Proposal Report

by

Team Mangalyaan

10875

December 2025

NIT Raipur, Raipur-492010

Contents

1 **Introduction and Mission Objectives** 3

2 **Overarching System Architecture and Design Philosophy** 4

3 **The ASCEND Aerial Vehicle: Design and Implementation** 5

4 **Autonomous Systems: Navigation, Perception, and Landing** .... 7.

5 **The Autonomous Solar-Powered Base Station** 11

6 **Integrated Mission Lifecycle and Safety Analysis** 12

7 **MECHANICAL PART OF DRONE** 13

8 **Drone Testing Methodologies** 16

9 **Novelty in the overall proposal** 18

10 **Project management** 18

11 **Conclusion** 18

ASCEND UAV System: A Technical Design Report for the ISRO Robotics Challenge – URSC 2026

1.0 Introduction and Mission Objectives

The ISRO Robotics Challenge – URSC 2026 (ASCEND) presents a formidable and realistic simulation of extraterrestrial exploration: the development of a fully autonomous Unmanned Aerial Vehicle (UAV) capable of navigating, mapping, and executing complex tasks in a GPS-denied environment analogous to a planetary surface. This challenge pushes the boundaries of autonomous systems, requiring solutions that are robust, self-reliant, and intelligent.

This document details the ASCEND system, a comprehensive solution engineered to meet and exceed the requirements of this challenge. The system is composed of two primary, deeply integrated components: an autonomous aerial vehicle and a complementary solar-powered base station. Together, they form a persistent exploration platform designed for longevity and operational independence.

The ASCEND system is designed to perform a complete mission cycle without any human intervention. Its primary capabilities include:

- **Autonomous GPS-Denied Navigation:** Establishing and maintaining local position awareness using onboard sensors.
- **Intelligent Feature Detection:** Identifying and documenting targets of interest using a sophisticated vision pipeline.
- **Autonomous Precision Landing:** Safely returning to and docking with the base station with high accuracy.
- **Autonomous Charging:** Replenishing its onboard battery to enable multi-sortie missions.
- **Reliable Data Transfer:** Securely synchronizing mission-critical data with the base station.

The purpose of this report is to provide a detailed technical analysis of the ASCEND system's design, its overarching architecture, the specific hardware selected, and the core algorithms that govern its autonomous behavior. This analysis will demonstrate the system's full compliance with all IRoC-U 2026 rules and its readiness for deployment in challenging Mars-analog environments. The following sections will begin with an overview of the system's high-level architecture and guiding design philosophy.

b) Comprehensive understanding behind choosing the rotorcraft:

Configuration	Advantages	Limitations
Single Rotor Helicopter	High efficiency	Mechanical complexity, control difficulty
Hexacopter	Redundancy, higher payload	Increased weight, power consumption

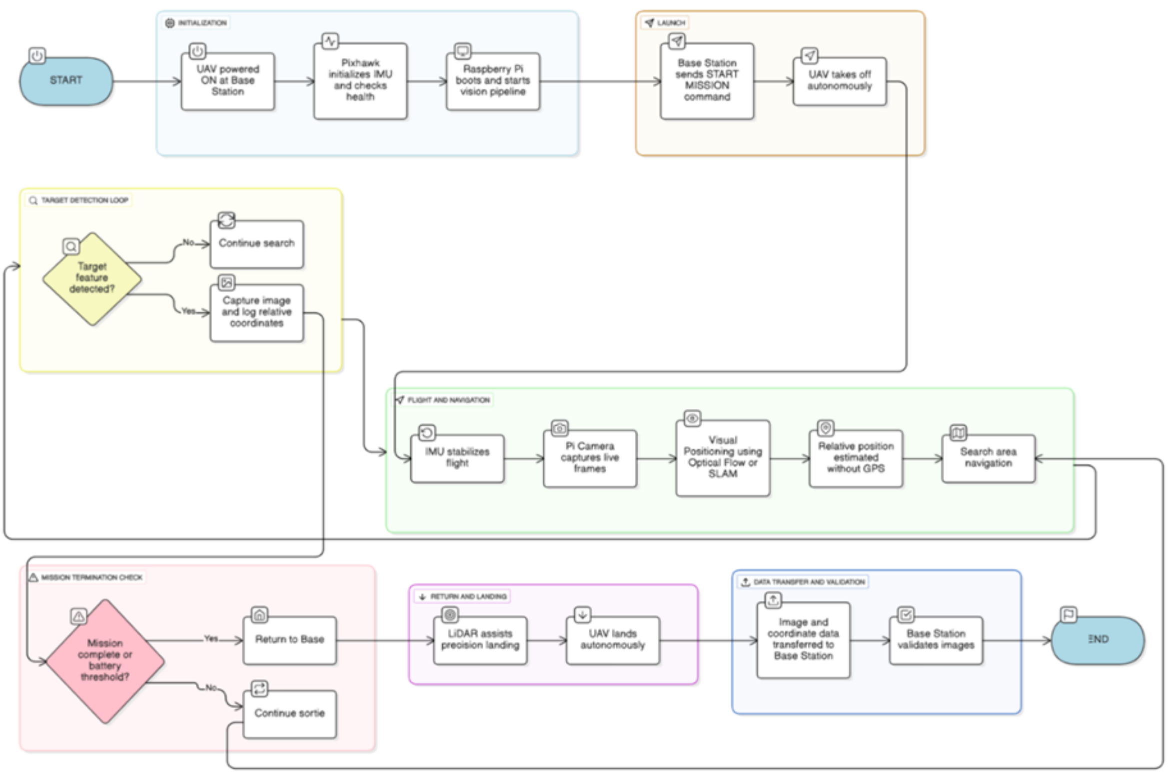
Coaxial Rotorcraft	Compact design	Complex aerodynamics and control
Quadcopter	**Simple structure, high maneuverability, low weight**	No motor redundancy

2.0 Overarching System Architecture and Design Philosophy

In the domain of autonomous robotics, particularly for applications where failure is not an option, the system architecture is of paramount strategic importance. A well-conceived architecture enhances not only performance but also the reliability and fault tolerance critical for planetary missions. The ASCEND system is built upon a foundation designed for resilience, modularity, and operational safety.

The system is organized into two primary subsystems that work in concert: the **ASCEND Aerial Vehicle**, responsible for exploration and data acquisition, and the **Autonomous Solar-Powered Base Station**, which serves as the logistical hub for power and data.

SYSTEM-LEVEL FLOWCHART (MISSION FLOW)



The central theme of the ASCEND design is a **Hybrid Control Architecture**. This philosophy intentionally decouples safety-critical, low-level functions from high-level, computationally intensive tasks. In practice, this means that essential operations like flight stabilization, motor control, and charging safety are managed by dedicated, real-time embedded controllers (a Pixhawk Flight Controller on the UAV and an MCU on the base station). In parallel, complex tasks such as visual navigation, object detection, and mission planning are handled by more powerful, non-real-time processors (a Raspberry Pi on the UAV and a laptop at the base station).

This architectural choice is deliberate and crucial for mission integrity. It ensures that the system can maintain a safe state and continue its core functions even if a high-level computing unit fails. For instance, a laptop crash at the base station will not interrupt or compromise the safety of an active charging cycle, as the embedded MCU continues to manage the process independently. This layered approach to control and computation forms the bedrock of the entire system, as will be detailed in the following breakdown of the aerial zvehicle.

3.0 The ASCEND Aerial Vehicle: Design and Implementation

The ASCEND Aerial Vehicle's design represents a tightly integrated system where mechanical stability, avionics reliability, and computational intelligence are engineered to deliver autonomous flight in unstructured environments. Its design prioritizes stability, efficiency, and the seamless integration of its diverse sensor and processing payloads.

3.1. Mechanical Design and Stability Analysis

The physical arrangement and mass distribution of components are critical to a UAV's flight performance. An optimized center of mass (COM) ensures stable, efficient flight and minimizes strain on the propulsion system. The component layout for the ASCEND vehicle has been meticulously planned to achieve these goals.

The table below summarizes the mass and distribution of the key components:

Component	Mass (g)	X (mm)	Y (mm)	Z (mm)
Drone Frame	300	0	0	0
8-Cell Battery	400	2	0	-25
MPPT Controller	147	20	0	40
Motors (Total)	120	0	0	0
Raspberry Pi 4 Model B	65	-50	0	40
Pixhawk FC	30	0	0	15
Telemetry	25	-100	0	40
Camera	20	120	0	0
PDB Board	12	0	0	-5
Regulators	10	0	0	0
TOTAL	**1129**			

The final calculated Center of Mass for this configuration is:

- **Xcom:** +0.35 mm

- **Ycom:** 0.00 mm
- **Zcom:** -0.11 mm

This specific component arrangement yields significant strategic benefits for flight dynamics:

- **Balance:** The near-zero Xcom value (+0.35 mm) indicates that the vehicle is almost perfectly balanced along its roll axis (front-to-back). This ensures level flight without requiring some motors to work harder than others, thereby increasing efficiency and flight time.
- **Stability:** A negative Zcom value (-0.11 mm) places the vehicle's center of mass slightly below the geometric center of the frame. This creates a natural "pendulum effect," where gravity passively helps the drone self-level and resist external disturbances, enhancing its inherent stability.
- **Vibration Control:** The strategic placement of the heaviest components—the Battery and MPPT controller—acts as a ballast. This helps to dampen high-frequency vibrations originating from the motors before they can propagate to sensitive electronics like the Pixhawk flight controller and its onboard Inertial Measurement Unit (IMU).

3.2. Avionics and Onboard Computing Hardware

The selection of avionics and onboard computing hardware is driven by the need for reliable flight control and powerful real-time data processing. The following table details the key components, their roles, and the justification for their inclusion in the ASCEND design.

Component	Hardware (Exact Name)	Primary Role & Justification
Flight Controller	Pixhawk 2.4.8	Core flight computer running the PX4/ArduPilot stack for real-time stabilization, motor control, and execution of high-level navigation commands.
Vision Processor	Raspberry Pi 4 Model B (4 GB RAM)	Onboard high-level processor dedicated to computationally intensive vision tasks, including the Visual Positioning System (VPS) and feature detection, providing navigation waypoints to the Pixhawk via MAVLink.
Vision Sensor	Raspberry Pi Camera Module v3	Provides the high-resolution image stream for the VPS, feature detection, and visual odometry algorithms.
Altitude Sensor	Benewake TF-Mini / TF-Mini-S LiDAR	Delivers direct, high-accuracy height-above-ground measurements, essential for safe navigation and precision landing by preventing vertical drift common with barometric sensors.
Telemetry Radio	3DR SiK Telemetry Radio – 915 MHz	Establishes the robust RF data link for MAVLink communication, enabling real-time mission monitoring and data relay between the UAV and base station.

The total estimated power consumption for this suite of onboard electronics is approximately **12–15 watts**, a critical metric for mission planning and battery selection.

4.0 Autonomous Systems: Navigation, Perception, and Landing

The most critical subsystems for any GPS-denied planetary exploration mission are those governing autonomous navigation and perception. The success of the ASCEND system is ultimately defined by the drone's ability to sense its environment, interpret the data it collects, and act decisively upon that information without any human guidance or external navigation aids.

4.1. GPS-Denied Navigation and Mapping Strategy

In the absence of GPS, the ASCEND vehicle determines its position within the operational area through a sophisticated process of sensor fusion. The system integrates data from multiple onboard sensors to build a coherent and continuous understanding of its local position relative to its starting point—the base station.

This fusion strategy can be summarized with the following relationship:

Local Position = IMU (Orientation) + Camera (X-Y Position via VPS/Visual Odometry) + LiDAR (Z-axis Height)

This fusion strategy leverages the strengths of each sensor to create a robust localization estimate. The **IMU** provides high-frequency orientation data to handle rapid movements, the **camera's visual odometry** corrects for low-frequency horizontal drift over distance, and the **LiDAR** provides an absolute, non-integrating measure of altitude, preventing vertical error accumulation.

This integrated approach enables a form of **Simultaneous Localization and Mapping (SLAM)**, where the drone simultaneously calculates its current position and builds a map of the surrounding environment, allowing for complex path planning and reliable return-to-base maneuvers.

4.2. Mission Execution Algorithms

The drone's complex mission logic is codified into distinct, state-based algorithms that manage different phases of the flight, from searching for targets to executing a safe and precise landing.

4.2.1. Algorithm 1: Search, Detection, and Odometry

This algorithm governs the primary exploration phase of the mission.

```
# MODEL LOADING - Loads detection & embedding models and computes seed
reference vector
START
Load YOLOv8_Model with weights "features_best.pt" in Detection mode
Load DINOv2_Model with weights "dinov2_vits_small_s4.pth" in Embedding mode
Seed_Reference_Images = { "target_feature.jpg" }
seed_embedding = MEAN( DINOv2_CLS_EMBED(DINOv2_Model, each image in
Seed_Reference_Images) )
```

```
# POSE SETUP - Sets origin coordinate and initializes path memory
pose = (0, 0, 0)
trajectory = EMPTY_LIST()
frame = Capture_Monocular_Camera()
prev_embedding = DINOv2_CLS_EMBED(DINOv2_Model, frame,)
trajectory.APPEND( pose.COPY() )

# PERCEPTION & INSTANCE VALIDATION - Detects colored patches and YOLO
objects, validates target via DINO similarity
WHILE system_active:
    frame = Capture_Monocular_Camera()
    curr_embedding = DINOv2_CLS_EMBED(DINOv2_Model, frame)
    IF COLOR_IN_ROI( inRange(Convert_HSV(frame), color_low, color_high) ) >
150:
        (Px, Py) = CENTROID( inRange(Convert_HSV(frame), color_low,
color_high) )
        Send_Navigation_Goal(Px, Py)
    ENDIF
    detections = YOLOv8_Infer(YOLOv8_Model, frame)
    FOR obj in detections:
        (x1, y1, x2, y2) = obj.bbox
        (Cx, Cy) = BBOX_CENTER(x1, y1, x2, y2)
        roi_vec = DINOv2_CLS_EMBED(DINOv2_Model, Crop_Image(frame, x1, y1,
x2, y2))
        # Cosine similarity between the two vectors of one of ROI and one
sample from basecamp
        IF COSINE_SIM(Seed_Embedding, roi_vec) > 0.80:
            Send_Navigation_Goal(Cx, Cy)
        ELSE
            ENDIF
    ENDFOR

    # ODOMETRY - Continuously matches frame embeddings to estimate motion
and update coordinates
    motion_transform = EstimateMotion(prev_embedding, curr_embedding,
RANSAC)
    IF TRANSFORM_VALID(motion_transform):
        (dx, dy, dz) = Decompose_Motion(motion_transform)
        pose = pose + (dx, dy, dz)
        trajectory.APPEND( pose.COPY() )
    ENDIF
    prev_embedding = curr_embedding
ENDWHILE

# PATH RETRACING - Follows stored coordinates back to origin using
waypoints
FOR waypoint IN REVERSED(trajectory):
    Drone.Navigate_To(waypoint)
ENDFOR
PRINT("Final Coordinates:", pose)
END
```

Plain-Language Explanation:

1. **Model Loading:** The algorithm begins by loading two pre-trained machine learning models: **YOLOv8** for general object detection and **DINOv2** for creating unique "embedding" vectors (digital fingerprints) from images. It computes a reference embedding from a seed image of the target.

2. **Perception & Instance Validation:** During flight, the drone continuously analyzes its camera feed. It first uses simple color filtering to find potential areas of interest. When it detects an object with YOLO, it crops that object from the image and uses DINOv2 to create an embedding. It then calculates the cosine similarity between the detected object's embedding and the reference seed embedding. If the similarity is above a threshold (e.g., 0.80), the target is validated.
3. **Odometry:** Simultaneously, the algorithm estimates the drone's motion. It tracks the displacement of distinct visual features between consecutive camera frames. By matching these features, it calculates a transformation matrix representing the drone's motion (dx, dy, dz), continuously updating its position relative to its origin.
4. **Path Retracing:** Upon completing the search phase, the drone accesses its stored trajectory. By reversing the sequence of waypoints, it navigates back along the path it took, ensuring a reliable return to its origin point.

4.2.2. Algorithm 2: Autonomous Precision Landing

To overcome the challenges of landing on unknown and potentially hazardous terrain, such as dust, slopes, and rocks, the ASCEND system employs a multi-phase precision landing algorithm. Each phase is designed to systematically de-risk the landing sequence, from coarse alignment down to final touchdown, ensuring the high precision required for docking with the base station.

```
# MODEL LOADING - Loads detection & sensor modules
Load LiDAR_Module in Ranging mode
Load IMU_Module in Stabilization mode
Load Vision_Module in Localization mode

# LANDING PARAMETERS - Safety thresholds
drift_threshold = 0.15      # max horizontal drift allowed (meters)
variance_threshold = 0.02   # max surface height variance allowed
touchdown_threshold = 0.10  # altitude (meters) for final touchdown trigger
battery_min_threshold = 20  # critical battery % for priority landing

# POSE SETUP - Sets origin and path memory
pose = (0, 0, 0)            # x, y, z
trajectory = EMPTY_LIST()

# SYSTEM ACTIVE LOOP - Autonomous landing pipeline
WHILE system_active:
    Frame = Vision.Module.capture()
    # Phase 1: RETURN & ALIGNMENT
    Vision_Module.navigate_to_base()
    pose.x, pose.y = measure_drift()
    IMU_Module.reduce_lateral_velocity()
    IF horizontal_drift(pose.x, pose.y) < drift_threshold:
        IMU_Module.stabilize_hover()
    ELSE:
        CONTINUE # keep correcting position
    ENDIF

    # Phase 2: SURFACE EVALUATION
    height_samples = []
    FOR i in range(10):
        h = LiDAR_Module.measure_height()
        height_samples.APPEND(h)
    ENDFOR
```

```
    surface_variance = VARIANCE(height_samples)
    IF surface_variance < variance_threshold AND
Vision.Module.is_surface_safe(frame):
        landing_zone = CENTER_OF_VALID_REGION(frame)
    ELSE:
        IMU_Module.ascend_slightly()
        CONTINUE # restart surface check
    ENDIF

    # Phase 3: CONTROLLED DESCENT
    IMU_Module.initiate_vertical_descent()
    WHILE pose.z > landing_zone.z:
        d = LiDAR_Module.measure_distance()
        IF battery_percent() < battery_min_threshold:
            IMU_Module.prioritize_immediate_landing()
        ENDIF
        IF d < touchdown_threshold:
            BREAK # move to final touchdown
        ENDIF
        IF NOT Vision.Module.is_path_clear():
            IMU_Module.abort_descent()
            IMU_Module.hover()
            BREAK # retry landing from hover
        ENDIF
        pose.z = update_altitude(pose.z, d)
    ENDWHILE
    trajectory.APPEND(pose.copy())

    # Phase 4: TOUCHDOWN & DISARMING
    IF d < touchdown_threshold AND IMU_Module.is_stable_on_ground():
        IMU_Module.reduce_final_descent_rate()
        IMU_Module.touchdown()
        IMU_Module.disarm_motors()
        BREAK # landing complete
    ENDWHALE
# FINAL STATUS - Output confirmation
PRINT("Landing Complete at Coordinates:", pose)
END
```

Plain-Language Explanation:

1. **Phase 1: Return & Alignment:** The drone navigates back to the base station's coordinates. It then actively works to minimize any horizontal drift, using its IMU and vision system to achieve a stable hover directly over the target landing zone. This addresses the cumulative drift inherent in GPS-denied navigation.
2. **Phase 2: Surface Evaluation:** Before committing to a descent, the drone takes multiple height samples using its LiDAR. It calculates the variance of these measurements to ensure the landing surface is flat and stable. This phase is a direct countermeasure to the risk of landing on hazardous, uneven terrain. If the surface is deemed unsafe, the drone will ascend slightly and re-evaluate.
3. **Phase 3: Controlled Descent:** Once the surface is confirmed safe, the drone begins a controlled, velocity-based descent. The logic includes a critical **abort** function: if the drone's tilt exceeds a safe threshold or the vision system loses confidence, it will immediately halt the descent and re-attempt the landing. The algorithm also includes priority logic to expedite landing if the battery level falls below a critical threshold.
4. **Phase 4: Touchdown & Disarm:** As the drone reaches the final touchdown altitude, it reduces its descent rate for a soft landing. The IMU detects the moment of stable

ground contact, at which point the algorithm immediately cuts power to the motors, preventing tipping and completing the landing sequence.

5.0 The Autonomous Solar-Powered Base Station

The autonomous base station is a cornerstone of the ASCEND system's design, elevating it from a single-mission platform to a persistent exploration asset. It is not merely a charging dock; it is an essential piece of infrastructure that provides the power, data management, and operational support necessary to enable mission longevity and data integrity in a resource-constrained planetary environment.

5.1. Power Generation and Management System

The base station's power architecture is designed for autonomy and resilience. It harnesses solar energy to power all its components and recharge the UAV, ensuring continuous operation. The power flows through the system as follows:

Solar Panel → MPPT Controller → Base Energy Storage (LiFePO4 Battery) → Regulated Power to MCU & Laptop

The key components of this system are selected for high performance and reliability in harsh conditions:

- **Solar Array:** Utilizes high-grade triple-junction solar cells for maximum energy conversion efficiency, even under the lower solar irradiance conditions of Mars.
- **MPPT Controller:** A high-efficiency (98%) Maximum Power Point Tracking controller ensures that the maximum possible energy is extracted from the solar panels under all conditions.
- **Energy Storage:** A thermally insulated Lithium Iron Phosphate (LiFePO4) battery stores the generated energy. This chemistry is chosen for its long cycle life and thermal stability, crucial for planetary environments.

A critical design feature is the power continuity architecture. The base station laptop is powered continuously from the primary base battery. The laptop's internal battery serves only as a short-term backup, eliminating the laptop's limited battery life as a single point of mission failure.

5.2. Charging and Docking System

A reliable charging interface is paramount. Several methods were evaluated to determine the most robust solution for a dusty, autonomous environment.

Method	Status	Rationale for Decision
Inductive Charging	Rejected	Prone to inefficiency due to dust accumulation and requires extremely precise alignment, which is challenging for autonomous systems.
Laser Power Beaming	Rejected	Presents significant safety concerns and is considered a technologically immature solution for this application.

Battery Swapping	Rejected	Involves a high degree of mechanical complexity, introducing multiple potential points of failure.
Contact-Based Charging	**Selected**	**Offers high charging efficiency, mechanical simplicity, and superior robustness and reliability in dusty environments.**

The final selected mechanism utilizes **spring-loaded, self-cleaning pogo-pin contacts**. As the drone lands, the compression of the pins creates a wiping action that clears dust and ensures a solid electrical connection.

To achieve the millimeter-level precision required for docking, the system adapts proven visual and thermal guidance principles. While these guidance systems were evaluated in the context of an inductive charging prototype, their principles are directly repurposed for our contact-based mechanism. The base station is equipped with **Nested ArUco markers** that provide a high-contrast visual target for fine-grained alignment during the final descent. An optional **IR beacon** serves as a robust fail-safe, providing a distinct heat signature that the drone can track if visual markers are obscured by dust or poor lighting, ensuring reliable docking across a range of conditions.

5.3. Communication and Data Transfer System

The system employs a **Hybrid Wireless-First Strategy** to balance flexibility with reliability. This strategy utilizes two distinct modes of communication:

- **During Flight:** A low-bandwidth **RF telemetry link (MAVLink over SiK Radio)** is used for transmitting real-time status updates, coordinates, and low-resolution preview images from the UAV to the base station.
- **During Docking:** Once the drone has landed, a high-speed wireless connection is established for the bulk synchronization of high-resolution images and mission logs. This transfer includes error checking (e.g., CRC, checksums) to ensure data integrity.

A critical fail-safe mechanism is built into this system to prevent data loss. In the event that the base station is unavailable or communication fails, all mission data, including captured images and logs, is stored securely on the drone's onboard memory. This ensures that no data is lost and can be retrieved later.

6.0 Integrated Mission Lifecycle and Safety Analysis

The true value of the ASCEND system is demonstrated by its ability to execute a complete, end-to-end mission cycle autonomously and to handle potential failures gracefully. This section integrates the preceding system descriptions to illustrate a full operational flow and highlight the built-in safety mechanisms that ensure mission robustness.

6.1. End-to-End Mission Flow

The entire operational sequence, from planning to post-flight analysis, is designed to be fully autonomous after initial setup. The lifecycle proceeds as follows:

1. **Mission Planning:** An operator loads initial mission parameters and seed images for target identification onto the base station laptop.

2. **Autonomous Flight & Search:** The drone undocks, takes off, and begins executing Algorithm 1, autonomously scanning the designated search area using its GPS-denied navigation system.
3. **Data Capture:** Upon positive identification of a target feature, the drone captures high-resolution verification images and logs the target's coordinates relative to the base station.
4. **Return & Precision Landing:** After completing its search pattern or reaching a low-battery state, the drone retraces its path and executes Algorithm 2 to perform a safe, autonomous, and precise landing on the base station's docking platform.
5. **Charging & Data Synchronization:** Once docked, the system initiates contact-based charging. Concurrently, it establishes a high-speed wireless link to transfer all captured mission data to the base station for validation and storage.
6. **Ready for Next Mission:** After the battery is fully recharged and all data is successfully synchronized, the ASCEND system enters a standby state, ready to receive commands for its next exploration sortie.

6.2. Failure Modes and Mitigation Strategies

The system's hybrid architecture and layered safety protocols are designed to anticipate and mitigate potential failures, ensuring mission continuity and protecting the hardware. The following table outlines key failure scenarios and their corresponding mitigation strategies.

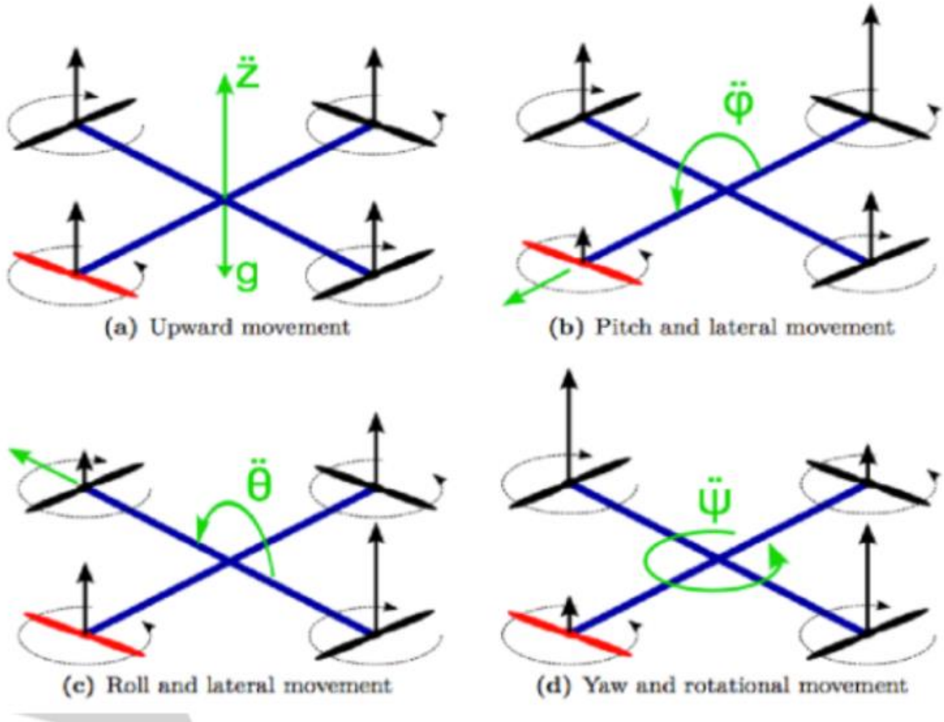
Potential Failure Scenario	Built-in Mitigation Strategy
Laptop Power Loss or Crash	The safety-critical Embedded MCU, which operates independently, continues to manage the charging process safely. It monitors battery temperature and voltage, ensuring no hazardous conditions arise. The mission is not aborted.
In-Flight Communication Loss	The drone is designed to continue its autonomous mission as planned. All captured images and telemetry data are stored on its onboard memory, preventing any data loss until communication can be re-established.
Misaligned Landing Attempt	The drone's precision landing algorithm (Algorithm 2) continuously monitors stability. If it detects excessive tilt, sudden altitude fluctuations, or an unsafe surface, it will automatically abort the landing, ascend to a safe height, and retry.
Battery Overheating During Charge	The base station's Embedded MCU constantly monitors the charging drone's battery temperature. If the temperature exceeds predefined safety thresholds, the MCU will automatically cut off the charging current to prevent damage.

7 MECHANICAL PART OF DRONE

FRAME OF DRONE – The frame should be light and rigid to host a LIPO battery, four brushless DC motors (BLDC), controller board, four propellers.

Main structure consists of a frame made of carbon composite materials to increase payload and decrease the weight.

7.1 MOVEMENT OF DRONE –

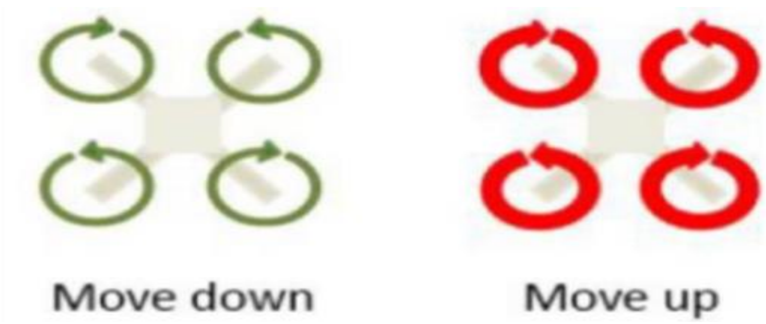


(a) Upward movement (b) Pitch and lateral movement

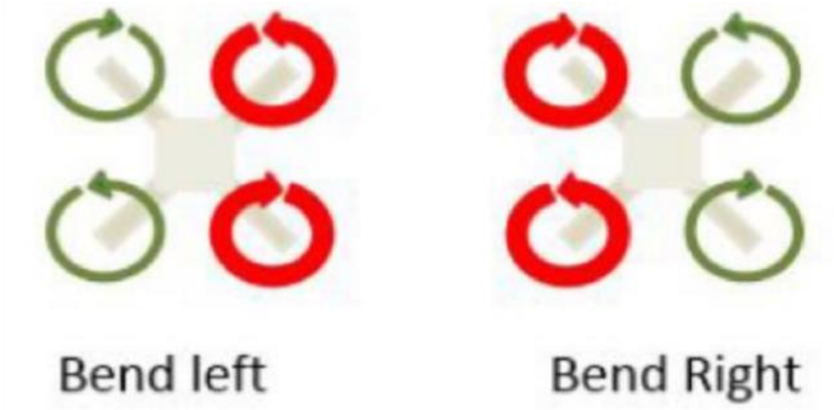
(c) Roll and lateral movement (d) Yaw and rotational movement

7.2 MECHANISM AND MOTION

1. TAKEOFF AND LANDING MOTION- Take-off is movement of quadcopter that lifts up from ground to hover position and vice versa for landing position . It is controlled by increasing or decreasing speed of four rotors simultaneously which means changing the vertical motion.



3.**Left and Right Motion**- Bend left and right motion of quadcopter is controlled by changing the roll angle ,Rotate left and right motion of quadcopter is controlled by changing the yaw angle.



Yaw angle control by increasing /decreasing counterclock rotor speed

7.3 Probable CAD Model of our drone-

8. Drone Testing Methodologies

1. Moment of Inertia (MOI) Analysis-Moment of Inertia (MOI) testing evaluates the drone's resistance to angular acceleration about a specific axis, influencing stability and maneuverability.

Formula: $I = \Sigma m r^2$

2. Axis-Wise Rotational Testing (Roll, Pitch, Yaw)-This test evaluates drone rotation about roll (X-axis), pitch (Y-axis), and yaw (Z-axis) to analyze and control effectiveness.

Relation: $I_{zz} > I_{xx} \approx I_{yy}$

3. Mass Distribution and Payload Placement Test -This testing studies the effect of payload placement on stability and rotational inertia.

Relation: $I \propto r^2$

4. Center of Mass (CoM) Determination Test-This test ensures symmetrical mass distribution for balanced thrust and stable flight.

Formula: $r_{cm} = (\Sigma r m) / (\Sigma m)$

5. Pendulum Test for Experimental MOI Measurement-In this experimental method, the drone is suspended and allowed to oscillate to measure MOI.

$T = 2\pi\sqrt{(h/g)}$

$g = 4\pi^2 h / T^2$

$I = m g D^2$

6. Error Analysis in Pendulum Testing-Error estimation ensures reliability of experimental results.

$\Delta T / T = ½ (\Delta h / h)$

7. Geometric Stability and Structural Configuration Test-This test evaluates how drone geometry affects stability.

Condition: Height < Length, Width

8.1 Quadcopter Drone Design Calculations

1. Frame Geometry and Propeller Clearance

Minimum distance between two adjacent propellers must be at least twice the propeller diameter to

avoid aerodynamic interference.

For a 10-inch propeller (25.4 cm diameter):

Motor-to-motor distance = 2 × 25.4 = 50.8 cm

2. Motor Position from Center

Motor radial distance from the center of the frame is half of the motor-to-motor distance.

Center-to-motor distance = 50.8 / 2 = 25.4 cm

4. Thrust Requirement

For stable flight, total thrust must be at least twice the total weight of the drone.

For a 2.5 kg drone:

Required total thrust = 5 kg

Required thrust per motor = 5 / 4 = 1.25 kg

5. Motor and Propeller Selection

6. Battery Endurance Calculation

Estimated flight time is calculated using usable battery capacity and average current draw.

For a 5200 Condition: Height < Length, Width

mAh battery with 20 A current draw:

Flight time = (5.2 × 0.8) / 20 = 0.208 hours ≈ 12.5 minutes

7. Geometric Stability and Structural Configuration Test

This test evaluates how drone geometry affects stability.

9 Novelty in the overall proposal

9.1 Novelty in System Design

Unlike conventional autonomous aerial platforms that rely heavily on GPS/GNSS for localization, the proposed ASCEND system employs a **sensor-driven navigation framework**, enabling reliable operation in environments where external positioning signals are unavailable or unreliable.

9.2 Novelty in Autonomy and Navigation Approach

The ASCEND system integrates:

- **Onboard sensor fusion** for state estimation using inertial and range/vision sensors
- **Real-time autonomous decision-making** without human intervention
- **Incremental autonomy validation**, ensuring stability and reliability

9.3 Novelty in Safety and Emergency Handling

A dedicated emergency response framework incorporating **Return-to-Home (RTH)** and **controlled emergency landing** ensures operational safety.

9.4 Novelty in Implementation Strategy

The system emphasizes:

- **Lightweight realization techniques** to comply with mass constraints
- **Modular hardware and software design** for ease of integration and testing
- **Cost-effective component selection** without compromising performance

10 Project management

No.	Task	Main Responsibility	Deadline for Completion	Secondary Responsibility
1	System design and architecture	Biswarup	Jan 2026	Abhay

2	Hardware procurement	Bhavishya	Jan 2026	Pranav
3	Mechanical assembly	Abhay	Feb 2026	Ayushman
4	Software and autonomy algorithms	Aditya Raj and Ayushman	Feb 2026	Bhavishya
5	System integration	Aditya Upadhyay	Mar 2026	Aditya Raj
6	Testing and validation	Pranav	Mar 2026	Biswarup
7	Documentation and reporting	Sayee	Mar 2026	Kritika

11 Conclusion

The ASCEND system represents a comprehensive and robustly engineered solution to the complex problem of autonomous planetary exploration. Through a carefully considered design, it successfully addresses the core challenges set forth by the ISRO Robotics Challenge – URSC 2026.

The system's architecture, founded on a philosophy of **hybrid control and layered redundancy**, directly mitigates the risks associated with GPS-denied navigation, autonomous operation, and long-duration missions in unstructured environments. By decoupling safety-critical functions from high-level computation, ASCEND ensures a high degree of fault tolerance and reliability. The meticulous mechanical design, advanced sensor fusion strategy, and intelligent perception algorithms combine to create a platform capable of true autonomy.

The proposed design is fully compliant with the IRoC-U 2026 rules, most notably its complete independence from GPS and its non-reliance on any pre-placed external markers for navigation or landing. Every component and algorithm has been selected and designed to contribute to a cohesive and resilient system. In conclusion, the ASCEND system is a robust, scalable, and well-reasoned solution poised for success in this challenge and, more importantly, serves as a powerful foundation for the development of future real-world planetary exploration missions.