones. cost analysis. Monte Carlo simulation. 5"/38 naval gun.

Contents

1	Introduction	1
	Introduction	1
2	Theoretical background	5
2.1	Conceptual foundations and scope	5
2.1.1	Angular momentum	5
2.1.2	Moment of inertia	8
2.1.3	Euler's equations	9
2.1.4	Direction cosines	10
2.2	The 6-DOF equations	11
2.2.1	Derivation of the equations	12
2.2.2	Aerodynamic force and moment coefficients	15
2.2.3	Auxiliary identities	17
2.2.4	The six scalar equations	19
2.2.5	General modeling considerations	20
2.2.6	Initial conditions	20
2.3	Statistical model and Monte Carlo estimation	22
3	Methodology	25
3.1	Physical configuration of the problem	25
3.1.1	Selection of the projectile and aerodynamic coefficients	25
3.1.2	Inertial parameters, initial conditions, and ammunition cost	27
3.1.3	Firing platform and target	28
3.2	Computational implementation of the simulation	29
3.2.1	Preparation of coefficients and physical parameters	30
3.2.2	Core numerical integration	30
3.2.3	Layer of numerical experiments	31

4 Numerical results and discussion	33
4.1 Deterministic reference shot	33
4.1.1 Time evolution of altitude and total angle of attack	34
4.1.2 Time evolution of velocities and the symmetry axis	36
4.2 Selection and spacing of engagement configurations	38
4.3 Monte Carlo simulations for hit probability	39
4.3.1 Geometric visualization of dispersion for a representative configuration	39
4.3.2 Estimated probabilities along the trajectory	40
4.4 Analysis of the expected engagement cost	41
4.4.1 Expected ammunition cost	41
4.4.2 Accumulated probability of success	42
4.4.3 Total expected cost and operating point	42
5 Conclusion	45
Annexes	
A Source code of the 6-DOF simulation	49
B Relevant results for shots against the <i>Sea Baby</i> drone	57
References	61

Chapter 1

Introduction

As noted by CROSBY (2024), maritime conflicts throughout the twentieth century reveal a recurring pattern: the employment of asymmetric naval strategies by forces at a numerical or technological disadvantage. During the two World Wars, for instance, Germany confronted the supremacy of the British Royal Navy by relying heavily on submarine warfare to interdict Allied convoys (BLAIR, 1996). In addition, heavy cruisers such as the *Admiral Graf Spee* were employed primarily against vulnerable targets, avoiding direct confrontation with superior forces (BIDLINGMAIER, 1971). Complementarily, fast battleships such as the *Scharnhorst* and the *Gneisenau* compelled the Allies to disperse escorts and to keep hunter-killer groups permanently mobilized, with a direct impact on logistical and strategic costs (HANDEL-MAZZETTI, 1956). This set of actions imposed on the defending forces the need to allocate at least one destroyer, frigate, or corvette to escort convoys of merchant ships, in addition to maintaining constant air cover along critical routes, which significantly increased operational costs and the escort burden (MARDER, 1969; CONNORS, 2019).

Analogous strategies re-emerged in the Middle East in the second half of the twentieth century. During the so-called Tanker War and in later incidents in the Persian Gulf, small fast craft armed with missiles or torpedoes began to attack civilian tankers and military vessels, forcing international coalitions to organize convoys protected by escorts (NAVIAS and HOOTON, 1996). In these operations, relatively inexpensive offensive platforms imposed relevant risks on much more expensive cargo vessels and combatants, shifting the financial burden to the side responsible for ensuring the security of navigation. This imbalance between the low cost of offensive platforms and the high cost of defensive means characterizes the economic-strategic logic typical of asymmetric warfare at sea, frequently cited in contemporary analyses of the security of oil-shipping routes and attacks on merchant vessels (BBC NEWS, 2019; PHILLIPS, 2024; MERCOLIANO, 2019).

At the beginning of the twenty-first century, this picture worsened with the spread of unmanned vehicles across multiple operational environments. In particular, the war in Ukraine and the recent attacks in the Black Sea have highlighted the intense use of unmanned aerial vehicles and unmanned surface vehicles as relatively inexpensive yet highly lethal attack vectors, capable of damaging or sinking much more expensive patrol

ships, frigates, and support vessels (KIRICHENKO, 2025b; KIRICHENKO, 2025a). Similarly, attacks in the Red Sea and adjacent areas, conducted with drones and short-range missiles, have exposed merchant ships and naval units to the risk of recurring engagements by low-cost threats (SELIGMAN and BERG, 2023). In several episodes, merchant or military vessels were threatened by swarms of low-cost drones, which required the use of guided munitions, surface-to-air missiles, or complex weapon systems whose unit cost is, in many cases, of the same order of magnitude as, or higher than, the cost of the neutralized threat itself (SELIGMAN and BERG, 2023; TREVITHICK, 2024). Thus, the asymmetry between the cost of offensive platforms and the cost of the defensive means required to preserve the integrity of high-value vessels is reinforced.

Remark. A recent example of the offensive use of naval drones against the so-called Russian *shadow fleet*—tankers used to circumvent economic sanctions—was reported by *The Kyiv Independent* on 29 November 2025, describing a Ukrainian naval-drone attack on the tankers *Kairos* and *Virat* in the Black Sea (THE KYIV INDEPENDENT, 2025). This episode occurred during the revision period of the present work and is included here in order to illustrate the timeliness and relevance of the investigated topic.

This historical panorama raises a central question: what is the expected cost, in terms of ballistic ammunition, that a modern escort ship must expend to neutralize a low-cost naval target, such as an unmanned surface drone? Such drones tend to have an acquisition cost far lower than that of frigates, destroyers, or support vessels employed in their defense, which amplifies the economic–strategic asymmetry underlying the problem (SELIGMAN and BERG, 2023). In this work, the focus is on the munitions fired by the main-battery naval guns of such escorts, weapon systems that combine a high rate of fire with a relatively affordable unit price when compared to guided missiles (LEONARDO DRS, 2020; TREVITHICK, 2024).

It is precisely this cost asymmetry that motivates the present undergraduate thesis. In general terms, the goal is to provide an estimate of the expected cost associated with the ballistic neutralization of a naval surface drone, understood here as a small, unmanned, and relatively low-cost vehicle engaged by main-battery naval guns.

To achieve this objective, we adopt a six-degrees-of-freedom ballistic model, or 6-DOF model, widely used in the literature to describe the motion of artillery projectiles that spin about their longitudinal axis (McCoy, 2012; CARLUCCI and JACOBSON, 2014; GKRITZAPIS *et al.*, 2007). In this formulation, the projectile is treated as a spinning rigid body, and the equations of motion couple the translational dynamics of the center of mass to the rotational dynamics about principal axes of inertia, under the action of aerodynamic forces and moments that depend on velocity, total angle of attack, and spin rate. When supplied with aerodynamic coefficients obtained experimentally or by numerical methods, this model can reproduce, with good fidelity, trajectories observed in ballistic tests (McCoy, 2012).

However, a purely deterministic model is not sufficient to reproduce the real dispersion observed in operations, since quantities such as variations in propellant charge, winds in multiple atmospheric layers, and aiming errors introduce relevant uncertainties into the problem (KHALIL *et al.*, 2009; LIU *et al.*, 2011). To incorporate them, this work combines 6-DOF modeling with Monte Carlo simulations. Critical parameters of the initial firing conditions, such as elevation angle and firing direction, are treated as random variables within ranges compatible with the operation of the weapon system. Each Monte Carlo

sample corresponds to a complete trajectory solved numerically, and the collection of such trajectories produces a cloud of impact points around the deterministic solution.

From these impact distributions, it is possible to estimate hit probabilities. By combining these probabilities with the unit cost of the munitions, we define a measure of the average neutralization cost that expresses the expected number of shots (or the associated monetary cost) required to neutralize a single drone under given engagement scenarios.

It is expected that this approach, although based on simplifying hypotheses and representative scenarios, will adequately reflect essential aspects of the physical and operational problem, contributing to a quantitative understanding of the impact that low-cost naval drones may have on the planning and employment of main-battery naval guns in asymmetric-warfare contexts.

Objectives

To state the aims of this work clearly, we distinguish below the general objective and a set of specific objectives.

The general objective of this undergraduate thesis is to *compute the expected neutralization cost of a naval surface drone, considering exclusively the cost of the main-battery ammunition and the cost associated with failure to neutralize the target*. To this end, we assume a sequence of up to n shots, each with probability p_i of neutralizing the target on the i -th shot, unit ammunition cost equal to c , and vessel cost E in the event that the target is not neutralized after all admissible shots. Under these conditions, the expected cost is modeled by

$$\mathbb{E}[C] = c(1 + (1-p_1) + (1-p_1)(1-p_2) + \dots + (1-p_1) \dots (1-p_{n-1})) + E(1-p_1) \dots (1-p_n), \quad (1.1)$$

The structure of (1.1) follows from the operational rule adopted: the firing sequence is interrupted as soon as the target is neutralized, so that the $(i+1)$ -th shot occurs only if the first i shots have not been successful. Thus, the expected ammunition cost equals the unit cost c multiplied by the *expected number of shots fired*.

In particular, the first shot occurs with probability 1, the second occurs only if the first fails, with probability $(1-p_1)$, the third occurs only if the first two fail, with probability $(1-p_1)(1-p_2)$, and so on, up to the n -th shot, which occurs with probability $(1-p_1) \dots (1-p_{n-1})$.

By summing these occurrence probabilities and multiplying by c , we obtain the first term of (1.1). Finally, the term $E(1-p_1) \dots (1-p_n)$ represents the cost associated with the scenario in which none of the n admissible shots neutralizes the target, whose probability is precisely the product of the failure probabilities over all shots.

From this general objective, we define the following specific objectives:

- to formulate and implement, in the Python programming language, a six-degrees-of-freedom ballistic model (6-DOF model) for projectiles, based on consolidated

references in the literature (McCoy, 2012; CARLUCCI and JACOBSON, 2014), in a reproducible manner;

- to collect and organize the aerodynamic coefficients, geometric parameters, and other relevant characteristics of the selected munitions and associated naval platforms, from technical reports, standards, and available reference material (WHYTE, 1973; HASELTINE, 1969; THOMAS and GOSWICK, 1967; GAO, 1990; SUPERVISOR OF SHIPBUILDING (SUPSHIP), SEATTLE, WASHINGTON, 1965; THE KYIV INDEPENDENT, 2024; SUTTON, 2024);
- to define a set of engagement positions and geometries in the form of a representative target-point *grid* for the possible positions occupied by naval drones approaching the escort ship, specifying distance bands and approach angles of operational interest;
- to apply Monte Carlo simulations to the 6-DOF model, varying the initial firing conditions within plausible ranges, in order to estimate impact distributions and hit probabilities associated with each point of the defined *grid* (CORRIVEAU, 2017; LIU *et al.*, 2011);
- to compute, for each engagement scenario considered, the expected neutralization cost according to (1.1) and to discuss the implications of these results for the employment of main-battery naval guns in asymmetric-warfare contexts.

Chapter 2

Theoretical background

This chapter, devoted to the literature review and to the theoretical background, focuses on the two main pillars of the proposed model. The first pillar, treated in greater detail, is the external-ballistics modeling of artillery projectiles, which provides the “engine” used to simulate six-degree-of-freedom trajectories and requires a more careful presentation, since it is a relatively specialized technical topic (CARLUCCI and JACOBSON, 2014; McCOY, 2012). The second pillar is the estimation of hit probability by means of Monte Carlo simulations, formulated in terms of Bernoulli variables and a simple expected-cost model—a framework that is already well established in the probability and applied-statistics literature (ROSS, 2014; KROESE *et al.*, 2011).

2.1 Conceptual foundations and scope

Following the advisor’s recommendation, this chapter presents a concise review of four physical concepts that support the six-degree-of-freedom ballistic formulation used in this work: (i) *angular momentum*; (ii) *moment of inertia*; (iii) *Euler’s equations*; and (iv) *direction cosines*, all considered in the context of rigid bodies. The main reference is GOLDSTEIN *et al.* (2002), complemented by LEVI (2014), in order to keep the notation consistent with the classical literature on rigid-body dynamics. In the next section, these concepts will be connected directly to the 6-DOF formulation for artillery projectiles, based on McCOY (2012) and CARLUCCI and JACOBSON (2014).

We adopt the standard notion of a rigid body as a system of material points that preserves, throughout the motion, the distances between any pair of points. Physically, this corresponds to an undeformable entity: its evolution in space is completely described by a combination of translation and rotation of the body as a whole (GOLDSTEIN *et al.*, 2002).

2.1.1 Angular momentum

As a starting point, we consider the classical decomposition of the total kinetic energy of a rigid body (in the notation of GOLDSTEIN *et al.* (2002)):

$$T = \frac{1}{2} M \|\vec{v}\|^2 + T'(\phi, \theta, \psi), \quad (2.1)$$

where the first term represents the kinetic energy associated with the motion of the center of mass (CM), and the second term aggregates the contribution of motion relative to the CM. The notation $T'(\phi, \theta, \psi)$ makes explicit the dependence on the body's attitude through the Euler angles; operationally, this term involves the rates of these angles through the instantaneous angular velocity of the body (GOLDSTEIN *et al.*, 2002).

For a minimal description of the motion, we adopt the set of generalized coordinates formed by the Cartesian coordinates of the CM (x, y, z) and three orientation parameters, here chosen as the Euler angles (ϕ, θ, ψ) , which are sufficient to characterize the evolution of an undeformable body (GOLDSTEIN *et al.*, 2002). In later chapters, the attitude will be described preferably by a direction-cosine matrix, but the formulation of GOLDSTEIN *et al.* (2002) in terms of Euler angles remains the basic conceptual reference.

Rate of change of a vector

A central result in rigid-body dynamics is the relationship between the time derivatives of a vector in two distinct reference frames. Let \vec{G} be a generic vector; GOLDSTEIN *et al.* (2002, §4.9) shows that its infinitesimal variation, observed from a frame fixed in space (“space”) and from a frame attached to the body (“body”), satisfies

$$(d\vec{G})_{\text{space}} = (d\vec{G})_{\text{body}} + d\vec{\Omega} \times \vec{G},$$

where $d\vec{\Omega}$ is the body's infinitesimal rotation. The proof that $(d\vec{G})_{\text{rot}} = d\vec{\Omega} \times \vec{G}$ is given in GOLDSTEIN *et al.* (2002, §4.8) and will not be repeated here. Dividing by dt , one obtains (cf. GOLDSTEIN *et al.* (2002, §4.9, eq. (4.82)))

$$\left(\frac{d\vec{G}}{dt} \right)_{\text{space}} = \left(\frac{d\vec{G}}{dt} \right)_{\text{body}} + \vec{\omega} \times \vec{G}, \quad \vec{\omega} = \frac{d\vec{\Omega}}{dt}, \quad (2.2)$$

Since \vec{G} is arbitrary, the relation (2.2) holds for the time derivative of any vector observed in two distinct coordinate systems.

Uniqueness of angular velocity and definition of angular momentum

Now consider two arbitrary points of the body, with position vectors \vec{r}_1 and \vec{r}_2 in the inertial frame, and define $\vec{R} = \vec{r}_2 - \vec{r}_1$. Applying (2.2) to \vec{R} , one obtains $\dot{\vec{R}} = \vec{\omega} \times \vec{R}$ (GOLDSTEIN *et al.*, 2002, §4.9, eq. (4.82)). Thus,

$$\left(\frac{d\vec{r}_2}{dt} \right)_{\text{space}} = \left(\frac{d\vec{r}_1}{dt} \right)_{\text{space}} + \left(\frac{d\vec{R}}{dt} \right)_{\text{space}} = \left(\frac{d\vec{r}_1}{dt} \right)_{\text{space}} + \vec{\omega}_1 \times \vec{R},$$

and, alternatively,

$$\left(\frac{d\vec{r}_1}{dt} \right)_{\text{space}} = \left(\frac{d\vec{r}_2}{dt} \right)_{\text{space}} - \left(\frac{d\vec{R}}{dt} \right)_{\text{space}} = \left(\frac{d\vec{r}_2}{dt} \right)_{\text{space}} - \vec{\omega}_2 \times \vec{R}.$$

Comparing the two expressions for the difference of velocities yields

$$(\vec{\omega}_1 - \vec{\omega}_2) \times \vec{R} = \vec{0}.$$

Since \vec{R} can be chosen as the vector between any two points of the body (with at least two non-collinear directions), it follows that $\vec{\omega}_1 = \vec{\omega}_2$. In other words, the *angular velocity* $\vec{\omega}$ is unique for a rigid body and is independent of the pair of points used in its definition (GOLDSTEIN *et al.*, 2002, §4.9).

Under these conditions, the *angular momentum* of the i -th particle is defined by

$$\vec{L} = m_i (\vec{r}_i \times \vec{v}_i), \quad (2.3)$$

where \vec{r}_i and \vec{v}_i are, respectively, the particle's position vector and velocity, and $\vec{p}_i = m_i \vec{v}_i$ is its linear momentum (LEVI, 2014, Def. 5.1).

Applying (2.2) to the position vector \vec{r}_i yields

$$\left(\frac{d\vec{r}_i}{dt} \right)_{\text{space}} = \left(\frac{d\vec{r}_i}{dt} \right)_{\text{body}} + \vec{\omega} \times \vec{r}_i.$$

Since \vec{r}_i is fixed in the body frame, $\left(\frac{d\vec{r}_i}{dt} \right)_{\text{body}} = \vec{0}$; therefore,

$$\vec{v}_i = \left(\frac{d\vec{r}_i}{dt} \right)_{\text{space}} = \vec{\omega} \times \vec{r}_i. \quad (2.4)$$

Substituting $\vec{v}_i = \vec{\omega} \times \vec{r}_i$ into (2.3) gives

$$\vec{L} = m_i [\vec{r}_i \times (\vec{\omega} \times \vec{r}_i)].$$

To make the structure of this term explicit, write $\vec{r} = (x, y, z)^\top$ and $\vec{\omega} = (\omega_x, \omega_y, \omega_z)^\top$. Then

$$\vec{r} \times (\vec{\omega} \times \vec{r}) = \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}.$$

Multiplying by m , the diagonal elements (times m) define the *moments of inertia* $I_{xx} = m(y^2 + z^2)$, $I_{yy} = m(x^2 + z^2)$, and $I_{zz} = m(x^2 + y^2)$. The off-diagonal elements define the *products of inertia* $I_{xy} = -mxy$, $I_{xz} = -mxz$, and $I_{yz} = -myz$.

Collecting these coefficients into the inertia matrix \mathbf{I} , one obtains the compact vector form

$$\boxed{\vec{L} = \mathbf{I} \vec{\omega}} \quad (2.5)$$

for the angular momentum of a rigid body (GOLDSTEIN *et al.*, 2002).

For a discrete system of particles, the angular momentum about a pole O is

$$\vec{L}_O = \sum_i \vec{r}_i \times m_i \vec{v}_i,$$

whereas, for a continuous body, one has the integral form

$$\vec{L}_O = \int_B \vec{r} \times \vec{v} dm, \quad (dm = \rho dV),$$

with all quantities measured from O (LEVI, 2014, Def. 2.2). In both cases, the inertia matrix I is a real symmetric 3×3 matrix and admits diagonalization by an orthonormal basis:

$$I' = R I R^\top = \text{diag}(I_1, I_2, I_3),$$

where R is the orthogonal matrix whose columns are the normalized eigenvectors and I_1, I_2, I_3 are the *principal moments of inertia* (GOLDSTEIN *et al.*, 2002; Levi, 2014).

2.1.2 Moment of inertia

For the i -th particle, the kinetic energy about a point is

$$T_i = \frac{1}{2} m_i \|\vec{v}_i\|^2. \quad (2.6)$$

Using the kinematic relation (2.4), we substitute *only one* of the velocity factors by $\vec{\omega} \times \vec{r}_i$, obtaining

$$T_i = \frac{1}{2} m_i \vec{v}_i \cdot (\vec{\omega} \times \vec{r}_i). \quad (2.7)$$

Writing $\vec{r}_i = (x_i, y_i, z_i)^\top$, $\vec{\omega} = (\omega_x, \omega_y, \omega_z)^\top$, and $\vec{v}_i = (v_{ix}, v_{iy}, v_{iz})^\top$, we compute

$$\vec{\omega} \times \vec{r}_i = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \omega_x & \omega_y & \omega_z \\ x_i & y_i & z_i \end{vmatrix} = \begin{bmatrix} \omega_y z_i - \omega_z y_i \\ \omega_z x_i - \omega_x z_i \\ \omega_x y_i - \omega_y x_i \end{bmatrix},$$

and then the scalar product

$$T_i = \frac{m_i}{2} \begin{bmatrix} v_{ix} & v_{iy} & v_{iz} \end{bmatrix} \begin{bmatrix} \omega_y z_i - \omega_z y_i \\ \omega_z x_i - \omega_x z_i \\ \omega_x y_i - \omega_y x_i \end{bmatrix} = \frac{m_i}{2} \left(v_{ix}(\omega_y z_i - \omega_z y_i) + v_{iy}(\omega_z x_i - \omega_x z_i) + v_{iz}(\omega_x y_i - \omega_y x_i) \right). \quad (2.8)$$

To obtain an equivalent alternative form, we compute separately the cross product $\vec{r}_i \times \vec{v}_i$:

$$\vec{r}_i \times \vec{v}_i = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_i & y_i & z_i \\ v_{ix} & v_{iy} & v_{iz} \end{vmatrix} = \begin{bmatrix} y_i v_{iz} - z_i v_{iy} \\ z_i v_{ix} - x_i v_{iz} \\ x_i v_{iy} - y_i v_{ix} \end{bmatrix}. \quad (2.9)$$

Multiplying by $\vec{\omega}$, we obtain

$$\frac{m_i}{2} \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix} \begin{bmatrix} y_i v_{iz} - z_i v_{iy} \\ z_i v_{ix} - x_i v_{iz} \\ x_i v_{iy} - y_i v_{ix} \end{bmatrix} = \frac{m_i}{2} \left(\omega_x(y_i v_{iz} - z_i v_{iy}) + \omega_y(z_i v_{ix} - x_i v_{iz}) + \omega_z(x_i v_{iy} - y_i v_{ix}) \right), \quad (2.10)$$

which coincides term-by-term with (2.8). Hence,

$$T_i = \frac{1}{2} m_i \vec{v}_i \cdot (\vec{\omega} \times \vec{r}_i) = \frac{1}{2} m_i \vec{\omega} \cdot (\vec{r}_i \times \vec{v}_i) \quad (2.11)$$

(GOLDSTEIN *et al.*, 2002).

The relation between angular momentum and angular velocity can be written as $\vec{L} = m_i(\vec{r}_i \times \vec{v}_i)$ and $\vec{L} = \mathbf{I} \vec{\omega}$ (cf. (2.3) and (2.5)). Separating direction and magnitude of the angular velocity, we define

$$\vec{\omega} = \omega \hat{\vec{n}}, \quad \|\hat{\vec{n}}\| = 1, \quad (2.12)$$

where $\hat{\vec{n}}$ is a unit vector specifying the rotation-axis direction and ω is the magnitude of the angular velocity.

Accordingly, the *scalar moment of inertia* about the axis $\hat{\vec{n}}$ is defined by

$$I_{\hat{n}} \stackrel{\text{def}}{=} m_i \|\hat{\vec{n}} \times \vec{r}_i\|^2 = m_i (\vec{r}_i \times \hat{\vec{n}}) \cdot (\vec{r}_i \times \hat{\vec{n}}), \quad (2.13)$$

i.e., mass times the square of the perpendicular distance to the axis ($\rho = \|\hat{\vec{n}} \times \vec{r}_i\|$; cf. eq. (5.18) in GOLDSTEIN *et al.* (2002)).

Multiplying and dividing (2.13) by ω^2 and using (2.4), we obtain

$$I_{\hat{n}} = m_i \|\hat{\vec{n}} \times \vec{r}_i\|^2 = \frac{m_i}{\omega^2} \|\omega(\hat{\vec{n}} \times \vec{r}_i)\|^2 = \frac{m_i}{\omega^2} \|\vec{\omega} \times \vec{r}_i\|^2 = \frac{m_i}{\omega^2} \|\vec{v}_i\|^2. \quad (2.14)$$

With (2.6) ($T_i = \frac{1}{2} m_i \|\vec{v}_i\|^2$), we directly obtain

$$\boxed{T_i = \frac{1}{2} I_{\hat{n}} \omega^2} \quad \iff \quad \boxed{I_{\hat{n}} = \frac{2 T_i}{\omega^2}}. \quad (2.15)$$

This expression highlights the analogy between mass and moment of inertia: while m measures resistance to linear acceleration (via $\vec{F} = m \vec{a}$), the scalar $I_{\hat{n}}$ measures resistance to angular acceleration about the axis $\hat{\vec{n}}$. The analogy also appears in the kinetic-energy forms: $T = \frac{1}{2} m v^2$ for translational motion and $T' = \frac{1}{2} I_{\hat{n}} \omega^2$ for rotational motion (LEVI, 2014).

2.1.3 Euler's equations

To describe rotational dynamics about a fixed point (or the center of mass), we start from Newton's second law for rotations, which equates the rate of change of the angular momentum \vec{L} to the total external torque \vec{N} in an inertial ("space") frame (GOLDSTEIN *et al.*, 2002):

$$\left(\frac{d\vec{L}}{dt} \right)_{\text{space}} = \vec{N}. \quad (2.16)$$

To work in the body-fixed frame, we again use (2.2), now applied to \vec{L} :

$$\left(\frac{d\vec{L}}{dt} \right)_{\text{space}} = \left(\frac{d\vec{L}}{dt} \right)_{\text{body}} + \vec{\omega} \times \vec{L}.$$

Substituting into (2.16) and omitting the subscript “body” for simplicity, we obtain the angular-momentum balance in the body frame:

$$\dot{\vec{L}} + \vec{\omega} \times \vec{L} = \vec{N}. \quad (2.17)$$

Expanding the cross product $\vec{\omega} \times \vec{L}$, the vector equation (2.17) takes the form

$$\begin{aligned} \dot{L}_x + \omega_y L_z - \omega_z L_y &= N_x, \\ \dot{L}_y + \omega_z L_x - \omega_x L_z &= N_y, \\ \dot{L}_z + \omega_x L_y - \omega_y L_x &= N_z, \end{aligned} \quad (2.18)$$

where (L_x, L_y, L_z) , $(\omega_x, \omega_y, \omega_z)$, and (N_x, N_y, N_z) are the components of \vec{L} , $\vec{\omega}$, and \vec{N} along the body axes.

The fundamental simplification occurs when one chooses a coordinate system aligned with the body’s *principal axes of inertia*. In this system, the inertia tensor is diagonal, $\mathbf{I} = \text{diag}(I_1, I_2, I_3)$, and the relation $\vec{L} = \mathbf{I} \vec{\omega}$ decomposes as $L_k = I_k \omega_k$ for $k = 1, 2, 3$. Substituting these relations into (2.18) and assuming that the principal axes are fixed in the body (so that $\dot{I}_k = 0$), one obtains *Euler’s equations* (GOLDSTEIN *et al.*, 2002):

$$\begin{aligned} I_1 \dot{\omega}_1 + (I_3 - I_2) \omega_2 \omega_3 &= N_1, \\ I_2 \dot{\omega}_2 + (I_1 - I_3) \omega_3 \omega_1 &= N_2, \\ I_3 \dot{\omega}_3 + (I_2 - I_1) \omega_1 \omega_2 &= N_3. \end{aligned} \quad (2.19)$$

These equations will form the core of the rotational part of the 6-DOF formulation used to describe the projectile in flight.

2.1.4 Direction cosines

To describe the orientation of a rigid body, it is convenient to introduce a rotation matrix (attitude matrix, or direction-cosine matrix) relating two orthonormal reference frames: a frame fixed in space and a frame fixed in the body (GOLDSTEIN *et al.*, 2002). Let (x, y, z) be the external system, with unit basis $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$, and (x', y', z') the body-attached system, with basis $\{\mathbf{i}', \mathbf{j}', \mathbf{k}'\}$. The direction cosines are defined by

$$\mathbf{e}'_p \cdot \mathbf{e}_q = \|\mathbf{e}'_p\| \|\mathbf{e}_q\| \cos \theta_{pq} = 1 \cdot 1 \cdot \cos \theta_{pq} = \cos \theta_{pq}, \quad p, q \in \{1, 2, 3\},$$

where $\mathbf{e}_1 = \mathbf{i}$, $\mathbf{e}_2 = \mathbf{j}$, $\mathbf{e}_3 = \mathbf{k}$ and $\mathbf{e}'_1 = \mathbf{i}'$, $\mathbf{e}'_2 = \mathbf{j}'$, $\mathbf{e}'_3 = \mathbf{k}'$.

With this convention, each unit vector of the body frame can be written in terms of

the unit vectors of the external frame:

$$\begin{aligned}\mathbf{i}' &= (\cos \theta_{11}) \mathbf{i} + (\cos \theta_{12}) \mathbf{j} + (\cos \theta_{13}) \mathbf{k}, \\ \mathbf{j}' &= (\cos \theta_{21}) \mathbf{i} + (\cos \theta_{22}) \mathbf{j} + (\cos \theta_{23}) \mathbf{k}, \\ \mathbf{k}' &= (\cos \theta_{31}) \mathbf{i} + (\cos \theta_{32}) \mathbf{j} + (\cos \theta_{33}) \mathbf{k},\end{aligned}\quad (2.20)$$

where, for example, $\cos \theta_{12} = \mathbf{i}' \cdot \mathbf{j}$ is the cosine of the angle between the axes x' and y .

Collecting the nine cosines into a matrix $\mathbf{C} = (\cos \theta_{pq})$, the rows of \mathbf{C} are the components of $\mathbf{i}', \mathbf{j}', \mathbf{k}'$ as seen in the external frame. Since both bases are orthonormal, \mathbf{C} is an orthogonal matrix and satisfies $\mathbf{C}^{-1} = \mathbf{C}^\top$.

Any vector \vec{r} can be written in both coordinate systems:

$$\vec{r} = x \mathbf{i} + y \mathbf{j} + z \mathbf{k} = x' \mathbf{i}' + y' \mathbf{j}' + z' \mathbf{k}'.$$

Projecting \vec{r} onto the primed axes and using (2.20), one finds

$$\begin{aligned}x' &= \vec{r} \cdot \mathbf{i}' = (\cos \theta_{11}) x + (\cos \theta_{12}) y + (\cos \theta_{13}) z, \\ y' &= \vec{r} \cdot \mathbf{j}' = (\cos \theta_{21}) x + (\cos \theta_{22}) y + (\cos \theta_{23}) z, \\ z' &= \vec{r} \cdot \mathbf{k}' = (\cos \theta_{31}) x + (\cos \theta_{32}) y + (\cos \theta_{33}) z,\end{aligned}\quad (2.21)$$

i.e., in matrix form,

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{C} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{C}^\top \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}.$$

In this notation, $\mathbf{i}', \mathbf{j}', \mathbf{k}'$ are unit vectors (body axes), whereas x', y', z' are the components of \vec{r} in that frame; they should not be confused.

2.2 The 6-DOF equations

This section presents the six-degree-of-freedom (6-DOF) formulation for the translational and rotational dynamics of projectiles in external ballistics. This type of modeling is widely recognized in the contemporary literature as a high-fidelity standard, because it integrates, within a single system, the center-of-mass trajectory, the rigid-body attitude, and the acting aerodynamic moments (McCoy, 2012, chap. 9). Reduced versions of this problem were already considered in the first half of the twentieth century, but routine use of full numerical integrations became consolidated only with the rise of postwar digital computation (McCoy, 2012).

Classical 6-DOF formulations are often written in terms of Euler angles. In this work, we instead adopt a parametrization by *direction-cosine matrices* (*direction cosine matrices*, DCM), as presented in Section 2.1.4. This choice avoids kinematic singularities and remains uniformly valid for large angles of attack, dispensing with reparametrizations during maneuvers with large deflections. From a physical standpoint, both descriptions (Euler and

DCM) are equivalent: they are simply different ways of parametrizing the same attitude, provided that the same force and moment fields are used (McCoy, 2012).

In the 6-DOF context, it is possible to obtain trajectory and flight-dynamics predictions with a high degree of realism, *provided that* the input parameters are consistent: aerodynamic force and moment coefficients, inertial properties \mathbf{I} , environmental and firing conditions, and appropriate numerical routines (McCoy, 2012; CARLUCCI and JACOBSON, 2014). Ultimately, the model's fidelity is limited by the quality of these coefficients and by the hypotheses adopted in their experimental determination.

2.2.1 Derivation of the equations

The adopted 6-DOF formulation is based on decomposing the motion of a rigid body into *translation* of the center of mass and *rotation* about the center of mass, as discussed in the rigid-body foundations (Section 2.1.1) and in GOLDSTEIN *et al.* (2002) and LEVI (2014). The specific operational form follows the external-ballistics treatment presented in CARLUCCI and JACOBSON (2014).

Translation of the center of mass. For translation, we use Newton's basic laws. A detailed review of these results is unnecessary here, as they are more elementary material; details can be found in the opening chapters of GOLDSTEIN *et al.* (2002) and LEVI (2014). In the adopted inertial frame, the balance of linear momentum of the center of mass is written, following CARLUCCI and JACOBSON (2014), as

$$m \dot{\vec{v}} = \vec{F}_{\text{aero}} + m \vec{g} + \vec{F}_{\text{Cor}} + \vec{F}_{\text{imp}}, \quad (2.22)$$

where m is the projectile's total mass, \vec{v} is the center-of-mass velocity, \vec{F}_{aero} is the sum of aerodynamic forces, $m \vec{g}$ is the weight, \vec{F}_{Cor} collects terms due to Earth's rotation when relevant, and \vec{F}_{imp} represents possible external impulses (e.g., propulsion). In classical ballistics of unpowered projectiles, one takes $\vec{F}_{\text{imp}} = \vec{0}$.

Rotation about the center of mass. For rotational motion, we employ the vector equation for the body's angular momentum about the center of mass,

$$\dot{\vec{L}} = \vec{M}_{\text{aero}}, \quad (2.23)$$

in which \vec{L} is the angular momentum and \vec{M}_{aero} is the sum of the acting aerodynamic moments (CARLUCCI and JACOBSON, 2014).

As seen in Section 2.1.1, the angular momentum can be written, in a body-fixed principal-axis system, as $\vec{L} = \mathbf{I} \vec{\omega}$, with \mathbf{I} evaluated at the center of mass. For a projectile with rotational symmetry about \mathbf{i}' , degenerate transverse axes, and body basis $\{\mathbf{i}', \mathbf{j}', \mathbf{k}'\}$, one has

$$\mathbf{I}_{\text{body}} = \text{diag}(I_P, I_T, I_T), \quad \vec{\omega} = \omega_1 \mathbf{i}' + \omega_2 \mathbf{j}' + \omega_3 \mathbf{k}',$$

so that

$$\vec{L} = I_p \omega_1 \mathbf{i}' + I_T \omega_2 \mathbf{j}' + I_T \omega_3 \mathbf{k}'. \quad (2.24)$$

The construction of principal axes and the relation $\vec{L} = \mathbf{I} \vec{\omega}$ are discussed in [GOLDSTEIN et al. \(2002, chap. 5\)](#).

Here, I_p denotes the projectile's *polar* moment of inertia about its longitudinal (symmetry) axis, whereas I_t denotes the *transverse* moment of inertia about any axis through the center of mass and perpendicular to that axis.

By the kinematic identity (2.4), for the unit vectors of the body basis one has

$$\frac{d\mathbf{i}'}{dt} = \vec{\omega} \times \mathbf{i}', \quad \text{with} \quad \vec{\omega} = \omega_1 \mathbf{i}' + \omega_2 \mathbf{j}' + \omega_3 \mathbf{k}'.$$

Writing $\omega_1 \rightarrow p$, $\omega_2 \rightarrow q$, $\omega_3 \rightarrow r$ only to facilitate the matrix calculation, we obtain the two expansions:

$$\frac{d\mathbf{i}'}{dt} = \begin{vmatrix} \mathbf{i}' & \mathbf{j}' & \mathbf{k}' \\ p & q & r \\ 1 & 0 & 0 \end{vmatrix} = r \mathbf{j}' - q \mathbf{k}' \iff \frac{d\mathbf{i}'}{dt} = \omega_3 \mathbf{j}' - \omega_2 \mathbf{k}'.$$

$$\mathbf{i}' \times \frac{d\mathbf{i}'}{dt} = \begin{vmatrix} \mathbf{i}' & \mathbf{j}' & \mathbf{k}' \\ 1 & 0 & 0 \\ 0 & r & -q \end{vmatrix} = \mathbf{i}'(0 \cdot (-q) - 0 \cdot r) - \mathbf{j}'(1 \cdot (-q) - 0 \cdot 0) + \mathbf{k}'(1 \cdot r - 0 \cdot 0) = q \mathbf{j}' + r \mathbf{k}' = \omega_2 \mathbf{j}' + \omega_3 \mathbf{k}'.$$

With these identities, the expression (2.24) can be rewritten compactly as

$$\vec{L} = I_p \omega_1 \mathbf{i}' + I_T \left(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt} \right)$$

(2.25)

Normalization by I_T . Following [CARLUCCI and JACOBSON \(2014\)](#), we introduce the vector

$$\vec{h} := \frac{\vec{L}}{I_T} = \frac{I_p}{I_T} \omega_1 \mathbf{i}' + \left(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt} \right). \quad (2.26)$$

Differentiating (2.26) in time and applying the product rule to the cross term,

$$\begin{aligned} \frac{d\vec{h}}{dt} &= \frac{I_p}{I_T} \left(\frac{d\omega_1}{dt} \mathbf{i}' + \omega_1 \frac{d\mathbf{i}'}{dt} \right) + \frac{d\mathbf{i}'}{dt} \times \frac{d\mathbf{i}'}{dt} + \mathbf{i}' \times \frac{d^2\mathbf{i}'}{dt^2} \\ &= \frac{I_p}{I_T} \left(\frac{d\omega_1}{dt} \mathbf{i}' + \omega_1 \frac{d\mathbf{i}'}{dt} \right) + \mathbf{i}' \times \frac{d^2\mathbf{i}'}{dt^2}, \end{aligned} \quad (2.27)$$

since $\frac{d\mathbf{i}'}{dt} \times \frac{d\mathbf{i}'}{dt} = \vec{0}$ (the cross product of a vector with itself).

The expression (2.25) is useful for conceptual analysis; however, for operational pur-

poses we return to (2.23) and use directly the definition (2.26), separating the aerodynamic moments into specific contributions.

Writing (2.23) in terms of \vec{h} ,

$$\frac{d\vec{h}}{dt} = \frac{\vec{M}_S}{I_T} + \frac{\vec{M}_R}{I_T} + \frac{\vec{M}_a}{I_T} + \frac{\vec{M}_{pa}}{I_T} + \frac{\vec{M}_q}{I_T}, \quad (2.28)$$

we obtain the operational form equivalent to equation (8:249) in CARLUCCI and JACOBSON (2014), where each term corresponds to a distinct moment contribution, discussed below.

Remark 1. Starting from (2.26),

$$\vec{h} = \frac{I_p}{I_T} \omega_1 \mathbf{i}' + \left(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt} \right).$$

We compute

$$\begin{aligned} \vec{h} \cdot \mathbf{i}' &= \left(\frac{I_p}{I_T} \omega_1 \mathbf{i}' + \mathbf{i}' \times \frac{d\mathbf{i}'}{dt} \right) \cdot \mathbf{i}' \\ &= \frac{I_p}{I_T} \omega_1 (\mathbf{i}' \cdot \mathbf{i}') + \left(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt} \right) \cdot \mathbf{i}' \\ &= \frac{I_p}{I_T} \omega_1 + 0. \end{aligned}$$

Since \mathbf{i}' is a unit vector, $\mathbf{i}' \cdot \mathbf{i}' = 1$.

Component-wise computation of $(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt}) \cdot \mathbf{i}' = 0$. Write $\mathbf{i}' = (i'_x, i'_y, i'_z)$ and $\frac{d\mathbf{i}'}{dt} = (\dot{i}'_x, \dot{i}'_y, \dot{i}'_z)$.

Then

$$\mathbf{i}' \times \frac{d\mathbf{i}'}{dt} = \begin{vmatrix} \mathbf{e}'_x & \mathbf{e}'_y & \mathbf{e}'_z \\ i'_x & i'_y & i'_z \\ \dot{i}'_x & \dot{i}'_y & \dot{i}'_z \end{vmatrix} = (i'_y \dot{i}'_z - i'_z \dot{i}'_y, \quad i'_z \dot{i}'_x - i'_x \dot{i}'_z, \quad i'_x \dot{i}'_y - i'_y \dot{i}'_x).$$

Hence

$$\begin{aligned} (\mathbf{i}' \times \frac{d\mathbf{i}'}{dt}) \cdot \mathbf{i}' &= i'_x (i'_y \dot{i}'_z - i'_z \dot{i}'_y) + i'_y (i'_z \dot{i}'_x - i'_x \dot{i}'_z) + i'_z (i'_x \dot{i}'_y - i'_y \dot{i}'_x) \\ &= 0, \end{aligned}$$

by term-by-term cancellation. Therefore,

$$\boxed{(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt}) \cdot \mathbf{i}' = 0}.$$

and thus

$$\boxed{\vec{h} \cdot \mathbf{i}' = \frac{I_p}{I_T} \omega_1} \quad (2.29)$$

Remark 2. Starting again from (2.26),

$$\vec{h} = \frac{I_p}{I_T} \omega_1 \mathbf{i}' + \left(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt} \right).$$

We compute

$$\begin{aligned} \vec{h} \times \mathbf{i}' &= \left(\frac{I_p}{I_T} \omega_1 \mathbf{i}' + \mathbf{i}' \times \frac{d\mathbf{i}'}{dt} \right) \times \mathbf{i}' \\ &= \underbrace{\left(\frac{I_p}{I_T} \omega_1 \mathbf{i}' \right) \times \mathbf{i}'}_{=0} + \left(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt} \right) \times \mathbf{i}' \\ &= \left(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt} \right) \times \mathbf{i}'. \end{aligned}$$

Writing $\mathbf{i}' = (i'_x, i'_y, i'_z)$ and $\frac{d\mathbf{i}'}{dt} = (u'_x, u'_y, u'_z)$, first

$$\mathbf{i}' \times \frac{d\mathbf{i}'}{dt} = \begin{vmatrix} \mathbf{e}'_x & \mathbf{e}'_y & \mathbf{e}'_z \\ i'_x & i'_y & i'_z \\ u'_x & u'_y & u'_z \end{vmatrix} = (A'_x, A'_y, A'_z), \quad \begin{cases} A'_x = i'_y u'_z - i'_z u'_y, \\ A'_y = i'_z u'_x - i'_x u'_z, \\ A'_z = i'_x u'_y - i'_y u'_x. \end{cases}$$

Then

$$(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt}) \times \mathbf{i}' = \begin{vmatrix} \mathbf{e}'_x & \mathbf{e}'_y & \mathbf{e}'_z \\ A'_x & A'_y & A'_z \\ i'_x & i'_y & i'_z \end{vmatrix} = (A'_y i'_z - A'_z i'_y, A'_z i'_x - A'_x i'_z, A'_x i'_y - A'_y i'_x) = (u'_x, u'_y, u'_z).$$

Therefore,

$$(\mathbf{i}' \times \frac{d\mathbf{i}'}{dt}) \times \mathbf{i}' = \frac{d\mathbf{i}'}{dt}.$$

We conclude that

$\vec{h} \times \mathbf{i}' = \frac{d\mathbf{i}'}{dt}$

(2.30)

2.2.2 Aerodynamic force and moment coefficients

This subsection does not discuss the experimental procedures used to determine aerodynamic coefficients; it only makes explicit *which* coefficients enter the formulation. Details on wind-tunnel tests and calibration from firing data can be found in CARLUCCI and JACOBSON (2014) and McCOY (2012).

The main objective is to make explicit the link between force and moment coefficients and the rotational terms introduced through the variable \vec{h} . The expressions below are presented in the same form used in CARLUCCI and JACOBSON (2014), with intermediate steps omitted for conciseness.

Aerodynamic forces

Let the flow velocity relative to the projectile be \vec{V} , with $V = \|\vec{V}\|$, reference area S , density ρ , and reference length d (e.g., the diameter). If \mathbf{i}' is the body's polar axis (unit vector) and \vec{h} is defined by (2.26), we use the following forces, in operational form:

(i) **Drag.**

$$\vec{F}_D = -\frac{1}{2} \rho S C_D V \vec{V}$$

(ii) **Lift.**

$$\vec{F}_L = \frac{1}{2} \rho S C_{L\alpha} \left[V^2 \mathbf{i}' - (\vec{V} \cdot \mathbf{i}') \vec{V} \right]$$

(iii) **Magnus.**

$$\vec{F}_M = -\frac{1}{2} \rho S d C_{Np\alpha} \frac{I_T}{I_P} (\vec{h} \cdot \mathbf{i}') (\mathbf{i}' \times \vec{V})$$

(iv) **Pitch damping.**

$$\vec{F}_q = \frac{1}{2} \rho V S d (C_{Nq} + C_{N\dot{\alpha}}) (\vec{h} \times \mathbf{i}')$$

Substituting the forces above into (2.22) and dividing by the mass yields the vector form of the translational acceleration:

$$\begin{aligned} \frac{d\vec{V}}{dt} &= \frac{\rho V S C_D}{2m} \vec{V} + \frac{\rho S C_{L\alpha}}{2m} \left[V^2 \mathbf{i}' - (\vec{V} \cdot \mathbf{i}') \vec{V} \right] - \frac{\rho S d C_{Np\alpha}}{2m} \frac{I_T}{I_P} (\vec{h} \cdot \mathbf{i}') (\mathbf{i}' \times \vec{V}) \\ &\quad + \frac{\rho V S d}{2m} (C_{Nq} + C_{N\dot{\alpha}}) (\vec{h} \times \mathbf{i}') + \vec{g} + \frac{1}{m} \vec{F}_{\text{Cor}}. \end{aligned} \quad (2.31)$$

Aerodynamic moments

Similarly, for aerodynamic moments we adopt S , d , the polar axis \mathbf{i}' , and the vector \vec{h} as in (2.26).

(i) **Spin damping moment.**

$$\vec{M}_S = \frac{1}{2} \rho V S d^2 C_{lp} \frac{I_T}{I_P} (\vec{h} \cdot \mathbf{i}') \mathbf{i}'$$

(ii) Rolling moment.

$$\vec{M}_R = \frac{1}{2} \rho V^2 S d \delta_F C_{l_\delta} \mathbf{i}'$$

(iii) Overturning moment.

$$\vec{M}_\alpha = \frac{1}{2} \rho S d V C_{M\alpha} (\vec{V} \times \mathbf{i}')$$

(iv) Magnus moment.

$$\vec{M}_{p\alpha} = \frac{1}{2} \rho S d^2 C_{M_{p\alpha}} \frac{I_T}{I_P} (\vec{h} \cdot \mathbf{i}') \left[\vec{V} - (\vec{V} \cdot \mathbf{i}') \mathbf{i}' \right]$$

(v) Pitch damping moment.

$$\vec{M}_q = \frac{1}{2} \rho V S d^2 (C_{M_q} + C_{M_{\dot{\alpha}}}) \left[\vec{h} - (\vec{h} \cdot \mathbf{i}') \mathbf{i}' \right]$$

Substituting these contributions into (2.28) yields

$$\begin{aligned} \frac{d\vec{h}}{dt} &= \frac{\rho V S d^2 C_{l_p}}{2 I_p} (\vec{h} \cdot \mathbf{i}') \mathbf{i}' + \frac{\rho V^2 S d \delta_F C_{l_\delta}}{2 I_T} \mathbf{i}' + \frac{\rho S d V C_{M_\alpha}}{2 I_T} (\vec{V} \times \mathbf{i}') \\ &+ \frac{\rho S d^2 C_{M_{p\alpha}}}{2 I_p} (\vec{h} \cdot \mathbf{i}') \left[\vec{V} - (\vec{V} \cdot \mathbf{i}') \mathbf{i}' \right] + \frac{\rho V S d^2 (C_{M_q} + C_{M_{\dot{\alpha}}})}{2 I_T} \left[\vec{h} - (\vec{h} \cdot \mathbf{i}') \mathbf{i}' \right]. \end{aligned} \quad (2.32)$$

2.2.3 Auxiliary identities

With the expressions (2.31) and (2.32), it is convenient to state some auxiliary component relations that will be used in the derivation of the final six scalar ODEs. From this point on, we adopt the inertial basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ (to avoid notational conflicts with the body basis $\{\mathbf{i}', \mathbf{j}', \mathbf{k}'\}$) and write:

$$\vec{h} = h_1 \mathbf{e}_1 + h_2 \mathbf{e}_2 + h_3 \mathbf{e}_3, \quad (2.33)$$

$$\mathbf{i}' = i'_1 \mathbf{e}_1 + i'_2 \mathbf{e}_2 + i'_3 \mathbf{e}_3, \quad (2.34)$$

$$\vec{V} = V_1 \mathbf{e}_1 + V_2 \mathbf{e}_2 + V_3 \mathbf{e}_3, \quad (\text{body velocity in the inertial frame}) \quad (2.35)$$

$$\vec{W} = W_1 \mathbf{e}_1 + W_2 \mathbf{e}_2 + W_3 \mathbf{e}_3, \quad (\text{wind velocity in the inertial frame}) \quad (2.36)$$

Define the *body-relative airspeed*

$$\vec{v} := \vec{V} - \vec{W} \Rightarrow \vec{v} = v_1 \mathbf{e}_1 + v_2 \mathbf{e}_2 + v_3 \mathbf{e}_3, \quad v_k = V_k - W_k,$$

with magnitude

$$v := \|\vec{v}\| = \sqrt{v_1^2 + v_2^2 + v_3^2}.$$

The identities below follow from elementary vector algebra and/or from using (2.29) and (2.30). They are recorded only in final form, similarly to the organization in CARLUCCI and JACOBSON (2014).

(i) Dot product $(\vec{v} \cdot \mathbf{i}')$.

$$(\vec{v} \cdot \mathbf{i}') = v_1 i'_1 + v_2 i'_2 + v_3 i'_3.$$

(ii) Term $(\vec{v} \cdot \mathbf{i}') \vec{v}$.

$$(\vec{v} \cdot \mathbf{i}') \vec{v} = (v_1^2 i'_1 + v_1 v_2 i'_2 + v_1 v_3 i'_3) \mathbf{e}_1 + (v_2 v_1 i'_1 + v_2^2 i'_2 + v_2 v_3 i'_3) \mathbf{e}_2 + (v_3 v_1 i'_1 + v_3 v_2 i'_2 + v_3^2 i'_3) \mathbf{e}_3.$$

(iii) Dot product $(\vec{h} \cdot \mathbf{i}')$.

$$(\vec{h} \cdot \mathbf{i}') = h_1 i'_1 + h_2 i'_2 + h_3 i'_3 = \frac{I_p}{I_T} \omega_1 \quad (\text{by (2.29)}).$$

(iv) Cross product $\mathbf{i}' \times \vec{v}$.

$$\mathbf{i}' \times \vec{v} = (i'_2 v_3 - i'_3 v_2) \mathbf{e}_1 + (i'_3 v_1 - i'_1 v_3) \mathbf{e}_2 + (i'_1 v_2 - i'_2 v_1) \mathbf{e}_3.$$

(v) Term $(\vec{h} \cdot \mathbf{i}') (\mathbf{i}' \times \vec{v})$.

$$(\vec{h} \cdot \mathbf{i}') (\mathbf{i}' \times \vec{v}) = \frac{I_p}{I_T} \omega_1 \left[(i'_2 v_3 - i'_3 v_2) \mathbf{e}_1 + (i'_3 v_1 - i'_1 v_3) \mathbf{e}_2 + (i'_1 v_2 - i'_2 v_1) \mathbf{e}_3 \right].$$

(vi) Term $(\vec{h} \cdot \mathbf{i}') \mathbf{i}'$.

$$(\vec{h} \cdot \mathbf{i}') \mathbf{i}' = \frac{I_p}{I_T} \omega_1 (i'_1 \mathbf{e}_1 + i'_2 \mathbf{e}_2 + i'_3 \mathbf{e}_3).$$

(vii) Cosine of the total angle of attack α_t .

$$\cos \alpha_t = \frac{\vec{v} \cdot \mathbf{i}'}{v} = \frac{v_1 i'_1 + v_2 i'_2 + v_3 i'_3}{\sqrt{v_1^2 + v_2^2 + v_3^2}}.$$

Remark. In classical external-ballistics terminology, the “total angle of attack” α_t (also called the total yaw angle) is the *magnitude* of the misalignment vector between the projectile’s symmetry axis and its velocity relative to the air, combining simultaneously the vertical component (α , vertical) and the lateral component (β , horizontal).

(viii) Term $(\vec{v} \cdot \mathbf{i}') \mathbf{i}'$.

$$(\vec{v} \cdot \mathbf{i}') \mathbf{i}' = (v_1 i'_1 + v_2 i'_2 + v_3 i'_3) \mathbf{e}_1 + (v_1 i'_1 + v_2 i'_2 + v_3 i'_3) \mathbf{e}_2 + (v_1 i'_1 + v_2 i'_2 + v_3 i'_3) \mathbf{e}_3.$$

2.2.4 The six scalar equations

With the auxiliary relations of Section 2.2.3 in hand, we perform the appropriate substitutions into (2.31) and (2.32). The full algebraic expansion can be found in CARLUCCI and JACOBSON (2014, chap. 8); here we record the final expressions already organized along the inertial basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, with $\vec{v} = \vec{V} - \vec{W}$ and $v = \|\vec{v}\|$. The result is a system of six ordinary differential equations equivalent to equations (8.275)–(8.280) of the reference text.

$$\begin{aligned} \frac{dV_1}{dt} = & -\frac{\rho v S C_D}{2m} v_1 + \frac{\rho S C_{L\alpha}}{2m} [v^2 i'_1 - v v_1 \cos \alpha_t] - \frac{\rho S d C_{Np\alpha} \omega_1}{2m} (v_3 i'_2 - v_2 i'_3) \\ & + \frac{\rho v S d (C_{Nq} + C_{N\dot{\alpha}})}{2m} (h_2 i'_3 - h_3 i'_2) + g_1 + \Lambda_1 \end{aligned} \quad (2.37)$$

$$\begin{aligned} \frac{dV_2}{dt} = & -\frac{\rho v S C_D}{2m} v_2 + \frac{\rho S C_{L\alpha}}{2m} [v^2 i'_2 - v v_2 \cos \alpha_t] - \frac{\rho S d C_{Np\alpha} \omega_1}{2m} (v_1 i'_3 - v_3 i'_1) \\ & + \frac{\rho v S d (C_{Nq} + C_{N\dot{\alpha}})}{2m} (h_3 i'_1 - h_1 i'_3) + g_2 + \Lambda_2 \end{aligned} \quad (2.38)$$

$$\begin{aligned} \frac{dV_3}{dt} = & -\frac{\rho v S C_D}{2m} v_3 + \frac{\rho S C_{L\alpha}}{2m} [v^2 i'_3 - v v_3 \cos \alpha_t] - \frac{\rho S d C_{Np\alpha} \omega_1}{2m} (v_2 i'_1 - v_1 i'_2) \\ & + \frac{\rho v S d (C_{Nq} + C_{N\dot{\alpha}})}{2m} (h_1 i'_2 - h_2 i'_1) + g_3 + \Lambda_3 \end{aligned} \quad (2.39)$$

$$\begin{aligned} \frac{dh_1}{dt} = & \frac{\rho v S d^2 C_{lp}}{2I_T} \omega_1 i'_1 + \frac{\rho v^2 S d \delta_F C_{l\delta}}{2I_T} i'_1 + \frac{\rho v S d C_{M\alpha}}{2I_T} (v_2 i'_3 - v_3 i'_2) \\ & + \frac{\rho S d^2 C_{M_{p\alpha}}}{2I_T} \omega_1 [v_1 - v i'_1 \cos \alpha_t] + \frac{\rho v S d^2 (C_{Mq} + C_{M\dot{\alpha}})}{2I_T} \left(h_1 - \left(\frac{I_p}{I_T} \omega_1 \right) i'_1 \right) \end{aligned} \quad (2.40)$$

$$\begin{aligned} \frac{dh_2}{dt} = & \frac{\rho v S d^2 C_{lp}}{2I_T} \omega_1 i'_2 + \frac{\rho v^2 S d \delta_F C_{l\delta}}{2I_T} i'_2 + \frac{\rho v S d C_{M\alpha}}{2I_T} (v_3 i'_1 - v_1 i'_3) \\ & + \frac{\rho S d^2 C_{M_{p\alpha}}}{2I_T} \omega_1 [v_2 - v i'_2 \cos \alpha_t] + \frac{\rho v S d^2 (C_{Mq} + C_{M\dot{\alpha}})}{2I_T} \left(h_2 - \left(\frac{I_p}{I_T} \omega_1 \right) i'_2 \right) \end{aligned} \quad (2.41)$$

$$\boxed{\begin{aligned}\frac{dh_3}{dt} = & \frac{\rho v S d^2 C_{lp}}{2I_T} \omega_1 i'_3 + \frac{\rho v^2 S d \delta_F C_{l\delta}}{2I_T} i'_3 + \frac{\rho v S d C_{M\alpha}}{2I_T} (v_1 i'_2 - v_2 i'_1) \\ & + \frac{\rho S d^2 C_{M\rho\alpha}}{2I_T} \omega_1 [v_3 - v i'_3 \cos \alpha_t] + \frac{\rho v S d^2 (C_{Mq} + C_{M\dot{\alpha}})}{2I_T} \left(h_3 - \left(\frac{I_p}{I_T} \omega_1 \right) i'_3 \right)\end{aligned}} \quad (2.42)$$

2.2.5 General modeling considerations

To reconstruct the center-of-mass position in inertial space, in addition to the six ODEs above, we integrate the velocity components:

$$x(t) = x_0 + \int_0^t V_1(\tau) d\tau, \quad (2.43)$$

$$y(t) = y_0 + \int_0^t V_2(\tau) d\tau, \quad (2.44)$$

$$z(t) = z_0 + \int_0^t V_3(\tau) d\tau. \quad (2.45)$$

Coriolis force. The 6-DOF formulation allows inclusion of Coriolis and Earth-rotation terms (see, e.g., CARLUCCI and JACOBSON (2014) and MCCOY (2012)). In this work, we set $\vec{F}_{\text{Cor}} = \vec{0}$, which is, as a first approximation, equivalent to operating at moderate latitudes.

Earth curvature and variation of gravity. Effects of Earth curvature and variation of g with altitude can be incorporated through geometric corrections and by modeling $g = g(y)$ (McCoy, 2012; CARLUCCI and JACOBSON, 2014). In particular, it is reported that, for ranges on the order of 1,8 km, curvature can introduce measurable differences in the trajectory. To keep the complexity compatible with the goals of this work, we adopt the approximations: (i) locally flat Earth over the region of interest and (ii) constant gravity $\vec{g} = (0, -g, 0)^T$. These hypotheses reduce accuracy at long ranges and high altitudes, but are adequate for the level of the present work.

2.2.6 Initial conditions

With the equations established and the general modeling hypotheses defined, we now specify the initial conditions required for numerical integration of the model. We collect only the quantities that enter explicitly in the initial step of the integrator.

Vector notation.

$$\vec{r}(0) = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}, \quad \vec{V}(0) = \begin{bmatrix} V_{1,0} \\ V_{2,0} \\ V_{3,0} \end{bmatrix}, \quad \vec{W}(0) = \begin{bmatrix} W_{1,0} \\ W_{2,0} \\ W_{3,0} \end{bmatrix}, \quad \vec{v}(0) = \vec{V}(0) - \vec{W}(0).$$

Aiming angles. Following the reference text in external ballistics (McCoy, 2012, chap. 9), we denote by ϕ_0 and θ_0 the *angles of departure* of the muzzle velocity at muzzle exit: ϕ_0 is the *vertical angle of departure* (*vertical angle of departure*), positive upward, and θ_0 is the *horizontal angle of departure* relative to the 1–2 plane (*horizontal angle of departure relative to the 1–2 plane*), positive to the right when looking in the firing direction. These two angles determine the direction of the initial center-of-mass velocity through (2.46).

The parameters α_0 and β_0 are, in turn, the small initial aerodynamic pitch (*pitch*) and yaw (*yaw*) angles of the projectile relative to the departure direction defined by (ϕ_0, θ_0) .

Initial velocity. The initial velocity $\vec{V}(0)$ follows directly from the aiming geometry defined by (θ_0, ϕ_0) . Decomposing the firing unit vector into Cartesian components and multiplying by V_0 ,

$$\vec{V}(0) = V_0 \begin{bmatrix} \cos \phi_0 \cos \theta_0 \\ \sin \phi_0 \cos \theta_0 \\ \sin \theta_0 \end{bmatrix}. \quad (2.46)$$

Body basis at muzzle exit. The aerodynamic misalignments (α_0, β_0) are added to the aiming angles to define the projectile-attached basis at the initial instant. The body's polar axis is given by

$$\mathbf{i}'_0 = \begin{bmatrix} \cos(\phi_0 + \alpha_0) \cos(\theta_0 + \beta_0) \\ \sin(\phi_0 + \alpha_0) \cos(\theta_0 + \beta_0) \\ \sin(\theta_0 + \beta_0) \end{bmatrix}. \quad (2.47)$$

To complete the orthonormal basis $\{\mathbf{i}'_0, \mathbf{j}'_0, \mathbf{k}'_0\}$, we introduce a normalization factor ensuring $\|\mathbf{j}'_0\| = \|\mathbf{k}'_0\| = 1$. Define

$$Q := \sin^2(\theta_0 + \beta_0) + \cos^2(\theta_0 + \beta_0) \cos^2(\phi_0 + \alpha_0), \quad (2.48)$$

and obtain

$$\mathbf{j}'_0 = \frac{1}{\sqrt{Q}} \begin{bmatrix} -\sin(\phi_0 + \alpha_0) \cos(\phi_0 + \alpha_0) \cos^2(\theta_0 + \beta_0) \\ \left(\cos^2(\phi_0 + \alpha_0) \cos^2(\theta_0 + \beta_0) + \sin^2(\theta_0 + \beta_0) \right) \\ -\sin(\phi_0 + \alpha_0) \cos(\theta_0 + \beta_0) \sin(\theta_0 + \beta_0) \end{bmatrix}. \quad (2.49)$$

$$\mathbf{k}'_0 = \frac{1}{\sqrt{Q}} \begin{bmatrix} -\sin(\theta_0 + \beta_0) \\ 0 \\ \cos(\phi_0 + \alpha_0) \cos(\theta_0 + \beta_0) \end{bmatrix}. \quad (2.50)$$

Let

$$\vec{\omega}(0) = v_{i0} \mathbf{i}'_0 + v_{j0} \mathbf{j}'_0 + v_{k0} \mathbf{k}'_0$$

be the projectile's initial angular velocity. By the kinematic identity (2.4),

$$\frac{d\mathbf{i}'_0}{dt} = \vec{\omega}(0) \times \mathbf{i}'_0 = v_{k0} \mathbf{j}'_0 - v_{j0} \mathbf{k}'_0. \quad (2.51)$$

Using (2.49)–(2.50) with the normalization Q of (2.48) and projecting onto $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, we obtain:

$$\dot{i}'_{1,0} = \frac{1}{\sqrt{Q}} \left[v_{j0} \sin(\theta_0 + \beta_0) - v_{k0} \cos^2(\theta_0 + \beta_0) \sin(\phi_0 + \alpha_0) \cos(\phi_0 + \alpha_0) \right], \quad (2.52)$$

$$\dot{i}'_{2,0} = \frac{v_{k0}}{\sqrt{Q}} \left[\cos^2(\phi_0 + \alpha_0) \cos^2(\theta_0 + \beta_0) + \sin^2(\theta_0 + \beta_0) \right], \quad (2.53)$$

$$\dot{i}'_{3,0} = \frac{1}{\sqrt{Q}} \left[-v_{j0} \cos(\theta_0 + \beta_0) \cos(\phi_0 + \alpha_0) - v_{k0} \sin(\phi_0 + \alpha_0) \cos(\theta_0 + \beta_0) \sin(\theta_0 + \beta_0) \right]. \quad (2.54)$$

Finally, following the same pattern as (2.25), we write the vector $\vec{h}(0)$ as a function of the initial components of \mathbf{i}'_0 and of their derivatives:

$$\vec{h}(0) = \begin{bmatrix} h_{1,0} \\ h_{2,0} \\ h_{3,0} \end{bmatrix} = \begin{bmatrix} \frac{I_P}{I_T} \omega_{1,0} i'_{1,0} + i'_{2,0} \dot{i}'_{3,0} - i'_{3,0} \dot{i}'_{2,0} \\ \frac{I_P}{I_T} \omega_{1,0} i'_{2,0} + i'_{3,0} \dot{i}'_{1,0} - i'_{1,0} \dot{i}'_{3,0} \\ \frac{I_P}{I_T} \omega_{1,0} i'_{3,0} + i'_{1,0} \dot{i}'_{2,0} - i'_{2,0} \dot{i}'_{1,0} \end{bmatrix} \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{bmatrix}. \quad (2.55)$$

Remark (axial spin-rate estimate). The initial axial spin rate can be estimated by

$$\omega_{1,0} = \frac{2\pi V_0}{n d}, \quad (2.56)$$

where V_0 is the muzzle velocity, d is the projectile diameter (caliber), and n is the number of calibers per turn.

2.3 Statistical model and Monte Carlo estimation

To use the cost function defined in the introduction, it is necessary, first, to estimate the hit probabilities associated with each firing configuration. The adopted strategy is to use Monte Carlo simulations to approximate these probabilities numerically (Ross, 2014; Kroese *et al.*, 2011).

Consider a set of m firing configurations, indexed by $i = 1, \dots, m$ (for example, combinations of elevation and target range). For each configuration i , we perform K independent simulations in which the initial firing parameters (elevation and azimuth) are randomly perturbed according to the adopted error model. In each simulation we record an indi-

cator variable

$$Y_{i,k} = \begin{cases} 1, & \text{if the shot hits the target in simulation } k, \\ 0, & \text{otherwise,} \end{cases} \quad k = 1, \dots, K. \quad (2.57)$$

Under the assumptions that, for a fixed configuration i , the hit probability is constant and the simulations are independent, we have

$$Y_{i,k} \sim \text{Bernoulli}(p_i), \quad k = 1, \dots, K, \quad (2.58)$$

where p_i denotes the *true* hit probability for configuration i . The number of hits in that configuration is

$$S_i := \sum_{k=1}^K Y_{i,k}, \quad (2.59)$$

and, since the sum of Bernoulli variables is binomial, we have

$$S_i \sim \text{Binomial}(K, p_i). \quad (2.60)$$

The natural estimator of the hit probability at i is the sample hit proportion,

$$\hat{p}_i := \frac{S_i}{K} = \frac{1}{K} \sum_{k=1}^K Y_{i,k}, \quad (2.61)$$

which is unbiased and has variance

$$\mathbb{E}[\hat{p}_i] = p_i, \quad \text{Var}(\hat{p}_i) = \frac{p_i(1 - p_i)}{K} \quad (2.62)$$

(Ross, 2014). For sufficiently large K , the distribution of \hat{p}_i can be approximated by a normal distribution, by virtue of the central limit theorem and the normal approximation to the binomial. Thus,

$$\hat{p}_i \stackrel{\text{d}}{\sim} \mathcal{N}\left(p_i, \frac{p_i(1 - p_i)}{K}\right). \quad (2.63)$$

In practice, one replaces p_i by \hat{p}_i in the variance, which leads to the approximate $1 - \alpha$ confidence interval for each individual hit probability:

$$\hat{p}_i \pm z_{1-\alpha/2} \sqrt{\frac{\hat{p}_i(1 - \hat{p}_i)}{K}}, \quad i = 1, \dots, m, \quad (2.64)$$

where $z_{1-\alpha/2}$ is the corresponding quantile of the standard normal distribution. Repeating this procedure for all configurations i yields a set of estimates $\hat{p}_1, \dots, \hat{p}_m$ together with their confidence intervals.

In summary, for each firing configuration i one obtains a point estimate \hat{p}_i of the hit probability, as well as an approximate $1 - \alpha$ confidence interval given by (2.64). These quantities are then used in the cost function defined in the Introduction:

$$\hat{C} = C(\hat{p}_1, \dots, \hat{p}_m),$$

in which the *true* hit probability at each position is replaced by the corresponding Monte Carlo estimate.

Chapter 3

Methodology

3.1 Physical configuration of the problem

This section describes the choice of the naval projectile, the aerodynamic coefficients, and the inertial parameters used in the 6-DOF model presented in Chapter 2, as well as the characterization of the firing platform and the target employed in the simulations.

3.1.1 Selection of the projectile and aerodynamic coefficients

For the 6-DOF equations of motion to produce physically plausible trajectories, it is necessary to have aerodynamic coefficients and initial conditions that are reasonably representative of a real projectile. With this objective, searches were conducted in public-domain technical reports, prioritizing naval ammunition for which coefficient tables as a function of Mach number and total angle of attack were available.

Among the identified documents, a technical report from Picatinny Arsenal stands out, presenting the computer program SPIN-73, developed to update the SPINNER code for computing ballistic coefficients of spin-stabilized projectiles from body geometry (WHYTE, 1973). This report includes tables of aerodynamic coefficients for several projectiles, among which are two 5-inch naval projectiles (5"/38 and 5"/54). Due to the limited quality of the digitization, only the table corresponding to the 5"/38 projectile proved sufficiently legible for numerical digitization. Thus, this projectile was adopted as the basis for the present work.

The tabulated coefficient values were manually copied into a spreadsheet from the figures in the report, rounded to two decimal places, as shown in Table 3.1.

The dataset obtained for the 5"/38 projectile includes force and moment coefficients in the body-fixed axis system, as a function of Mach number and total angle of attack. The SPIN-73 tables for the 5"/38 projectile provide:

- axial force C_X and normal force $C_{N\alpha}$ coefficients, as well as terms associated with the Magnus force;
- pitching-moment, pitch-damping, spin-damping, and Magnus-moment coefficients.

By construction, there is no fin-induced moment coefficient, since the projectile considered is a body of revolution without fins; in this case, the fin-induced roll moment is zero when the angle is zero (McCoy, 2012). In addition, the force coefficients associated with pitch damping (“pitch damping force”) are not explicitly provided as an isolated quantity in the table. In several 6-DOF trajectory studies oriented toward global dispersion analysis, the focus falls on modeling pitching moments and spin damping, and explicit inclusion of the pitch-damping force component is less common (McCoy, 2012). Following this simplified practice, this force contribution was neglected, while the pitch-damping moment was retained to ensure dynamic stability.

Another relevant aspect is that the coefficients tabulated in WHYTE (1973) are defined in the body axis system, whereas the 6-DOF model developed in Section 2.2.4 is formulated in terms of drag C_D and lift $C_{L\alpha}$ coefficients in the axis system aligned with the projectile’s relative velocity. To reconcile these definitions, standard geometric transformations between body axes and trajectory axes were used, which, in terms of the total angle of attack α_t , can be written as

$$C_D = C_{N\alpha} \sin^2 \alpha_t - C_X \cos \alpha_t, \quad C_{L\alpha} = C_{N\alpha} \cos \alpha_t + C_X. \quad (3.1)$$

These expressions are consistent with the relations presented in McCoy (2012), and were implemented directly in the simulation code.

Finally, some columns of the SPIN-73 table present coefficients decomposed into sinusoidal components of the total angle of attack (for example, terms proportional to $\sin \alpha_t$, $\sin^2 \alpha_t$, and $\sin^3 \alpha_t$). In the developed program, the effective coefficient used at each time step is reconstructed from the appropriate linear combination of these components, according to the parametrization indicated in the report itself (WHYTE, 1973) and in McCoy (2012).

Mach	CX0	CX2	CNA	CMA	CPN	CYP	CNPA	CNPA3	CNPA5	CPF1	CPF5	CNPA-5	CMQ	CLP
0.01	0.17	2.64	1.80	3.40	0.77	-0.77	-0.39	89.34	-855.95	2.21	3.23	0.42	-9.42	-0.03
0.60	0.17	2.64	1.80	3.51	0.76	-0.77	-0.39	89.34	-855.95	2.21	3.23	0.42	-9.42	-0.03
0.80	0.17	3.17	1.82	3.69	0.69	-0.77	-0.24	81.18	-774.34	2.40	3.35	0.49	-9.42	-0.03
0.90	0.19	3.70	1.87	3.86	0.63	-0.86	0.03	59.25	-555.01	2.75	3.37	0.56	-11.75	-0.02
0.95	0.24	4.10	2.04	4.14	0.68	-1.09	0.28	46.50	-427.51	2.97	3.36	0.71	-14.26	-0.02
1.00	0.37	4.63	2.14	3.93	0.85	-1.00	0.38	31.71	-279.63	3.09	3.38	0.67	-16.32	-0.02
1.05	0.45	5.15	2.22	3.74	1.03	-0.91	0.45	18.96	157.13	3.21	3.40	0.63	-17.98	-0.02
1.10	0.45	5.71	2.27	3.71	1.07	-0.86	0.48	13.35	-96.03	3.27	3.41	0.61	-19.36	-0.02
1.20	0.45	6.28	2.35	3.65	1.16	-0.77	0.48	9.38	-56.25	3.33	3.45	0.56	-20.60	-0.02
1.35	0.44	5.69	2.47	3.49	1.30	-0.77	0.49	7.74	-39.93	3.35	3.45	0.57	-20.12	-0.02
1.50	0.42	5.09	2.61	3.57	1.34	-0.77	0.50	6.93	-31.77	3.36	3.45	0.57	-18.99	-0.02
1.75	0.40	4.52	2.71	3.41	1.45	-0.77	0.51	6.11	-23.61	3.37	3.45	0.57	-18.99	-0.02
2.00	0.37	3.94	2.83	3.31	1.53	-0.77	0.52	5.30	-15.45	3.38	3.45	0.57	-18.99	-0.02
2.50	0.33	3.22	2.95	3.21	1.62	-0.77	0.52	4.48	-7.29	3.39	3.45	0.57	-18.99	-0.02
3.00	0.29	2.67	2.93	3.07	1.66	-0.77	0.53	3.66	0.87	3.40	3.45	0.57	-18.99	-0.02
4.00	0.25	2.19	2.83	3.06	1.63	-0.77	0.53	3.66	0.87	3.40	3.45	0.57	-18.99	-0.02
5.00	0.23	1.71	2.73	3.04	1.60	-0.77	0.53	3.66	0.87	3.40	3.45	0.57	-18.99	-0.02

Table 3.1: Dimensionless aerodynamic coefficients of the 5"/38 naval projectile, extracted from WHYTE (1973), as a function of Mach number.

3.1.2 Inertial parameters, initial conditions, and ammunition cost

In the report containing the aerodynamic coefficient tables for the 5"/38 projectile, neither the mass nor the projectile moments of inertia are visible. To address this gap, we resorted to the study by [HASELTINE \(1969\)](#), which analyzes the yaw motion of the 5-inch naval projectile Mk 41 (5"/54) by means of *yaw* waves. Among the experimental data presented are mass and moments of inertia of 5-inch projectiles, with values on the order of

$$69.10 \text{ lb}, \quad 240 \text{ lb} \cdot \text{in}^2 \quad (\text{axial axis}), \quad 2619 \text{ lb} \cdot \text{in}^2 \quad (\text{transverse axes}).$$

These values were converted to SI units during code execution.

As a working hypothesis, it was assumed that these magnitudes are representative of the order of magnitude of the inertial parameters of a 5"/38 naval projectile.

To construct the cost function presented in equation 1.1, it is necessary to specify the cost per round. An estimate of the unit cost of 5"/38 ammunition was obtained from a U.S. General Accounting Office report on Department of Defense ammunition budgets, which indicates values on the order of 860 dollars per projectile (in then-year dollars) ([GAO, 1990](#)). This value was updated to current dollars using an inflation calculator based on U.S. Bureau of Labor Statistics data ([US INFLATION CALCULATOR, 2025](#)), and the result was adopted as the reference cost c per shot in the cost model, about 2 thousand current dollars.

Regarding muzzle velocity, the literature on the 5"/38 gun indicates typical initial velocities around $V_0 \approx 792 \text{ m/s}$ ([DiGIULIAN, 2012](#)), whereas values near $V_0 \approx 807 \text{ m/s}$ are cited for the 5"/54 gun ([WIKIPEDIA CONTRIBUTORS, 2025](#)). In this work, since the adopted mass and moments of inertia come from a 5"/54 projectile while the aerodynamic coefficients were extracted from a 5"/38 projectile, the muzzle velocity was fixed at $V_0 = 807 \text{ m/s}$ as a personal choice.

In addition to the axial spin component, the 6-DOF model requires initial values for the transverse components of angular velocity, associated with the initial yaw and pitch. As discussed by [McCoy \(2012\)](#), in applications aimed at detailed reproduction of experimental data these values are usually chosen so that the yaw and pitch angles assume values compatible with measurements at the first point of maximum yaw along the trajectory. In this work, the simplified choice of 5 rad/s was adopted for both, merely to ensure a moderate excitation of the yaw and pitch modes. It is acknowledged that this choice was not systematically calibrated and constitutes more of a personal choice than a fine tuning to real firing regimes.

The external environment was modeled in a simplified manner. A locally flat Earth is assumed, with no effects of planetary rotation (absence of Coriolis terms) and no relative motion of the air with respect to the firing platform (zero wind).

Air density is taken as constant and equal to the typical sea-level value in the standard atmosphere,

$$\rho = 1,225 \text{ kg/m}^3,$$

and gravitational acceleration is taken as

$$g = 9,81 \text{ m/s}^2.$$

These values are used both in the formulation of the aerodynamic forces and in the weight term in the equations of motion.

For the computation of Mach number, a constant speed of sound is also assumed,

$$a = 340 \text{ m/s},$$

so that the Mach number is simply $M = \|\vec{v}_{\text{rel}}\|/a$, where \vec{v}_{rel} is the projectile velocity relative to the air. The variation of the speed of sound with temperature or altitude is therefore neglected. This assumption reduces computational cost and is sufficient for the level of detail required in this Undergraduate Thesis.

3.1.3 Firing platform and target

To contextualize the results of the 6-DOF model in a concrete operational scenario, a blue-water escort equipped with a 5-inch/38 main gun was chosen as the reference platform. A representative example is the *Garcia*-class, built for the U.S. Navy in the 1960s and equipped with two Mk 30 5"/38 guns ([SUPERVISOR OF SHIPBUILDING \(SUPSHIP\), SEATTLE, WASHINGTON, 1965](#)). Four ships of this class were later transferred to the Brazilian Navy, where they served as *Pará*-class destroyers until the last unit was decommissioned in 2008 ([PODER NAVAL, 2023](#)).

Analyses published in *Naval History* indicate construction costs on the order of 28 million dollars per ship for the *Garcia* and *Brooke* classes in the mid-1960s ([U.S. NAVAL INSTITUTE, 1989](#)). Updating this figure to current prices using the same inflation calculator already cited ([US INFLATION CALCULATOR, 2025](#)) yields an equivalent cost of approximately 289 million dollars. This value is used as an approximation of the surface platform cost in the cost model.

As the target, the Ukrainian naval drone *Sea Baby* was adopted, an unmanned surface vehicle (USV) employed in the Russo-Ukrainian War. According to technical compilations on maritime drones ([SUTTON, 2024](#)), the *Sea Baby* has a length on the order of 6 m and a beam close to 2 m, resulting in a typical projected area of approximately 12 m². The same set of sources indicates an explosive payload capacity of about 900 kg and operational use in long-range attacks against naval and infrastructure targets.

As for cost, reporting on crowdfunding campaigns for producing *Sea Baby* drones indicates unit costs on the order of 8.5 million hryvnias per unit ([THE KYIV INDEPENDENT, 2024](#)), which corresponds to approximately 220 thousand dollars at the exchange rate prevailing at the time. This value was adopted as an estimate of the unit cost of the target in the cost model developed in this work.

These parameters provide the basis for modeling the fire decision problem. The representative weapon is a 5"/38 gun installed on an escort with a cost comparable to that of a *Garcia*-class frigate, while the target is an explosive USV of the *Sea Baby* type.

Initial position of the gun on the platform

In the inertial coordinate system adopted in this work, where the y -axis is vertical (positive upward) and the $x-z$ plane is horizontal, the position of the geometric center of the gun muzzle at the firing instant was set to

$$(x_0, y_0, z_0) = (0, 10, 0) \text{ m.}$$

The value $y_0 = 10$ m was chosen as a reasonable approximation for the installation height of the main gun on an escort of similar size to the *Garcia* class. General arrangement drawings of this class indicate that the bow gun is located at a height on the order of 7 m above the waterline (**SUPERVISOR OF SHIPBUILDING (SUPSHIP), SEATTLE, WASHINGTON, 1965**). The 10 m value used in the simulations therefore represents a slightly higher idealization, compatible with design variations among different ships and with the simplification of considering the hull rigidly supported in calm seas.

The target (*Sea Baby*) is modeled with its geometric center located at the sea surface, that is, with initial vertical coordinate $y = 0$. Thus, the initial position condition of the projectile in the 6-DOF model corresponds to a shot fired from a fixed platform located 10 m above the free surface.

In the following sections, the Monte Carlo simulation scheme, the estimation of hit probabilities, and the construction of the cost function associated with different engagement policies are described in detail.

3.2 Computational implementation of the simulation

The simulations were implemented in Python 3, using libraries widely employed in scientific computing, such as `numpy` and `math` for low-level numerical operations, `scipy` for differential-equation integration and interpolation, `pandas` for reading and manipulating tabular data, `matplotlib` for generating figures, and `dataclasses` for the structured definition of parameter classes.

The source code was developed with the support of generative AI tools (in particular, programming assistants such as ChatGPT and Claude), mainly used to refactor the code into an object-oriented style, assist with plotting, and standardize module organization. The physical and mathematical logic of the model, as well as all parameter choices, were defined by the author and manually verified.

In general, the implementation can be decomposed into three main components: (i) preparation of aerodynamic coefficients and physical parameters, (ii) the core numerical integration of the equations of motion, and (iii) the layer of numerical experiments responsible for the Monte Carlo simulations discussed in Section 2.3.

Remark: Firing-angle convention. In the code interface, input firing data are specified in the usual artillery form: the *elevation angle* E of the barrel relative to the horizontal plane and the *azimuth angle* A relative to the ship's centerline plane. The formulation adopted

in the 6-DOF model follows the convention of McCoy (2012), which uses a pair of initial angles (θ_0, ϕ_0) associated with the ballistic axis system. To reconcile these two descriptions, the code performs an explicit transformation from (E, A) to (θ_0, ϕ_0) , given by

$$\theta_0 = \arcsin(\cos E \sin A), \quad \phi_0 = \arcsin\left(\frac{\sin E}{\cos \theta_0}\right),$$

with E and A expressed in radians. These expressions are implemented in the `calculate_firing_angles` routine, ensuring that the user can work with the more intuitive elevation and azimuth angles, while the 6-DOF integrator receives the angles θ_0 and ϕ_0 in the convention of McCoy (2012).

3.2.1 Preparation of coefficients and physical parameters

The code is structured around classes representing the main physical elements of the problem, such as ammunition, weapon, and platform (ship), whose attributes are initialized from the parameters discussed in Section 3.1. The tabulated aerodynamic coefficients for the 5"/38 projectile, described in Table 3.1, are read from a CSV file and organized into two-dimensional grids as a function of Mach number and total angle of attack.

For each relevant aerodynamic quantity (drag, normal force, and moments), a smooth interpolator is constructed using cubic-spline routines from the `scipy` library, in order to allow efficient evaluation of coefficients at intermediate Mach and total-angle-of-attack values. When the original table provides force coefficients in the body axis system, these are converted into drag C_D and lift $C_{L\alpha}$ coefficients using the geometric transformations presented in equation (3.1) and discussed in McCoy (2012).

Parameters originally provided in Imperial units (mass, moments of inertia, muzzle velocity, among others) are converted to SI units during the initialization step of the ammunition and weapon objects, ensuring dimensional consistency in all equations. Since the total angle of attack relevant to the simulated engagements remains small, the aerodynamic coefficients were pre-tabulated on a total-angle-of-attack grid limited to $\alpha_t \leq 10^\circ$. This preprocessing significantly reduces the computational cost of interpolation during integration, without introducing relevant error for the firing scenarios considered.

3.2.2 Core numerical integration

The second part of the code corresponds to the 6-DOF simulation core. Given the initial firing conditions, muzzle velocity, initial barrel orientation (elevation and azimuth), initial spin rate about the symmetry axis, and initial position relative to sea level, the state variables of the model are constructed according to the formulation presented by McCoy (2012).

At each time step, the code computes the Mach number and total angle of attack from the projectile velocity relative to the air. These values are used to query the interpolators built in the previous step, yielding instantaneous aerodynamic coefficients. Next, the corresponding aerodynamic forces and moments are computed and inserted into the system of ordinary differential equations for translational and rotational motion. In addition to

the equations of Section 2.2.4, the system integrated by the routine also includes the three kinematic equations for the center-of-mass position, given by (2.43)–(2.45), and the vector equation for the evolution of the unit vector \vec{i} along the projectile axis, given by (2.30).

The system is integrated using the `solve_ivp` routine from `scipy`, employing the DOP853 method, an explicit eighth-order Runge–Kutta scheme with adaptive step size. Relative and absolute tolerances on the order of 10^{-7} were specified to ensure trajectory accuracy without making the simulation time prohibitive. The integrator is run until the projectile reaches the sea surface (when the vertical coordinate of the center of mass becomes zero) or until a predefined maximum flight time is reached, which prevents excessively long integrations in cases of anomalous trajectories.

3.2.3 Layer of numerical experiments

With the 6-DOF simulation core implemented, the numerical experiments consist of additional routines that use the solver as a *black box*, without altering its internal structure. This layer of numerical experiments can be organized into: (i) a deterministic sweep of the firing parameter space (elevation and azimuth), (ii) the selection of representative engagement points in range, and (iii) Monte Carlo simulations for estimating hit probabilities, statistical analysis, and cost.

The first step is a regular-grid sweep of the firing parameter space. Elevation angles ranging from $45,0^\circ$ down to $-15,0^\circ$, with step $0,1^\circ$, and azimuth angles ranging from $-1,65^\circ$ to $0,0^\circ$, with step $0,05^\circ$, were considered. For each pair (elevation, azimuth), the 6-DOF model is integrated once, and the impact point relative to the target is recorded. From this set of simulations, for each elevation the azimuth values that minimize the lateral deviation of the impact point relative to the target symmetry axis were selected, analogously to a grid-based minimum search. The results of this step were stored in a table for subsequent filtering.

In the second step, this table is filtered to produce a set of engagement points distributed approximately every 100 m in range, starting from the maximum impact distance and proceeding to shorter distances. A practical minimum elevation of $-1,5^\circ$ was adopted, below which the firing geometry ceases to be representative of the scenario of interest. This range spacing is consistent, in order of magnitude, with the combination of target speed and the weapon rate of fire: the *Sea Baby* naval drone has a maximum speed close to 90 km/h, i.e., about 1,5 km/min (SUTTON, 2024), while the 5"/38 gun has a typical rate of fire of 12–15 rounds per minute (DiGIULIAN, 2012). Thus, intervals of approximately 100 m in range correspond, roughly, to the distances traveled by the target between successive shots in a continuous engagement scenario.

Once the set of (range, firing configuration) pairs to be analyzed is defined, the stochastic simulation stage is performed. For each selected point, $K = 1000$ Monte Carlo replicas are carried out, in which the only assumed sources of randomness are aiming errors in elevation and azimuth. In each replica, starting from the nominal angles E_0 (elevation) and A_0 (azimuth), effective angles are generated as

$$E = E_0 + \varepsilon_\theta, \quad A = A_0 + \varepsilon_\phi,$$

where $\varepsilon_\theta \sim \mathcal{N}(0, \sigma_\theta^2)$ and $\varepsilon_\phi \sim \mathcal{N}(0, \sigma_\phi^2)$ are independent Gaussian random variables with standard deviations $\sigma_\theta = 0,1^\circ$ and $\sigma_\phi = 0,05^\circ$. In the absence of specific angular-dispersion data for the 5"/38 gun under equivalent conditions, these values were adopted as plausible order-of-magnitude approximations, with the goal of representing moderate variability in the fire-control pointing.

The value $K = 1000$ was chosen to control sampling uncertainty in estimating the hit probability p_i . For each range, the estimator \hat{p}_i is simply the mean of K Bernoulli random variables, so that

$$\text{Var}(\hat{p}_i) = \frac{p_i(1 - p_i)}{K} \leq \frac{0,25}{K},$$

since the variance of a Bernoulli is maximized at $p_i = 0,5$. With $K = 1000$, one obtains $\text{Var}(\hat{p}_i) \leq 0,00025$ and therefore a maximum standard deviation of approximately 0,016. Invoking the normal approximation for sample proportions (central limit theorem) (Ross, 2014; KROESE *et al.*, 2011), an approximate 95 % confidence interval for p_i has width on the order of

$$\hat{p}_i \pm 1,96 \times 0,016 \approx \hat{p}_i \pm 0,03,$$

i.e., about 3 percentage points in the worst case. Thus, $K = 1000$ represents a trade-off between computational cost and acceptable statistical precision for the hit-probability estimates.

For each replica, the 6-DOF core is integrated until impact, and the result is classified as a hit or miss based on the relative position between the impact point and the effective area of the target. For each range in the grid, one thus obtains an empirical estimate of the hit probability \hat{p}_i , given by the relative frequency of hits among the K corresponding simulations. These values feed, in the following sections, both the neutralization cost function and the statistical inference procedures for the cost.

Chapter 4

Numerical results and discussion

This chapter presents and discusses the numerical results obtained with the 6-DOF simulator described in Chapter 3, varying only the effective elevation and azimuth firing angles (E, A).

The organization follows a progression from simple to complex. First, in Section 4.1, we analyze a deterministic reference shot, with no noise in the initial conditions, with the goal of verifying that the implemented model reproduces orders of magnitude compatible with reference data from the literature.

4.1 Deterministic reference shot

As a first sanity check of the simulator, we consider a test shot in which no stochastic perturbations are imposed on the initial firing angles, i.e., $\varepsilon_\theta = \varepsilon_\phi = 0$. The nominal elevation and azimuth values are taken as

$$E_0 = 43.3^\circ, \quad A_0 = 0^\circ,$$

so that the projectile is fired in a vertical plane containing the line of sight.

For this elevation angle, firing tables available for the 5 in/38 gun indicate a maximum range on the order of 16.6 km, with a flight time of approximately 64.4 s and a maximum altitude around 5.4 km (DiGIULIAN, 2012). It is worth noting that the initial conditions used in those tables may differ moderately from those adopted in this work (for instance, in muzzle velocity and in details of the aerodynamic model). Even so, these values serve as a reference for an order-of-magnitude comparison.

In the corresponding numerical simulation, the shot with $E_0 = 43.3^\circ$ and $A_0 = 0^\circ$ yields the following results:

$$R \approx 16.69 \text{ km}, \quad h_{\max} \approx 5.75 \text{ km}, \quad T \approx 66.05 \text{ s},$$

where R denotes the horizontal range to impact, h_{\max} the maximum altitude reached by the projectile, and T the total flight time. Relative to the reference values, the relative

differences are on the order of 0.3% in range, 6.3% in maximum altitude, and 2.6% in flight time, which is compatible with the expected uncertainties arising from differences in parameterization between the models.

Figure 4.1 shows the complete three-dimensional trajectory of this deterministic shot, highlighting the initial position at the muzzle and the impact point at sea level. The expected ballistic behavior is observed: the projectile is accelerated by gravity throughout the flight, describing an arc whose maximum height and range are compatible with the orders of magnitude discussed above. Note also the lateral drift due to the projectile's spin.

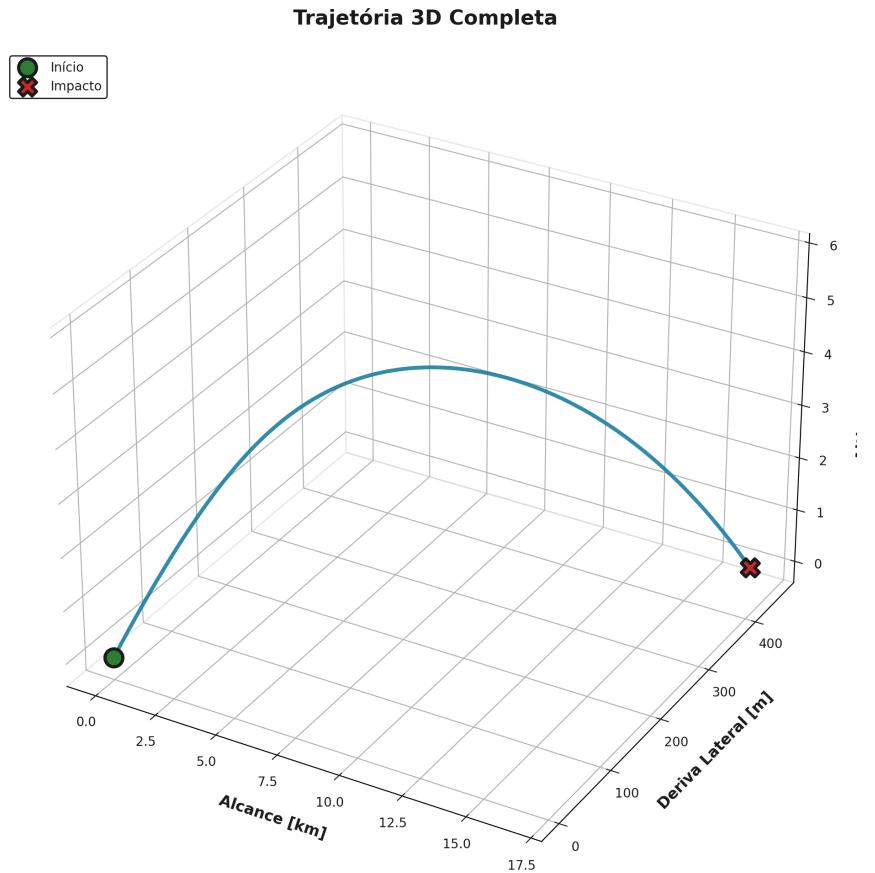


Figure 4.1: Complete three-dimensional trajectory of the reference shot, with nominal elevation $E_0 = 43.3^\circ$ and azimuth $A_0 = 0^\circ$, with no noise in the initial conditions. The green marker indicates the muzzle exit position and the red marker indicates the impact point at sea level.

4.1.1 Time evolution of altitude and total angle of attack

Beyond the aggregate values of range, maximum altitude, and flight time, it is instructive to analyze the time evolution of kinematic and aerodynamic quantities along the reference shot. Figure 4.2 shows the projectile altitude as a function of time, while Figure 4.3 shows the variation of the total angle of attack $\alpha(t)$ along the same trajectory.

In Figure 4.2, one observes that the altitude increases monotonically from muzzle exit until reaching a maximum near half of the total flight time, then decreases until

4.1 | DETERMINISTIC REFERENCE SHOT

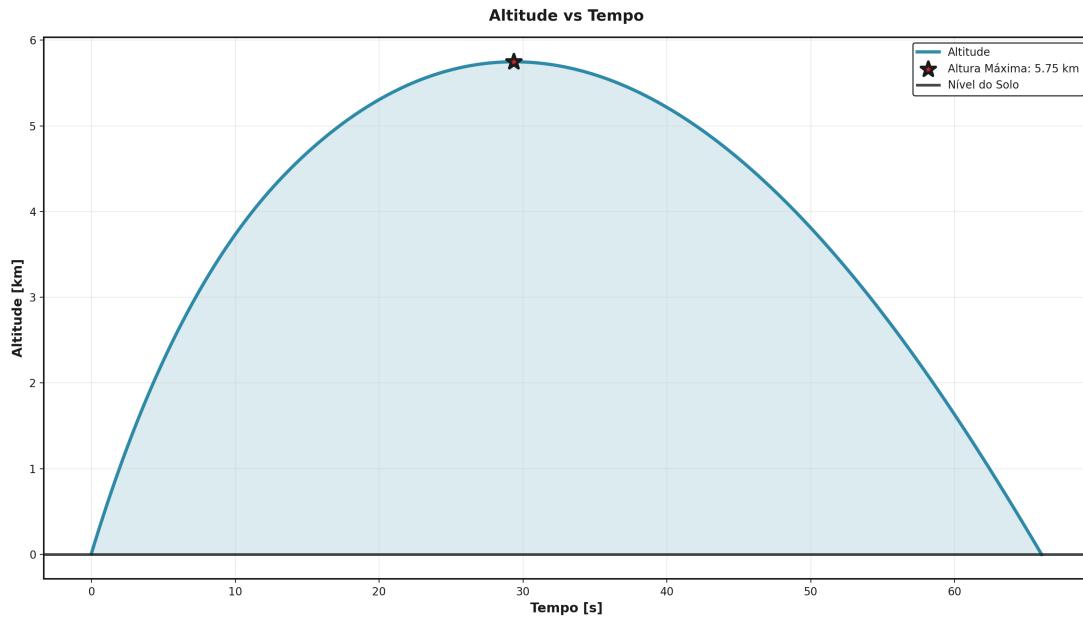


Figure 4.2: Projectile altitude as a function of time for the reference shot with $E_0 = 43.3^\circ$ and $A_0 = 0^\circ$. The marker highlights the maximum altitude reached, $h_{\max} \approx 5.75$ km, attained approximately halfway through the flight time.

returning to sea level at the impact instant. The curve shape is compatible with the expected ballistic behavior: one can note that the vertical component of the motion is dominated by gravitational acceleration, with small asymmetries introduced by aerodynamic drag and lift.

The total angle-of-attack behavior shown in Figure 4.3 exhibits several relevant features. Immediately after firing, there is a brief interval during which α_t reaches values up to about 7° . This initial peak is associated with the accommodation between the projectile's initial orientation and the effective velocity direction shortly after leaving the tube. Subsequently, the total angle of attack decays rapidly to values below 0.5° , reflecting the stabilizing action of spin and aerodynamic damping moments.

Over most of the flight, $\alpha_t(t)$ remains small, always on the order of a few degrees. A gentle maximum is observed approximately halfway through the flight time, when the trajectory gradually transitions from the ascending to the descending phase. This slight increase is related to trajectory curvature: while the velocity vector rotates downward under gravity, the projectile's symmetry axis does not follow at the same rate, resulting in a small misalignment between orientation and velocity. In the final phase of flight, aerodynamic moments tend to realign the projectile with the direction of motion, causing the total angle of attack to return to values close to zero.

Taken together, Figures 4.2 and 4.3 indicate that the numerical solution obtained for the deterministic reference shot exhibits time evolution consistent with the expected physical behavior, reinforcing confidence in the implemented model for the statistical analyses discussed in the following sections.

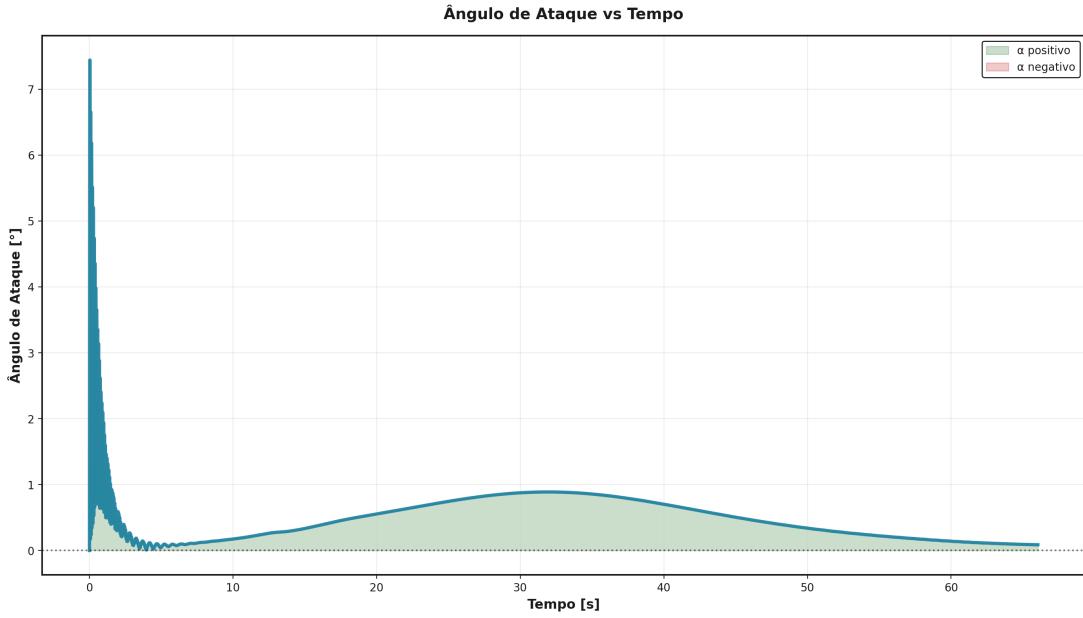


Figure 4.3: Total angle of attack $\alpha_t(t)$ as a function of time for the reference shot. A transient regime is observed shortly after muzzle exit, followed by small-magnitude values over most of the trajectory.

4.1.2 Time evolution of velocities and the symmetry axis

Still within the deterministic reference shot, it is instructive to analyze the time evolution of the velocity components and the orientation of the projectile's symmetry axis. Figure 4.4 presents the speed magnitude $\|\vec{V}\|$ and its components in an inertial basis (V_1, V_2, V_3) , while Figure 4.5 shows the components of the polar-axis vector $\vec{i}' = (i'_1, i'_2, i'_3)$.

In Figure 4.4, one observes that the speed magnitude $\|\vec{V}\|$ starts at approximately 800 m/s and decays rapidly in the first few seconds due to the strong effect of aerodynamic drag in the supersonic regime. The component V_1 , associated with the approximate firing direction, decreases monotonically throughout the flight, reflecting the gradual loss of speed along the nominal launch direction. The vertical component V_2 is initially positive and of large magnitude, crosses zero around half of the flight time (coinciding with the instant of maximum altitude in Figure 4.2), and becomes negative during the descending phase, as expected for ballistic motion under gravity. The lateral component V_3 remains close to zero throughout the flight, with small oscillations, which is consistent with the choice of azimuth $A_0 = 0^\circ$ and the absence of crosswind in the model.

Figure 4.5 illustrates the evolution of the polar-axis components \vec{i}' in inertial coordinates. Shortly after muzzle exit, a brief transient regime with small-amplitude oscillations in the three components is observed, characterizing the typical nutation of gyroscopically stabilized projectiles. Within a short time, however, these oscillations are strongly damped, and the components vary smoothly over the remainder of the flight. In particular, the lateral component i'_3 quickly tends to values near zero, indicating that the symmetry axis remains essentially contained in the vertical plane of the trajectory, while i'_1 and i'_2 evolve coherently with the gradual rotation of the projectile from an initial orientation “pointing upward” to a slightly downward-tilted orientation in the final phase of flight.

4.1 | DETERMINISTIC REFERENCE SHOT

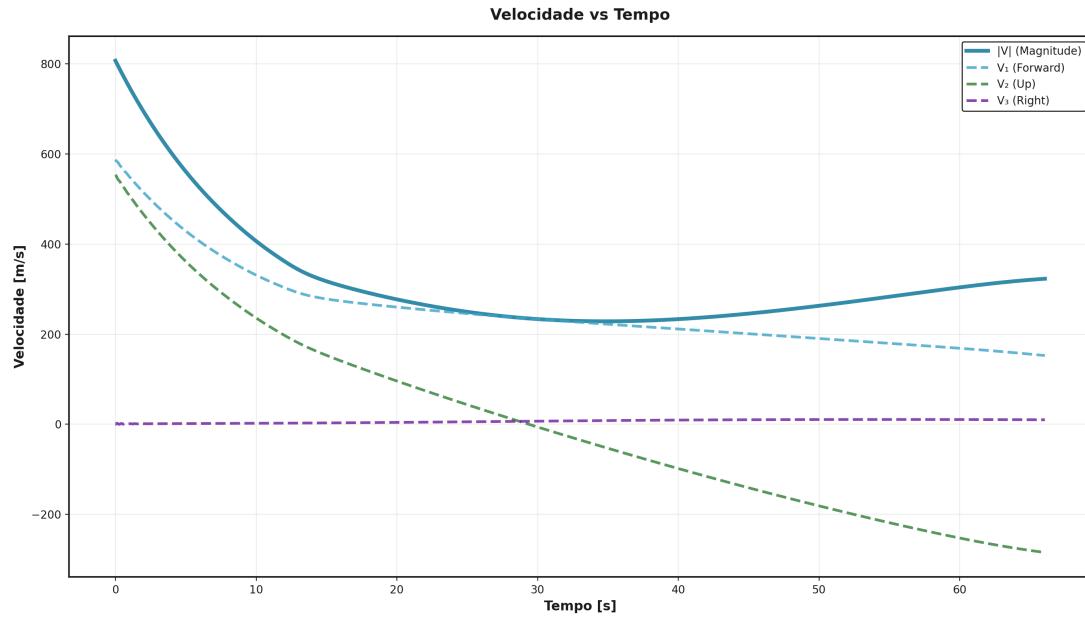


Figure 4.4: Velocity components as a function of time for the reference shot. The solid curve represents the magnitude $\|\vec{V}\|$, while dashed curves indicate the components V_1 (approximately the horizontal firing direction), V_2 (vertical), and V_3 (lateral).

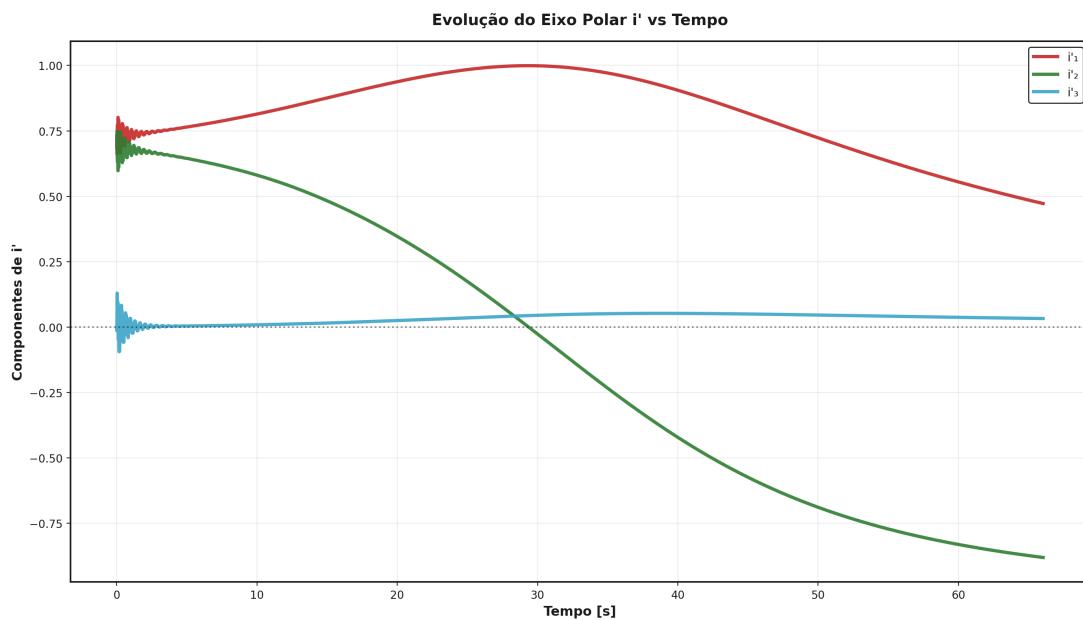


Figure 4.5: Components of the symmetry axis \vec{i}' as a function of time for the reference shot. An initial transient nutation is observed, followed by a slow variation associated with trajectory curvature.

The difference between the symmetry-axis direction and the instantaneous velocity direction is precisely the total angle of attack analyzed in Figure 4.3. As discussed above, this difference remains small at all times (on the order of a few degrees), indicating that the projectile closely follows the trajectory curvature throughout the shot, without exhibiting significant aerodynamic deviations or instabilities.

4.2 Selection and spacing of engagement configurations

As described in Section 3.2, the engagement points along the trajectory were chosen so that the distance between consecutive impacts would be approximately constant, with a target of 100 m. The primary tolerance band is taken as the interval 80 to 120 m (i.e., ± 20 m around the target) and, more permissively, an expanded band between 70 and 130 m.

Figure 4.6 presents the distribution of distances between consecutive points obtained after applying this procedure. In total, $N = 162$ distances were selected (163 points in all), with mean distance $\bar{d} \approx 101.6$ m and standard deviation of approximately 20.6 m. The minimum and maximum observed values were, respectively, 67.7 m and 148.2 m.

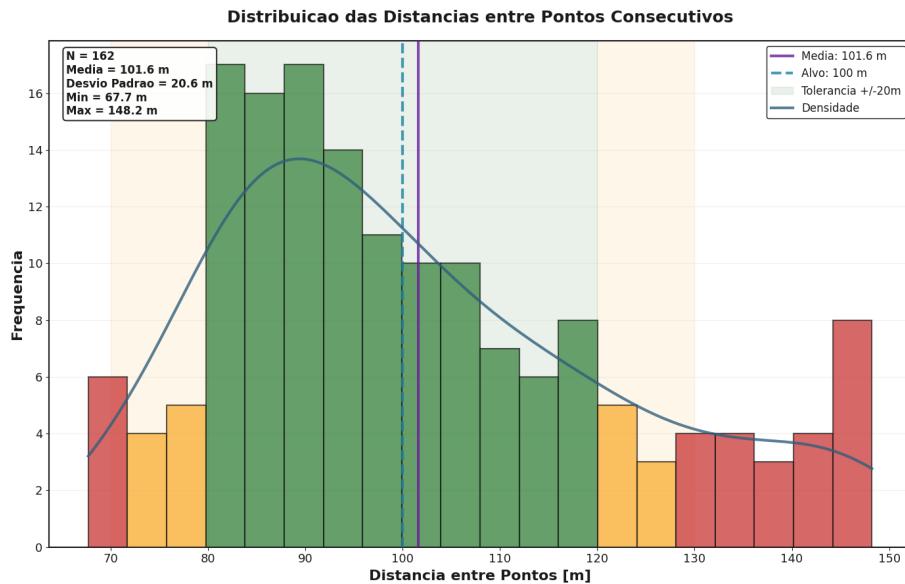


Figure 4.6: Distribution of distances between consecutive engagement points along the trajectory of the reference shot. The dashed line indicates the 100 m target. The region shaded in green corresponds to the primary tolerance band, between 80 and 120 m (± 20 m around the target), while the yellow bands cover the expanded interval from 70 to 130 m. Bars are colored according to the bin center: green within the primary band, yellow in the expanded band, and red outside it.

One observes that most distances between consecutive points remain within the primary tolerance band [80, 120] m, concentrating around the target value 100 m. Few intervals fall only in the expanded band [70, 80] or [120, 130] m, and only a small fraction of points is classified as outside tolerance (red). Thus, the constructed grid of points is sufficiently uniform to sample the trajectory with good resolution, without generating

an excessive number of firing configurations for the Monte Carlo simulations presented in the following sections.

4.3 Monte Carlo simulations for hit probability

As described in Section 3.2, uncertainties in the initial firing conditions are modeled by independent Gaussian perturbations in the elevation and azimuth angles. For each engagement configuration $i = 1, \dots, m$, K independent simulations are performed, resulting in K impact points. The hit probability is then estimated by $\hat{p}_i = N_i^{(\text{hit})}/K$, where $N_i^{(\text{hit})}$ is the number of impacts whose impact point lies within the geometric contour of the Sea Baby drone ($6 \text{ m} \times 2 \text{ m}$). This section first presents a geometric visualization of dispersion for a representative configuration and then the global results for the estimated probabilities along the full trajectory.

4.3.1 Geometric visualization of dispersion for a representative configuration

Figure 4.7 illustrates, in top view, the dispersion of impact points for a reference configuration with nominal elevation $E_0 = 18^\circ$ and azimuth $A_0 = -0.6^\circ$. In this example, only $K = 50$ simulations were generated, with the specific objective of providing a geometric illustration of the effect of angular noise on the impact pattern. The green rectangle represents the Sea Baby drone contour ($6 \text{ m} \times 2 \text{ m}$), centered around the nominal impact point (marked by a red cross), while blue circles indicate the impact points from the simulations.

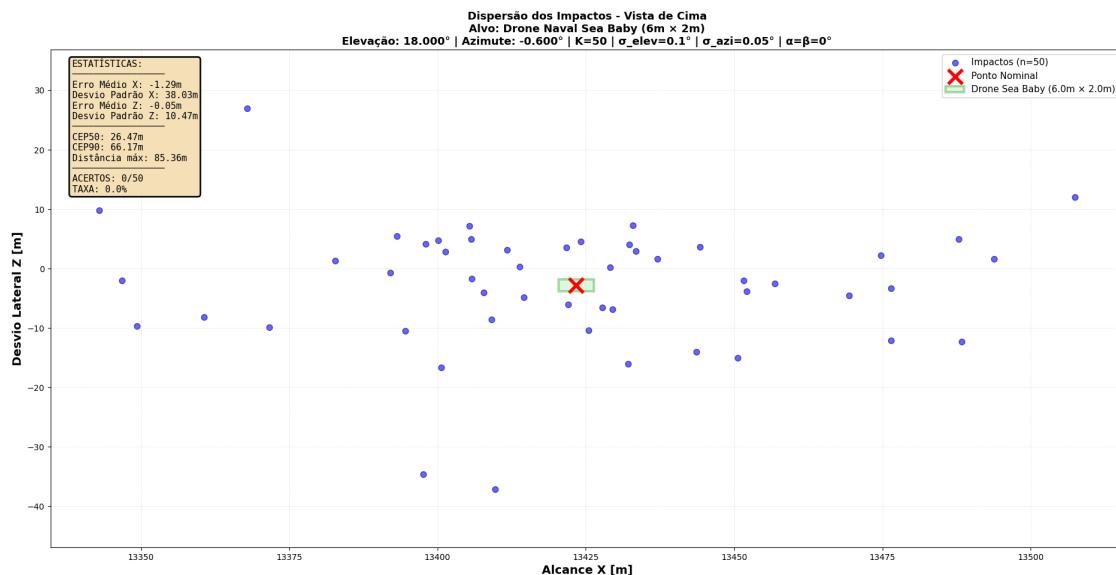


Figure 4.7: Top-view impact dispersion for the configuration with nominal elevation $E_0 = 18^\circ$ and azimuth $A_0 = -0.6^\circ$, considering $K = 50$ simulations with angular noise. The green rectangle represents the Sea Baby drone contour ($6 \text{ m} \times 2 \text{ m}$), the red cross marks the nominal impact point, and blue circles indicate the simulated impact points.

Even with relatively modest standard deviations in elevation and azimuth (0.1° and 0.05°), Figure 4.7 shows a dispersion of tens of meters around the nominal point. The impact pattern is approximately elliptical, elongated in the range direction X and narrower in the lateral direction Z .

4.3.2 Estimated probabilities along the trajectory

For the quantitative analysis of neutralization probability, we used $K = 1000$ simulations for each of the $m = 163$ engagement configurations described in Section 4.2. Figure 4.8 presents the estimated probabilities \hat{p}_i along the projectile trajectory, ordered by distance to the target, together with approximate 95% confidence intervals. These intervals were constructed using the normal approximation to the binomial distribution: $\hat{p}_i \pm z_{0.975} \sqrt{\hat{p}_i(1 - \hat{p}_i)/K}$, with $z_{0.975} \approx 1.96$.

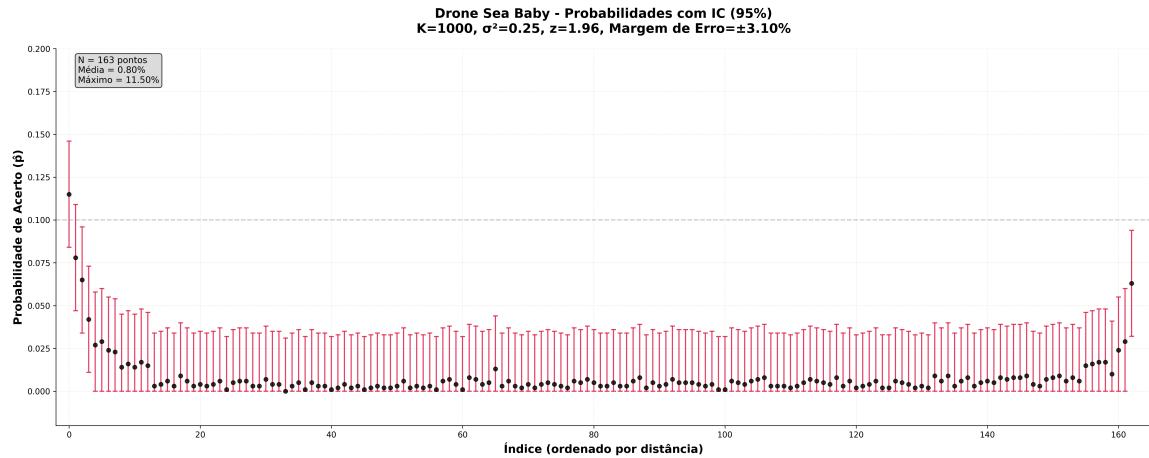


Figure 4.8: Estimated hit probabilities \hat{p}_i for the Sea Baby drone as a function of the engagement-configuration index (ordered by distance to the target). For each point, approximate 95% confidence intervals are shown based on the binomial distribution with $K = 1000$ simulations per configuration. The dashed line indicates the 10% hit-probability level.

In total, we obtain $N = 163$ estimates, with mean hit probability $\bar{p} \approx 0.8\%$ and maximum value around 11.5%. One observes that for most configurations, especially in the intermediate distance region, the estimates \hat{p}_i remain very close to zero, and all confidence intervals lie well below the 10% threshold indicated in the figure. In practical terms, this means that, with the considered levels of angular error, a single shot has a low probability of neutralizing the drone over much of the possible engagement trajectory.

An interesting behavior occurs in the vicinity of the extreme distances considered. Near maximum range, the values of \hat{p}_i tend to be slightly larger than those at immediately shorter distances, although they remain modest in absolute terms. A heuristic explanation for this effect is related to the fact that, near the elevation angle that maximizes range, any elevation error will reduce the distance. In other words, shots tend to land closer to one another.

It is important to emphasize that this maximum-range scenario is unlikely from an operational standpoint. It is highly improbable that a ship would attempt to engage a suicide drone precisely near the ballistic maximum range of its main gun.

4.4 Analysis of the expected engagement cost

This section uses the cost function defined in Chapter 1 to quantify, in an aggregate manner, the expected cost of an engagement against the Sea Baby drone along the entire simulated trajectory. Recall that, for a prefix with n firing opportunities and hit probabilities $\{p_1, \dots, p_n\}$, the model considers the expected cost

$$\mathbb{E}[C] = c(1 + (1 - p_1) + (1 - p_1)(1 - p_2) + \dots + (1 - p_1) \dots (1 - p_{n-1})) + E \prod_{i=1}^n (1 - p_i),$$

where c is the unit cost of each round and E is the value of the defending platform, interpreted as a penalty in the case of total failure (destruction of the ship after all firing opportunities have been exhausted).

In the numerical scenarios considered, we adopted $c = \text{US\$ } 2,000$ per shot and $E = \text{US\$ } 289,000,000$, which implies a ratio $E/c \approx 1.4 \times 10^5$; that is, the loss of the platform is equivalent, in monetary terms, to roughly one hundred and forty thousand rounds of ammunition. The probabilities $\{p_i\}$ used at each point correspond to the estimates obtained in Section 4.3.2 for the Sea Baby drone.

4.4.1 Expected ammunition cost

Figure 4.9 shows the behavior of the term associated exclusively with ammunition expenditure, $c \mathbb{E}[N]$, as a function of the maximum number n of shots allowed. The curve represents the expected ammunition cost for prefixes of length $n = 1, 2, \dots, 163$, where 163 is the total number of firing configurations considered along the trajectory.

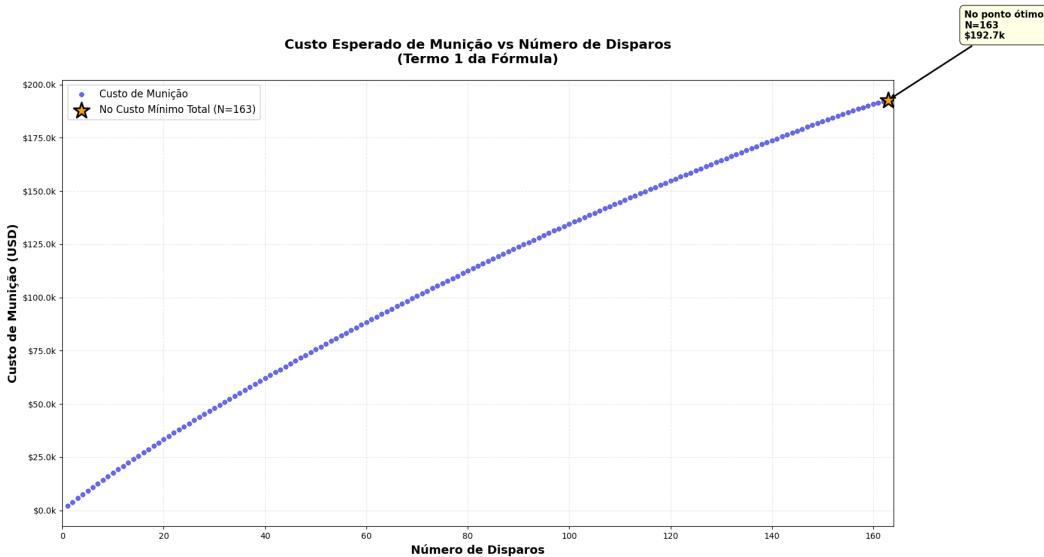


Figure 4.9: Expected ammunition cost $c \mathbb{E}[N]$ as a function of the maximum number of shots n . The star indicates the point at which the total expected cost (ammunition + penalty for platform loss) reaches its minimum, which coincides with using all $n = 163$ firing opportunities.

For the case in which all $n = 163$ opportunities are used in the model, we obtain an

expected number of shots $E[N] \approx 96.36$, resulting in an expected ammunition cost on the order of $c E[N] \approx \text{US\$ } 192,700$. The curve in Figure 4.9 is increasing in n , as expected: the more firing opportunities are made available, the larger the average number of rounds actually expended over the engagement. Note, however, that this cost remains small relative to the scale of the problem when compared to the platform value.

4.4.2 Accumulated probability of success

Figure 4.10 presents the accumulated probability of success $P(\text{success}) = 1 - \prod_{i=1}^n (1-p_i)$ as a function of the maximum number of shots n . Each blue point represents the probability of neutralizing the drone in up to n consecutive shots, considering that the probabilities p_i are those estimated via Monte Carlo simulation for the Sea Baby.

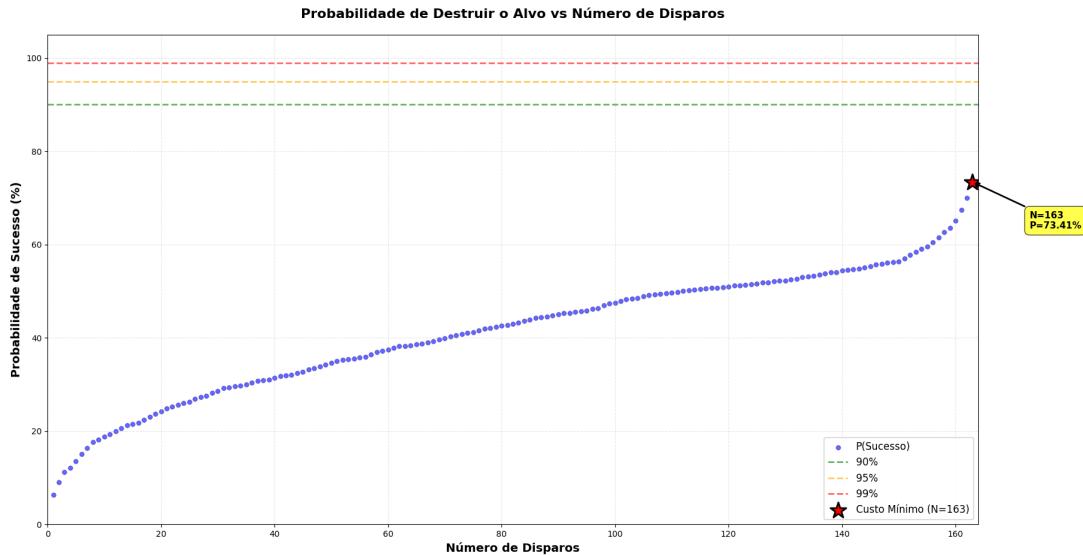


Figure 4.10: Accumulated probability of destroying the target as a function of the maximum number of shots n . Dashed lines indicate reference levels of 90%, 95%, and 99%. The star marks the point $n = 163$, corresponding to the use of all available configurations.

With only one shot ($n = 1$), the success probability is on the order of 6.3%, reflecting the low individual hit probability discussed earlier. As n increases, the accumulated probability of destroying the target grows monotonically, reaching approximately 18.8% at $n = 10$, 47.5% at $n = 100$, and about 73.4% when all $n = 163$ opportunities are considered. Even at the model limit (163 shots possible over the drone's approach), the 90% success level is not reached.

4.4.3 Total expected cost and operating point

Figure 4.11 synthesizes the two competing effects described above, presenting the total expected cost $E[C]$ as a function of n . For each prefix of length n , both the expected ammunition cost (term 1) and the expected cost associated with the eventual loss of the platform (term 2), weighted by the probability of total failure, are considered simultaneously.

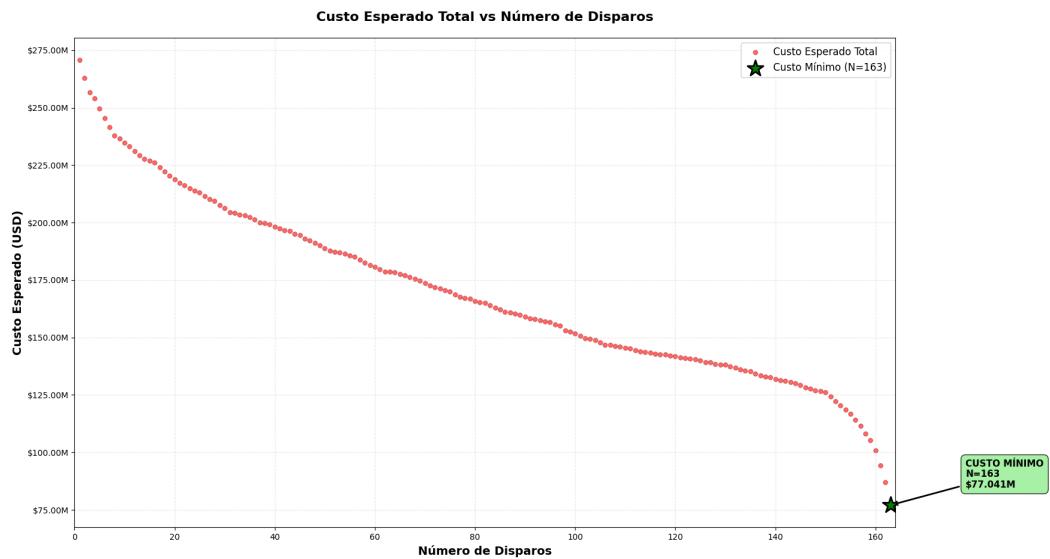


Figure 4.11: Total expected cost $\mathbb{E}[C]$ as a function of the maximum number of shots n . The green star indicates the minimum-cost point, which occurs at $n = 163$, with expected cost of approximately US\$ 77.0 million.

The numerical results show that, for $n = 1$, the expected cost is on the order of US\$ 270.8 million, very close to the platform value E . This occurs because, with only one shot, the probability of total failure is high (about 93.7%), so most of the expected cost comes from the term associated with ship loss. As the maximum number of shots increases, the probability of total failure decreases substantially, and the expected cost $\mathbb{E}[C]$ decreases monotonically: for $n = 50$ we obtain an expected cost around US\$ 188.8 million, for $n = 100$ about US\$ 151.8 million, and finally, for $n = 163$, the approximate minimum cost of US\$ 77.0 million, with success probability of approximately 73.4% and residual total-failure probability of 26.6%.

This result highlights that, for the considered cost ratio ($E/c \approx 1.4 \times 10^5$), the dominant term in the cost function is the expected value associated with platform loss. The ammunition cost at the optimal point, about US\$ 192.7 thousand, represents less than 0.3% of the total expected cost of US\$ 77.0 million. Consequently, under the model assumptions (absence of additional constraints such as ammunition limitations, barrel heating, collateral risk, etc.), the policy that minimizes the expected cost is simply to use all available firing opportunities against the target, i.e., the most aggressive engagement strategy possible over the drone's approach.

Chapter 5

Conclusion

The central question of this work was to estimate the *expected cost of neutralizing* a naval surface drone using only a ship's main gun, taking into account, on the one hand, the cost of the ammunition employed and, on the other, the cost associated with a potential failure to neutralize the target. To answer this question, a six-degrees-of-freedom trajectory model was developed for the naval projectile, coupled with Monte Carlo simulations that made it possible to estimate hit probabilities as a function of engagement distance. These probabilities were then incorporated into a cost function that combines ammunition expenditure with the economic risk of losing the defending platform.

In the reference scenario considered, under the simplifying assumptions described in the methodology and restricting exclusively to the use of the main gun, we found that, in the case in which the entire planned firing sequence is actually executed, the total expected neutralization cost is on the order of US\$ 77 million. Of this amount, the expected contribution of ammunition cost is relatively modest, approximately US\$ 192.7 thousand, while the dominant term is the cost associated with the possibility that the drone is not neutralized and, consequently, with the loss of the ship.

If the focus is *only* on ammunition cost, the result initially appears favorable: the expected amount spent on projectiles (about US\$ 192.7 thousand) is lower than the typical unit cost attributed to the drone, on the order of US\$ 220 thousand. From this purely ballistic and per-shot price perspective, employing the main gun with relatively inexpensive ammunition would seem compatible with the idea of "neutralizing a cheap target with even cheaper ammunition." However, this reading ignores the central component of the problem: the risk of neutralization failure and the resulting cost of losing the defending platform.

Indeed, the simulations indicate that, even after applying the entire firing sequence considered, a neutralization failure probability of around 26% remains. In other words, approximately one quarter of the simulated attacks result in a drone that passes through the entire firing sequence without being hit, leading to the destruction of the ship. It is precisely this risk that raises the total expected cost to the order of tens of millions of dollars, dominated by the value of the naval platform. In practice, therefore, even if ammunition cost in isolation is lower than the drone's cost, an engagement method based *only* on the main gun is not economically effective, given the high residual probability

of catastrophic loss.

One way to reinforce this interpretation is to consider, for illustrative purposes, an alternative weapons system: suppose, for example, a missile system whose unit cost is on the order of US\$ 10 million, but with a 99% hit probability. Assuming a defending platform cost of approximately US\$ 289 million, the expected neutralization cost in this hypothetical scenario would be on the order of

$$C_{\text{missile}} \approx 0.01 \times 289 \text{ million} + 10 \text{ million} \approx 12.9 \text{ million US\$},$$

a value significantly lower than the roughly US\$ 77 million associated with exclusive use of the main gun in the studied scenario. Even though these numbers are only illustrative, they show that a more expensive system per shot can be, in expected-cost terms, more economical when it drastically reduces the probability of ship loss.

In this sense, the main message of this work is that the cost asymmetry between low-value naval drones and high-value naval platforms does manifest itself forcefully. Even when relatively inexpensive ammunition is used, the risk of losing the ship keeps the expected engagement cost high. Thus, the apparent “economy” of using only simple ballistic projectiles from the main gun proves misleading: although the per-shot cost is low, the total expected cost, weighted by the risk of failure, is not competitive.

Naturally, the study has important limitations. The most significant is the exclusive consideration of the main gun, imposed by the availability of ballistic and aerodynamic data. In many modern ships, neutralizing surface drones would tend to involve a combination of systems, such as high-rate point defenses (*Close-In Weapon Systems* – CIWS), stabilized machine guns, and, eventually, short-range missiles. These systems can operate at much higher rates of fire and, in some cases, with still relatively inexpensive ammunition, which may substantially reduce the risk of non-neutralization for a given number of shots. However, in the absence of reliable aerodynamic coefficients, rates of fire, and engagement envelopes for these systems, it was not possible to include them quantitatively in the present cost model.

Another relevant simplification is the use of purely ballistic projectiles, without terminal guidance, fired by a single type of gun. In practice, guided or assisted munitions exist that may have significantly higher unit costs, but also higher hit probabilities, especially against small and maneuvering targets. Similarly, different calibers and weapon systems, such as the OTO Melara 76 mm gun, widely used by navies around the world, could offer distinct combinations of range, rate of fire, and accuracy. The lack, in the public domain, of complete sets of aerodynamic coefficients and reliable operational data for these systems prevented their inclusion in the present study.

In light of these limitations, the obtained results should be interpreted as a conservative estimate, referring to a specific engagement scenario with a single surface drone, under simplified conditions and using only the main gun with unguided projectiles. Within this scope, however, the conclusion is robust: the strategy of relying exclusively on the main gun, even with low-cost ammunition, does not eliminate the cost asymmetry in favor of the attacker.

The natural extensions of this work point in several directions. First, the developed

methodology can be extended to include models of other weapon systems, such as CIWS and short-range missiles, provided that adequate ballistic and operational data are available. This would allow a comparison, within the same expected-cost framework, of different defense architectures against naval drones, exploring combinations of main gun, point defenses, and missiles.

Second, it would be of great interest to incorporate guided munitions and different calibers into the model, including 76 mm guns and other systems widely employed on modern escorts. Obtaining or constructing aerodynamic coefficients for these projectiles would enable hit-probability and expected-cost estimates much closer to operational reality.

Third, the approach proposed here can be adapted to the study of aerial threats, such as armed aerial drones, evaluating the expected neutralization cost with anti-aircraft ammunition and mixed defense systems. Given the relevance of this class of threats in modern asymmetric warfare, extending the model to aerial targets represents a promising research direction.

Finally, we consider that the objectives defined in the introduction were achieved. The work presented a physical and statistical model capable of estimating, albeit under simplifying assumptions, the expected cost of neutralizing a surface drone using a ship's main gun, and showed that, under the studied conditions, this strategy is not economically efficient when the risk of losing the defending platform is taken into account. The results reinforce the need for layered defense architectures and for systems with high hit probability, in order to mitigate the cost asymmetry that characterizes the use of naval drones in contemporary conflicts.

Appendix A

Source code of the 6-DOF simulation

This appendix presents the main structure of the Python code used for the 6-DOF ballistic simulation described in this work. It is not the complete code due to size limitations.

The full source code is available at:

<https://github.com/OAfundador/6-DOF-ballistic-model>

Program A.1 Main structure of the 6-DOF ballistic simulator in Python.

```

1  #
2  =====
3  # SIMULADOR BALISTICO 6-DOF
4  #
5  =====
6  import numpy as np
7  from math import sqrt, pi, sin, cos, atan2, acos
8  from dataclasses import dataclass
9  import matplotlib.pyplot as plt
10 from mpl_toolkits.mplot3d import Axes3D
11 from scipy.integrate import solve_ivp
12 import pandas as pd
13 from scipy.interpolate import interp1d, RectBivariateSpline
14 import warnings
15 warnings.filterwarnings('ignore')
16 #
17 # CLASSE SIMULADOR BALISTICO
18 #
19 =====

```

cont →

→ *cont*

```

20     class BallisticSimulator:
21         """
22             Classe principal que gerencia a simulacao balistica 6-DOF.
23         """
24
25     def __init__(self, projectile, weapon, environment, aero_coeffs):
26         """
27             Inicializa o simulador.
28
29             Parametros:
30             -----
31                 projectile : Projectile
32                     Objeto projetil
33                 weapon : Weapon
34                     Objeto arma
35                 environment : Environment
36                     Condicoes ambientais
37                 aero_coeffs : RealAerodynamicCoefficients
38                     Coeficientes aerodinamicos
39         """
40
41         self.projectile = projectile
42         self.weapon = weapon
43         self.environment = environment
44         self.aero_coeffs = aero_coeffs
45
46         # Resultado da simulacao
47         self.result = None
48
49     def build_initial_conditions(self, alpha0_deg=0.0, beta0_deg=0.0,
50                                 w_j0=5.0, w_k0=5.0):
51         """
52             Constrói condicoes iniciais para a simulacao.
53             IMPORTANTE: Considera a velocidade da embarcacao se a arma estiver
54                 montada em uma.
55
56             Parametros:
57             -----
58                 alpha0_deg : float
59                     Angulo de arfagem inicial em graus
60                 beta0_deg : float
61                     Angulo de guinada inicial em graus
62                 w_j0 : float
63                     Velocidade angular em j' [rad/s]
64                 w_k0 : float
65                     Velocidade angular em k' [rad/s]
66
67             Retorna:
68             -----
69                 array : vetor de estado inicial [V1, V2, V3, h1, h2, h3, i1, i2, i3, x,
70                               y, z]
71         """
72
73         # Angulos de tiro

```

cont →

```

→ cont
70     theta0, phi0 = self.weapon.calculate_firing_angles()
71     alpha0 = np.radians(alpha0_deg)
72     beta0 = np.radians(beta0_deg)
73
74     # Velocidade inicial DO PROJETIL RELATIVA A ARMA (velocidade na boca)
75     V0 = self.weapon.muzzle_velocity
76     V1_rel = V0 * cos(theta0) * cos(phi0)
77     V2_rel = V0 * cos(theta0) * sin(phi0)
78     V3_rel = V0 * sin(theta0)
79
80     # Velocidade da plataforma (embarcacao)
81     platform_velocity = self.weapon.get_velocity()
82
83     # VELOCIDADE ABSOLUTA = Velocidade relativa + Velocidade da plataforma
84     V1_0 = V1_rel + platform_velocity[0]
85     V2_0 = V2_rel + platform_velocity[1]
86     V3_0 = V3_rel + platform_velocity[2]
87
88     # Spin inicial
89     w_i0 = self.projectile.calculate_initial_spin(V0)
90
91     # Eixo polar i'
92     phi_eff = phi0 + alpha0
93     theta_eff = theta0 + beta0
94
95     i1_0 = cos(phi_eff) * cos(theta_eff)
96     i2_0 = cos(theta_eff) * sin(phi_eff)
97     i3_0 = sin(theta_eff)
98
99
100    # Eixos j' e k'
101    Q = sin(theta_eff)**2 + cos(theta_eff)**2 * cos(phi_eff)**2
102    sqrt_Q = sqrt(Q)
103
104    j1_0 = -(sin(phi_eff) * cos(phi_eff) * cos(theta_eff)**2) / sqrt_Q
105    j2_0 = (cos(theta_eff)**2 * cos(phi_eff)**2 + sin(theta_eff)**2) /
106        sqrt_Q
107    j3_0 = -(sin(theta_eff) * cos(theta_eff) * sin(phi_eff)) / sqrt_Q
108
109    k1_0 = -sin(theta_eff) / sqrt_Q
110    k2_0 = 0.0
111    k3_0 = (cos(phi_eff) * cos(theta_eff)) / sqrt_Q
112
113    # di'/dt
114    di1_dt = (w_j0 * sin(theta_eff) -
115                w_k0 * cos(theta_eff)**2 * sin(phi_eff) * cos(phi_eff)) /
116                    sqrt_Q
117
116    di2_dt = (w_k0 / sqrt_Q) * (cos(theta_eff)**2 * cos(phi_eff)**2 +
117                                sin(theta_eff)**2)
118
119    di3_dt = (-w_j0 * cos(theta_eff) * cos(phi_eff))

```

cont →

→ *cont*

```

120          - w_k0 * sin(phi_eff) * cos(theta_eff) * sin(theta_eff)) /
121                      sqrt_Q
122
123      # Momento angular h
124      omega1_inertial = w_i0
125
126      I_P = self.projectile.I_P
127      I_T = self.projectile.I_T
128
129      term1_h1 = (I_P / I_T) * omega1_inertial * i1_0
130      term1_h2 = (I_P / I_T) * omega1_inertial * i2_0
131      term1_h3 = (I_P / I_T) * omega1_inertial * i3_0
132
133      term2_h1 = i2_0 * di3_dt - i3_0 * di2_dt
134      term2_h2 = i3_0 * di1_dt - i1_0 * di3_dt
135      term2_h3 = i1_0 * di2_dt - i2_0 * di1_dt
136
137      h1_0 = term1_h1 + term2_h1
138      h2_0 = term1_h2 + term2_h2
139      h3_0 = term1_h3 + term2_h3
140
141      # Posicao inicial (posicao absoluta da arma)
142      abs_position = self.weapon.get_absolute_position()
143      x0, y0, z0 = abs_position
144
145      # Monta vetor de estado
146      y0_vec = np.array([V1_0, V2_0, V3_0,
147                         h1_0, h2_0, h3_0,
148                         i1_0, i2_0, i3_0,
149                         x0, y0, z0], dtype=float)
150
151      return y0_vec
152
153  def rhs(self, t, y):
154      """
155          Lado direito das equacoes diferenciais (RHS).
156
157          Parametros:
158          -----
159          t : float
160              Tempo
161          y : array
162              Vetor de estado [V1, V2, V3, h1, h2, h3, i1, i2, i3, x, y, z]
163
164          Retorna:
165          -----
166          array : derivadas do vetor de estado
167          """
168          V1, V2, V3, h1, h2, h3, i1, i2, i3, x, ypos, z = y
169
170          # Velocidade relativa (com vento)

```

cont →

```

→ cont
171     v1 = V1 - self.environment.W1
172     v2 = V2 - self.environment.W2
173     v3 = V3 - self.environment.W3
174     v = sqrt(v1*v1 + v2*v2 + v3*v3)
175
176     # Numero de Mach
177     mach = v / self.environment.sound_speed
178
179     # Angulo de ataque
180     cos_alpha_t = (v1*i1 + v2*i2 + v3*i3) / v
181     cos_alpha_t = np.clip(cos_alpha_t, -1.0, 1.0)
182     alpha_rad = acos(cos_alpha_t)
183
184     # Obter coeficientes aerodinamicos
185     coeffs = self.aero_coeffs.get_coefficients(mach, alpha_rad)
186
187     C_D = coeffs['CD_total'] # Drag Force Coefficient
188     C_Lalpha = coeffs['CLA_total'] # Lift Force Coefficient
189     C_Nalpha = coeffs['CYP'] #Magnus Force Coefficient
190     C_Nq = 0 # Pitching Dumping Force Coefficient
191     C_Nalpha_dot = 0 # Pitching Dumping Force Coefficient (segunda
                       componente)
192     C_lp = coeffs['CLP'] # Spin Dumping Moment Coefficient
193     C_Malpha = coeffs['CMA'] # Pitching Moment Coefficient
194     C_Mpalpha = coeffs['CNP_total'] # Magnus Moment Coefficient
195     C_Mq = coeffs['CMQ'] #Pitching Dumping Moment Coefficient
196     C_Malpha_dot = 0
197
198     C_l_delta = 0.0
199     delta_F = 0.0
200
201     # Parametros do projetil e ambiente
202     m = self.projectile.mass
203     S = self.projectile.S
204     d = self.projectile.diameter
205     I_P = self.projectile.I_P
206     I_T = self.projectile.I_T
207     rho = self.environment.rho
208     g = self.environment.g
209
210     # = (I_T/I_P) (h i')
211     h_dot_i = (h1*i1 + h2*i2 + h3*i3)
212     omega1 = (I_T/I_P) * h_dot_i
213
214     # Equacoes de forca (dV/dt)
215     dV1 = (
216         - (rho*v*S*C_D)/(2*m) * v1
217         + (rho*S*C_Lalpha)/(2*m) * ( (v*v)*i1 - v*v1*cos_alpha_t )
218         - (rho*S*d*C_Nalpha*omega1)/(2*m) * ( v3*i2 - v2*i3 )
219         + (rho*v*S*d*(C_Nq + C_Nalpha_dot))/(2*m) * ( h2*i3 - h3*i2 )
220     )
221     dV2 = (

```

cont →

```

→ cont

222      - (rho*v*S*C_D)/(2*m) * v2
223      + (rho*S*C_Lalpha)/(2*m) * ( (v*v)*i2 - v*v2*cos_alpha_t )
224      - (rho*S*d*C_Nalpha*omega1)/(2*m) * ( v1*i3 - v3*i1 )
225      + (rho*v*S*d*(C_Nq + C_Nalpha_dot))/(2*m) * ( h3*i1 - h1*i3 )
226      - g
227  )
228  dv3 = (
229      - (rho*v*S*C_D)/(2*m) * v3
230      + (rho*S*C_Lalpha)/(2*m) * ( (v*v)*i3 - v*v3*cos_alpha_t )
231      - (rho*S*d*C_Nalpha*omega1)/(2*m) * ( v2*i1 - v1*i2 )
232      + (rho*v*S*d*(C_Nq + C_Nalpha_dot))/(2*m) * ( h1*i2 - h2*i1 )
233  )
234

235 # Equacoes de momento (dh/dt)
236 dh1 = (
237      (rho*v*S*d**2*C_lp*omega1)/(2*I_T) * i1
238      + (rho*v**2*S*d*delta_F*C_l_delta)/(2*I_T) * i1
239      + (rho*v*S*d*C_Malpha)/(2*I_T) * ( v2*i3 - v3*i2 )
240      + (rho*S*d**2*C_Malpha*omega1)/(2*I_T) * ( v1 - v*i1*cos_alpha_t )
241      + (rho*v*S*d**2*(C_Mq+C_Malpha_dot))/(2*I_T) * ( h1 - ((I_P/I_T)*
242          omega1)*i1 )
243  )
244 dh2 = (
245      (rho*v*S*d**2*C_lp*omega1)/(2*I_T) * i2
246      + (rho*v**2*S*d*delta_F*C_l_delta)/(2*I_T) * i2
247      + (rho*v*S*d*C_Malpha)/(2*I_T) * ( v3*i1 - v1*i3 )
248      + (rho*S*d**2*C_Malpha*omega1)/(2*I_T) * ( v2 - v*i2*cos_alpha_t )
249      + (rho*v*S*d**2*(C_Mq+C_Malpha_dot))/(2*I_T) * ( h2 - ((I_P/I_T)*
250          omega1)*i2 )
251  )
252 dh3 = (
253      (rho*v*S*d**2*C_lp*omega1)/(2*I_T) * i3
254      + (rho*v**2*S*d*delta_F*C_l_delta)/(2*I_T) * i3
255      + (rho*v*S*d*C_Malpha)/(2*I_T) * ( v1*i2 - v2*i1 )
256      + (rho*S*d**2*C_Malpha*omega1)/(2*I_T) * ( v3 - v*i3*cos_alpha_t )
257      + (rho*v*S*d**2*(C_Mq+C_Malpha_dot))/(2*I_T) * ( h3 - ((I_P/I_T)*
258          omega1)*i3 )
259
260  # Equacao de orientacao (di'/dt)
261  di1 = h2*i3 - h3*i2
262  di2 = h3*i1 - h1*i3
263  di3 = h1*i2 - h2*i1
264
265  # Equacao de posicao
266  dx, dy, dz = V1, V2, V3
267
268  return np.array([dv1, dv2, dv3, dh1, dh2, dh3, di1, di2, di3, dx, dy,
269                  dz], dtype=float)

def simulate(self, max_time=100.0, alpha0_deg=0.0, beta0_deg=0.0,
270             w_j0=5.0, w_k0=5.0, rtol=1e-7, atol=1e-8):

```

cont →

```

→ cont
270     """
271     Executa a simulacao balistica.
272
273     Parametros:
274     -----
275     max_time : float
276         Tempo maximo de simulacao [s]
277     alpha0_deg : float
278         Angulo de arfagem inicial [deg]
279     beta0_deg : float
280         Angulo de guinada inicial [deg]
281     w_j0 : float
282         Velocidade angular em j' [rad/s]
283     w_k0 : float
284         Velocidade angular em k' [rad/s]
285     rtol : float
286         Tolerancia relativa do integrador
287     atol : float
288         Tolerancia absoluta do integrador
289
290     Retorna:
291     -----
292     SimulationResult : objeto contendo os resultados
293     """
294     print("\n" + "="*80)
295     print("INICIANDO SIMULACAO")
296     print("="*80)
297
298     # Construir condicoes iniciais
299     y0 = self.build_initial_conditions(alpha0_deg, beta0_deg, w_j0, w_k0)
300
301     # Evento de impacto no solo
302     def ground_event(t, y):
303         return y[10] # y position
304     ground_event.direction = -1
305     ground_event.terminal = True
306
307     print("\nIntegrando trajetoria...")
308
309     # Resolver EDO
310     sol = solve_ivp(self.rhs, (0.0, max_time), y0,
311                     method='DOP853',
312                     rtol=rtol, atol=atol,
313                     events=ground_event,
314                     max_step=0.1)
315
316     if sol.success:
317         print(f" Integracao bem-sucedida!")
318         print(f" Tempo de voo: {sol.t[-1]:.2f} s")
319     else:
320         print(f" Erro na integracao: {sol.message}")
321

```

cont →

→ *cont*

```
322     # Criar objeto de resultado
323     self.result = SimulationResult(sol, self)
324
325     return self.result
```

Appendix B

Relevant results for shots against the *Sea Baby drone*

This appendix presents, for the *Sea Baby drone* target, the list of firing points considered in the Monte Carlo simulation, including elevation, azimuth, range, the estimated hit rate, and the lower and upper bounds of the 95% confidence interval.

Table B.1: Resultados da simulação para o alvo Drone Sea Baby.

Ponto	Elevação (°)	Azimute (°)	Alcance (m)	Taxa de acerto (%)	p_{inf} (%)	p_{sup} (%)
1	39.6	-1.35	16797	6.3	3.2	9.4
2	36.2	-1.20	16716	2.9	0.0	6.0
3	34.8	-1.15	16635	2.4	0.0	5.5
4	33.7	-1.10	16551	1.0	0.0	4.1
5	32.8	-1.10	16469	1.7	0.0	4.8
6	32.0	-1.05	16387	1.7	0.0	4.8
7	31.2	-1.00	16295	1.6	0.0	4.7
8	30.5	-1.00	16207	1.5	0.0	4.6
9	29.9	-0.95	16126	0.6	0.0	3.7
10	29.3	-0.95	16039	0.8	0.0	3.9
11	28.7	-0.95	15947	0.6	0.0	3.7
12	28.2	-0.90	15866	0.9	0.0	4.0
13	27.7	-0.90	15782	0.8	0.0	3.9
14	27.2	-0.90	15694	0.7	0.0	3.8
15	26.7	-0.85	15602	0.3	0.0	3.4
16	26.2	-0.85	15507	0.4	0.0	3.5
17	25.7	-0.85	15408	0.9	0.0	4.0
18	25.3	-0.80	15326	0.8	0.0	3.9
19	24.9	-0.80	15242	0.8	0.0	3.9
20	24.5	-0.80	15156	0.7	0.0	3.8
21	24.1	-0.80	15067	0.8	0.0	3.9
22	23.7	-0.75	14976	0.5	0.0	3.6
23	23.3	-0.75	14882	0.6	0.0	3.7
24	22.9	-0.75	14787	0.5	0.0	3.6
25	22.5	-0.75	14689	0.3	0.0	3.4
26	22.1	-0.70	14588	0.8	0.0	3.9
27	21.7	-0.70	14486	0.6	0.0	3.7
28	21.3	-0.70	14381	0.3	0.0	3.4
29	21.0	-0.70	14300	0.9	0.0	4.0
30	20.7	-0.65	14219	0.6	0.0	3.7
31	20.4	-0.65	14136	0.9	0.0	4.0
32	20.1	-0.65	14051	0.2	0.0	3.3
33	19.8	-0.65	13966	0.3	0.0	3.4
34	19.5	-0.65	13879	0.2	0.0	3.3
35	19.2	-0.60	13791	0.4	0.0	3.5
36	18.9	-0.60	13701	0.5	0.0	3.6

Table B.1: Resultados da simulação para o alvo Drone Sea Baby (continuação).

Ponto	Elevação (°)	Azimute (°)	Alcance (m)	Taxa de acerto (%)	p_{inf} (%)	p_{sup} (%)
37	18.6	-0.60	13610	0.6	0.0	3.7
38	18.3	-0.60	13517	0.2	0.0	3.3
39	18.0	-0.60	13423	0.2	0.0	3.3
40	17.7	-0.60	13328	0.6	0.0	3.7
41	17.4	-0.55	13231	0.4	0.0	3.5
42	17.1	-0.55	13133	0.3	0.0	3.4
43	16.8	-0.55	13033	0.2	0.0	3.3
44	16.5	-0.55	12932	0.6	0.0	3.7
45	16.2	-0.55	12830	0.3	0.0	3.4
46	15.9	-0.55	12725	0.8	0.0	3.9
47	15.6	-0.50	12620	0.4	0.0	3.5
48	15.3	-0.50	12512	0.5	0.0	3.6
49	15.0	-0.50	12403	0.6	0.0	3.7
50	14.7	-0.50	12292	0.7	0.0	3.8
51	14.4	-0.50	12180	0.5	0.0	3.6
52	14.1	-0.45	12066	0.3	0.0	3.4
53	13.8	-0.45	11949	0.2	0.0	3.3
54	13.5	-0.45	11832	0.3	0.0	3.4
55	13.2	-0.45	11712	0.3	0.0	3.4
56	13.0	-0.45	11631	0.3	0.0	3.4
57	12.8	-0.45	11549	0.8	0.0	3.9
58	12.6	-0.45	11466	0.7	0.0	3.8
59	12.4	-0.40	11382	0.6	0.0	3.7
60	12.2	-0.40	11297	0.4	0.0	3.5
61	12.0	-0.40	11211	0.5	0.0	3.6
62	11.8	-0.40	11124	0.6	0.0	3.7
63	11.6	-0.40	11037	0.1	0.0	3.2
64	11.4	-0.40	10948	0.1	0.0	3.2
65	11.2	-0.40	10857	0.4	0.0	3.5
66	11.0	-0.40	10766	0.3	0.0	3.4
67	10.8	-0.40	10673	0.4	0.0	3.5
68	10.6	-0.35	10580	0.5	0.0	3.6
69	10.4	-0.35	10484	0.5	0.0	3.6
70	10.2	-0.35	10388	0.5	0.0	3.6
71	10.0	-0.35	10290	0.7	0.0	3.8
72	9.8	-0.35	10191	0.4	0.0	3.5
73	9.6	-0.35	10090	0.3	0.0	3.4
74	9.4	-0.35	9987	0.5	0.0	3.6
75	9.2	-0.35	9883	0.2	0.0	3.3
76	9.0	-0.30	9777	0.8	0.0	3.9
77	8.8	-0.30	9669	0.6	0.0	3.7
78	8.6	-0.30	9559	0.3	0.0	3.4
79	8.4	-0.30	9447	0.3	0.0	3.4
80	8.2	-0.30	9333	0.5	0.0	3.6
81	8.0	-0.30	9217	0.3	0.0	3.4
82	7.8	-0.30	9098	0.3	0.0	3.4
83	7.6	-0.30	8977	0.5	0.0	3.6
84	7.4	-0.30	8853	0.7	0.0	3.8
85	7.2	-0.25	8726	0.5	0.0	3.6
86	7.0	-0.25	8597	0.6	0.0	3.7
87	6.8	-0.25	8464	0.2	0.0	3.3
88	6.7	-0.25	8396	0.3	0.0	3.4
89	6.6	-0.25	8328	0.4	0.0	3.5
90	6.5	-0.25	8258	0.5	0.0	3.6
91	6.4	-0.25	8188	0.4	0.0	3.5
92	6.3	-0.25	8117	0.2	0.0	3.3
93	6.2	-0.25	8044	0.4	0.0	3.5
94	6.1	-0.25	7971	0.2	0.0	3.3
95	6.0	-0.25	7897	0.3	0.0	3.4
96	5.9	-0.25	7822	0.6	0.0	3.7
97	5.8	-0.25	7746	0.3	0.0	3.4
98	5.7	-0.25	7669	1.3	0.0	4.4
99	5.6	-0.25	7591	0.5	0.0	3.6
100	5.5	-0.20	7512	0.4	0.0	3.5
101	5.4	-0.20	7431	0.7	0.0	3.8
102	5.3	-0.20	7350	0.8	0.0	3.9
103	5.2	-0.20	7267	0.1	0.0	3.2
104	5.1	-0.20	7183	0.4	0.0	3.5

Table B.1: Resultados da simulação para o alvo Drone Sea Baby (continuação).

Ponto	Elevação (°)	Azimute (°)	Alcance (m)	Taxa de acerto (%)	p_{inf} (%)	p_{sup} (%)
105	5.0	-0.20	7098	0.7	0.0	3.8
106	4.9	-0.20	7012	0.6	0.0	3.7
107	4.8	-0.20	6924	0.1	0.0	3.2
108	4.7	-0.20	6836	0.3	0.0	3.4
109	4.6	-0.20	6745	0.2	0.0	3.3
110	4.5	-0.20	6654	0.3	0.0	3.4
111	4.4	-0.20	6561	0.2	0.0	3.3
112	4.3	-0.20	6467	0.6	0.0	3.7
113	4.2	-0.20	6371	0.3	0.0	3.4
114	4.1	-0.20	6274	0.2	0.0	3.3
115	4.0	-0.20	6175	0.2	0.0	3.3
116	3.9	-0.20	6075	0.3	0.0	3.4
117	3.8	-0.15	5973	0.2	0.0	3.3
118	3.7	-0.15	5870	0.1	0.0	3.2
119	3.6	-0.15	5765	0.3	0.0	3.4
120	3.5	-0.15	5659	0.2	0.0	3.3
121	3.4	-0.15	5550	0.4	0.0	3.5
122	3.3	-0.15	5440	0.2	0.0	3.3
123	3.2	-0.15	5328	0.1	0.0	3.2
124	3.1	-0.15	5215	0.3	0.0	3.4
125	3.0	-0.15	5100	0.3	0.0	3.4
126	2.9	-0.15	4982	0.5	0.0	3.6
127	2.8	-0.15	4863	0.1	0.0	3.2
128	2.7	-0.15	4742	0.5	0.0	3.6
129	2.6	-0.15	4619	0.3	0.0	3.4
130	2.5	-0.15	4494	0.0	0.0	3.1
131	2.4	-0.15	4367	0.4	0.0	3.5
132	2.3	-0.15	4238	0.4	0.0	3.5
133	2.2	-0.15	4108	0.7	0.0	3.8
134	2.1	-0.10	3975	0.3	0.0	3.4
135	2.0	-0.10	3840	0.3	0.0	3.4
136	1.9	-0.10	3703	0.6	0.0	3.7
137	1.8	-0.10	3564	0.6	0.0	3.7
138	1.7	-0.10	3424	0.5	0.0	3.6
139	1.6	-0.10	3282	0.1	0.0	3.2
140	1.5	-0.10	3138	0.6	0.0	3.7
141	1.4	-0.10	2993	0.4	0.0	3.5
142	1.3	-0.10	2846	0.3	0.0	3.4
143	1.2	-0.10	2699	0.4	0.0	3.5
144	1.1	-0.10	2550	0.3	0.0	3.4
145	1.0	-0.10	2402	0.6	0.0	3.7
146	0.9	-0.10	2254	0.9	0.0	4.0
147	0.8	-0.10	2107	0.3	0.0	3.4
148	0.7	-0.10	1961	0.6	0.0	3.7
149	0.6	-0.10	1819	0.4	0.0	3.5
150	0.5	-0.10	1679	0.3	0.0	3.4
151	0.4	-0.10	1545	1.5	0.0	4.6
152	0.3	-0.10	1415	1.7	0.0	4.8
153	0.2	-0.10	1294	1.4	0.0	4.5
154	0.1	-0.10	1179	1.6	0.0	4.7
155	0.0	-0.10	1074	1.4	0.0	4.5
156	-0.1	-0.05	976	2.3	0.0	5.4
157	-0.2	-0.10	889	2.4	0.0	5.5
158	-0.3	-0.05	812	2.9	0.0	6.0
159	-0.4	-0.05	740	2.7	0.0	5.8
160	-0.6	-0.05	624	4.2	1.1	7.3
161	-0.8	-0.05	533	6.5	3.4	9.6
162	-1.1	-0.05	430	7.8	4.7	10.9
163	-1.5	-0.05	341	11.5	8.4	14.6

References

- [BBC News 2019] BBC News. *Gulf Crisis: Are We Heading for a New Tanker War?* BBC News. July 18, 2019. URL: <https://www.bbc.com/news/world-middle-east-48709049> (visited on 05/30/2025) (cit. on p. 1).
- [BIDLINGMAIER 1971] Gerhard BIDLINGMAIER. *KM Admiral Graf Spee. Warship Profile No. 4.* Windsor: Profile Publications, 1971 (cit. on p. 1).
- [BLAIR 1996] Clay BLAIR. *Hitler's U-Boat War: The Hunters, 1939–1942.* New York: Random House, 1996 (cit. on p. 1).
- [CARLUCCI and JACOBSON 2014] Donald E. CARLUCCI and Sidney S. JACOBSON. *Ballistics: Theory and Design of Guns and Ammunition.* 2nd ed. Boca Raton: CRC Press/Taylor & Francis, 2014 (cit. on pp. 2, 4, 5, 12–15, 18–20).
- [CONNORS 2019] Matthew CONNORS. *The Kriegsmarine and Compound War at Sea in WWII.* Acesso em: 30 mai. 2025. Center for International Maritime Security. Jan. 2019. URL: <http://cimsec.org/the-kriegsmarine-and-compound-war-at-sea-in-wwii/> (cit. on p. 1).
- [CORRIVEAU 2017] Daniel CORRIVEAU. “Validation of the nato armaments ballistic kernel for use in small-arms fire control systems”. *Defence Technology* 13.3 (2017), pp. 188–199. DOI: [10.1016/j.dt.2017.04.006](https://doi.org/10.1016/j.dt.2017.04.006) (cit. on p. 4).
- [CROSBY 2024] Andrew CROSBY. *Asymmetric Naval Strategies: Overcoming Power Imbalances to Contest Sea Control.* Center for International Maritime Security (CIMSEC). Apr. 24, 2024. URL: <https://cimsec.org/overcoming-relative-naval-power-weakness-to-contest-command-of-the-sea/> (visited on 05/30/2025) (cit. on p. 1).
- [DiGIULIAN 2012] Tony DiGIULIAN. *United States of America 5“/38 (12.7 cm) Mark 12.* Ficha técnica do canhão naval 5“/38, incluindo dados de cadência de tiro e elevação. NavWeaps.com. 2012. URL: http://www.navweaps.com/Weapons/WNUS_5-38_mk12.php (visited on 11/17/2025) (cit. on pp. 27, 31, 33).
- [GKRITZAPIS *et al.* 2007] Dimitris GKRITZAPIS, Elias PANAGIOTOPoulos, Dionysios P. MARGARIS, and Dimitris G. PAPANIKAS. “A six degree of freedom trajectory analysis of spin-stabilized projectiles”. In: *AIP Conference Proceedings*. Vol. 963. 1. AIP Publishing, 2007, pp. 1187–1194. DOI: [10.1063/1.2835958](https://doi.org/10.1063/1.2835958) (cit. on p. 2).

- [GOLDSTEIN *et al.* 2002] Herbert GOLDSTEIN, Charles P. POOLE, and John L. SAFKO. *Classical Mechanics*. 3rd ed. Addison-Wesley (Pearson Education, Inc.), 2002. ISBN: 978-0201657029 (cit. on pp. 5–10, 12, 13).
- [HANDEL-MAZZETTI 1956] Paul HANDEL-MAZZETTI. “The scharnhorst-gneisenau team at its peak”. *U.S. Naval Institute Proceedings* 82.8 (1956), pp. 854–857 (cit. on p. 1).
- [HASELTINE 1969] W. R. HASELTINE. *Yawing Motion of 5.0 Mk 41 Projectile Studied by Means of Yaw Sondes*. Technical Publication NWC TP 4779. AD0862065. China Lake, CA: Naval Weapons Center, Aug. 1969. URL: <https://apps.dtic.mil/sti/tr/pdf/AD0862065.pdf> (visited on 11/13/2025) (cit. on pp. 4, 27).
- [KHALIL *et al.* 2009] M. KHALIL, H. ABDALLA, and O. KAMAL. “Dispersion analysis for spinning artillery projectile”. In: *13th International Conference on Aerospace Sciences & Aviation Technology (ASAT-13)*. Cairo, Egypt: MTC, 2009, pp. 1–12 (cit. on p. 2).
- [KIRICHENKO 2025a] David KIRICHENKO. *Ukraine’s Innovative Defense Tech Sector Is the Country’s Trump Card*. Acesso em: 30 mai. 2025. Atlantic Council. Mar. 2025. URL: <https://www.atlanticcouncil.org/blogs/ukrainealert/ukraines-innovative-defense-tech-sector-is-the-countrys-trump-card> (cit. on p. 2).
- [KIRICHENKO 2025b] David KIRICHENKO. *Ukraine’s Marauding Sea Drones Bewilder Russia*. Acesso em: 30 mai. 2025. Center for European Policy Analysis. Jan. 2025. URL: <https://cepa.org/article/ukraines-marauding-sea-drones-bewilder-russia> (cit. on p. 2).
- [KROESE *et al.* 2011] Dirk P. KROESE, Thomas TAIMRE, and Zdravko I. BOTEV. *Handbook of Monte Carlo Methods*. Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons, 2011. doi: [10.1002/9781118014967](https://doi.org/10.1002/9781118014967) (cit. on pp. 5, 22, 32).
- [LEONARDO DRS 2020] LEONARDO DRS. *76 mm Shipboard Gun – 76/62 Super Rapide Brochure*. 2020. URL: https://www.leonardodrs.com/wp-content/uploads/2023/08/76mm-shipboard_brochure.pdf (visited on 04/05/2025) (cit. on p. 2).
- [LEVI 2014] Mark LEVI. *Classical Mechanics with Calculus of Variations and Optimal Control: An Intuitive Introduction*. Vol. 69. Student Mathematical Library. Providence, RI: American Mathematical Society, 2014. ISBN: 978-0-8218-9138-4 (cit. on pp. 5, 7–9, 12).
- [LIU *et al.* 2011] Yong LIU, Kai LI, Xiaolong SONG, and Xuefeng YAO. “Monte carlo ballistic simulation applied to dispersion analysis of starting control points for antitank missile”. In: *2011 International Conference on Electronics, Communications and Control (EMEIT)*. IEEE. 2011, pp. 2929–2932. doi: [10.1109/EMEIT.2011.6023691](https://doi.org/10.1109/EMEIT.2011.6023691) (cit. on pp. 2, 4).
- [MARDER 1969] Arthur J. MARDER. *From the Dreadnought to Scapa Flow: Volume IV, 1917 – Year of Crisis*. London: Oxford University Press, 1969 (cit. on p. 1).

REFERENCES

- [McCoy 2012] Robert L. McCoy. *Modern Exterior Ballistics: The Launch and Flight Dynamics of Symmetric Projectiles*. Atglen, PA: Schiffer Publishing, 2012 (cit. on pp. 2, 4, 5, 11, 12, 15, 20, 21, 26, 27, 30).
- [MERCOLIANO 2019] Salvatore R. MERCOLIANO. *Escorting in the Persian Gulf: Firefighting, Policing, or Bodyguarding?* Acesso em: 30 mai. 2025. Center for International Maritime Security. Aug. 2019. URL: <http://cimsec.org/escorting-in-the-persian-gulf-firefighting-policing-or-bodyguarding/> (cit. on p. 1).
- [NAVIAS and HOOTON 1996] Martin S. NAVIAS and E. R. HOOTON. *Tanker Wars: The Assault on Merchant Shipping during the Iran–Iraq Conflict, 1980–1988*. London: I.B. Tauris, 1996 (cit. on p. 1).
- [PHILLIPS 2024] Stephen PHILLIPS. *Revisiting the Tanker War*. Acesso em: 30 mai. 2025. War on the Rocks. Mar. 2024. URL: <https://warontherocks.com/2024/03/revisiting-the-tanker-war/> (cit. on p. 1).
- [PODER NAVAL 2023] PODER NAVAL. *Os navios da classe Garcia na Marinha do Brasil*. Portuguese. Artigo com histórico e dados de serviço dos navios da classe *Garcia* na Marinha do Brasil. Oct. 29, 2023. URL: <https://www.naval.com.br/blog/2023/10/29/os-navios-classe-garcia-na-marinha-do-brasil/> (visited on 11/13/2025) (cit. on p. 28).
- [Ross 2014] Sheldon M. Ross. *Introduction to Probability Models*. 11th ed. Oxford: Academic Press, 2014. ISBN: 9780124080809 (cit. on pp. 5, 22, 23, 32).
- [SELIGMAN and BERG 2023] Lara SELIGMAN and Matt BERG. *A \$2M missile vs. a \$2,000 drone: Pentagon worried over cost of Houthi attacks*. 2023. URL: <https://www.politico.com/news/2023/12/19/missile-drone-pentagon-houthi-attacks-iran-00132480> (visited on 04/04/2025) (cit. on p. 2).
- [SUPERVISOR OF SHIPBUILDING (SUPSHIP), SEATTLE, WASHINGTON 1965] SUPERVISOR OF SHIPBUILDING (SUPSHIP), SEATTLE, WASHINGTON. *USA – DEG-1 USS Brooke: Booklet of General Plans*. Booklet of General Plans. Guided missile escort destroyer, Brooke-class. Seattle, WA: U.S. Navy, Dec. 1965. URL: <https://archive.org/details/deg1bogp1965> (visited on 11/13/2025) (cit. on pp. 4, 28, 29).
- [SUTTON 2024] H. I. SUTTON. *Overview of Maritime Drones (USVs) of the Russo–Ukraine War*. Artigo em linha no *Covert Shores* com dimensões e descrições técnicas de diversos USVs, incluindo o *Sea Baby*. 2024. URL: <http://www.hisutton.com/Russia-Ukraine-USVs-2024.html> (visited on 11/13/2025) (cit. on pp. 4, 28, 31).
- [THE KYIV INDEPENDENT 2024] THE KYIV INDEPENDENT. *Crowdfunding campaign raises \$7.8 million for Ukrainian Sea Baby naval drones*. Notícia que cita o custo unitário aproximado de um drone *Sea Baby* (8,5 milhões de hryvnias) e suas características básicas. Feb. 23, 2024. URL: <https://kyivindependent.com/crowdfunding-campaign-raises-7-8/> (visited on 11/13/2025) (cit. on pp. 4, 28).

- [THE KYIV INDEPENDENT 2025] THE KYIV INDEPENDENT. 'Successful' Ukrainian naval drone strike disables 2 Russian shadow fleet tankers, source says. Nov. 29, 2025. URL: <https://kyivindependent.com/ukrainian-naval-drones-behind-successful-strike-on-2-russian-shadow-fleet-tankers/> (visited on 11/29/2025) (cit. on p. 2).
- [THOMAS and GOSWICK 1967] M. A. THOMAS and T. E. GOSWICK. *Naval Gunfire Dispersion*. Technical Memorandum K-26/67. AD0651998. Dahlgren, VA: U.S. Naval Weapons Laboratory, May 1967. URL: <https://apps.dtic.mil/sti/trecms/pdf/AD0651998.pdf> (visited on 11/13/2025) (cit. on p. 4).
- [TREVITHICK 2024] Joseph TREVITHICK. *Phalanx CIWS Costs \$3,500 Per Second in Ammo To Fire*. 2024 (cit. on p. 2).
- [U.S. NAVAL INSTITUTE 1989] U.S. NAVAL INSTITUTE. "Garcias and brookes: one last look". *Naval History Magazine* (Apr. 1989). Artigo com panorama histórico e estimativas de custo das classes *Garcia* e *Brooke*. URL: <https://www.usni.org/magazines/naval-history-magazine/1989/april/garcias-and-brookes-one-last-look> (visited on 11/13/2025) (cit. on p. 28).
- [GAO 1990] UNITED STATES GENERAL ACCOUNTING OFFICE. *Defense Budget: Potential Reductions to DOD's Ammunition Budgets*. Report GAO/NSIAD-90-256. Washington, DC: U.S. General Accounting Office, Sept. 1990. URL: <https://www.gao.gov/assets/nsiad-90-256.pdf> (visited on 11/13/2025) (cit. on pp. 4, 27).
- [US INFLATION CALCULATOR 2025] US INFLATION CALCULATOR. *Inflation Calculator / Find US Dollar's Value From 1913–2025*. Calculadora de inflação baseada em dados oficiais do U.S. Bureau of Labor Statistics. 2025. URL: <https://www.usinflationcalculator.com/> (visited on 11/13/2025) (cit. on pp. 27, 28).
- [WHYTE 1973] Robert H. WHYTE. *SPIN-73: An Updated Version of the SPINNER Computer Program*. Technical Report TR-4588. AD0915628. Dover, NJ: Picatinny Arsenal, Nov. 1973. URL: <https://apps.dtic.mil/sti/citations/AD0915628> (visited on 11/13/2025) (cit. on pp. 4, 25, 26).
- [WIKIPEDIA CONTRIBUTORS 2025] WIKIPEDIA CONTRIBUTORS. *5-inch/54 caliber Mark 42 gun*. Artigo com dados básicos de desempenho do canhão naval 5-inch/54 Mark 42. Wikipedia, The Free Encyclopedia. 2025. URL: https://en.wikipedia.org/wiki/5-inch/54-caliber_Mark_42_gun (visited on 11/20/2025) (cit. on p. 27).