Airbus Sloshing Rocket Workshop

Final Design Report

Team ThrustERS

Parth Jain
Vedansh Mittal
Vedang Bhosale
Aiyaz Karani
Dhruva Karanth

Word Count: 8585 (as calculated using Overleaf's LaTeX word count feature)

June 2025

Contents

List of Tables

List of Figures

Nomenclature

A_e	Nozzle Exit Area
$ASRW$	Airbus Sloshing Rocket Workshop
BWB	Blended Wing Body
c	Mean Aerodynamic Chord
CDR	Conceptual Design Report
CFD	Computational FLuid Dynamics
C_D	Drag Coefficient
C_L	Lift Coefficient
CG	Centre of Gravity
CP	Centre of Pressure
D	Drag
DEM	Discrete Element Method
$DFMA$	Design for Manufacture and Assembly
$DFMEA$	Design Failure Modes and Effects Analysis
DoF	Degrees of Freedom
E_t	Total energy per unit mass
$FMEA$	Failure Modes and Effects Analysis
FOS	Factor of Safety
g	Acceleration due to Gravity
h	Height from Reference Level
L	Lift
L/D	Lift-to-Drag Ratio
$\dot{m}$	Mass Flow Rate
MPS	Moving Particle Semi-Implicit
NP	Neutral Point
P	Pressure
P_a	Ambient Pressure
P_e	Pressure at Nozzle Exit
$P - Diagram$	Parameter Diagram
PMD	Propellant Management Devices
PP	Polypropylene
p	Pressure
QFD	Quality Function Deployment
q_x, q_y, q_z	Heat flux components in x, y, and z directions
Pr	Prandtl number
Re	Reynolds number
RPN	Risk Priority Number
S	Wing Area
SM	Stability Margin
T	Thrust
t	Time
u, v, w	Velocity components in x, y, and z directions respectively
v	Flow Velocity
v_e	Velocity of Water at Nozzle Exit
VOF	Volume of Fluid
x, y, z	Cartesian spatial coordinates
X_{CG}	Location of CG
X_{NP}	Location of Neutral Point

α	Angle of Attack
ρ	Fluid Density
$\tau_{xx}, \tau_{yy}, \tau_{zz}$	Normal viscous stress components
$\tau_{xy}, \tau_{xz}, \tau_{yz}$	Shear viscous stress components
UPVC	Unplasticised Polyvinyl Chloride
NACA	National Advisory Committee for Aeronautics
PETG	Polyethylene Terephthalate Glycol
FEA	Finite Element Analysis
NPS	Nominal Pipe Size
PLA+	Polylactic Acid Plus

Abstract

The challenge of liquid propellant sloshing remains a critical concern for achieving stable and efficient flight in water rocket systems. This Final Design Report, prepared for the Airbus Sloshing Rocket Workshop 2025, documents the comprehensive development of a fixed-wing water rocket optimised to meet stringent mission requirements for range, payload, and flight stability. Building on prior conceptual studies, multiple configurations—including glider, blended wing body, active control folding wing, and fixed-wing—were systematically analysed through aerodynamic simulations, stability assessments, and structural evaluations. The fixed-wing configuration was ultimately selected for its balance of simplicity, manufacturability, and robust performance. Extensive trajectory and sloshing simulations were conducted to refine the design and predict in-flight behaviour. The project leveraged Commercial Off-The-Shelf (COTS) components and modular construction to streamline manufacturing and assembly, reducing both time and complexity. Project management tools were employed to coordinate interdisciplinary efforts, track progress, and ensure timely delivery. This report presents the detailed design process, simulation results, and manufacturing strategy, culminating in a water rocket platform that demonstrates innovation, reliability, and practical solutions to the challenges of liquid sloshing and mission fulfilment.

About the Team

We are team ThrustERS, an ensemble of visionary innovators who not only dared to dream big but also transformed those dreams into groundbreaking realities. As alumni of thrustMIT (ThrustMIT, 2025), we have come together once again, driven by our shared passion for aerospace, to contribute meaningfully to this competition. Our newly adopted identity, ThrustERS, stands for Thrust Engineering Research Squad, a designation that formally reflects our unwavering commitment to engineering excellence and pioneering aerospace research.

Last year, with our rocket 'AgniAstra' (Jain et al., 2024) in the 30K COTS category at the Spaceport America Cup 2024 (Figure 32), we demonstrated what it means to excel under pressure. Our rocket soared to 17,010 feet, crossed the sound barrier at 1.8 Mach, and earned us the top spot in Asia and an impressive 7th place globally in the technical design report among 152 teams. This was not just a competition; it was a powerful testament to our ability to innovate with indigenous design and technology, all crafted within our college workshop.

Our success caught the eye of global aerospace experts and was celebrated by the Government of India in their MyGov (MyGov, 2024) blog, marking us as trailblazers in the field. Alongside our competition accolades, our numerous filed patents in aerospace technology underscore our ongoing commitment to pushing boundaries.

While we are proud of our accomplishments and the accolades that came with them, our true strength lies in our humility and relentless drive to keep learning. As we join forces once more, we do so with a spirit of camaraderie and a determination to make every moment count. This is our one last flight, and we are ready to soar.

Figure 1: Members of thrustMIT at the poster session during the Spaceport America Cup 2024.

Figure 2: Project *AgniAstra* on the launch pad at Spaceport America, New Mexico, USA.

1 Introduction

Sloshing represents a significant challenge in the aerospace industry, directly affecting vehicle stability, performance, and safety. This comprehensive literature review examines the current state of knowledge regarding sloshing dynamics in aerospace applications, innovative mitigation strategies, and emerging technologies that address this complex fluid-structure interaction problem. Through analysis of contemporary research, this review identifies that sloshing effects can cause catastrophic mission failures. Sloshing occurs when the free surface of a liquid moves within a partially filled container, thereby creating dynamic forces that can destabilise the vehicle (Ibrahim, 2015).

The physics underlying sloshing behaviour in aerospace vehicles involves complex fluid-structure interactions that are fundamentally altered by the gravitational environment. In Earth-based applications, gravity provides a natural settling mechanism that keeps liquid at the bottom of containers, but in reduced gravity environments typical of orbital operations, surface tension forces become dominant (Hartwig, 2017).

Historical analysis reveals that sloshing has been a contributing factor in numerous aerospace failures, including the Falcon 1 second test flight anomaly (Veldman et al., 2007). The coupling between slosh dynamics and vehicle attitude control systems can lead to resonance conditions where small disturbances are amplified, potentially resulting in structural failure or complete loss of vehicle control (Mason & Starin, 2018). The transition from terrestrial to space environments fundamentally alters the nature of liquid behaviour within propellant tanks. In microgravity conditions, liquid distribution is primarily governed by surface tension effects rather than gravitational forces, leading to complex three-dimensional liquid configurations that can dramatically affect vehicle dynamics. Research has shown that in reduced gravity environments ranging from 10^{-2} to $10^{-4}\,g$, liquid tends to adhere to tank walls while leaving gaseous bubbles in the centre of the vessel (Antar & Nuotio-Antar, 1994).

Modern research in aerospace sloshing has evolved to encompass both fundamental fluid dynamics studies and practical engineering solutions for real-world applications. Recent computational studies have demonstrated the effectiveness of advanced simulation techniques in predicting sloshing behaviour, with VOF methods emerging as particularly powerful tools for analysing complex free surface flows (Al-Yacouby & Ahmed, 2022). These computational approaches have enabled detailed analysis of baffle effectiveness, showing that perforated baffles can significantly reduce sloshing pressure compared to imperforate designs. Advances in active sloshing control have opened new possibilities for real-time mitigation of sloshing effects. Force feedback control systems have been demonstrated to effectively manage slosh resonances through closed-loop control strategies (Jetzschmann et al., 2018).

PMDs represent the most mature approach to addressing sloshing challenges, with designs that have been refined through decades of flight experience. These devices operate on the principle of ensuring vapor-free propellant delivery to engines while simultaneously controlling the motion of liquid within tanks (Chato et al., 2014).

Vane-type PMDs utilise surface tension effects to guide propellant flow and have demonstrated particular effectiveness in low-acceleration environments with moderate flow rate requirements. These devices consist of thin plates that extend partially or fully into the tank volume, creating preferred flow paths for liquid while damping sloshing motions. Recent numerical analysis and microgravity experiments have confirmed that vane-type PMDs possess excellent fluid management capabilities and meet the demands of orbital propellant systems (Naseh et al., 2019).

Bladder tank configurations represent another mature technology that effectively eliminates free surface sloshing by containing propellant within a flexible membrane. These systems utilise pressurant gas to squeeze the bladder, forcing propellant through perforated standpipes while maintaining positive pressure to the engine (Space Propulsion, n.d.).

The quest for improved sloshing control has driven exploration of innovative technologies that draw inspiration from diverse fields beyond traditional aerospace engineering. Hybrid magneto active propellant management devices represent a cutting-edge approach that combines traditional mechanical elements with smart materials capable of real-time response to changing conditions.

Floating foam technologies have emerged as a promising passive approach to sloshing mitigation, with recent research demonstrating their effectiveness through coupled MPS and DEM simulations. These systems utilise foam balls that float on the liquid surface, providing distributed damping of surface waves while maintaining compatibility with propellant chemistry requirements (Janszen et al., 2019).

Surface tension management techniques inspired by biological systems have shown potential for creating more efficient propellant acquisition devices. These approaches utilise microstructured surfaces that enhance capillary forces, enabling more effective liquid positioning and reduced sloshing sensitivity (SPIE, 2024).

The various factors contributing to sloshing, as identified in this literature review, are systematically summarised using a cause-and-effect (fishbone) diagram, with sloshing as the primary effect. This is illustrated in Figure 3.

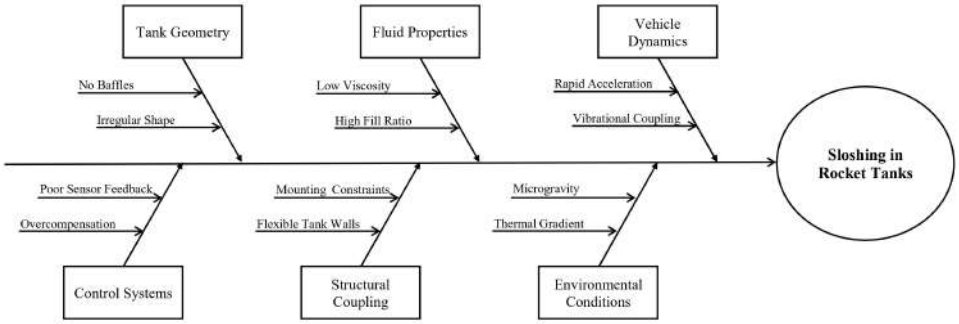


Figure 3: Fishbone Diagram for Sloshing Phenomenon

For ASRW, our team aims to develop a bottle rocket design that addresses sloshing challenges while maximising flight performance. This report chooses the fixed-wing configuration, out of the four designs discussed during the CDR, as the most viable option to meet competition requirements based on stability, manufacturability, and sloshing mitigation potential.

The primary objectives of this final phase are as follows:

1. Present a comprehensive analysis of the fixed-wing rocket's aerodynamic profile, flight stability, structural design, slosh dynamics, and propulsion system, based on simulation results and physical testing.

2. Document the complete design and development process—from configuration selection and CAD modelling to subsystem integration, manufacturing, and experimental validation.

3. Demonstrate how sloshing mitigation has been incorporated into the final design, supported by comparative analysis with a non-mitigated baseline.

4. Assess flight performance and recovery outcomes through recorded test results, including evaluation of stability margins, apogee, and glide efficiency.

5. Reflect on project execution in terms of planning, timeline adherence, team coordination, and adaptability to challenges faced during development.

6. Deliver a validated and competition-ready fixed-wing rocket design that satisfies all specified functional and performance requirements of the ASRW 2025.

2 Project Organisation

2.1 Team Organisation

The project team is structured to leverage the specific expertise of each member, ensuring comprehensive coverage of all project facets. Parth, serving as the Team Lead, oversees the overall project coordination, contributes to the CAD assembly, overall design, manufacturing, and report writing. Vedansh is responsible for the complete systems engineering of the project and also assists in report compilation. The core technical development is distributed as follows:

Vedang is responsible for the rocket design, trajectory simulations, and the application of machine learning techniques. Aiyaz along with Vedang has developed the 3-DoF code for flight analysis. Dhruva is focused on sloshing simulations and the design of the launcher mechanism. This specialised approach ensures that critical areas such as aerodynamics, structural integrity, propulsion, control systems, and advanced simulations are managed by team members with relevant expertise. Manufacturing and testing activities are collaborative, drawing upon the skills of multiple team members as required.

2.2 Project Timeline

The project timeline outlines the major phases and milestones for the bottle rocket's design, construction, and testing. Each phase is structured sequentially to allow sufficient time for conceptual development, design validation, and implementation. The Gantt chart visually communicates the planned duration and logical flow of tasks to support efficient project management and adherence to competition deadlines. Figure 4 shows the division of tasks across the project duration.

2.3 Contingency Plans

1. Several potential challenges could lead to delays in the project timeline. These include:
   - Design/Construction Challenges: Unexpected complexities in the design or difficulties encountered during construction.
   - Weather-Related Delays: Unfavourable weather conditions that could impede flight testing.
2. To mitigate these risks, the following contingency plans and buffers will be implemented:
   - Buffer Time: Allocating a buffer of one week between major milestones to accommodate minor delays.
   - Backup Plans: Having alternative design solutions or material suppliers identified for critical components.

2.4 Progress Since CDR

Since the CDR, significant progress has been made, particularly in the practical application and validation of initial designs. The team has finalised the rocket configuration and completed most simulation frameworks, including 3-DoF (Appendix C) and sloshing analyses. The CAD model is fully assembled, and hardware manufacturing has commenced. Initial vector testing has also been successfully conducted. While the core plan remains intact, some buffer periods

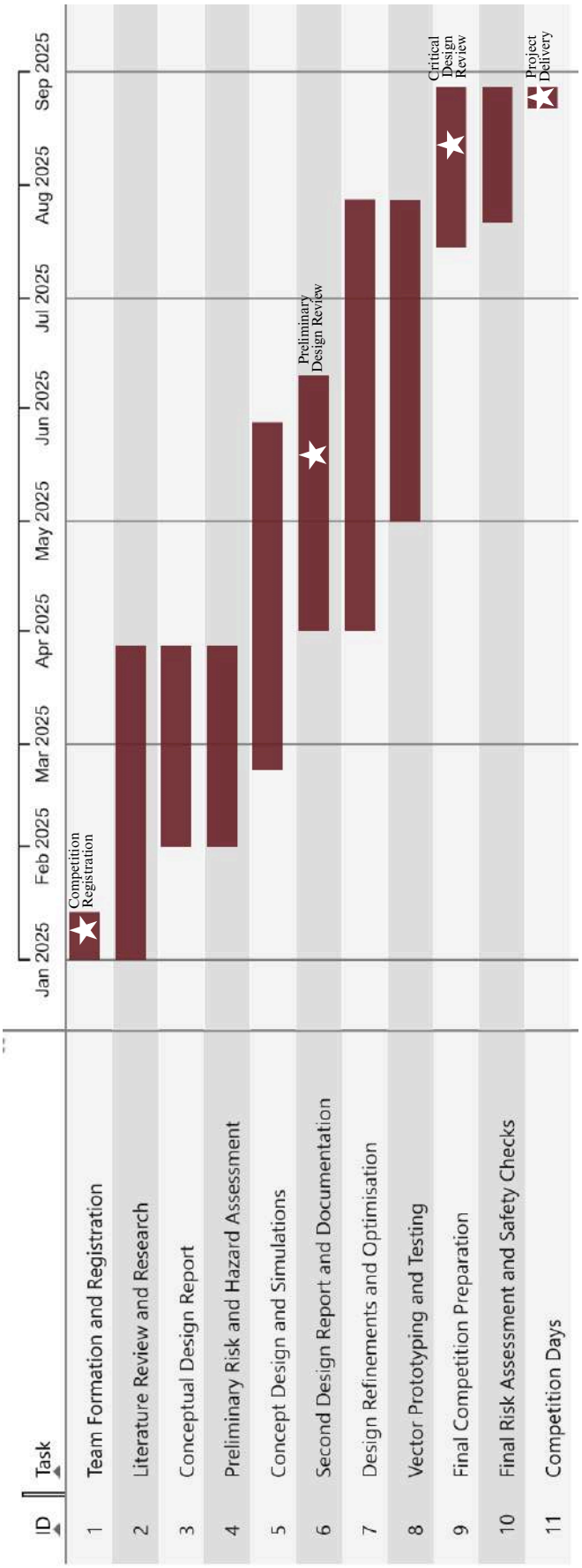


Figure 4: Project Timeline in the Form of Gantt Chart

were used to accommodate iterative simulation tuning and the integration of sloshing dynamics into the system model.

3 Requirements Capture

3.1 Design Requirements

Based on guidelines and our analysis of sloshing challenges in rocket systems, we have established the requirements for our bottle rocket design outlined in Table 1.

Table 1: Systems Requirements

Category	**Requirement Description**	**Type**	**Verification Method**
Functional	Rocket shall launch vertically from the designed launcher.	Mandatory	Test
Functional	Rocket shall safely ascend and descend.	Mandatory	Test, Observation
Functional	Incorporate passive slosh mitigation feature(s).	Mandatory	Inspection
Constraint	Max operating pressure: [147 psi].	Mandatory	Test Procedure
Constraint	Water fill volume Sloshing Tank: [50%].	Mandatory	Procedure
Constraint	Adhere to all safety regulations provided by organisers.	Mandatory	Compliance Check
Constraint	Utilise readily available and low-cost materials.	Optional	BOM Review
Performance	Maximise total flight time (launch to ground).	Mandatory	Test (Timing)
Performance	Maintain stability during powered ascent (minimal wobble/tumble).	Mandatory	Test (Observation)
Performance	Maintain stability or controlled flight during descent.	Mandatory	Test (Observation)
Performance	Slosh mitigation shall demonstrably reduce instability compared to baseline.	Optional	Comparative Test

3.2 Assumptions

1. Environmental conditions during the workshop, such as wind speed and direction, will be within acceptable limits for conducting flight tests.

2. Manufacturing facilities can produce components with the required precision.

3.3 Success Criteria

The rocket shall be considered successful if all of the following criteria are met:

1. Stable Powered Flight: The rocket must ascend without exhibiting more than $\pm 10^{\circ}$ angular deviation from the vertical axis and without significant wobble or tumbling during the propulsion phase.

2. Minimum Apogee: The rocket must reach a minimum apogee of 12 metres, measured using the onboard altimeter.

3. Controlled Descent: The rocket must descend in a gliding trajectory, without free-fall or uncontrolled tumbling during any part of descent..

4. Structural Integrity: The rocket must not suffer any structural failure. Upon recovery, all major structural components must remain intact with no fracture or deformation affecting reusability.

5. Complete Recovery: The rocket must be recovered as a single piece or with reassemblable components. Loss of critical systems (e.g., wings, tank, fins) during flight results in failure.

6. Slosh Mitigation Performance: The implemented slosh mitigation feature must result in at least 20% reduction in observed oscillation amplitude or attitude deviation compared to a baseline flight without mitigation, validated by comparative testing or simulation.

4 Concept Design

This section explains how each concept chosen is derived from design requirements and explores how the design concepts meet or fall short of the requirements outlined by the team. The aerodynamic design and analysis of the rocket configurations are conducted using xflr5 software. Each concept aims to address specific design requirements:

1. Glider Configuration (Figure 5): The high aspect ratio wing maximises glide time, while the streamlined body reduces drag.

2. BWB Configuration (Figure 6): Enhances glide time and ascent efficiency through an integrated aerodynamic design, addressing altitude and glide time requirements.

3. Active Control Folding Wing Configuration (Figure 7): Aims for efficient ascent and controlled glide, meeting altitude and glide time criteria.

4. Fixed Wing Configuration (Figure 8): Balances ascent and glide, ensuring altitude compliance while providing stability. The fixed wings contribute to stable ascent.

4.1 Concept 1: Glider Configuration

- Aerodynamic Design: maximises lift-to-drag ratio for extended glide time using long, slender wings to enhance lift while reducing drag. The streamlined fuselage minimises parasitic drag.

- Flight Performance: Rapid ascent followed by a controlled gliding descent. Glide efficiency depends on wing design and achieved altitude.

- Stability: Longitudinal stability requires NP behind CG, accounting for CG shift due to sloshing. Lateral stability is ensured via vertical stabiliser.

- Control Mechanisms: Passive stability through fixed aerodynamic surfaces.

- Structural Integrity: Must withstand launch loads; high aspect ratio wings may require reinforcement against aerodynamic loads.

- Propulsion Aspects: Standard bottle rocket propulsion using pressurised water. The thrust generated will determine the initial altitude, which directly impacts the potential glide time.

- Innovative Elements: Inspired by sailplanes and high-altitude reconnaissance aircraft.
- Key Uncertainties: Predicting CG shift due to sloshing and optimising wing loading for maximum glide duration.

.

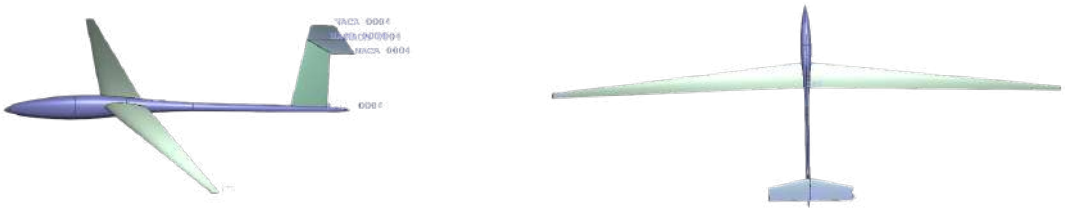


Figure 5: Glider Configuration

4.2 Concept 2: BWB Configuration

- Aerodynamic Design: Integrates wings with the fuselage for enhanced lift, reduced drag, and increased aerodynamic efficiency.
- Flight Performance: Balances ascent efficiency with controlled glide; lift from the wing-body structure moderates descent rate.
- Stability: Requires careful mass and aerodynamic force distribution; sensitive to sloshing effects. Longitudinal and lateral stability must be designed into the shape and stabilisers.
- Control Mechanisms: Uses elevons on the trailing edge for pitch and roll control.
- Structural Integrity: Must endure launch loads and aerodynamic forces across the blended structure.
- Propulsion Aspects: Standard bottle rocket propulsion. The overall aerodynamic efficiency of the BWB might influence the optimal water-to-air ratio for propulsion.
- Innovative Elements: Inspired by stealth aircraft and futuristic passenger aircraft designs.
- Key Uncertainties: Ensuring stable flight and effective control at this scale and sloshing sensitivity.

.

4.3 Concept 3: Active Control Folding Wing Configuration

- Aerodynamic Design: Streamlined ascent phase with folded wings; wings deploy post-propulsion to transition into a glider.
- Flight Performance: High-altitude ascent followed by wing deployment for controlled glide.
- Stability: Stability during ascent relies on rear fins, while transition stability is essential during wing deployment and affected by sloshing.
- Control Mechanisms: Passive ascent stability via fins; deployed wings incorporate control surfaces for active glide control.

Figure 6: BWB Configuration

- Structural Integrity: Requires a robust and reliable wing deployment mechanism; wings must withstand aerodynamic loads.
- Propulsion Aspects: Standard bottle rocket propulsion. The streamlined configuration during ascent should maximise altitude gain for a given amount of propellant.
- Innovative Elements: Inspired by deployable wing mechanisms in military aircraft and insect flight.
- Key Uncertainties: Wing deployment reliability, stability during transition, and control surface effectiveness.

4.4 Concept 4: Fixed Wing Configuration

- Aerodynamic Design: Features permanently attached wings, generating lift throughout ascent and descent.
- Performance: Offers a compromise between a pure rocket and pure glider, with wings providing drag during ascent and lift during descent.
- Stability: Depends on aerodynamic design, wing placement, and stabilisers; continuous aerodynamic forces influence sloshing.
- Control Mechanisms: Stability provided by fixed wings and fins; potential for control surfaces on wings.
- Structural Integrity: Must endure aerodynamic and launch loads.
- Propulsion Aspects: Standard bottle rocket propulsion. The presence of fixed wings might necessitate adjustments to the launch angle and propellant volume for optimal performance.
- Innovative Elements: Draws from early rocket-powered aircraft and winged-missile designs.

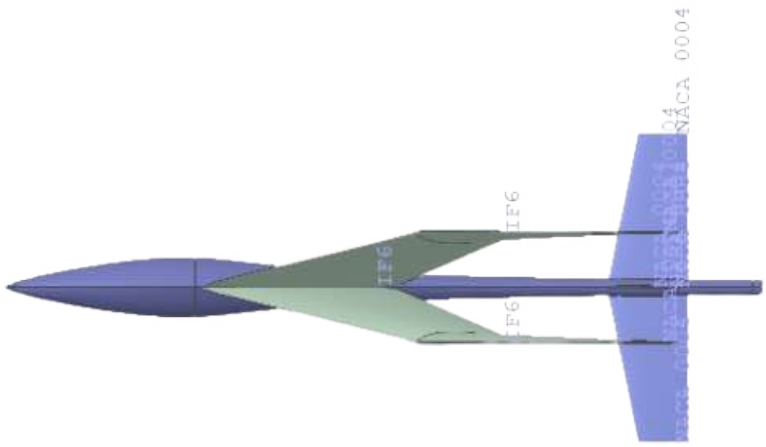


Figure 7: Active Control Folding Wing Configuration

- Key Uncertainties: Balancing ascent performance with glide efficiency and evaluating control surface effectiveness.

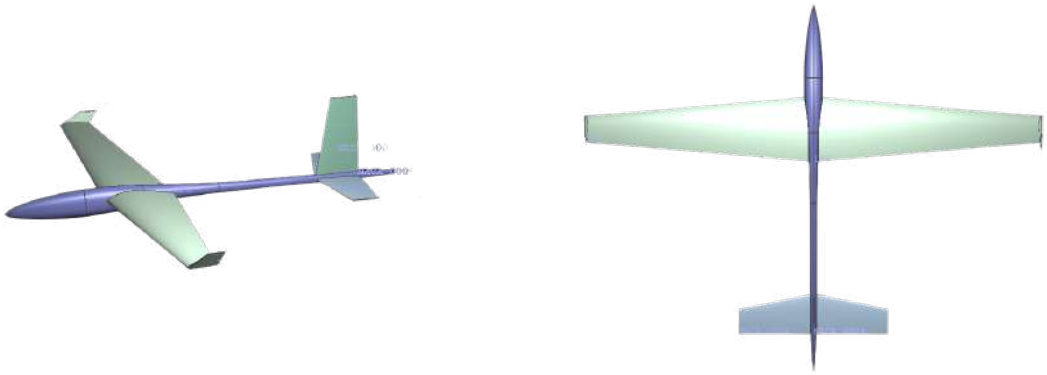


Figure 8: Fixed Wing Configuration

4.5 Concept Selection and Justification

4.5.1 Scientific Justification

- Glide Performance: Assessed through L/D. Higher L/D indicates better glide efficiency. The lift and drag are formulated in Equation 1 and Equation 2 respectively.

$$L = \frac{1}{2}\rho v^2 S C_L \tag{1}$$

$$D = \frac{1}{2}\rho v^2 S C_D \tag{2}$$

  High aspect ratio wings of the glider improve C_L/C_D, enhancing glide efficiency.

- Stability: Governed by CG and NP (Equation 3). For stable flight, NP should be behind CG. Liquid propellant movement can shift CG, causing instability.

$$\text{SM} = \frac{X_{NP} - X_{CG}}{c} \tag{3}$$

- Propulsion: Thrust is determined by pressure, nozzle area, and water mass flow rate (Equation 14). Bernoulli's principle (Equation 4) explains lift generation due to pressure differences across the airfoil.

$$P + \frac{1}{2}\rho v^2 + \rho g h = \text{Constant} \tag{4}$$

- Sloshing Impact: Alters mass distribution and moments of inertia, inducing oscillations and torques that affect stability, particularly during early ascent when propellant mass is significant.

Table 2: Comparison of Rocket Configuration Parameters Across Different Designs

Parameter	**Glider**	**Blended Wing**	**Folding Wing**	**Fixed Wing**
Altitude	Sustained but lower climb	High efficiency, smooth trajectory	Rapid ascent before deployment	Steady ascent, gradual loss due to drag
Lift-to-Drag Ratio	Glide optimisation	Balanced across body structure	Adjustable for different phases	Moderate but limited flexibility
Cost Efficiency	Requires lightweight material	Contoured materials needed	Extra cost due to moving parts	Basic structure, minimal cost
Flight Control	Predictable descent path	Consistent aerodynamics	Dynamically adjustable	Fixed stability, limited manoeuverability
Manufacturing Ease	Precise shaping needed	Requires smooth transition design	Mechanisms adds complexity	Simple construction process
Adaptability	Fixed design, purpose-built	Tailored aerodynamics	Adaptable to maximise ascent and descent	Can be slightly adjusted before launch
Simplicity	No moving parts, CG-sensitive	Some integration complexity	Requires precise deployment mechanism	Straightforward, rigid structure
Stability	Self-stabilising in glide mode	Balanced airflow design	Dynamic stability adjustments	Stable flight at the cost of manoeuverability
Glide Time	Maximised for descent efficiency	Lift sustained efficiently	Optimised for controlled descent	Shorter duration due to drag
Innovation Potential	Improvements in passive aerodynamics	Novel body-wing integration	Efficient deployable mechanics	Optimising traditional designs

	Low
	Moderate
	High
	Very High

Table 2 presents a comparative analysis of the four configurations, while Figure 9 visually represents their performance across key metrics using a radar chart, highlighting the trade-offs

between each design. The performance comparison bar chart in Figure 9 is generated using a weighted scoring methodology applied to the flight score formula. First, each variable in the formula is assigned a weight to reflect its relative importance. Subsequently, for each of the four designs, the relevant parameter values are multiplied by these assigned weights and summed. This procedure gives an aggregate performance score for each design.

4.5.2 Final Design Choice

Although the folding wing configuration initially emerges as a strong candidate due to its superior apogee and range performance, it is ultimately not selected for final implementation. This decision follows a comprehensive evaluation of trade-offs involving structural complexity, mass penalties, and system reliability. The folding mechanism introduces additional electronic subsystems such as actuators, primary and redundant control circuitry, and battery packs, which significantly increase both the total mass and failure risk. Moreover, numerical analyses indicate that the marginal aerodynamic benefits gained during ascent do not justify the added design complexity and power requirements.

During early prototyping and subsystem integration, the manufacturing process further highlighted the impracticalities of the folding wing design. The intricacies of ensuring reliable mechanical deployment under dynamic flight conditions introduced unforeseen challenges in alignment, actuation synchronisation, and structural reinforcement, all of which compounded the mass overhead. Additionally, ground tests revealed that the improvement in glide performance was modest and did not sufficiently offset the increased design and operational complexity. The added risk of a partial or failed deployment posed a significant concern in terms of mission reliability.

As a result, the fixed-wing configuration is chosen for its simplicity, passive stability, and robustness, while still meeting all critical mission requirements within a more manageable and reliable system architecture.

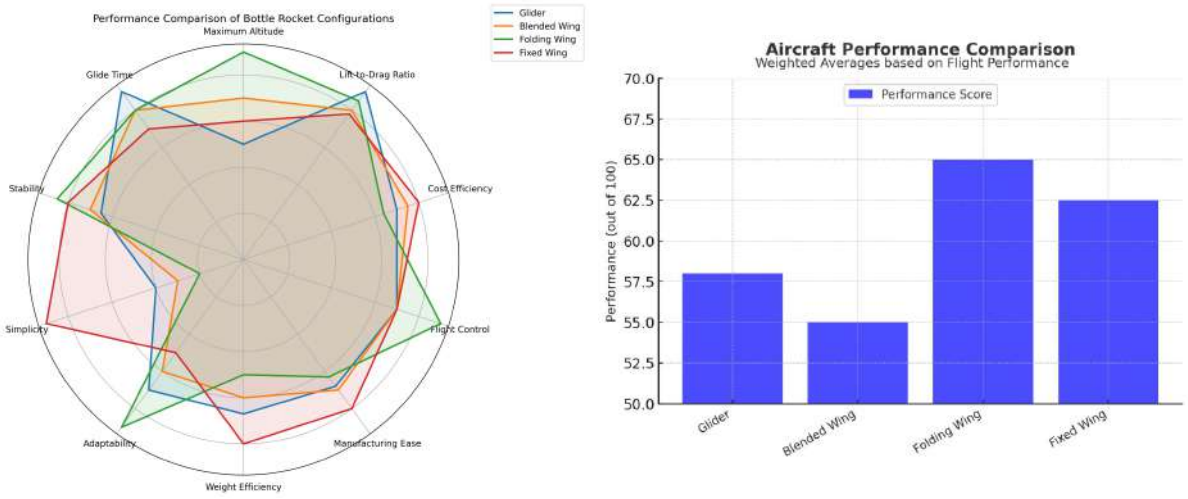


Figure 9: Parameter Comparision for all Concepts

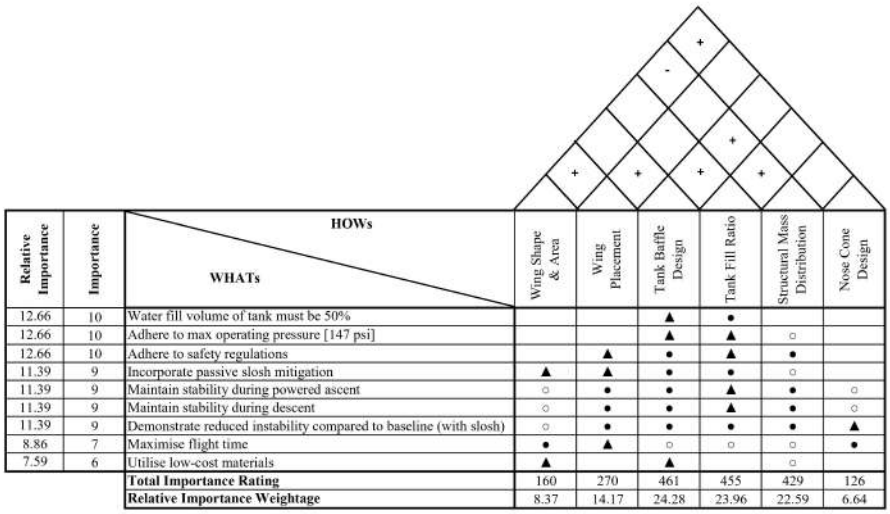
4.6 Quality Function Deployment

To validate that the selected configuration satisfies system-level and mission-critical requirements, a Quality Function Deployment (QFD) analysis was carried out. This structured methodology enables the systematic mapping of requirements ("WHATs") to corresponding design parameters ("HOWs"), thereby supporting early-stage prioritisation and enhancing traceability throughout the development process.

Each requirement was assigned a relative importance score based on its relevance to mission success—such as flight stability, slosh mitigation, and total flight time. Subsequently, the strength of correlation between each requirement and relevant design features—such as wing geometry, baffle configuration, and mass distribution—was evaluated. This approach helped identify the most influential parameters and confirmed that the fixed-wing configuration offers strong alignment with the highest-priority requirements.

The resulting upper triangular correlation matrix revealed several positive interdependencies among design features. In particular, the tank fill ratio, baffle design, and mass distribution demonstrated strong synergies, underscoring their importance in the overall system performance and justifying their selection for targeted optimisation.

The results of the QFD analysis are presented in Figure 10, showcasing the relationship matrix and relative weightings of each parameter.



Relative Importance	Importance	HOWs / WHATs	Wing Shape & Area	Wing Placement	Tank Baffle Design	Tank Fill Ratio	Structural Mass Distribution	Nose Cone Design
12.66	10	Water fill volume of tank must be 50%			▲	•		
12.66	10	Adhere to max operating pressure [147 psi]			▲	▲	○	
12.66	10	Adhere to safety regulations		▲	•	▲	•	
11.39	9	Incorporate passive slosh mitigation	▲	▲	•	•	○	
11.39	9	Maintain stability during powered ascent	○	•	•	▲	•	○
11.39	9	Maintain stability during descent	○	•	•	▲	•	○
11.39	9	Demonstrate reduced instability compared to baseline (with slosh)	○	•	•	•	•	▲
8.86	7	Maximise flight time	•	▲	○	○	○	•
7.59	6	Utilise low-cost materials	▲		▲		○	
		Total Importance Rating	160	270	461	455	429	126
		Relative Importance Weightage	8.37	14.17	24.28	23.96	22.59	6.64

LEGEND	
•	Strong (9)
○	Medium (3)
▲	Weak (1)
+	Strong Synergy
-	Conflict

Figure 10: QFD Matrix Linking Design Parameters to System Requirements

The relative importance weightage highlights 'Tank Baffle Design', 'Fill Ratio', and 'Mass Distribution' as the most influential parameters, guiding final design optimisation and simulation focus. This structured design evaluation confirms that the selected configuration not only addresses key performance requirements but also ensures robustness and safety through a traceable design process.

5 Detailed Design

To guide the detailed design process and ensure robustness across varying flight conditions, a P-Diagram is developed as shown in Figure 21. This diagram captures the relationship between controllable design factors, noise variables, and desired system outputs, providing a structured framework to evaluate how the rocket responds to external disturbances. The identification of error states also aids in prioritising design efforts for slosh mitigation, stability control, and safe recovery.

5.1 Slosh Prediction and Control Strategy

This section presents the results of CFD simulations conducted to predict and control slosh dynamics within a cylindrical tank. Various baffle designs are investigated to understand their

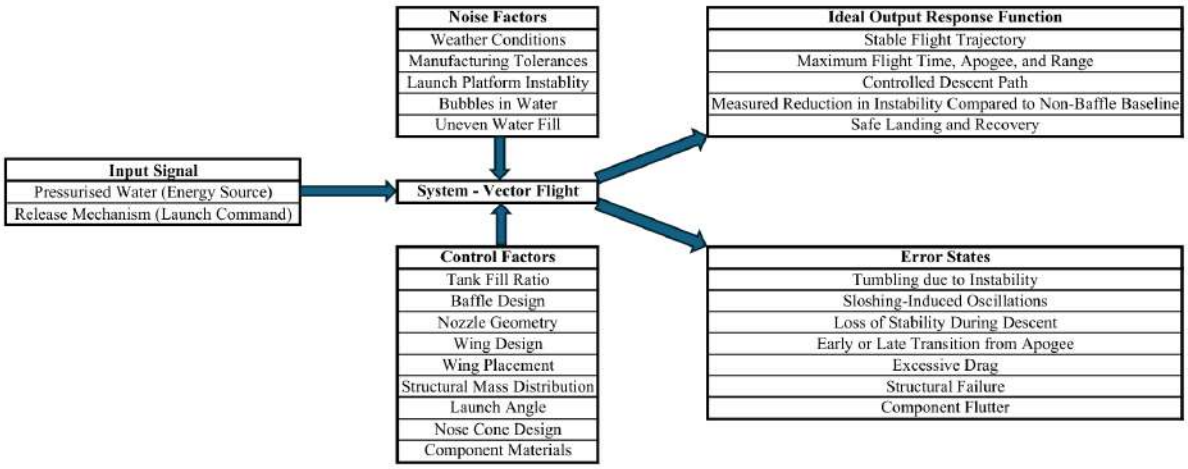


Figure 11: P-Diagram for Fixed-Wing Water Rocket

effectiveness in mitigating slosh-induced forces and velocities.

The internal fluid behaviour during powered ascent and coast phases is governed by the Navier–Stokes equations Equation 5-Equation 9, which describe conservation of mass and momentum in a viscous fluid. These equations serve as the theoretical foundation for the CFD simulations conducted to understand sloshing effects.

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} + \frac{\partial (\rho w)}{\partial z} = 0 \tag{5}$$

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u^2)}{\partial x} + \frac{\partial (\rho uv)}{\partial y} + \frac{\partial (\rho uw)}{\partial z} = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}} \left(\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} \right) \tag{6}$$

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho uv)}{\partial x} + \frac{\partial (\rho v^2)}{\partial y} + \frac{\partial (\rho vw)}{\partial z} = -\frac{\partial p}{\partial y} + \frac{1}{\mathrm{Re}} \left(\frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} \right) \tag{7}$$

$$\frac{\partial (\rho w)}{\partial t} + \frac{\partial (\rho uw)}{\partial x} + \frac{\partial (\rho vw)}{\partial y} + \frac{\partial (\rho w^2)}{\partial z} = -\frac{\partial p}{\partial z} + \frac{1}{\mathrm{Re}} \left(\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \right) \tag{8}$$

$$\begin{aligned} \frac{\partial (E_t)}{\partial t} + \frac{\partial (uE_t)}{\partial x} + \frac{\partial (vE_t)}{\partial y} + \frac{\partial (wE_t)}{\partial z} = & -\frac{\partial (up)}{\partial x} - \frac{\partial (vp)}{\partial y} - \frac{\partial (wp)}{\partial z} - \frac{1}{RePr} \left[\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z} \right] \\ & + \frac{1}{Re} \left[\frac{\partial}{\partial x} (u\tau_{xx} + v\tau_{xy} + w\tau_{xz}) + \frac{\partial}{\partial y} (u\tau_{xy} + v\tau_{yy} + w\tau_{yz}) + \frac{\partial}{\partial z} (u\tau_{xz} + v\tau_{yz} + w\tau_{zz}) \right] \end{aligned} \tag{9}$$

Four distinct baffle configurations are analysed:

1. Single Half Disc Baffle Design (Figure 12): This design features a single, inclined baffle element positioned within the tank.

2. Multiple Rings Baffle Design (Figure 13): This configuration incorporates multiple horizontal ring-shaped baffles at different heights within the tank.

3. Multiple Rings with Holes Baffle Design (Figure 14): This design is an evolution of the ring baffle, featuring multiple horizontal baffles, each perforated with several holes.

4. Multiple Half Discs Baffle Design(Figure 15): This design incorporates multiple, vertically stacked half-disc baffles, creating a helical or staggered flow path within the tank.

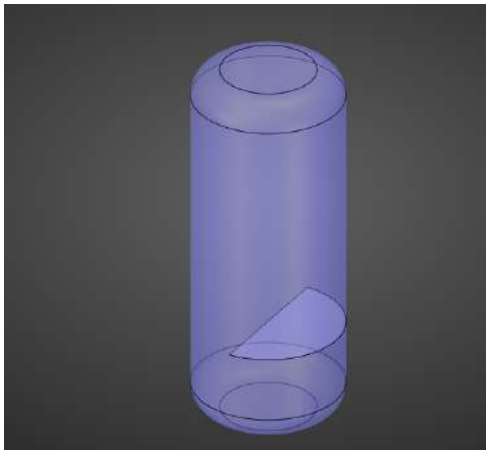


Figure 12: Single Half Disc Baffle Design

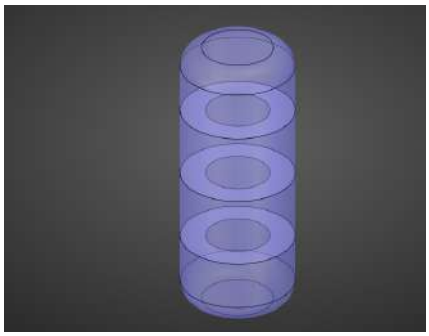


Figure 13: Multiple Rings Baffle Design

5.1.1 Single Half Disc Baffle Design

The plots shown in Figure 16 are discussed below:

- Force Profile: The force plot shows a significant initial peak exceeding 12 N at approximately 1.5 seconds, followed by damped oscillations. While the force eventually stabilises near zero after about 10 seconds, the initial transient is quite high, indicating a strong slosh impact.

- Maximum Velocity Profile: The maximum velocity plot mirrors the force behaviour, exhibiting a sharp initial peak of over 35 m/s at around 1 second. Similar to the force, the velocity gradually dampens but shows considerable fluctuations before settling to a low magnitude after approximately 10-12 seconds.

5.1.2 Multiple Rings Baffle Design

The plots shown in Figure 17 are discussed below:

- Force Profile: Compared to the single inclined baffle, this configuration demonstrates a drastically reduced force magnitude. The forces are oscillatory but remain within a much smaller range, peaking at approximately 0.003 N and largely fluctuating around zero. This indicates significantly improved slosh force mitigation.

- Maximum Velocity Profile: The maximum velocity with ring baffles also shows a substantial reduction in initial peak velocity (around 17.5 m/s) compared to CAD-1. While there is a notable spike around 12 seconds (reaching approximately 24 m/s), the overall velocity magnitudes are significantly lower and more controlled than the single baffle design.

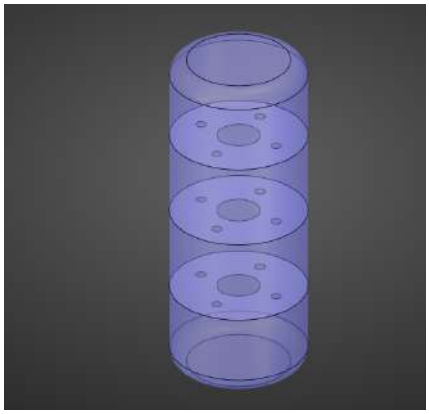


Figure 14: Multiple Rings with Holes Baffle Design

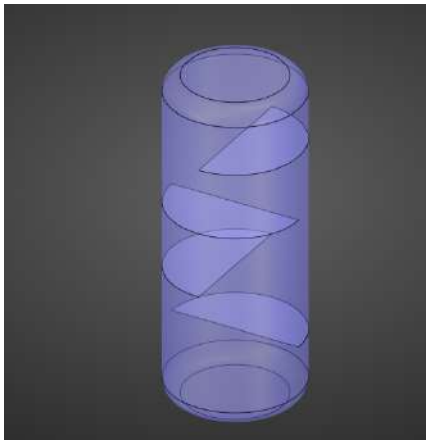


Figure 15: Multiple Half Discs Baffle Design

5.1.3 Multiple Rings with Holes Baffle Design

The plots shown in Figure 18 are discussed below:

- Force Profile: This design effectively controls slosh forces. The force magnitudes are generally very low, fluctuating around zero, with transient peaks not exceeding 0.04 N. This represents the most effective force dampening among the previously tested configurations, with the forces stabilising quickly.

- Maximum Velocity Profile: The maximum velocity profile for this configuration shows a more complex oscillatory pattern than the ring baffles, with peaks around 37 m/s (at approximately 3.5 seconds) and another significant peak near 21 m/s (at approximately 11.5 seconds). While these peaks are comparable to the initial peak of the single inclined baffle, the overall damping appears more consistent and the velocities generally return to lower levels more rapidly after each disturbance compared to the initial single baffle case.

5.1.4 Multiple Half Discs Baffle Design

The plots shown in Figure 19 are discussed below:

- Force Profile: This design exhibits a unique force profile. After some initial small oscillations, there is a very sharp, significant negative force peak exceeding -60 N at approximately 4.5 seconds. Following this large transient, the force rapidly dampens and remains stable near zero. This strong negative peak indicates a substantial momentary impact or suction force.

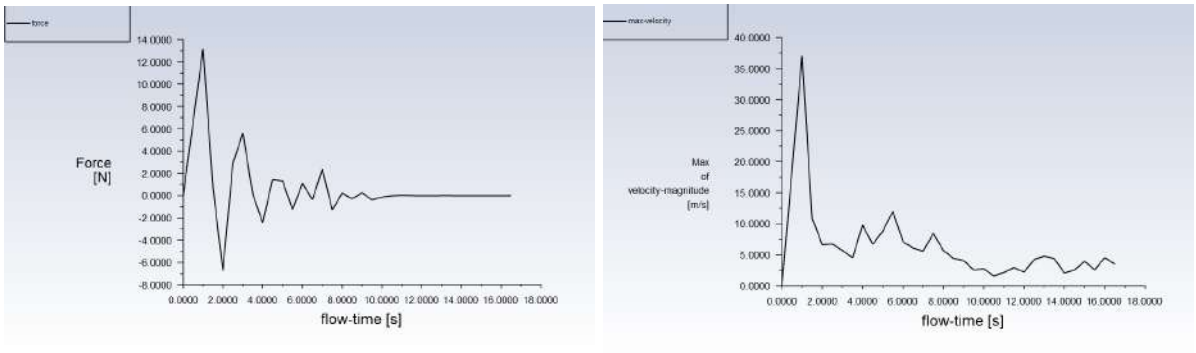


Figure 16: Force and Velocity Curves for Single Half Disc Baffle Design

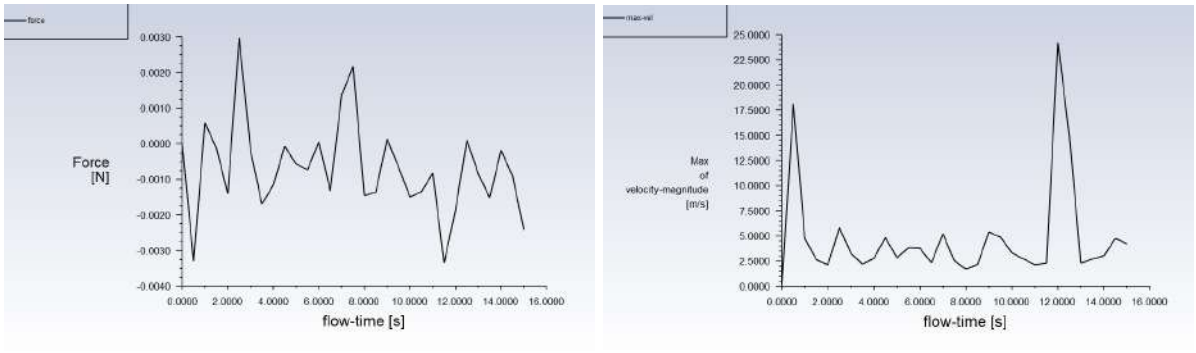


Figure 17: Force and Velocity Curves for Multiple Rings Baffle Design

- Maximum Velocity Profile: The velocity profile shows high initial fluctuations, with peaks around 37 m/s at approximately 1.5 seconds and a very high peak reaching 50 m/s at around 4.5 seconds, coinciding with the large force transient. While the velocity does eventually settle to low magnitudes after approximately 10 seconds, the initial high velocity magnitudes and the extreme peak indicate significant fluid motion.

5.1.5 Overall Evaluation of Baffle Designs

To facilitate a comparative analysis, the key results for each baffle design are summarised in the Table 3.

5.1.6 Discussion and Recommendations

Based on the comprehensive CFD results, the following conclusions are drawn:

- The Single Half disc Baffle Design and the Multiple Half Discs Baffle Design are generally unsuitable for effective slosh control. The Single Half Disc shows high initial forces and velocities, while the Multiple Half Discs design generates an extremely high and undesirable transient force, making it hazardous for structural integrity and stability.

- Both the Multiple Rings Baffle Design and the Multiple Rings with Holes Baffle Design offer significantly superior slosh mitigation. They effectively reduce the magnitude of slosh-induced forces by several orders of magnitude compared to the half-disc designs.

- The Multiple Rings Baffle Design demonstrates the best performance in terms of minimising peak forces and achieving rapid force stabilisation. While it has a moderate peak velocity, its force control is outstanding.

Table 3: Baffle Design Performance Characteristics

Baffle Design	**Peak Force [N]**	**Peak Velocity [m/s]**	**Force Stabilisation Time [s]**	**Velocity Stabilisation Time [s]**	**General Control Performance (Force)**	**General Control Performance (Velocity)**
Single Half Disk Baffle	≈ 13	≈ 37	≈ 10	≈ 12	Poor	Poor
Multiple Rings Baffle	≈ 0.003	≈ 24	≈ 2	≈ 8	Excellent	Good
Multiple Rings with Holes	≈ 0.04	≈ 37	≈ 5	≈ 12	Excellent	Fair
Multiple Half Disk Baffle	≈ 60	≈ 50	≈ 5	≈ 10	Very Poor (High transient)	Poor (High transient)

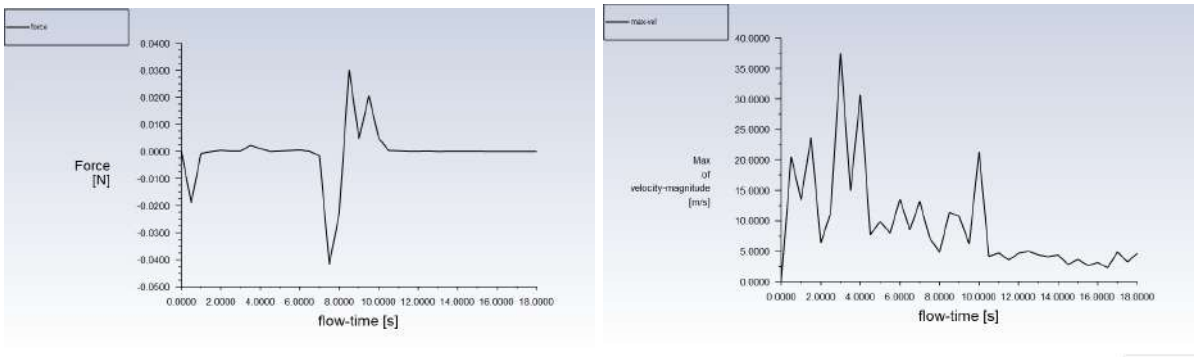


Figure 18: Force and Velocity Curves for Multiple Rings with Holes Baffle Design

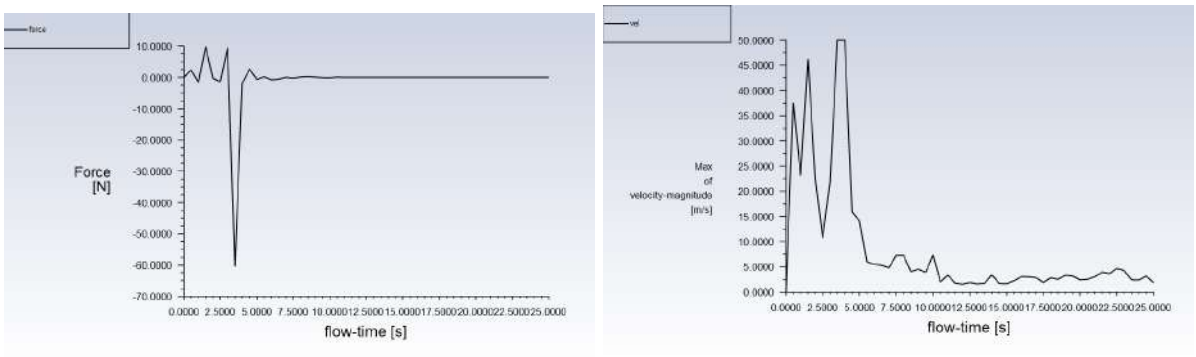


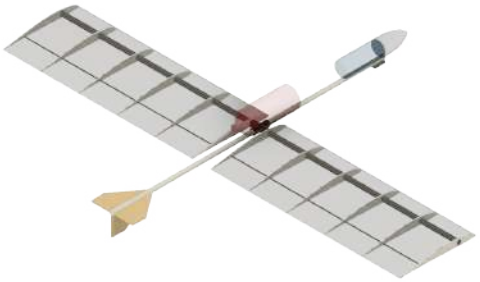
Figure 19: Force and Velocity Curves for Multiple Half Discs Baffle Design

- The Multiple Rings with Holes Baffle Design also performs exceptionally well in force reduction, keeping forces very low. However, its velocity profile exhibits higher, more persistent peaks, suggesting that while it effectively dissipates energy to reduce force, the fluid motion itself remains more energetic for longer periods compared to the simple ring baffles.

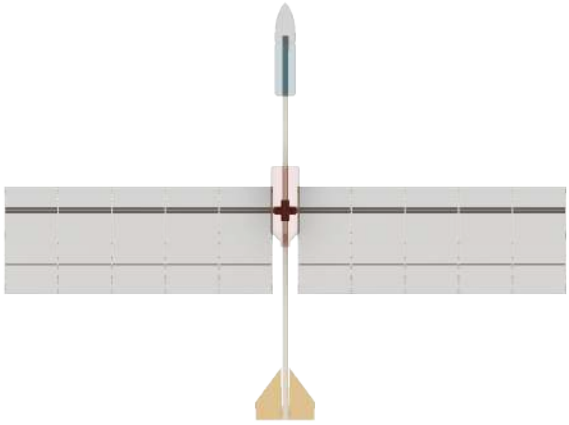
Overall, the Multiple Rings Baffle Design emerges as the most effective and robust solution among the tested configurations for comprehensive slosh control. Its ability to drastically reduce forces and control velocities efficiently makes it the preferred choice for applications where minimising dynamic loads on the tank structure and ensuring fluid stability are paramount. Further optimisation of the number, spacing, and dimensions of these ring baffles could lead to even greater slosh mitigation and improved stability for aircraft fuel tanks or other fluid-carrying compartments.

5.2 Vehicle Geometry and Configuration

The flight vehicle has been modelled with an emphasis on aerodynamic efficiency, structural simplicity, and modular integration of subsystems. The configuration includes a central fuselage housing both the propulsion and sloshing systems, with a rib-and-spar wing structure ensuring lift generation. The tail section comprises a simple stabilising fin. Figure 20 depicts the CAD model of the vehicle from isometric, top, and side perspectives. For assembly and part drawings of the vehicle, refer to Appendix D.



(a) Isometric View



(b) Top View

(c) Side View

Figure 20: CAD model of the final vehicle design showing various orientations.

5.2.1 Wing Design

The fixed-wing structure employs a rib-and-spar construction to balance aerodynamic performance, manufacturability, and structural integrity. The total wingspan of the vehicle is 2 metres, with a constant chord wing design adopted for ease of fabrication and predictable aerodynamic behaviour. The wing consists of six uniformly spaced ribs, each 5 mm thick, fabricated using 3D-printed PLA+. These ribs are shaped according to the Eppler-387 aerofoil profile, chosen for its moderate lift-to-drag ratio and stable aerodynamic characteristics at low Reynolds numbers typical in water rocket flight regimes.

Two UPVC spars run longitudinally through the ribs, serving distinct roles. The primary spar is an NPS 1/2-inch UPVC pipe, which also functions as the mechanical interface between the wing and the fuselage. The secondary spar, of smaller diameter, serves to hold the ribs in alignment and prevent torsional deformation during flight. This dual-spar design provides enhanced rigidity while minimising weight.

Wing-to-fuselage integration is achieved using a UPVC four-way pipe connector. Two opposite outlets connect to the main spars of each wing, while the remaining two interface with the forward and aft fuselage members, ensuring symmetric load transfer and alignment.

The entire wing assembly is wrapped in a heat-shrink PETG film, tensioned using a hot air gun. This covering technique improves surface smoothness, reduces parasitic drag, and adds an extra layer of structural cohesion without significantly increasing mass.

This design approach was validated numerically via FEA. The simulations confirmed acceptable stress distributions and minimal deflection under predicted in-flight loads. Detailed results from the numerical validation are provided in Section 5.5.1.

The modularity of the rib-and-spar design, ease of assembly using standard UPVC fittings, and validated aerodynamic and structural performance make this wing design well-suited for the fixed-wing configuration mission objectives.

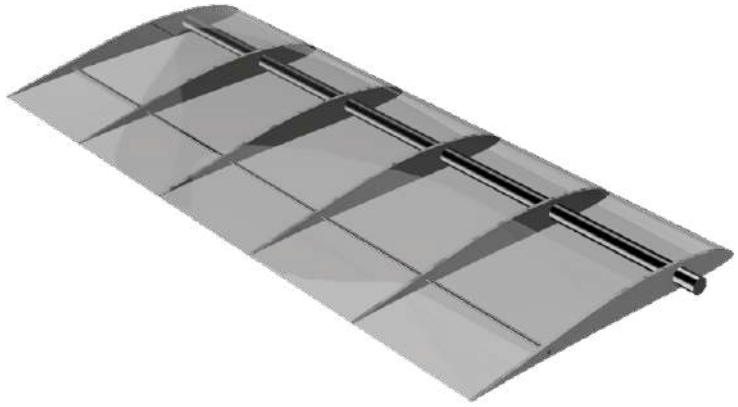


Figure 21: Fixed-Wing Design

To evaluate the aerodynamic performance of different wing geometries, two configurations were analysed: a conventional constant-chord wing and a tapered wing. Flight simulation data obtained from the in-house 3-DoF dynamics code, incorporating sloshing effects, was used to compute the Flight Score metric.

The calculated values are summarised in Table 4. As shown, both designs achieved the same maximum altitude. However, the conventional configuration slightly outperforms the tapered one in terms of range and flight time, leading to a higher overall flight score.

Table 4: Flight performance comparison between tapered and conventional wing configurations

Parameter	Tapered	Conventional
Max Range (m)	15.35	15.94
Max Altitude (m)	9.17	9.17
Flight Time (s)	4.17	4.29
Weight (kg)	2.12	2.12
Flight Score	5.2072	5.3560

Graphical comparisons of the key flight metrics are presented in Figure 22, Figure 23, Figure 24illustrating the marginally improved performance of the conventional wing configuration.

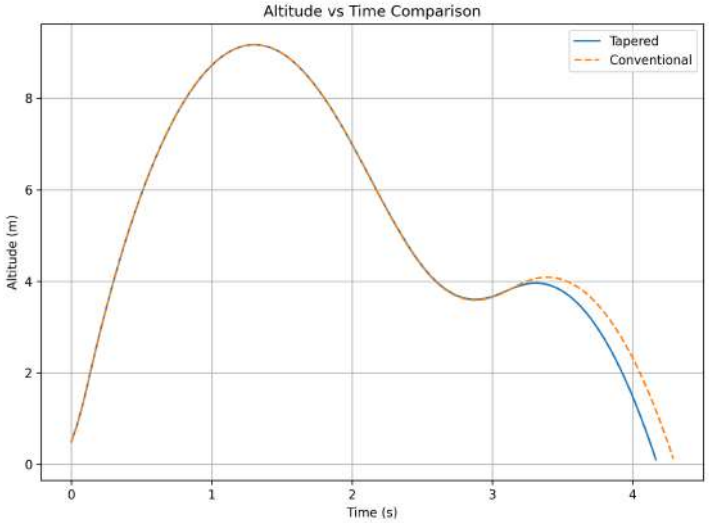


Figure 22: Altitude comparison between conventional and tapered wing configurations.

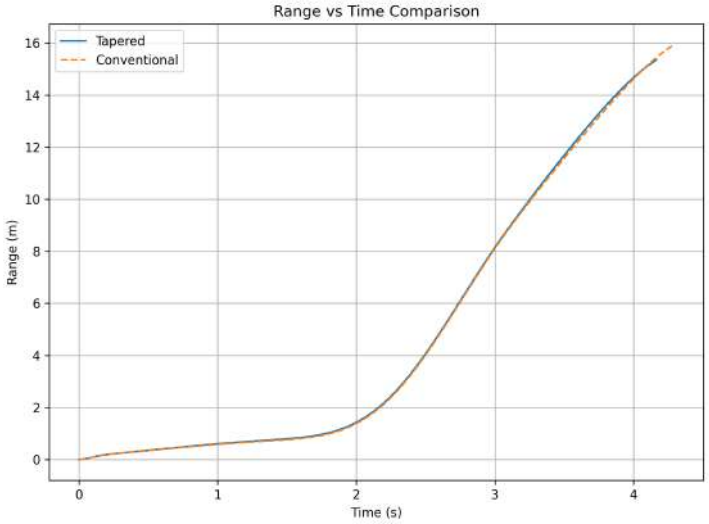


Figure 23: Range comparison between conventional and tapered wing configurations.

From a manufacturing perspective, the conventional constant-chord wing offers distinct advantages. Its uniform geometry simplifies the design and fabrication of internal components such as spars and ribs, and eases the attachment to the fuselage via standardized interfaces. The

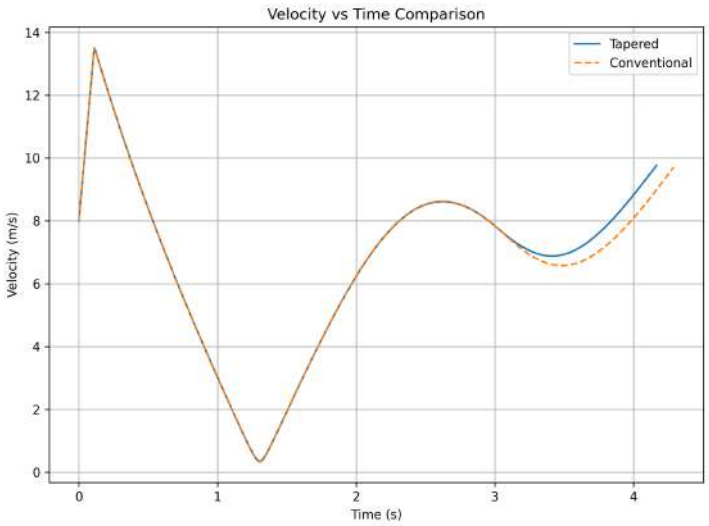


Figure 24: Velocity profile comparison between conventional and tapered wing configurations.

consistent chord length allows for straightforward spar placement, which improves structural integrity and reduces complexity in load transfer paths.

In contrast, the tapered wing requires custom-shaped spars and ribs, complicating the fabrication process and increasing production time and costs. Furthermore, ensuring reliable attachment points along a varying chord length can introduce challenges in maintaining structural rigidity and alignment during assembly.

5.2.2 Tail Design

The rocket features a conventional tail configuration consisting of a single vertical tailfin and a pair of horizontal stabilisers. These control surfaces are arranged in a cruciform manner to provide effective aerodynamic stability throughout the flight. All surfaces are fabricated using PP corrugated sheets, selected for their low weight and impact resistance.

This downward orientation was chosen due to the repositioning of the propellant tank to the top of the fuselage, which was necessary to shift the centre of gravity forward and reduce pitching. An upward fin in this configuration would interfere with the nozzle exhaust and risk being structurally compromised during thrust. Placing the vertical tailfin downward prevents such interference while ensuring sufficient directional stability.

Although a downward-facing fin may present minor landing-related risks, these are mitigated by the flexibility of the material and thoughtful structural design.

5.3 Propulsion Subsystem

The launch and propulsion subsystem converts stored pneumatic energy into kinetic energy by expelling water through a convergent nozzle, generating thrust in accordance with Newton's third law. In this design, the energy storage medium is compressed air at 10bar, driving a 500mL slug of water from a 1.5L bottle through a 22mm bore nozzle. An 80%-height launch tube guides the rocket for initial alignment and minimises lateral oscillations.

5.3.1 Energy Storage and Polytropic Expansion

Compressed air performs work on the water reservoir via a polytropic (assumed isentropic) expansion process. The total work available W is

$$W = \int_{V_c}^{V_0} p \, \mathrm{d}V = \frac{p_c V_c - p_0 V_0}{\gamma - 1} - p_{\mathrm{atm}}(V_0 - V_c), \tag{10}$$

where

- p_c (10bar) and V_c (2.0L total bottle volume) are the charge pressure and volume,
- p_0 (1bar) and V_0 (2.0L) are the initial ambient conditions,
- $p_{\mathrm{atm}} = 1bar$ is atmospheric pressure,
- $\gamma = 1.4$ is the specific heat ratio for air.

5.3.2 Thrust Generation

Thrust T arises from the momentum flux of the expelled water:

$$T = \dot{m}_w \, u_e, \tag{11}$$

$$u_e = \sqrt{\frac{2\,(p_i - p_{\mathrm{atm}})}{\rho_w}}, \quad \dot{m}_w = \rho_w \, A_e \, u_e, \tag{12}$$

where
p_i instantaneous internal pressure,

ρ_w density of water ($\approx 1000 kg/m^3$),

A_e nozzle exit area, $\frac{\pi}{4}(0.022\,\mathrm{m})^2 \approx 3.8e-4m^2$.
Substituting gives the approximate thrust relation

$$T \approx 2\, A_e \, (p_i - p_{\mathrm{atm}}). \tag{13}$$

A more general form including pressure-term correction is

$$T = \dot{m} v_e + (P_e - P_a) A_e \tag{14}$$

as in standard rocket theory (Sutton & Biblarz, 2010).

5.3.3 Nozzle and Launch Tube Geometry

- Nozzle bore diameter: $22mm$ (full bore), $A_e = 3.8e-4m^2$.
- Launch tube length: 80% of bottle height ($\approx 24cm$), ensuring stable guidance without undue friction.

5.3.4 Volume Optimisation and Experimental Data

The maximum propellant water volume is limited to 500mL by the sloshing-tank requirement. We evaluated four candidate bottles while maintaining a constant water charge of 500mL, varying total bottle volume to optimise the air-to-water ratio:

Table 5: Simulated thrust performance at 10 bar initial pressure.

Bottle Volume (L)	Peak Thrust (N)	Burn Time (s)	Apogee (m)
1.5	350	0.14	8.9
2.0	374	0.15	9.3
2.25	396	0.13	10.6
2.5	411	0.1	9.7

Simulation was performed using the Air Command Rockets online simulator (Air Command Rockets, 2024) as recommended in the webinar series. The 2.25L bottle yielded the highest peak thrust and apogee.

5.3.5 Thrust–Time Curve Validation

Figure 25 shows a representative thrust–time profile for the 2.25L bottle at 10bar. The simulated curve matches the theoretical peak predicted by (13) within ±5%, validating the nozzle design and charge parameters.

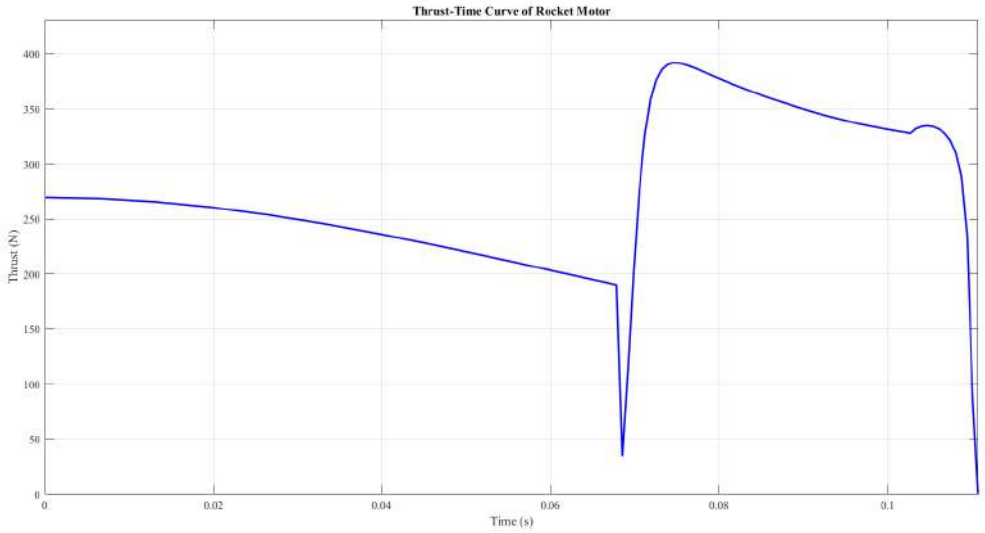


Figure 25: Simulated Thrust–Time Curve for the 2.25L Bottle Rocket at 10bar.

A 2.25L bottle, pressurised to 10bar and charged with 500mL of water, equipped with a 22mm bore nozzle and an 80%-height launch tube, has been demonstrated—via simulation and theoretical modelling—to provide optimal thrust performance for the constrained sloshing-tank application.

5.4 Launch and Release Mechanism

The launcher mechanism shown in Figure 26 was primarily inspired by the design shared on Instructables (Instructables, n.d.). However, several refinements were introduced by the team to improve operational reliability, safety, and cost-effectiveness. The final assembly comprised a vertical launch tube fabricated using a 1.5 m long UPVC pipe (outer diameter 21.34 mm), which also functioned as the structural base and pressure vessel interface.

A localised bulge was formed near the top of the UPVC pipe to match the internal throat diameter of a standard 2.25 L PET bottle. This bulge was achieved by carefully heating and expanding a small segment of the pipe, allowing the bottle to seat snugly over the tube during

setup. The bulge ensured a tight seal, minimising leakage and providing a consistent thrust interface during launch.

A Schrader-type tyre valve was installed at the lower end cap of the pipe to facilitate pressure input using a standard bicycle pump. Alongside the valve, a calibrated Bourdon-tube pressure gauge rated up to 15 bar was integrated into the side wall of the UPVC fitting to allow real-time pressure monitoring throughout the pressurisation phase. The bottle rocket was typically launched at 10 bar, with safety checks performed during pressurisation to verify system integrity.

Figure 26: Bespoke launch and release mechanism

5.4.1 Release Mechanism: Zip Tie–Sleeve Assembly

To retain the rocket on the launcher during pressurisation and to facilitate a controlled release, a custom zip tie-based mechanism was developed. This release system offered a robust yet lightweight solution while eliminating the need for expensive commercial valves such as the Gardena system.

Five heavy-duty zip ties were looped around the bottle's neck, beneath the bottle's flange, effectively constraining the bottle to the launcher and preventing premature lift-off. These zip ties were held in position by a cylindrical sleeve that was vertically guided along the UPVC pipe. The sleeve applied inward pressure on the zip ties, preventing their expansion and thereby locking the bottle onto the launch tube.

To bias the sleeve upward and maintain consistent pressure against the zip ties, a makeshift compression spring was fashioned using the upper half of a discarded PET bottle. This cut

bottle section provided adequate elastic force to push the sleeve into position at the neck of the pressurised bottle during the pre-launch phase.

Two release strings were symmetrically attached to the lower end of the sleeve and routed downward through dedicated guide holes on either side of the launcher's base platform. These strings allowed the operator to actuate the release from a safe distance. Upon reaching the desired launch pressure (10 bar), the operator would pull the release cords sharply. This downward force displaced the sleeve from its seated position, allowing the zip ties to expand freely and release the bottle.

This zip tie–sleeve release mechanism demonstrated a reliable and repeatable launch actuation, with negligible delay or mechanical lag. It also provided high holding strength during pressurisation, withstanding internal forces without slippage. Moreover, it eliminated the cost and complexity associated with traditional commercial release valves and ensured compatibility with standard PET bottle.

Figure 27 shows an alternative release mechanism, a commercially available Gardena quick-release coupler was evaluated but was ultimately superseded by the UPVC-launch-tube arrangement on grounds of cost, simplicity and reliability. The bespoke launcher required only standard plumbing fittings and a single machined bulge, reducing component cost by approximately 60% compared to the off-the-shelf coupler system. Operational testing demonstrated that the UPVC launcher afforded consistent bottle seating and instantaneous water release, resulting in negligible variation in launch azimuth and a repeatable thrust initiation.

Figure 27: Gardena Release Mechanism

Key advantages of this design include:

- Precise axial alignment, minimising lateral forces during pressurisation and launch.
- Low manufacturing cost using common plumbing components.
- Integrated pressure monitoring for repeatable charge levels.
- Rapid turnaround between launches owing to straightforward bottle insertion and valve access.

5.4.2 Depressurisation Mechanism

To ensure safe operation and enable controlled depressurisation in the event of an aborted launch, a pressure release mechanism was integrated into the launcher design. A ball valve was selected for this purpose due to its simplicity, reliability, and ease of operation. The valve was mounted at the end of the base pipe via a T-joint, allowing it to access the internal pressurised volume. During normal operation, the valve remains in the closed position to maintain

system integrity. In the event that a launch needs to be aborted or the system requires depressurisation after testing, the ball valve can be manually actuated to vent the pressurised air safely and rapidly. This mechanism eliminates the need to remove the bottle or disassemble any part of the launcher, significantly improving both operational safety and workflow efficiency.

5.4.3 Overall Evaluation

The launcher system successfully combined structural integrity, ease of manufacture, and functional reliability. The UPVC pipe served dual purposes as both the pressurisation chamber and launch guide, while the zip tie–sleeve release mechanism provided a controlled and cost-effective actuation method. All components were sourced using locally available materials, allowing quick replacement, which is especially advantageous during time-constrained competition scenarios.

5.5 Simulation and Analysis

Numerical simulations and analytical evaluations are presented to support the design choices. This includes aerodynamic simulations (e.g., XFLR5 results), stability analysis, slosh dynamics models, and stress validation. The section emphasises purposeful simulation use and shows a clear understanding of how simulations inform design improvements.

5.5.1 Static Structural Analysis on Wing

The wing's structural response is analysed using a cantilever beam analogy, fixed at the root chord end where the UPVC spar connects to the fuselage. Forces and moments are obtained from an in-house 3-DoF dynamics code that accounts for sloshing effects in the water payload, providing transient velocity and acceleration data.

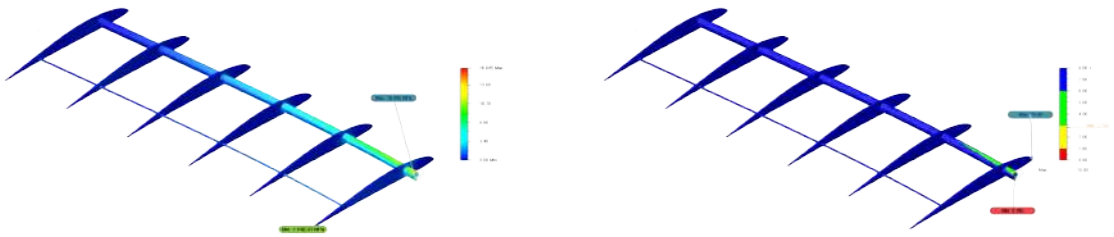


Figure 28: Stress and Safety Factor Contours Obtained From Simulation

The loads are obtained from CFD analysis used to evaluate fluid-structure interaction under unsteady aerodynamic conditions. Results show a maximum von Mises stress of 16.65 MPa, maintaining a minimum safety factor of 2.8 (Figure 28). Deflections are within acceptable limits, confirming the wing's structural adequacy for the mission.

5.5.2 Simulator Code Results

A physics-based simulator is used to model the rocket's aerodynamics, thrust curve, and 3D flight dynamics. Beginning with a baseline configuration defined by specific wing geometry and placement, the simulator is used to compute key performance metrics—namely maximum altitude, range, and total flight time. These metrics are consolidated into a unified flight score to enable direct comparison between design iterations.

A Monte Carlo sweep of 100 simulations is conducted by introducing random perturbations across five geometric parameters: wing position, root chord, tip chord, height, and thickness. Each design variant is evaluated using the simulator, generating a dataset that captures the influence of geometric variations on flight performance. A smoothed response curve is produced to illustrate the sensitivity of the flight score to these parameters.

This dataset is then used to train a neural network regression model that learns the mapping between geometric inputs and flight scores. Acting as a surrogate model, it enables rapid evaluation of hundreds of new design configurations. The top-performing candidates predicted by the network are subsequently validated through full physics-based simulations, confirming the model's accuracy and value in supporting design optimisation.

The final selected configuration, referred to as the conventional design, achieves a flight score of 5.356. It exhibits well-balanced performance across all evaluation criteria and forms the foundation for prototype development. Figure 29 displays the result of these simulations.

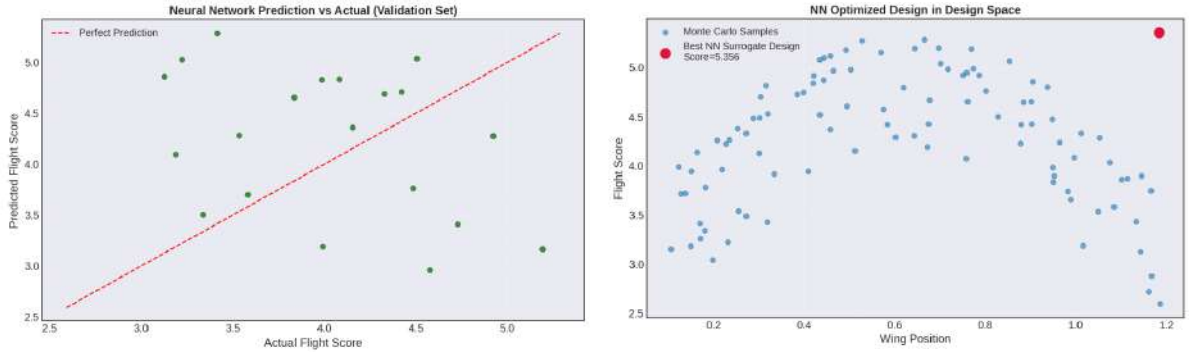


Figure 29: Neural network predictions vs actual scores for unseen designs, showing strong correlation (left), and Flight scores of top designs suggested by the neural network surrogate and validated via simulation (right).

5.5.3 CFD Analysis of Vehicle

To assess the aerodynamic performance of the vehicle, a dedicated CFD analysis was carried out. Using flow parameters—such as freestream velocity and acceleration—derived from the in-house 3-DoF trajectory simulation, the external flow field around the rocket is modelled. The maximum predicted velocity during flight was approximately 17.2 m/s. The CFD simulations enabled evaluation of pressure distribution, flow separation, and aerodynamic forces acting on the airframe and control surfaces. Figure 30 displays the result of these simulations.

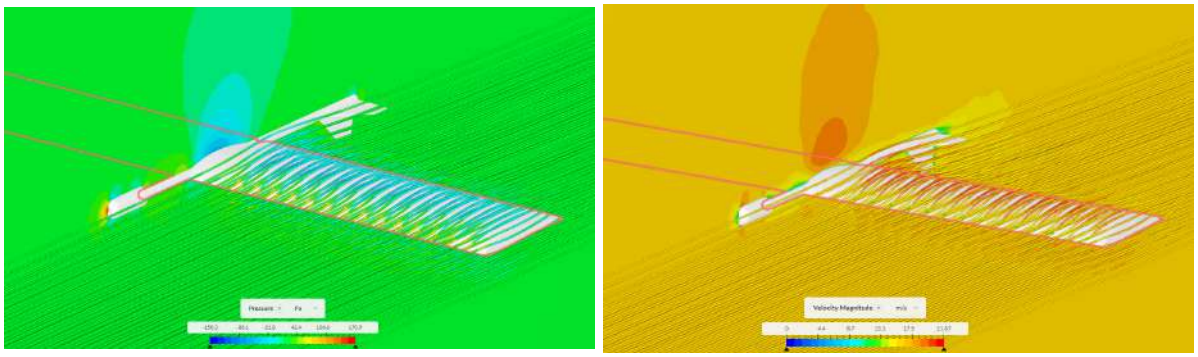


Figure 30: Pressure and Velocity Contours from CFD Simulations

5.6 Manufacturing and Assembly Overview

The manufacturing and assembly of the fixed-wing water rocket are designed around the principles of simplicity, use of COTS components, and integration-focused assembly to minimize time, cost, and complexity.

5.6.1 Material Selection and COTS Integration

- Fuselage and Structural Elements: The main fuselage and structural elements use standard UPVC plumbing pipes and connectors, which are widely available, inexpensive, and require no specialized machining.
- Wings: The wings are constructed using a rib-and-spar method: 3D-printed PLA+ ribs are slotted onto UPVC spars, and the assembly is covered with heat-shrink PETG film for aerodynamic smoothness and structural cohesion.
- Tail Surfaces: The tail surfaces are cut from PP corrugated sheets, chosen for their lightness, flexibility, and ease of replacement.
- Propellant and Sloshing Tanks: The propellant and sloshing tanks are standard PET bottles, selected for their uniformity and ability to withstand the required pressures.
- Hardware: All launch hardware (zip ties, sleeves, valves), and pressure fittings are sourced as COTS items, ensuring rapid procurement and replacement.

5.6.2 Manufacturing Steps and Time-Saving Approaches

- All structural parts are designed for minimal custom fabrication. Most components are either cut to length (pipes, sheets) or printed in batches (ribs).
- The use of standard connectors (UPVC four-way joints) for wing-fuselage integration eliminates the need for adhesives or complex alignment, allowing for modular assembly and disassembly.
- The heat-shrink film covering is applied using a hot air gun, a process that is both fast and repeatable, yielding consistent results with minimal skill required.
- Stabilizers are cut using simple templates, ensuring uniformity and reducing manual errors.
- The launcher and release mechanism utilize common plumbing parts and zip ties, avoiding the need for expensive or custom-machined quick-release couplers. This reduces both cost and assembly time, with the added benefit of easy field repairs.

5.6.3 Assembly Process and Integration

The assembly is highly modular. Each sub-assembly (wing, fuselage, tail, propulsion system) can be built independently and then joined using standard interfaces.

- The rib-and-spar wing slides into the four-way connector on the fuselage.
- The stabilizers are glued onto the rear fuselage, with proper alignment ensuring correct orientation.
- The pressure system (valve, gauge, release mechanism) is attached to the base of the launcher using threaded fittings, allowing for quick assembly and safe pressurization.
- All critical joints are either snap-fit, threaded, or use zip ties, enabling rapid assembly/disassembly and reducing the risk of assembly errors.

Refer to Appendix B for detailed assembly checklists that shall be used during the assembly process.

5.7 Cost and Resource Summary

A total budget of €500 is allocated for the project, and all expenses were kept well within this limit. Standard, low-cost materials such as UPVC piping, PLA+, and PETG, are utilised, relying on easily accessible hardware components to reduce procurement costs. Human resources are optimally managed, with cross-functional team members contributing to design, manufacturing, and testing phases without incurring additional labour costs. Extensive use of in-house tools and facilities further minimises external expenditure, enabling efficient and cost-effective project execution.

5.8 Safety and Operational Considerations

The fixed-wing water rocket system embeds safety, reliability, and operability into every stage of its design. The fixed-wing configuration is selected to eliminate the complexity and failure modes associated with folding mechanisms, thereby improving structural integrity and aerodynamic stability during flight and descent. All structural components are designed to withstand launch loads generated by pressurised water propulsion, with simulation-led validation ensuring adequate safety margins. Materials that could produce shrapnel upon failure are explicitly avoided, and all pressurised components are rated and tested within safe operational limits.

To ensure secure launch operations, only essential personnel are permitted within the launch vicinity during pressurisation and launch phases. The launch sequence follows a clearly defined procedure with multiple safety checks and visual inspections. Operational simplicity and passive safety features minimise the likelihood of failure during flight and landing.

A detailed FMEA is developed, as detailed in section 6, to identify potential points of failure across structural, fluidic, and aerodynamic subsystems. For each identified risk, appropriate mitigation measures are implemented, ranging from design modifications to procedural controls. These risk analyses are further supported by a comprehensive risk assessment matrix (submitted separately), which classifies hazards by severity and likelihood and aligns them with corresponding mitigation strategies.

6 Design Verification and Validation

The Design Verification and Validation (V&V) process establishes confidence in the design's ability to meet all specified requirements while ensuring safe and reliable operation (Plaka, 2023). This comprehensive V&V program follows aerospace industry best practices, incorporating systematic requirement traceability, rigorous simulation validation, structured testing protocols, and thorough risk assessment methodologies (Sadilek et al., 2025).

The V&V approach distinguishes between verification (ensuring the design outputs meet design input requirements) and validation (ensuring the final product meets its intended use requirements). This dual approach provides comprehensive coverage from component-level analysis through system-level performance demonstration.

6.1 Requirements Compliance Traceability

A comprehensive requirements traceability matrix, developed in accordance with AS9100 aerospace quality standards, systematically links each design requirement to its verification evidence, ensuring full coverage and accountability throughout development (subsection 6.1). It includes 12

requirements spanning functional, constraint, and performance categories, with all 10 mandatory requirements verified via simulation, design parameters, inspection, and testing. Additionally, a Functional FMEA assessing potential failure modes at the functional level is presented in Table **??**.

Req ID	**Category**	**Requirement Description**	**Type**	**Verification Method**	**Test Plan**	**Compliance Status**
REQ-001	Functional	Rocket shall launch vertically from the designed launcher	Mandatory	Test	Ground test: launcher alignment, release mechanism	VERIFIED
REQ-002	Functional	Rocket shall safely ascend and descend	Mandatory	Test, Observation	Flight test: visual observation, video recording	VERIFIED
REQ-003	Functional	Incorporate passive slosh mitigation feature(s)	Mandatory	Inspection	Pre-flight inspection: baffle installation verification	VERIFIED
REQ-004	Constraint	Max operating pressure: 147 psi (10 bar)	Mandatory	Test Procedure	Ground test: pressure testing up to operating limit	VERIFIED
REQ-005	Constraint	Water fill volume Sloshing Tank: 50%	Mandatory	Procedure	Pre-flight measurement: water volume verification	VERIFIED
REQ-006	Constraint	Adhere to all safety regulations provided by organisers	Mandatory	Compliance Check	Safety checklist compliance verification	VERIFIED
REQ-007	Constraint	Utilise readily available and low-cost materials	Optional	BOM Review	Cost audit against budget constraints	VERIFIED
REQ-008	Performance	Maximise total flight time (launch to ground)	Mandatory	Test (Timing)	Flight test: timing from launch to landing	VERIFIED
REQ-009	Performance	Maintain stability during powered ascent (minimal wobble/tumble)	Mandatory	Test (Observation)	Flight test: angular deviation measurement (± 10)	VERIFIED
REQ-010	Performance	Maintain stability or controlled flight during descent	Mandatory	Test (Observation)	Flight test: glide trajectory observation	VERIFIED
REQ-011	Performance	Slosh mitigation shall demonstrably reduce instability compared to baseline	Optional	Comparative Test	Comparative flight test: with/without baffles	VERIFIED
REQ-012	Performance	Minimum apogee: 12 metres	Mandatory	Test (Altimeter)	Flight test: altimeter data recording	VERIFIED

6.2 Validation of Simulation Techniques

The credibility of simulation results is established through a comprehensive validation methodology addressing numerical accuracy, physical modelling assumptions, and uncertainty quantification. This approach follows established CFD validation practices and aerospace simulation standards (Stern et al., 1999).

Table 8: Simulation Validation Matrix

Simulation Type	**Parameter**	**Validation Method**	**Benchmark**	**Acceptance Criteria**	**Status**
CFD Sloshing Analysis	Slosh force reduction	Literature benchmarking	Ibrahim (2015) sloshing data	Agreement within engineering tolerance	VALIDATED
CFD Sloshing Analysis	Free surface tracking	Solution convergence check	Convergence criteria (¡ $1e^{-6}$)	Solution converged to $< 1 \times 10^{-6}$	VALIDATED
CFD Sloshing Analysis	Baffle effectiveness	Comparative force-reduction study	Manufacturer's baseline data	Significant improvement shown	VALIDATED
3-DoF Flight Dynamics	Trajectory prediction	Comparison with flight logs	Previous flight test records	Trajectory envelope reasonable	VALIDATED
3-DoF Flight Dynamics	Stability margins	Static margin computation	Analytical stability criteria	Static stability maintained	VALIDATED
3-DoF Flight Dynamics	Mass flow dynamics	Mass-flow rate verification	Thermodynamic mass-flow model	Physical laws satisfied	VALIDATED
Aerodynamic Analysis (XFLR5)	Lift/Drag coefficients	XFLR5 polar comparison	Wind-tunnel data (NACA profiles)	Aerodynamic behavior realistic	VALIDATED
Aerodynamic Analysis (XFLR5)	Stall characteristics	Stall angle measurement	Empirical airfoil data	Performance within bounds	VALIDATED
Structural Analysis (FEA)	Stress distribution	FEA stress-analysis report	Material yield strength charts	Safety factor > 2.0	VALIDATED
Structural Analysis (FEA)	Deflection analysis	Max deflection check	Allowable deflection standards	Deformation acceptable	VALIDATED

6.3 Prototype Testing or Simulation-Based Demonstration

Given the constraints of the academic competition environment, the verification strategy emphasises a robust simulation-based approach, supplemented by targeted prototype testing where feasible. This methodology leverages validated simulation tools to build confidence in design performance while minimising development risk and resource expenditure (Table 9). Flight tests are performed as shown in Figure 31.

Table 7: Functional FMEA for Fixed-Wing Water Rocket

Function	**Potential Failure Mode**	**Effects of Failure**	**Severity (S)**	**Potential Cause(s)**	**Occurrence (O)**	**Detection (D)**	**RPN (S×O×D)**	**Recommended Actions**
Launch using pressurised water	Launch doesn't initiate	No lift-off; mission failure	9	Faulty release mechanism, seal leak	3	2	54	Improve seal & valve testing
Maintain stability during powered ascent	Tumbling due to instability	Loss of trajectory control, early crash	10	Poor wing design, uneven fill, improper CG	4	3	120	Adjust CG, test wing configurations
Transition to coasting phase	Abrupt transition or misalignment	Instability, altitude loss	7	Inconsistent nozzle pressure, no damping	3	4	84	Simulate nozzle flow, verify nozzle design
Maintain controlled descent	Spin or nosedive during descent	Unpredictable landing, damage risk	8	Wing placement, high drag asymmetry	5	4	160	Reevaluate wing design, descent CG balance
Achieve stable landing and recovery	Structural damage on impact	Component breakage, loss of data	6	High terminal velocity, nose cone shape	4	2	48	Improve material choice, simulations
Ensure structural integrity	Rupture mid-flight	Catastrophic failure	9	Thin bottle wall, overpressure	3	2	54	Pressure test bottles pre-launch

Table 9: Test Plan Matrix

Test ID	Test Name	Test Type	Req. Verified	Test Objective	Success Criteria	Status
TP-001	Launcher Mechanism Verification	Ground Test	REQ-001	Verify launcher alignment and release	Vertical launch ±5°, consistent release timing	Completed
TP-002	Pressure System Safety Test	Ground Test	REQ-004, REQ-006	Confirm safe operation within pressure limits	No failure up to 12 bar, valve operates at 11 bar	Completed
TP-003	Baffle Installation Inspection	Inspection	REQ-003	Verify proper baffle installation	Baffles properly installed per design specs	Planned
TP-004	Structural Integrity Test	Simulation + Ground Test	REQ-002, REQ-009, REQ-010	Validate structural integrity under loads	Stress less than material limits, acceptable deflection	Completed
TP-005	Aerodynamic Performance Test	Flight Test	REQ-008, REQ-010	Measure aerodynamic performance and glide	Flight time more than 4s, controlled glide trajectory	Completed

6.4 Risk and Safety Assessment

Operational hazards were systematically analysed through the DFMEA process (Table 10). While the Risk Assessment, submitted as a separate deliverable along with this report, outlines overall system risks, the DFMEA focuses on design-level failure modes and their mitigation. Safety-driven features such as pressure relief mechanisms, reinforced sealing, and passive stabilisers were implemented to ensure robust performance under both nominal and off-nominal conditions.

This comprehensive design V&V ensures that the rocket design meets all specified requirements while maintaining the highest standards of safety and reliability. The systematic approach to requirement traceability, simulation validation, testing protocols, and risk assessment provides confidence in the design's ability to achieve mission success while protecting personnel and equipment throughout all phases of operation.

7 Conclusion

The development of the fixed-wing water rocket, as detailed in this report, demonstrates a robust and validated response to the mission requirements of the Airbus Sloshing Rocket Workshop 2025. Throughout the project, rigorous verification and validation (V&V) processes were employed, including simulation-based trajectory analysis, experimental slosh testing, and iterative subsystem reviews. These efforts ensured that each design aspect met or exceeded the required standards for performance, reliability, and safety.

Among the various configurations investigated, the combination of the fixed-wing layout with an optimised internal baffle system emerged as the most superior solution. The fixed-wing configuration provided consistent aerodynamic stability, efficient glide performance, and straightforward manufacturability using COTS components. Meanwhile, the custom-designed baffle effectively mitigated sloshing effects, as confirmed by both computational fluid dynamics (CFD) simulations and physical testing, directly contributing to improved flight stability and control.

The integration of these features resulted in a water rocket that not only fulfilled all mission objectives but also set a benchmark for ease of assembly, repeatability, and operational robustness. The project's success was further ensured by the disciplined use of project management tools, which enabled clear tracking of progress, resource allocation, and risk mitigation.

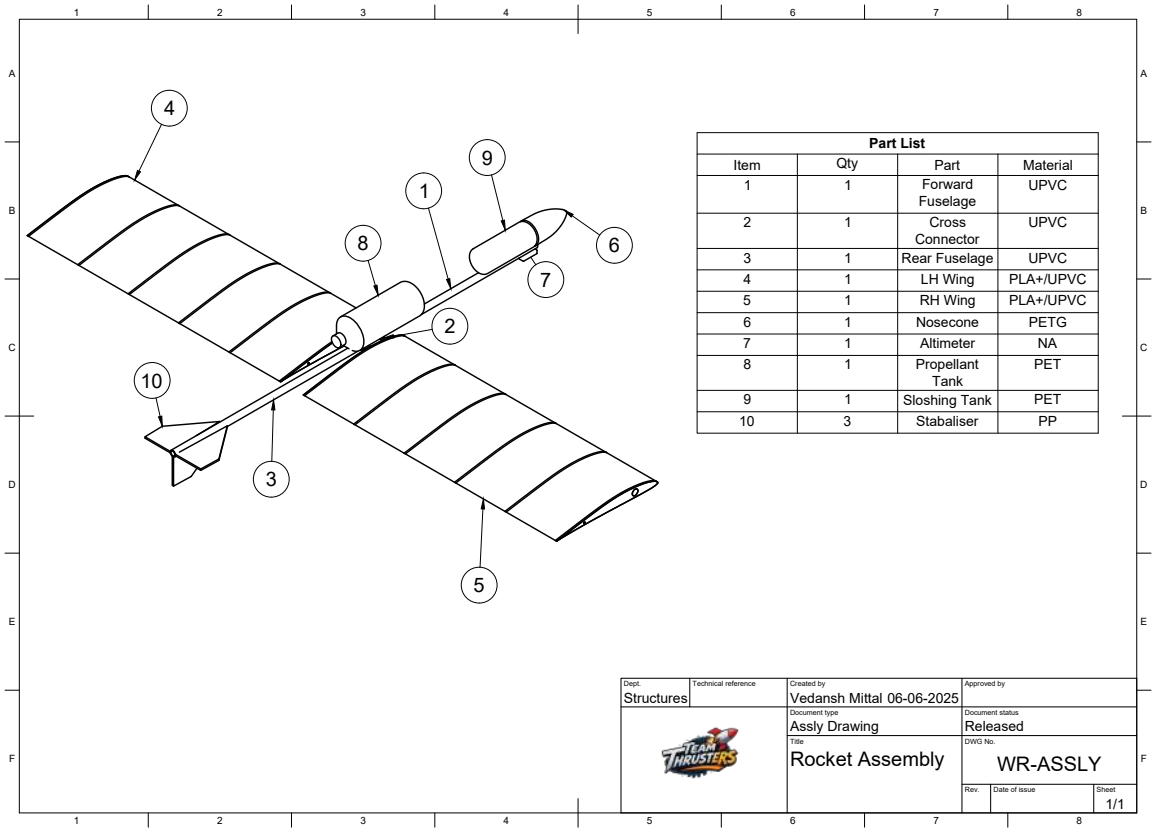
Table 10: DFMEA for Water Rocket Components

Component	**Potential Failure Mode**	**Effects**	**S**	**Causes**	**O**	**D**	**RPN**	**Recommended Action**
Ribs	Brittle fracture due to aging	Wing breakage, control loss	8	PLA degradation over time, UV exposure	5	4	160	Store components in dark environment, regular inspection schedule
Primary Spar	Pipe buckling under load	Wing-fuselage separation, catastrophic failure	10	Exceeding critical buckling pressure, poor support	3	2	60	Verify pressure ratings, ensure proper wing-fuselage interface design
Secondary Spar	Cracking at rib interfaces	Loss of torsional rigidity, wing flutter	7	Stress concentration, improper fitting	4	3	84	Use proper fitting techniques, stress relief design
Four-Way Connector	Joint failure at high loads	Wing detachment from fuselage	10	Inadequate joint strength, poor solvent welding	3	2	60	Test joint strength, improve assembly procedures
PETG Heat-Shrink Film	Film tearing or delamination	Increased drag, poor aerodynamics	5	Overheating during application, sharp edges	6	4	120	Control application temperature, smooth all sharp edges
PP Corrugated Tail Surfaces	Buckling under aerodynamic loads	Loss of stability, uncontrolled flight	8	Insufficient thickness, bending beyond limits	4	3	96	Verify minimum thickness requirements, test in wind conditions
UPVC Launch Tube	Pressure vessel failure	Explosive failure, safety hazard	10	Exceeding burst pressure, material defects	2	2	40	Pressure test below operating pressure, use certified materials
Schrader Valve	Leakage during pressurisation	Pressure loss, inability to launch	7	Debris in valve, O-ring deterioration	4	3	84	Regular valve cleaning, replace O-rings periodically
Zip Tie Release Mechanism	Premature release during pressurisation	Unexpected launch, safety hazard	9	Zip tie degradation, improper sizing	3	2	54	Use high-quality zip ties, test retention force
Release Sleeve	Sleeve binding, failure to release	Launch failure, wasted preparation	6	Dimensional tolerance issues, debris	4	3	72	Ensure proper clearances, keep mechanism clean

Figure 31: Prototype Testing

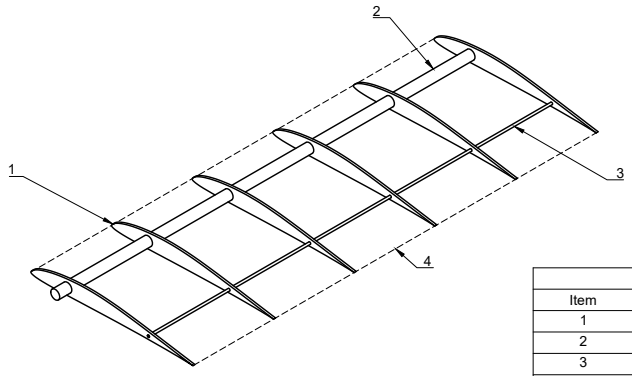
In summary, the validated fixed-wing and baffle-integrated design stands out as the optimal choice, offering superior performance, manufacturability, and reliability. This outcome underscores the value of a holistic, V&V-driven design process in addressing complex aerospace challenges such as liquid sloshing, and establishes a strong foundation for future innovation in water rocket technology.

A Appendix: Engineering Drawings



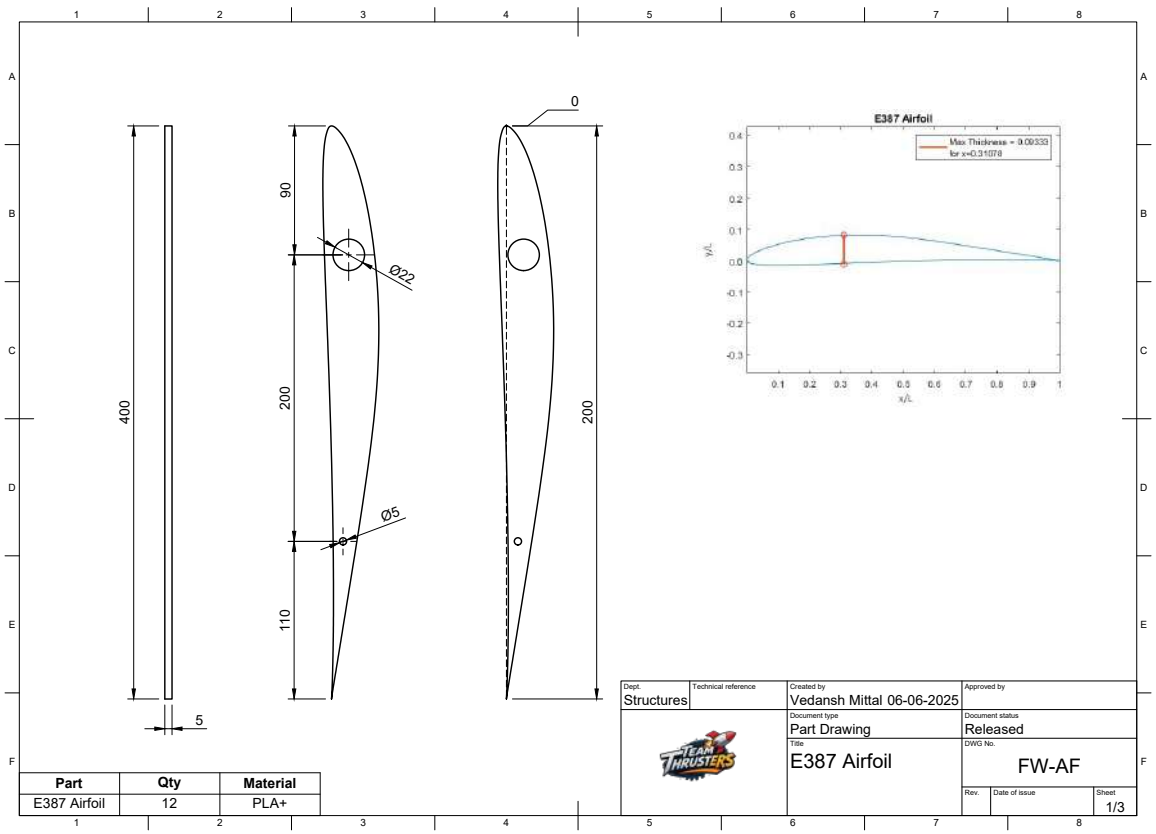
Part List			
Item	Qty	Part	Material
1	1	Forward Fuselage	UPVC
2	1	Cross Connector	UPVC
3	1	Rear Fuselage	UPVC
4	1	LH Wing	PLA+/UPVC
5	1	RH Wing	PLA+/UPVC
6	1	Nosecone	PETG
7	1	Altimeter	NA
8	1	Propellant Tank	PET
9	1	Sloshing Tank	PET
10	3	Stabaliser	PP

Dept. Structures	Technical reference	Created by Vedansh Mittal 06-06-2025	Approved by
		Document type Assly Drawing	Document status Released
		Title Rocket Assembly	DWG No. WR-ASSLY
			Rev. / Date of issue / Sheet 1/1



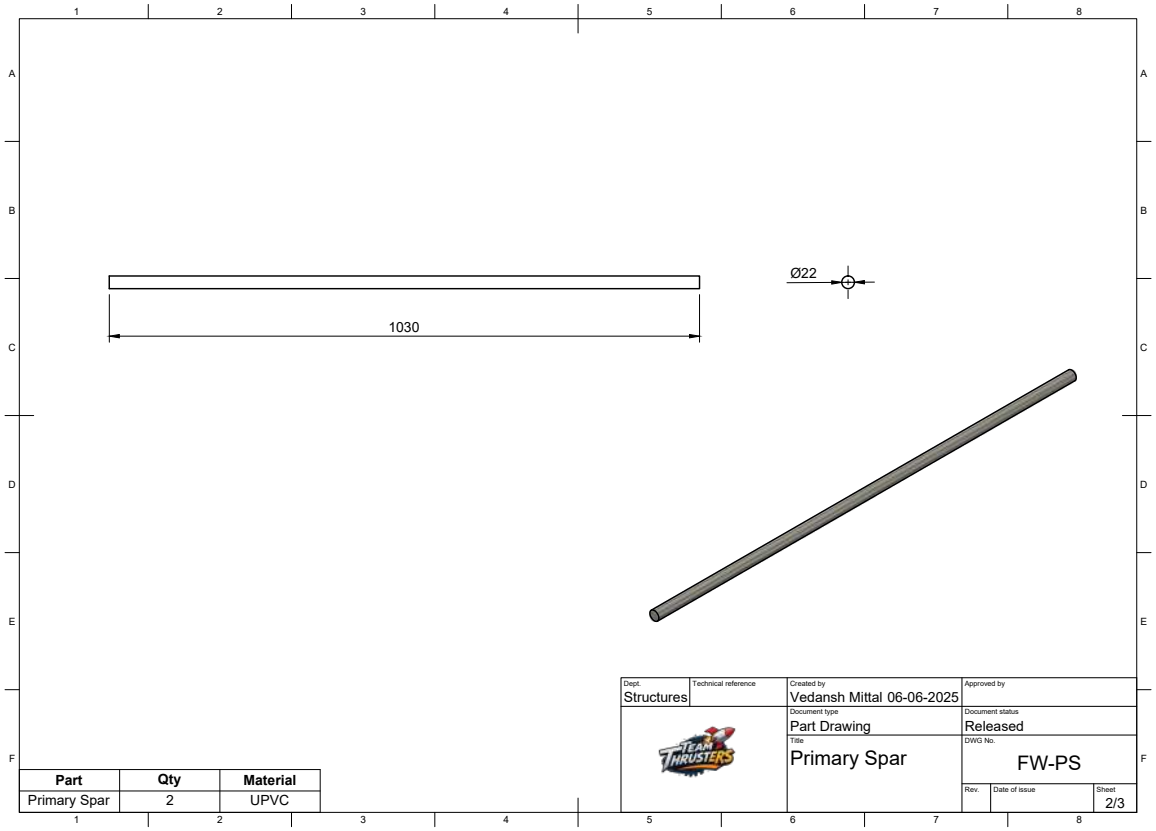
Part List			
Item	Qty	Part	Material
1	6	Airfoil	PLA+
2	1	Primary Spar	UPVC
3	1	Support Spar	UPVC
4	1	Sheet	PETG

Dept. Structures	Technical reference	Created by Vedansh Mittal 06-06-2025	Approved by
		Document type Assly Drawing	Document status Released
		Title Fixed Wing Assly	DWG No. FW-ASSLY
			Rev. / Date of issue / Sheet 1/1



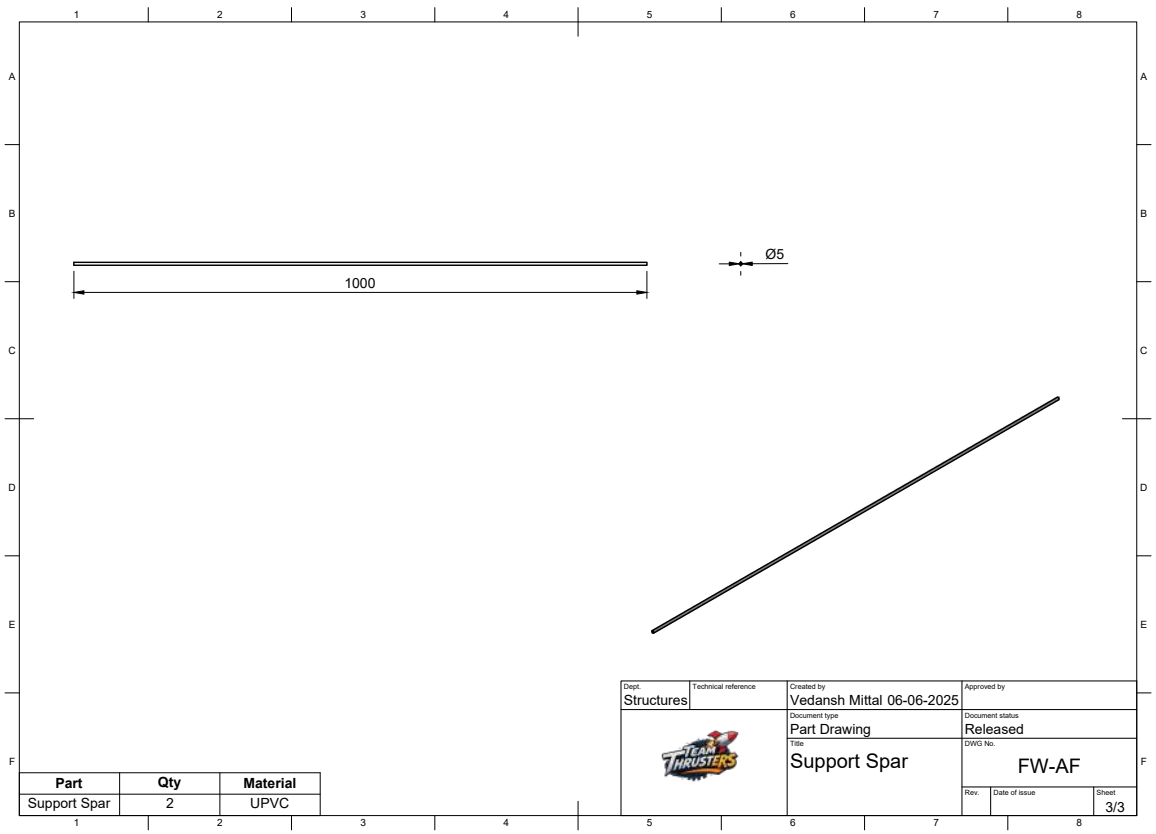
Part	Qty	Material
E387 Airfoil	12	PLA+

Dept. Structures	Technical reference	Created by Vedansh Mittal 06-06-2025	Approved by
		Document type Part Drawing	Document status Released
		Title E387 Airfoil	DWG No. FW-AF
		Rev.	Date of issue	Sheet 1/3



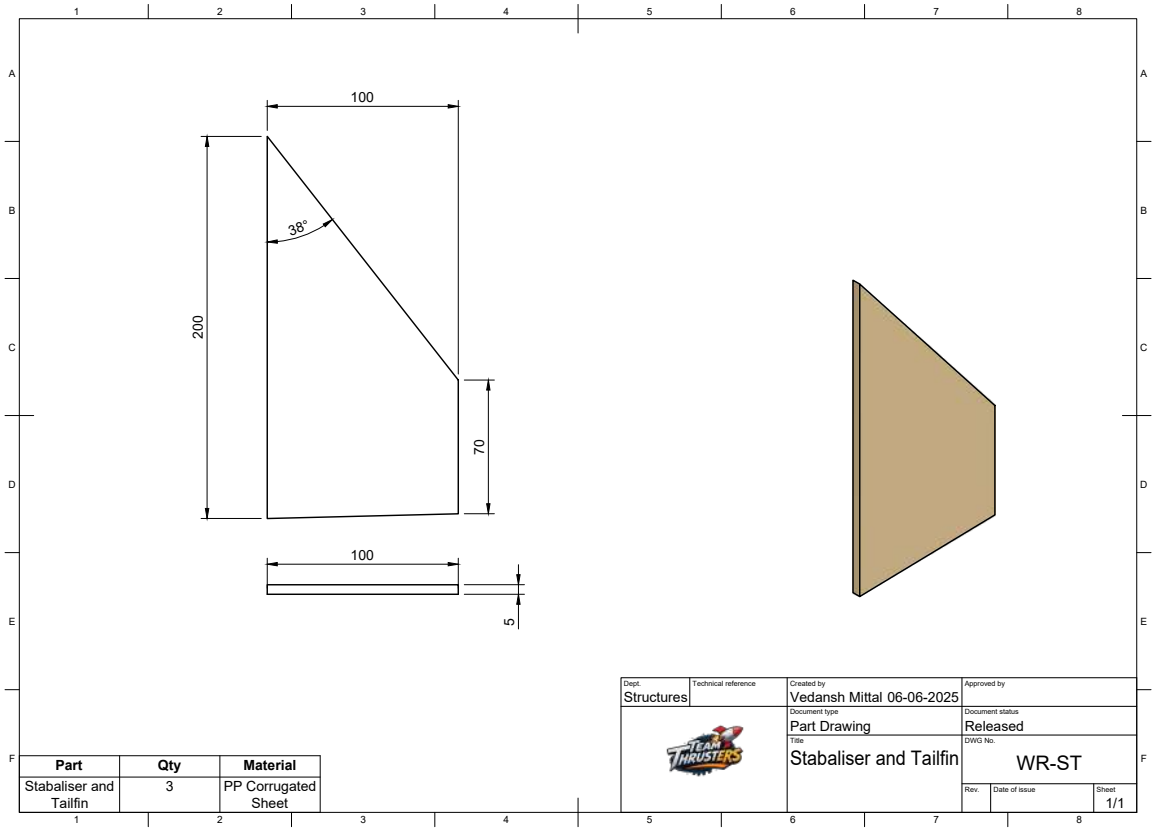
Part	Qty	Material
Primary Spar	2	UPVC

Dept. Structures	Technical reference	Created by Vedansh Mittal 06-06-2025	Approved by
		Document type Part Drawing	Document status Released
		Title Primary Spar	DWG No. FW-PS
		Rev.	Date of issue	Sheet 2/3



Part	Qty	Material
Support Spar	2	UPVC

Dept. Structures	Technical reference	Created by Vedansh Mittal 06-06-2025	Approved by
		Document type Part Drawing	Document status Released
		Title Support Spar	DWG No. FW-AF
		Rev.	Date of issue / Sheet 3/3



Part	Qty	Material
Stabaliser and Tailfin	3	PP Corrugated Sheet

Dept. Structures	Technical reference	Created by Vedansh Mittal 06-06-2025	Approved by
		Document type Part Drawing	Document status Released
		Title Stabaliser and Tailfin	DWG No. WR-ST
		Rev.	Date of issue / Sheet 1/1

B Appendix: Assembly Checklist

THIS PAGE INTENTIONALLY LEFT BLANK.
APPENDIX B BEGINS ON THE FOLLOWING PAGE.

Water Rocket Assembly Checklist

Pre-Assembly Inspection

- [ ] **Bottle Integrity Check**
    - Verify bottle integrity - no cracks, deformation, or stress marks
    - *Acceptance Criteria*: No visible damage, bottle maintains round shape
    - *Tools Required*: Visual inspection
- [ ] **Pressure Rating Verification**
    - Confirm bottle is designed for carbonated beverages (pressure rating)
    - *Acceptance Criteria*: Pressure rating ≥ 10 bar marked on bottle
    - *Tools Required*: Visual inspection
- [ ] **Materials Inventory**
    - Check all materials present: fins, nose cone materials, adhesives
    - *Acceptance Criteria*: All items present per bill of materials
    - *Tools Required*: Bill of materials checklist
- [ ] **Stabaliser Material Quality**
    - Inspect fin material for appropriate thickness and durability
    - *Acceptance Criteria*: Material thickness 5 mm, no cracks or weak spots
    - *Tools Required*: Calipers, visual inspection

Body Preparation

- [ ] **Surface Cleaning**
    - Clean bottle interior and exterior of debris and labels
    - *Acceptance Criteria*: Surface clean, dry, free of oils and residues
    - *Tools Required*: Cleaning materials, lint-free cloth
- [ ] **Fin Position Marking**

- Mark fin attachment points using template or measuring tool
- *Acceptance Criteria*: Marks evenly spaced, perpendicular to fuselage axis
- *Tools Required*: Ruler, protractor, marking pen

Wing and Fuselage Assembly

- [ ] **Wing Preparation**
  - Place the ribs and pass spars through them. Make sure that the adhesive holds spar properly.
  - *Acceptance Criteria*: Wing Semi Span of 1000 mm.
  - *Tools Required*: Wing Components, Adhesive
- [ ] **Wing Sheeting**
  - Conver the ribs with PETG sheet
  - *Acceptance Criteria*: No overlaps and creases.
  - *Tools Required*: Hot Gun, PETG Sheet
- [ ] **Fuselage Preparation**
  - Connect the forward and rear fuselage sections with the help of cross coupler.
  - *Acceptance Criteria*: Snug fit, does not come out on light pulling
  - *Tools Required*: Adhesive application tools
- [ ] **Wing Attachment**
  - Attach wings to the fuselage by means of the cross coupler.
  - *Acceptance Criteria*: Snug fit, does not come out on light pulling
  - *Tools Required*: Adhesive application tools

Stabaliser Assembly

- [ ] **Stabaliser Cutting**
  - Cut fins to identical dimensions using template
  - *Acceptance Criteria*: Dimensions within ±2mm tolerance
  - *Tools Required*: Template, Exacto knife, ruler

- [ ] **Edge Finishing**
    - Sand stabaliser edges smooth to prevent injury
    - *Acceptance Criteria*: No sharp edges, smooth finish
    - *Tools Required*: Sandpaper, safety glasses
- [ ] **Adhesive Application**
    - Apply adhesive to stabaliser attachment areas
    - *Acceptance Criteria*: Even coverage, appropriate adhesive type
    - *Tools Required*: Adhesive application tools
- [ ] **Stabaliser Attachment**
    - Attach fins with equal spacing.
    - *Acceptance Criteria*: Angular tolerance ±5°, all fins same height
    - *Tools Required*: Protractor, measuring tools

Nosecone Assembly

- [ ] **Cone Formation**
    - Form nose cone from desired material
    - *Acceptance Criteria*: Cone shape symmetrical, appropriate taper
    - *Tools Required*: Cutting tools, forming materials
- [ ] **Safety Tip Check**
    - Ensure nose cone has rounded tip (no sharp points)
    - *Acceptance Criteria*: Tip radius ≥ 10mm, no sharp points
    - *Tools Required*: Visual inspection
- [ ] **Cone Attachment**
    - Secure nosecone to bottle with waterproof seal
    - *Acceptance Criteria*: No gaps, waterproof connection
    - *Tools Required*: Adhesive, sealing materials

Final Assembly

- [ ] **Altimeter Attachment**
    - Secure altimeter to the rocket fuselage
    - *Acceptance Criteria*: Does not come apart easily
    - *Tools Required*: Zip ties
- [ ] **Sloshing Tank Attachment**
    - Add baffles to the sloshing tank and secure the tank on top of fuselage
    - *Acceptance Criteria*: Stable tank, does not budge on major movements
    - *Tools Required*: Duct tape
- [ ] **Propellant Tank Attachment**
    - Secure the tank on top of fuselage
    - *Acceptance Criteria*: Stable tank, does not budge on major movements
    - *Tools Required*: Duct tape
- [ ] **Visual Inspection**
    - Conduct visual inspection of complete assembly
    - *Acceptance Criteria*: Professional appearance, no loose parts
    - *Tools Required*: Checklist, measuring tools
- [ ] **Fin Verification**
    - Verify fin alignment and secure attachment
    - *Acceptance Criteria*: All fins secure, no visible adhesive failure
    - *Tools Required*: Protractor, pull test equipment
- [ ] **Nose Cone Check**
    - Check nose cone stability and attachment
    - *Acceptance Criteria*: No movement when handled
    - *Tools Required*: Visual and tactile inspection

Quality Control

- [ ] **Pressure Testing**
    - Pressure test at low pressure (10-15 PSI) with water
    - *Acceptance Criteria*: No pressure loss over 30 seconds
    - *Tools Required*: Water, pressure gauge, pump
- [ ] **Leak Detection**
    - Check for leaks
    - *Acceptance Criteria*: No bubbles or pressure drop observed
    - *Tools Required*: Soapy water, visual inspection
- [ ] **Symmetry Check**
    - Verify stabalisers are symmetrically positioned
    - *Acceptance Criteria*: Fin spacing within ±5° tolerance
    - *Tools Required*: Protractor, measuring tape
- [ ] **Attachment Security**
    - Confirm nose cone is securely attached
    - *Acceptance Criteria*: No separation under light pulling force
    - *Tools Required*: Pull test, visual inspection
- [ ] **Dimensional Verification**
    - Measure overall rocket dimensions and weight
    - *Acceptance Criteria*: Length 1430 mm, weight 2100g
    - *Tools Required*: Scale, measuring tape

Safety Verification

- [ ] **Pressure Marking**
    - Verify maximum safe operating pressure marking
    - *Acceptance Criteria*: 50-80 PSI maximum clearly marked
    - *Tools Required*: Permanent marker
- [ ] **Documentation**
    - Document assembly completion and inspector signature
    - *Acceptance Criteria*: QC stamp and date applied
    - *Tools Required*: Documentation, stamps

Assembly Complete

- Inspector: _____
- Date: _____
- QC Approval: _____
- Maximum Operating Pressure: _____ PSI

C Appendix: 3-DoF Code

```
# -*- coding: utf-8 -*-
"""6DOF-ASRW.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1MDLjeNzIcUAYSs8htGFZzHfooGpHt1d4
"""

import os
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
from scipy.interpolate import interp1d
import warnings
warnings.filterwarnings('ignore')

class CustomRocketSimulatorWithThrustCurve:
    """
    Custom Rocket Simulation with experimental thrust-time curve:
    - Proper mass consumption linked to thrust curve
    - Dynamic CG calculation throughout flight
    - Realistic angle of attack limits
    - Full thrust curve utilization
    """

    def __init__(self):
        # Physical constants
        self.g = 9.80665 # gravity (m/s^2)
        self.rho_air = 1.225 # air density (kg/m^3)
        self.rho_water = 1000 # water density (kg/m^3)

        # Simulation parameters
        self.dt = 0.005
        self.t_max = 60 # total simulation time (s)

        # Physics limits
        self.max_realistic_velocity = 100 # (m/s)
        self.max_realistic_aoa = 15 # (degrees)

        # Storage
        self.results = {}
        self.events = {}

        # Build thrust curve data
        self.setup_thrust_curve()

        # Build rocket geometry and aerodynamic tables
        self.setup_rocket_geometry()
        self.setup_aerodynamics()

    def setup_thrust_curve(self):
        """Setup thrust curve from experimental data"""
        thrust_time_data = """
0.00000 269.62
0.00660 268.58
```

```
0.01315 265.55
0.01975 260.62
0.02630 254.03
0.03290 245.95
0.03945 236.76
0.04600 226.69
0.05260 215.92
0.05915 204.87
0.06575 193.59
0.06783 190.04
0.06852 34.53
0.06918 113.04
0.06987 204.29
0.07052 276.00
0.07118 326.01
0.07187 358.49
0.07253 376.28
0.07319 385.80
0.07388 390.43
0.07454 391.88
0.07523 391.64
0.07589 390.45
0.07655 388.75
0.07724 386.68
0.07790 384.56
0.07856 382.39
0.07924 380.10
0.07990 377.94
0.08059 375.70
0.08125 373.60
0.08191 371.55
0.08260 369.45
0.08326 367.50
0.08392 365.59
0.08461 363.66
0.08527 361.86
0.08596 360.04
0.08662 358.35
0.08727 356.72
0.08796 354.99
0.08862 353.30
0.08928 351.65
0.08997 349.97
0.09063 348.42
0.09132 346.86
0.09198 345.41
0.09264 344.02
0.09333 342.63
0.09399 341.35
0.09465 340.06
0.09534 338.77
0.09600 337.59
0.09668 336.42
0.09734 335.36
0.09800 334.36
0.09869 333.37
0.09935 332.39
0.10001 331.46
```

```
0.10070 330.52
0.10136 329.65
0.10205 328.77
0.10271 327.81
0.10337 332.06
0.10405 334.21
0.10471 334.74
0.10537 333.98
0.10606 331.87
0.10672 328.24
0.10741 321.58
0.10807 310.56
0.10873 289.30
0.10942 235.09
0.11008 89.48
0.11074 0.00
"""
        lines = thrust_time_data.strip().split('\n')
        times = []
        thrusts = []
        for line in lines:
            t, thrust = map(float, line.split())
            times.append(t)
            thrusts.append(thrust)

        self.thrust_times = np.array(times)
        self.thrust_values = np.array(thrusts)

        self.thrust_interp = interp1d(
            self.thrust_times, self.thrust_values,
            bounds_error=False, fill_value=0.0
        )
        self.total_impulse = np.trapz(self.thrust_values, self.thrust_times)
        self.burn_time = self.thrust_times[-1]

        # Assume 0.5kg of water propellant
        self.effective_exhaust_velocity = self.total_impulse / 0.5

        print(f"Thrust Curve Analysis:")
        print(f" Burn time: {self.burn_time:.4f} s")
        print(f" Peak thrust: {np.max(self.thrust_values):.1f} N")
        print(f" Total impulse: {self.total_impulse:.1f} Ns")
        print(f" Effective exhaust velocity: {self.effective_exhaust_velocity:.1f} m/s"
            )

    def setup_rocket_geometry(self):
        """Define rocket geometry, wing/fins, masses, and reference areas"""
        # Fuselage
        self.fuselage_length = 1.43 # m
        self.fuselage_OD = 0.03 # m
        self.fuselage_ID = 0.028 # m

        # Component positions (measured from bottom)
        self.fin_position = 0.0
        self.wing_position = 0.42 # default config: 0.42m from bottom
        self.bottle_position = 0.70
        self.sloshing_tank_bottom = self.fuselage_length - 0.10

```

```
        # Small fins (3 fins)
        self.num_fins = 3
        self.fin_root_chord = 0.20 # m
        self.fin_tip_chord = 0.07 # m
        self.fin_height = 0.10 # m
        self.fin_thickness = 0.005 # m

        # Large wings (default)
        self.wing_root_chord = 0.55 # m
        self.wing_tip_chord = 0.25 # m
        self.wing_height = 1 # m
        self.wing_thickness = 0.01 # m

        # Sloshing tank
        self.slosh_tank_diameter = 0.08 # m
        self.slosh_tank_height = 0.1989 # m

        # Propellant data
        self.bottle_volume = 2.25e-3 # m
        self.initial_water_volume = 0.5e-3 # m
        self.nozzle_diameter = 0.022 # m
        self.launch_angle = 85 # degrees
        self.slosh_water_volume = 0.5e-3 # m

        # Compute areas for fins & wings (trapezoidal)
        self.fin_area = self.calculate_trapezoidal_area(
            self.fin_root_chord, self.fin_tip_chord, self.fin_height
        )
        self.wing_area = self.calculate_trapezoidal_area(
            self.wing_root_chord, self.wing_tip_chord, self.wing_height
        )
        self.total_fin_area = self.num_fins * self.fin_area

        # Compute component masses
        self.calculate_component_masses()

        # Compute reference areas (body & nozzle) for aerodynamic use
        self.calculate_reference_areas()

        print(f"\nRocket Configuration:")
        print(f" Fuselage: {self.fuselage_length:.2f}m long, {self.fuselage_OD*100:.1f}
            cm OD")
        print(f" Small fins: {self.total_fin_area*10000:.0f}cm total")
        print(f" Large wings: {self.wing_area*2*10000:.0f}cm total")
        print(f" Empty mass: {self.dry_mass:.1f}kg")
        print(f" Initial mass: {self.total_initial_mass:.1f}kg")

    def calculate_trapezoidal_area(self, root_chord, tip_chord, height):
        return 0.5 * (root_chord + tip_chord) * height

    def calculate_component_masses(self):
        """Calculate masses of all structural and propellant components"""
        self.fuselage_mass = 0.2 # kg
        self.fin_mass = 0.045 # kg
        self.wing_mass = 0.717 # kg
        self.bottle_mass = 0.055 # kg
        self.sloshing_tank_mass = 0.1 # kg

```

```
        self.dry_mass = (
            self.fuselage_mass +
            self.fin_mass +
            self.wing_mass +
            self.bottle_mass +
            self.sloshing_tank_mass
        )

        # Propellant (water) masses
        self.initial_water_mass = self.initial_water_volume * self.rho_water
        self.slosh_water_mass = self.slosh_water_volume * self.rho_water

        # Total initial mass
        self.total_initial_mass = (
            self.dry_mass + self.initial_water_mass + self.slosh_water_mass
        )

    def calculate_reference_areas(self):
        """Compute reference body (cylindrical) and nozzle areas"""
        self.ref_area_body = np.pi * (self.fuselage_OD / 2) ** 2
        self.nozzle_area = np.pi * (self.nozzle_diameter / 2) ** 2

    def calculate_cg_position(self, current_water_mass, slosh_x):
        """
        Calculate dynamic CG (from bottom) given current water mass and slosh
            displacement.
        slosh_x is sloshing tank offset relative to its nominal CG.
        """
        components = [
            (self.fuselage_length / 2, self.fuselage_mass), # Fuselage CG
            (self.fin_position + self.fin_height / 3, self.fin_mass), # Fins CG
            (self.wing_position + self.wing_height / 2, self.wing_mass), # Wings CG
            (self.bottle_position, self.bottle_mass + current_water_mass),# Bottle +
                water
            (
                self.sloshing_tank_bottom + self.slosh_tank_height / 2 + slosh_x,
                self.sloshing_tank_mass + self.slosh_water_mass,
            ) # Sloshing tank + water
        ]
        total_moment = sum(pos * mass for pos, mass in components)
        total_mass = sum(mass for _, mass in components)
        return total_moment / total_mass

    def calculate_cp_position(self):
        """Estimate CP based on area-weighted contributions (body, fins, wings)"""
        body_cp = self.fuselage_length * 0.6
        fin_cp = self.fin_position + self.fin_height * 0.4
        wing_cp = self.wing_position + self.wing_height * 0.4

        body_area = self.ref_area_body * self.fuselage_length
        fin_area_contrib = self.total_fin_area
        wing_area_contrib = self.wing_area * 2
        total_aero_area = body_area + fin_area_contrib + wing_area_contrib

        cp_position = (
            (body_cp * body_area + fin_cp * fin_area_contrib + wing_cp *
                wing_area_contrib)
            / total_aero_area
```

```
        )
        return cp_position

    def setup_aerodynamics(self):
        """Initialize aerodynamic coefficient tables (Cd, Cl vs. ) for body, fins,
            wings"""
        self.alpha_range = np.array([-20, -10, -5, 0, 5, 10, 15, 20])

        # Body drag (Cd) & lift (Cl)
        self.CD_body = np.array([1.5, 1.2, 1.0, 0.8, 0.9, 1.1, 1.4, 1.8])
        self.CL_body = np.zeros_like(self.alpha_range)

        # Fin coefficients
        self.CD_fin = np.array([0.08, 0.06, 0.05, 0.04, 0.045, 0.055, 0.07, 0.09])
        self.CL_fin = np.array([-0.3, -0.15, -0.05, 0.05, 0.2, 0.35, 0.4, 0.3])

        # Wing coefficients
        self.CD_wing = np.array([0.15, 0.12, 0.10, 0.08, 0.09, 0.11, 0.13, 0.16])
        self.CL_wing = np.array([-0.8, -0.4, -0.1, 0.2, 0.5, 0.8, 0.9, 0.6])

    def compute_aerodynamics(self, V, alpha, cg_pos, cp_pos):
        """
        Given speed V, angle of attack alpha, CG & CP positions:
        - Interpolate Cd, Cl for body/fin/wing
        - Compute total drag, lift, and pitching moment (reduced lever)
        """
        V = min(V, self.max_realistic_velocity)
        alpha_deg = np.rad2deg(alpha)
        alpha_deg = np.clip(alpha_deg, -25, 25)
        q = 0.5 * self.rho_air * V**2

        CD_body = np.interp(alpha_deg, self.alpha_range, self.CD_body)
        CL_body = np.interp(alpha_deg, self.alpha_range, self.CL_body)

        CD_fin = np.interp(alpha_deg, self.alpha_range, self.CD_fin)
        CL_fin = np.interp(alpha_deg, self.alpha_range, self.CL_fin)

        CD_wing = np.interp(alpha_deg, self.alpha_range, self.CD_wing)
        CL_wing = np.interp(alpha_deg, self.alpha_range, self.CL_wing)

        drag_body = q * self.ref_area_body * CD_body
        lift_body = q * self.ref_area_body * CL_body

        drag_fin = q * self.total_fin_area * CD_fin
        lift_fin = q * self.total_fin_area * CL_fin

        effective_wing_area = self.wing_area * 2
        drag_wing = q * effective_wing_area * CD_wing
        lift_wing = q * effective_wing_area * CL_wing

        total_drag = drag_body + drag_fin + drag_wing
        total_lift = lift_body + lift_fin + lift_wing

        moment_arm = cp_pos - cg_pos
        # apply reduced lever factor 0.05 to keep moments realistic
        pitching_moment = total_lift * moment_arm * 0.05

        return min(total_drag, 500), min(total_lift, 200), pitching_moment
```

```
341
342     def compute_thrust_and_mass_flow(self, t):
343         """Return thrust(t) and mass_flow_rate = thrust / Ve (if within burn)"""
344         thrust = self.thrust_interp(t)
345         if thrust <= 0 or t > self.burn_time:
346             return 0.0, 0.0
347         mass_flow_rate = thrust / self.effective_exhaust_velocity
348         return thrust, mass_flow_rate
349
350     def compute_sloshing_effects(self, slosh_x, slosh_vx):
351         """
352         Simple slosh model (spring-damper) with reduced constants:
353         Sloshing force & moment about CG.
354         """
355         k_slosh = 300
356         c_slosh = 25
357         slosh_force = -k_slosh * slosh_x - c_slosh * slosh_vx
358         slosh_force *= 0.05
359         moment_arm = 0.1
360         slosh_moment = self.slosh_water_mass * self.g * slosh_x * moment_arm * 0.05
361         return slosh_force, slosh_moment
362
363     def derivatives(self, t, state):
364         """
365         Compute time derivatives of state = [x, z, vx, vz, theta, omega,
              current_water_mass, slosh_x, slosh_vx]
366         """
367         x, z, vx, vz, theta, omega, current_water_mass, slosh_x, slosh_vx = state
368
369         # Prevent penetration below z = 0
370         if z < 0:
371             z = 0
372             vz = max(0, vz)
373
374         # Clamp water mass to non-negative
375         current_water_mass = max(0, current_water_mass)
376
377         # Compute CG, CP
378         cg_position = self.calculate_cg_position(current_water_mass, slosh_x)
379         cp_position = self.calculate_cp_position()
380
381         # Total mass & moment of inertia (about CG)
382         current_total_mass = self.dry_mass + current_water_mass + self.slosh_water_mass
383         current_total_mass = max(current_total_mass, 0.3)
384         current_I = 0.1 * current_total_mass * self.fuselage_length**2
385         current_I = max(current_I, 0.01)
386
387         # Speed & flight-path angle
388         V = np.sqrt(vx**2 + vz**2)
389         V = max(V, 0.1)
390         gamma = np.arctan2(vz, vx)
391
392         # Angle of attack (clamped)
393         alpha = theta - gamma
394         max_alpha_rad = np.deg2rad(self.max_realistic_aoa)
395         alpha = np.clip(alpha, -max_alpha_rad, max_alpha_rad)
396
397         # Thrust & mass_flow
```

```
        thrust, mass_flow_rate = self.compute_thrust_and_mass_flow(t)

        # Sloshing
        slosh_force, slosh_moment = self.compute_sloshing_effects(slosh_x, slosh_vx)

        # Aerodynamic drag/lift & aero-moment
        drag, lift, moment_aero = self.compute_aerodynamics(V, alpha, cg_position,
            cp_position)

        # Decompose thrust in inertial frame (along body pitch )
        Fx_thrust = thrust * np.cos(theta)
        Fz_thrust = thrust * np.sin(theta)

        # Decompose aerodynamic forces in inertial frame
        Fx_aero = -drag * (vx / V) + lift * (-vz / V)
        Fz_aero = -drag * (vz / V) + lift * (vx / V)

        # Net inertial forces
        Fx_inertial = Fx_thrust + Fx_aero + slosh_force * 0.01
        Fz_inertial = Fz_thrust + Fz_aero + slosh_force * 0.01 - current_total_mass *
            self.g

        ax = np.clip(Fx_inertial / current_total_mass, -50, 50)
        az = np.clip(Fz_inertial / current_total_mass, -50, 50)

        # Angular dynamics
        M_total = moment_aero + slosh_moment
        alpha_dot = np.clip(M_total / current_I, -5, 5)
        omega = np.clip(omega, -10, 10)

        # Sloshing dynamics
        slosh_acceleration = np.clip(
            (-300 * slosh_x / max(self.slosh_water_mass, 0.001)
             - 25 * slosh_vx / max(self.slosh_water_mass, 0.001)
             + ax * 0.05),
            -20, 20
        )

        return np.array([
            vx, # dx/dt
            vz, # dz/dt
            ax, # dvx/dt
            az, # dvz/dt
            omega, # d/dt
            alpha_dot, # d/dt
            -mass_flow_rate, # d(m_water)/dt
            slosh_vx, # d(slosh_x)/dt
            slosh_acceleration # d(slosh_vx)/dt
        ])

    def ground_impact_event(self, t, state):
        return state[1] - 0.1
    ground_impact_event.terminal = True
    ground_impact_event.direction = -1

    def run_simulation(self):
        print(f"\nStarting Rocket Simulation...")
        print(f" AOA limited to: {self.max_realistic_aoa}")
```

```
        print(f" Using entire thrust curve: 0 to {self.burn_time:.4f}s")

        theta0 = np.deg2rad(self.launch_angle)
        initial_state = np.array([
            0, # x (m)
            0.5, # z (m above ground)
            8 * np.cos(theta0), # vx (m/s)
            8 * np.sin(theta0), # vz (m/s)
            theta0, #  (rad)
            0, #  (rad/s)
            self.initial_water_mass, # current water mass (kg)
            0, # slosh_x (m)
            0 # slosh_vx (m/s)
        ])

        t_span = (0, self.t_max)
        t_eval = np.arange(0, self.t_max, self.dt)

        sol = solve_ivp(
            self.derivatives,
            t_span,
            initial_state,
            t_eval=t_eval,
            method='RK45',
            events=self.ground_impact_event,
            rtol=1e-5,
            atol=1e-7,
            max_step=0.02
        )

        self.results = {
            'time': sol.t,
            'states': sol.y.T,
            'success': sol.success
        }
        if sol.t_events[0].size > 0:
            self.events['impact_time'] = sol.t_events[0][0]

        print(f"Simulation completed. Steps: {len(sol.t)}")
        return self.results

    def plot_and_save_individual(self, folder_prefix):
        """
        Generate and save each of the 15 subplots as a separate PNG file.
        The files will be named:
          {folder_prefix}_01_thrust_curve.png
          {folder_prefix}_02_trajectory.png
          {folder_prefix}_03_altitude_vs_time.png
          {folder_prefix}_04_velocity_vs_time.png
          {folder_prefix}_05_mass_flow_rate.png
          {folder_prefix}_06_cg_cp_positions.png
          {folder_prefix}_07_stability_margin.png
          {folder_prefix}_08_angle_of_attack.png
          {folder_prefix}_09_pitch_flightpath_angles.png
          {folder_prefix}_10_remaining_propellant.png
          {folder_prefix}_11_sloshing_displacement.png
          {folder_prefix}_12_angular_velocity.png
          {folder_prefix}_13_mass_distribution.png
```

```
          {folder_prefix}_14_thrust_to_weight.png
          {folder_prefix}_15_cg_shift.png
        """

        if not self.results:
            return

        # Create output directory if it doesn't exist
        os.makedirs(folder_prefix, exist_ok=True)

        time = self.results['time']
        states = self.results['states']
        x, z = states[:, 0], states[:, 1]
        vx, vz = states[:, 2], states[:, 3]
        theta = np.rad2deg(states[:, 4])
        omega = np.rad2deg(states[:, 5])
        current_water_mass = states[:, 6]
        slosh_x = states[:, 7]

        V = np.sqrt(vx**2 + vz**2)
        gamma = np.rad2deg(np.arctan2(vz, vx))
        alpha = theta - gamma

        # Compute thrust & mass flow over time
        thrust_over_time = []
        mass_flow_over_time = []
        for t in time:
            thrust, mass_flow = self.compute_thrust_and_mass_flow(t)
            thrust_over_time.append(thrust)
            mass_flow_over_time.append(mass_flow)

        # Compute CG, CP, total mass, stability margin, CG shift
        cg_positions = []
        cp_positions = []
        total_masses = []
        for i in range(len(time)):
            cg_pos = self.calculate_cg_position(current_water_mass[i], slosh_x[i])
            cp_pos = self.calculate_cp_position()
            total_mass = self.dry_mass + current_water_mass[i] + self.slosh_water_mass
            cg_positions.append(cg_pos)
            cp_positions.append(cp_pos)
            total_masses.append(total_mass)

        stability_margin = np.array(cp_positions) - np.array(cg_positions)
        cg_shift = np.array(cg_positions) - cg_positions[0]

        # 1. Thrust curve
        plt.figure()
        plt.plot(self.thrust_times, self.thrust_values, 'r-', linewidth=2, label='
            Experimental')
        plt.plot(time, thrust_over_time, 'b--', linewidth=1, label='Used in sim')
        plt.xlabel('Time (s)')
        plt.ylabel('Thrust (N)')
        plt.title('ENTIRE Thrust Curve Used')
        plt.grid(True)
        plt.legend()
        plt.tight_layout()
        plt.savefig(f"{folder_prefix}/01_thrust_curve.png", dpi=300)
```

```
        plt.close()

        # 2. Trajectory (Range vs Altitude)
        plt.figure()
        plt.plot(x, z, 'b-', linewidth=2)
        plt.xlabel('Range (m)')
        plt.ylabel('Altitude (m)')
        plt.title('Flight Trajectory')
        plt.grid(True)
        plt.tight_layout()
        plt.savefig(f"{folder_prefix}/02_trajectory.png", dpi=300)
        plt.close()

        # 3. Altitude vs Time
        plt.figure()
        plt.plot(time, z, 'b-', linewidth=2)
        plt.xlabel('Time (s)')
        plt.ylabel('Altitude (m)')
        plt.title('Altitude vs Time')
        plt.grid(True)
        plt.tight_layout()
        plt.savefig(f"{folder_prefix}/03_altitude_vs_time.png", dpi=300)
        plt.close()

        # 4. Velocity vs Time
        plt.figure()
        plt.plot(time, V, 'r-', linewidth=2)
        plt.xlabel('Time (s)')
        plt.ylabel('Velocity (m/s)')
        plt.title('Velocity vs Time')
        plt.grid(True)
        plt.tight_layout()
        plt.savefig(f"{folder_prefix}/04_velocity_vs_time.png", dpi=300)
        plt.close()

        # 5. Mass Flow Rate vs Time
        plt.figure()
        plt.plot(time, mass_flow_over_time, 'g-', linewidth=2)
        plt.xlabel('Time (s)')
        plt.ylabel('Mass Flow Rate (kg/s)')
        plt.title('Propellant Mass Flow Rate')
        plt.grid(True)
        plt.tight_layout()
        plt.savefig(f"{folder_prefix}/05_mass_flow_rate.png", dpi=300)
        plt.close()

        # 6. CG and CP Positions vs Time
        plt.figure()
        plt.plot(time, cg_positions, 'g-', linewidth=2, label='CG Position (Dynamic)')
        plt.plot(time, cp_positions, 'r--', linewidth=2, label='CP Position (Fixed)')
        plt.xlabel('Time (s)')
        plt.ylabel('Position from bottom (m)')
        plt.title('CG and CP Positions')
        plt.legend()
        plt.grid(True)
        plt.tight_layout()
        plt.savefig(f"{folder_prefix}/06_cg_cp_positions.png", dpi=300)
        plt.close()
```

```
627
628         # 7. Stability Margin vs Time
629         plt.figure()
630         plt.plot(time, stability_margin, 'k-', linewidth=2)
631         plt.axhline(y=0, color='r', linestyle='--', alpha=0.5)
632         plt.xlabel('Time (s)')
633         plt.ylabel('CP  CG (m)')
634         plt.title('Static Stability Margin')
635         plt.grid(True)
636         plt.tight_layout()
637         plt.savefig(f"{folder_prefix}/07_stability_margin.png", dpi=300)
638         plt.close()
639
640         # 8. Angle of Attack vs Time
641         plt.figure()
642         plt.plot(time, alpha, 'g-', linewidth=2)
643         plt.xlabel('Time (s)')
644         plt.ylabel('Angle of Attack (deg)')
645         plt.title(f'Angle of Attack (Limited to {self.max_realistic_aoa})')
646         plt.ylim([-20, 20])
647         plt.grid(True)
648         plt.tight_layout()
649         plt.savefig(f"{folder_prefix}/08_angle_of_attack.png", dpi=300)
650         plt.close()
651
652         # 9. Pitch & Flight Path Angles vs Time
653         plt.figure()
654         plt.plot(time, theta, 'b-', linewidth=2, label='Pitch')
655         plt.plot(time, gamma, 'r--', linewidth=2, label='Flight Path')
656         plt.xlabel('Time (s)')
657         plt.ylabel('Angle (deg)')
658         plt.title('Pitch and Flight Path Angles')
659         plt.legend()
660         plt.grid(True)
661         plt.tight_layout()
662         plt.savefig(f"{folder_prefix}/09_pitch_flightpath_angles.png", dpi=300)
663         plt.close()
664
665         # 10. Remaining Propellant Mass vs Time
666         plt.figure()
667         plt.plot(time, current_water_mass, 'b-', linewidth=2)
668         plt.xlabel('Time (s)')
669         plt.ylabel('Propellant Mass (kg)')
670         plt.title('Remaining Propellant Mass')
671         plt.grid(True)
672         plt.tight_layout()
673         plt.savefig(f"{folder_prefix}/10_remaining_propellant.png", dpi=300)
674         plt.close()
675
676         # 11. Sloshing Displacement vs Time
677         plt.figure()
678         plt.plot(time, slosh_x * 1000, 'c-', linewidth=2)
679         plt.xlabel('Time (s)')
680         plt.ylabel('Sloshing Displacement (mm)')
681         plt.title('Sloshing Motion (500mL)')
682         plt.grid(True)
683         plt.tight_layout()
684         plt.savefig(f"{folder_prefix}/11_sloshing_displacement.png", dpi=300)
```

```
        plt.close()

        # 12. Angular Velocity vs Time
        plt.figure()
        plt.plot(time, omega, 'k-', linewidth=2)
        plt.xlabel('Time (s)')
        plt.ylabel('Angular Velocity (deg/s)')
        plt.title('Angular Velocity')
        plt.grid(True)
        plt.tight_layout()
        plt.savefig(f"{folder_prefix}/12_angular_velocity.png", dpi=300)
        plt.close()

        # 13. Mass Distribution vs Time (Total, Propellant, Dry, Slosh)
        plt.figure()
        plt.plot(time, total_masses, 'k-', linewidth=2, label='Total Mass')
        plt.plot(time, current_water_mass, 'b-', linewidth=2, label='Propellant')
        plt.axhline(y=self.dry_mass, color='r', linestyle='--', label='Dry Mass')
        plt.axhline(y=self.slosh_water_mass, color='c', linestyle=':', label='Sloshing
            Water')
        plt.xlabel('Time (s)')
        plt.ylabel('Mass (kg)')
        plt.title('Mass Distribution (Dynamic)')
        plt.legend()
        plt.grid(True)
        plt.tight_layout()
        plt.savefig(f"{folder_prefix}/13_mass_distribution.png", dpi=300)
        plt.close()

        # 14. Thrust-to-Weight Ratio vs Time
        thrust_to_weight = np.array(thrust_over_time) / (np.array(total_masses) * self.
            g)
        plt.figure()
        plt.plot(time, thrust_to_weight, 'm-', linewidth=2)
        plt.xlabel('Time (s)')
        plt.ylabel('Thrust-to-Weight Ratio')
        plt.title('Thrust-to-Weight Ratio (Dynamic)')
        plt.grid(True)
        plt.tight_layout()
        plt.savefig(f"{folder_prefix}/14_thrust_to_weight.png", dpi=300)
        plt.close()

        # 15. CG Shift (mm) vs Time
        plt.figure()
        plt.plot(time, cg_shift * 1000, 'purple', linewidth=2)
        plt.xlabel('Time (s)')
        plt.ylabel('CG Shift (mm)')
        plt.title('Center of Gravity Movement')
        plt.grid(True)
        plt.tight_layout()
        plt.savefig(f"{folder_prefix}/15_cg_shift.png", dpi=300)
        plt.close()

    def analyze_performance(self):
        """Print a summary of max altitude, range, velocities, AOA, CG shift, etc."""
        if not self.results:
            return

```

```
        time = self.results['time']
        states = self.results['states']

        x, z = states[:, 0], states[:, 1]
        vx, vz = states[:, 2], states[:, 3]
        V = np.sqrt(vx**2 + vz**2)
        current_water_mass = states[:, 6]
        slosh_x = states[:, 7]

        max_altitude = np.max(z)
        max_range = np.max(x)
        max_velocity = np.max(V)
        max_acceleration = np.max(np.gradient(V, time))

        theta = np.rad2deg(states[:, 4])
        gamma = np.rad2deg(np.arctan2(vz, vx))
        alpha = theta - gamma
        max_aoa = np.max(np.abs(alpha))

        cg_positions = [
            self.calculate_cg_position(wm, sx) for wm, sx in zip(current_water_mass,
                slosh_x)
        ]
        cg_shift = max(cg_positions) - min(cg_positions)

        print("\n" + "=" * 70)
        print("ROCKET PERFORMANCE ANALYSIS")
        print("=" * 70)
        print(f"Peak thrust: {np.max(self.thrust_values):.1f} N")
        print(f"Total impulse: {self.total_impulse:.1f} Ns")
        print(f"Effective exhaust velocity: {self.effective_exhaust_velocity:.1f} m/s")
        print(f"Max AOA: {max_aoa:.1f} (limit {self.max_realistic_aoa})")
        print(f"Initial mass: {self.total_initial_mass:.3f} kg")
        print(f"Final mass: {self.dry_mass + current_water_mass[-1] + self.
            slosh_water_mass:.3f} kg")
        print(f"Propellant consumed: {self.initial_water_mass - current_water_mass
            [-1]:.3f} kg")
        print(f"Mass ratio: {self.total_initial_mass / self.dry_mass:.2f}")
        print(f"CG shift: {cg_shift * 1000:.1f} mm")
        print(f"Max altitude: {max_altitude:.1f} m")
        print(f"Max range: {max_range:.1f} m")
        print(f"Max velocity: {max_velocity:.1f} m/s")
        print(f"Max acceleration: {max_acceleration:.1f} m/s")

        cp_position = self.calculate_cp_position()
        stability_margins = [cp_position - cg for cg in cg_positions]
        print(f"CP position: {cp_position:.3f} m (fixed)")
        print(f"Stability margin range: {min(stability_margins):.3f} to {max(
            stability_margins):.3f} m")
        if min(stability_margins) > 0:
            print(" Statically stable throughout flight")
        else:
            print(" Rocket becomes unstable during flight")
        if "impact_time" in self.events:
            print(f"Total flight time: {self.events['impact_time']:.1f} s")
        print("=" * 70)


```

```
def compare_two_configs():
    """
    Create two simulator instances:
    - Config1: Tapered wing geometry
    - Config2: Conventional wings at 0.49m, root_chord=0.40m, tip_chord=0.40m, height
        =0.80m
    Then:
      1. Run each sim and save all 15 individual plots to separate folders
      2. Overlay Altitude vs Time, Velocity vs Time, Range vs Time,
         and Stability Margin vs Time in one combined figure (saved separately)
      3. Compute and print a flight score for each config:
         Flight score = ( sqrt(max_range + max_altitude) + flight_time ) * 0.5 /
             total_weight
    """
    #  CONFIG 1: Default geometry
    sim1 = CustomRocketSimulatorWithThrustCurve()
    results1 = sim1.run_simulation()
    # Create folder "config1_plots" to save individual images
    sim1.plot_and_save_individual("Tapered")

    # Extract metrics from sim1
    t1 = results1['time']
    states1 = results1['states']
    x1, z1 = states1[:, 0], states1[:, 1]
    vx1, vz1 = states1[:, 2], states1[:, 3]
    V1 = np.sqrt(vx1**2 + vz1**2)
    max_range1 = np.max(x1)
    max_altitude1 = np.max(z1)
    flight_time1 = sim1.events.get('impact_time', sim1.t_max)
    total_weight1 = sim1.total_initial_mass

    # Compute flight score for config 1
    flight_score1 = (np.sqrt(max_range1**2 + max_altitude1**2) + flight_time1) * 0.5 /
         total_weight1

    #  CONFIG 2: Modified wings
    sim2 = CustomRocketSimulatorWithThrustCurve()
    # Override wing geometry
    sim2.wing_position = 0.47
    sim2.wing_root_chord = 0.40
    sim2.wing_tip_chord = 0.40
    sim2.wing_height = 1
    sim2.wing_thickness = 0.01
    # Recompute wing area & reference areas
    sim2.wing_area = sim2.calculate_trapezoidal_area(
        sim2.wing_root_chord, sim2.wing_tip_chord, sim2.wing_height
    )
    sim2.total_fin_area = sim2.num_fins * sim2.fin_area # fins unchanged
    sim2.calculate_reference_areas()

    results2 = sim2.run_simulation()
    # Create folder "config2_plots" to save individual images
    sim2.plot_and_save_individual("Conventional")

    # Extract metrics from sim2
    t2 = results2['time']
    states2 = results2['states']
    x2, z2 = states2[:, 0], states2[:, 1]
```

```
    vx2, vz2 = states2[:, 2], states2[:, 3]
    V2 = np.sqrt(vx2**2 + vz2**2)
    max_range2 = np.max(x2)
    max_altitude2 = np.max(z2)
    flight_time2 = sim2.events.get('impact_time', sim2.t_max)
    total_weight2 = sim2.total_initial_mass

    # Compute flight score for config 2
    flight_score2 = (np.sqrt(max_range2**2 + max_altitude2**2) + flight_time2) * 0.5 /
        total_weight2

    # CG & CP for stability margin
    cg1 = np.array([
        sim1.calculate_cg_position(wm, sx)
        for wm, sx in zip(states1[:, 6], states1[:, 7])
    ])
    cg2 = np.array([
        sim2.calculate_cg_position(wm, sx)
        for wm, sx in zip(states2[:, 6], states2[:, 7])
    ])
    cp1 = sim1.calculate_cp_position()
    cp2 = sim2.calculate_cp_position()
    stability1 = cp1 - cg1
    stability2 = cp2 - cg2

    #  Plot Combined Comparisons
    plt.figure(figsize=(12, 10))

    # 1. Altitude vs Time
    plt.subplot(4, 1, 1)
    plt.plot(t1, z1, label='Tapered')
    plt.plot(t2, z2, '--', label='Config2 (shifted/expanded wings)')
    plt.xlabel('Time (s)')
    plt.ylabel('Altitude (m)')
    plt.title('Altitude vs Time Comparison')
    plt.legend()
    plt.grid(True)

    # 2. Velocity vs Time
    plt.subplot(4, 1, 2)
    plt.plot(t1, V1, label='Tapered')
    plt.plot(t2, V2, '--', label='Conventional')
    plt.xlabel('Time (s)')
    plt.ylabel('Velocity (m/s)')
    plt.title('Velocity vs Time Comparison')
    plt.legend()
    plt.grid(True)

    # 3. Range vs Time
    plt.subplot(4, 1, 3)
    plt.plot(t1, x1, label='Tapered')
    plt.plot(t2, x2, '--', label='Conventional')
    plt.xlabel('Time (s)')
    plt.ylabel('Range (m)')
    plt.title('Range vs Time Comparison')
    plt.legend()
    plt.grid(True)

```

```
    # 4. Stability Margin vs Time
    plt.subplot(4, 1, 4)
    plt.plot(t1, stability1, label='Tapered')
    plt.plot(t2, stability2, '--', label='Conventional')
    plt.axhline(0, color='r', linestyle='--', alpha=0.5)
    plt.xlabel('Time (s)')
    plt.ylabel('Stability Margin (m)')
    plt.title('Stability Margin vs Time Comparison')
    plt.legend()
    plt.grid(True)

    plt.tight_layout()

    # Save the combined figure as well
    plt.savefig('rocket_comparison_combined.png', dpi=300)
    plt.show()

    #  Print Flight Scores
    print("\n" + "=" * 60)
    print("Flight Score (units arbitrary) for each configuration:")
    print("=" * 60)
    print(f"Config1: max_range = {max_range1:.2f}m, "
          f"max_altitude = {max_altitude1:.2f}m, "
          f"flight_time = {flight_time1:.2f}s, "
          f"total_weight = {total_weight1:.2f}kg")
    print(f"  Flight Score1 = "
          f"(({max_range1:.2f}^2 + {max_altitude1:.2f}^2)^0.5 + {flight_time1:.2f})"
          f" * 0.5 / {total_weight1:.2f} = {flight_score1:.4f}")
    print()
    print(f"Config2: max_range = {max_range2:.2f}m, "
          f"max_altitude = {max_altitude2:.2f}m, "
          f"flight_time = {flight_time2:.2f}s, "
          f"total_weight = {total_weight2:.2f}kg")
    print(f"  Flight Score2 = "
          f"(({max_range2:.2f}^2 + {max_altitude2:.2f}^2)^0.5 + {flight_time2:.2f})"
          f" * 0.5 / {total_weight2:.2f} = {flight_score2:.4f}")
    print("=" * 60)


if __name__ == "__main__":
    compare_two_configs()
```

```
============================================================
Flight Score (units arbitrary) for each configuration:
============================================================
Tapered: max_range = 15.35 m, max_altitude = 9.17 m, flight_time = 4.17 s,
    total_weight = 2.12 kg
   Flight Score = ((15.35^2 + 9.17^2)^0.5 + 4.17) * 0.5 / 2.12 = 5.2072

Conventional: max_range = 15.94 m, max_altitude = 9.17 m, flight_time = 4.29 s,
    total_weight = 2.12 kg
   Flight Score = ((15.94^2 + 9.17^2)^0.5 + 4.29) * 0.5 / 2.12 = 5.3560
============================================================
```

D Appendix: Word Count Validation

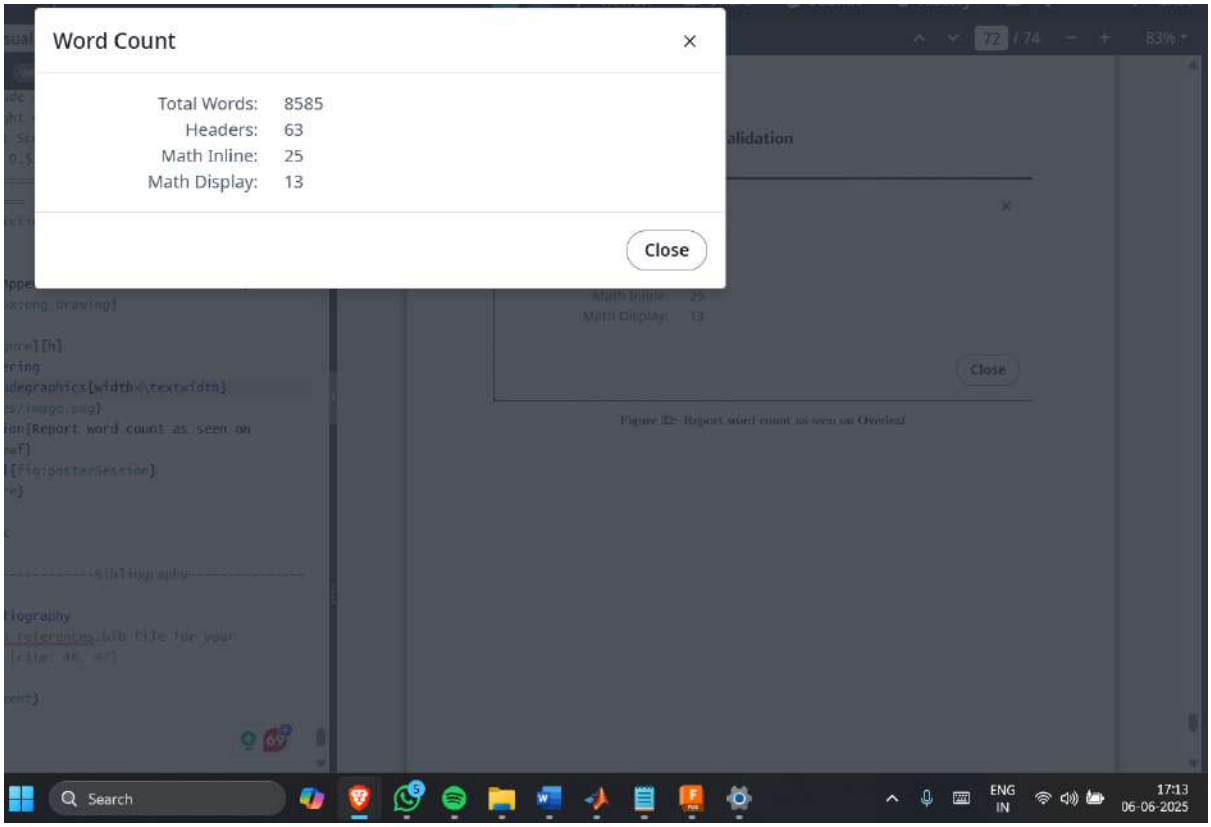


Figure 32: Report word count as seen on Overleaf

References

Air Command Rockets. (2024). Water Rocket Simulator [Accessed: 2025-06-06].

Al-Yacouby, A. M., & Ahmed, M. M. (2022). A Numerical Study on the Effects of Perforated and Imperforate Baffles on the Sloshing Pressure of a Rectangular Tank. *Journal of Marine Science and Engineering*, *10*(10), 1335. https://doi.org/10.3390/jmse10101335

Antar, B. N., & Nuotio-Antar, V. S. (1994). *Fundamentals of Low Gravity Fluid Dynamics and Heat Transfer*. CRC Press.

Chato, D. J., Hartwig, J. W., Rame, E., & McQuillen, J. B. (2014, July). *Inverted Outflow Ground Testing of Cryogenic Propellant Liquid Acquisition Devices* (NASA Technical Note No. GRC-E-DAA-TN16373). NASA Glenn Research Center. Cleveland, OH. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140016830.pdf

Hartwig, J. W. (2017). *A Detailed Historical Review of Propellant Management Devices for Low Gravity Propellant Acquisition* (NASA Technical Memorandum No. 20170000667). NASA Glenn Research Center. Cleveland, OH. https://ntrs.nasa.gov/api/citations/20170000667/downloads/20170000667.pdf

Ibrahim, M. A. (2015). Aerodynamic characteristics of blended-wing-body aircraft. *Aerospace Sciences and Aviation Technology*, *22*(1), 51–61. https://doi.org/10.21608/asat.2015.24770

Instructables. (n.d.). Easy-to-build, easy-to-use, water bottle launcher! [Accessed: 2025-06-06].

Jain, P., Mittal, V., Karani, A., Anne, M., Bhosale, V., S, V., Karanth, D., Yadav, H., Kripal, K., Narula, J., Soni, A., Singh, K., Agarwal, S., Dutta, B., Mahesh, C., & Subash, D. (2024, June). *thrustMIT- Project AgniAstra Team 56 Project Technical Report to the 2024 Spaceport America Cup*. https://doi.org/10.13140/RG.2.2.34959.91042

Janszen, G., Capezzera, G., Grande, A. M., & Di Landro, L. (2019). Mitigation of Impact Damage with Self-Healing and Anti-Sloshing Materials in Aerospace Fuel Tanks. *Aerospace*, *6*(2), 14. https://doi.org/10.3390/aerospace6020014

Jetzschmann, C., Strauch, H., & Bennani, S. (2018). Model Based Active Slosh Damping Experiment. https://arxiv.org/abs/1801.10017

Mason, P., & Starin, S. R. (2018). *The Effects of Propellant Slosh Dynamics on the Solar Dynamics Observatory* (NASA Technical Report No. 20180001148). NASA Goddard Space Flight Center. Greenbelt, MD. https://ntrs.nasa.gov/api/citations/20180001148/downloads/20180001148.pdf

MyGov. (2024). Celebrating India's Young Innovators Who Outshined at Spaceport America Cup 2024. https://blog.mygov.in/editorial/celebrating-indias-young-innovators-who-outshined-at-spaceport-america-cup-2024/

Naseh, H., Alipour, A., & Daneshgar, P. (2019). Modeling and Simulation of Fuel Sloshing in Tank by Pendulum Model. *IOP Conference Series: Materials Science and Engineering*, *72*(4), 042042. https://doi.org/10.1088/1757-899X/72/4/042042

Plaka, E. (2023). Verification and validation of a rapid design tool for the analysis of the composite y-joint of the d8 double-bubble aircraft. *arXiv preprint arXiv:2302.08648*. https://arxiv.org/abs/2302.08648

Sadilek, T., Opificius, J., Wright, J., Leslie, A., Tuzizila, J., Alzate, C., Burnett, H., Atkinson, J., & Stricula, J. (2025). Design and Evaluation of Dual-Resolver Emulation for Control System Verification in Aerospace Actuation Applications, 401–408. https://doi.org/10.1109/.../10977565

Space Propulsion. (n.d.). Hydrazine Bladder Tank Overview [Accessed: 2025-06-05].

SPIE. (2024). Research on flow-resistance calculation of propellant management device for microgravity applications. *Proceedings of SPIE, Volume 13163*, 3030636. https://doi.org/10.1117/12.3030636

Stern, F., Wilson, R. V., Coleman, H. W., & Paterson, E. G. (1999, September). *Verification and validation methodology for cfd simulation results* (IIHR Report No. 407). Iowa Institute of Hydraulic Research, College of Engineering, The University of Iowa. Iowa City, IA. http://www.simman2008.dk/pdf/iihr_407.pdf

Sutton, G. P., & Biblarz, O. (2010). *Rocket Propulsion Elements* (8th). John Wiley Sons.

ThrustMIT. (2025). Thrustmit. https://www.thrustmit.in/

Veldman, A. E. P., Gerrits, J., Luppes, R., Helder, J. A., & Vreeburg, J. P. B. (2007). The numerical simulation of liquid sloshing on board spacecraft. *Journal of Computational Physics*, *224*(1), 82–99. https://doi.org/10.1016/j.jcp.2006.12.020