Bibliography

[1] B. Yamansavascilar, A. Ozgovde, and C. Ersoy, “Aircompsim: A discrete event simulator for air computing,” 2025. Accessed via project documentation.

[2] B. Yamansavascilar, A. Ozgovde, and C. Ersoy, “Air computing: A survey on a new generation computation paradigm,” *Computer Networks*, vol. 251, p. 110653, 2024.

[3] B. Yamansavascilar, A. Ozgovde, and C. Ersoy, “Deepair: A multi-agent deep reinforcement learning-based scheme for an unknown user location problem,” *IEEE Access*, vol. 12, pp. 192195–192197, 2024.

[4] B. Yamansavascilar, A. Ozgovde, and C. Ersoy, “Llms are everywhere: Ubiquitous utilization of ai models through air computing,” *LLM Magazine*, 2025. Not published yet.

CMPE 492
AirCompSim - Energy-Efficient Air Computing Simulation

Huriye Ceylin Gebes
Mehmet Tuluyhan Sozen

Advisor:
Cem Ersoy
Atay Ozgovde

Contents

Chapter 1

INTRODUCTION

1.1 Broad Impact

Air computing has become increasingly relevant as demand grows for fast and flexible computational resources beyond fixed infrastructure. In particular, the use of Unmanned Aerial Vehicles (UAVs) for extending edge computing capabilities offers a dynamic solution for serving users in motion or in remote environments.

AirCompSim is an open-source simulation framework that enables experimentation with these systems. However, the current implementation does not account for energy — a critical real-world limitation for UAV operation. Without energy awareness, simulation results risk overestimating system capacity and reliability.

This project addresses that gap. Our goal is to introduce realistic energy consumption modeling into AirCompSim and to enable energy-based behavior for UAVs. These improvements will help ensure that simulations better reflect the practical constraints of UAV-assisted networks and contribute to more sustainable, accurate system design. The results can benefit scenarios such as disaster recovery, rural deployments, and energy-constrained mission planning. In the final version of our simulator, we were able to observe how energy-aware UAV behavior improved sustainability in high-load scenarios and led to more realistic performance insights, particularly for dynamic mission planning.

1.2 Ethical Considerations

Simulating UAV systems without energy limits can result in misleading expectations, especially in contexts like emergency services where overpromising performance can be harmful. Adding energy awareness into simulations helps anticipate system behavior more accurately and supports responsible planning.

There is also an environmental dimension to consider. UAV energy efficiency has implications for overall carbon footprint and sustainability. Our improvements to the simulator allow researchers and developers to explore trade-offs between performance and energy usage, enabling more informed design decisions that account for both technical and environmental concerns.

Chapter 2

PROJECT DEFINITION AND PLANNING

2.1 Project Definition

The aim of this project is to improve AirCompSim by incorporating energy consumption models and energy-aware decision logic for UAVs. The current version of AirCompSim includes features such as dynamic task offloading, mobility-aware simulation, and UAV-edge-cloud interactions, but it lacks any representation of energy usage.

Our contribution is to introduce a system in which each UAV tracks its energy levels, and makes operational decisions based on that state. The total energy cost (E_{total}) of a UAV over time will be calculated using:

$$E_{total} = E_{flight} + E_{hover} + E_{compute} + E_{communication}$$

This allows us to model how different behaviors affect battery depletion under realistic conditions.

In addition, UAVs will follow a tiered energy mode logic:

- **High Energy Mode:** UAVs operate at full capacity, actively seeking tasks, and flying to high-demand zones without major restrictions.
- **Mid Energy Mode:** UAVs behave conservatively, focusing on queued tasks while limiting unnecessary movement.
- **Low Energy Mode:** UAVs reduce mobility, avoid long distances, and stay near charging stations or safe landing spots.
- **Critical Energy Mode:** All active tasks are halted and the UAV returns directly to the nearest recharge location.

This structured behavior enables the simulator to mimic how UAVs might operate in real-world energy-constrained environments. The enhancements will provide researchers with the ability to test policies that trade off between performance and energy availability — a crucial step toward making air computing systems practical and sustainable.

2.2 Project Planning

This section outlines the project's time and resource estimates, success metrics, risk assessments, and teamwork contributions. The objective is to deliver an energy-aware extension of AirCompSim that is realistic, modular, and easily extensible for future research. We also planned further extensions such as adaptive recharge placement, task urgency-aware offloading prioritization, and forecasting UAV fatigue using energy-time tracking.

2.2.1 Project Time and Resource Estimation

The estimated timeline for the project is divided into five major phases:

Phase	**Duration**	**Tasks**
Research and Design	4 weeks	Literature review, understanding AirCompSim codebase, energy modeling design
Implementation	4 weeks	UAV energy model integration, behavior state machine, simulation logic changes
Testing and Evaluation	2 weeks	Scenario development, comparative performance analysis, debugging
Reporting and Presentation	1 week	Final report writing, visualizations, presentation preparation
Further Implementations	2 weeks	Identify additional features (e.g., recharge station logic, renewable energy modeling) in collaboration with project advisors

The final phase includes consultation with our academic advisors, where we will explore possible directions for extended functionality. These may involve implementing renewable energy-aware UAV scheduling, dynamic recharge station placement, and enhanced modularity for future research use cases.

Feedback from advisors will also help align the simulator's evolution with ongoing research in air computing and sustainable system design.

Resources required include:

- AirCompSim source code and documentation
- Computational resources for simulation (standard desktop/laptop)
- Libraries for visualization (e.g., Matplotlib, Seaborn)
- Communication tools (e.g., Git, Trello) for collaboration and version control

2.2.2 Success Criteria

The project will be considered successful if it meets the following objectives:

- Accurate integration of energy consumption modeling for UAV operations
- Implementation of energy-aware behavior modes (High, Mid, Low, Critical)
- Dynamic adjustment of UAV decision-making based on energy status
- Clear logging and visualization of energy usage trends across simulations
- Performance evaluation comparing original vs. energy-aware simulations

Additionally, the system should be modular enough to allow further development, such as support for renewable charging stations or workload forecasting.

2.2.3 Risk Analysis

Risk	**Likelihood**	**Impact**	**Mitigation Strategy**
Codebase complexity	Medium	Medium	Start with small modifications and modular design
Incomplete energy models	Medium	High	Use simplified but adjustable energy cost formulas; validate with literature
Time constraints	Medium	High	Apply agile cycles; prioritize core features first
Simulator bugs	Low	Medium	Implement incremental testing during development

2.2.4 Team Work (if applicable)

This project is a team effort involving two contributors, **Huriye Ceylin Gebes** and **Mehmet Tuluyhan Sozen**. Responsibilities were divided as follows:

- **Huriye Ceylin Gebes**: Responsible for simulator integration, algorithm implementation, system testing, and report drafting.
- **Mehmet Tuluyhan Sozen**: Handled energy modeling research, algorithm implementation, and evaluation strategy.

Team coordination was maintained via weekly meetings, Git version control, and shared task boards. Collaborative decision-making ensured consistent progress and alignment with project goals.

Chapter 3

RELATED WORK

Air computing is an emerging paradigm that aims to provide computational services using aerial platforms such as Unmanned Aerial Vehicles (UAVs), High Altitude Platforms (HAPs), and Low Earth Orbit (LEO) satellites. These systems are particularly useful in environments lacking stable infrastructure or requiring rapid deployment. Several studies have explored the architectural design, application domains, and challenges associated with air computing.

The simulator at the core of our project, *AirCompSim*, was introduced by Yamansavascilar et al. in 2025 [1]. It provides a discrete event simulation environment for evaluating task offloading strategies in air computing, considering user mobility, UAV policies, and hybrid edge-cloud deployments. However, one major limitation is that the simulator does not account for energy consumption of UAVs or edge servers. All tasks, movements, and computations are performed without considering their energy cost, which significantly limits the realism of simulation results, particularly in mission-critical or long-duration scenarios.

In a broader context, the survey by Yamansavascilar et al. (2024) outlines the foundations of air computing and highlights open challenges, including orchestration, latency optimization, and energy efficiency [2]. The authors emphasize that energy limitations, especially in UAV-based deployments, are a significant barrier to scalability and sustainability. Despite identifying this gap, the survey does not propose concrete mechanisms or simulation tools to model or manage energy-aware behavior.

Another relevant study is *DeepAir* (2024), which proposes a reinforcement learning-based multi-agent solution to task assignment under uncertainty [3]. While it offers a dynamic and adaptive approach to user-aware offloading, the model does not explicitly consider energy as part of the learning objective. Integrating energy metrics into such strategies remains an open research direction — and one our project may help facilitate by providing an energy-aware simulation base.

Finally, Yamansavascilar et al. (2025) present applied use cases in their article in LLM Magazine, focusing on how air computing can support AI-driven services in disaster recovery, rural outreach, and urban scalability scenarios [4]. Although our work does not directly simulate LLM workloads, the study reinforces the need for realistic energy modeling in aerial systems — particularly when deploying computation-intensive services from UAVs or satellite nodes.

In summary, while existing literature provides valuable insight into system design and optimization for air computing, there is a clear lack of energy-aware simulation infrastructure. Our project extends AirCompSim to address this limitation and enable new research directions in battery-constrained UAV operations, recharge-aware mobility, and sustainable offloading policies.

Chapter 4

METHODOLOGY

Our project aims to extend the existing AirCompSim simulation framework by integrating energy consumption modeling and implementing energy-aware UAV behavior. The current version of AirCompSim supports task offloading and mobility-based user scenarios but does not consider energy as a constraint. Our methodology focuses on building realistic energy tracking features and making UAV decisions dependent on remaining energy levels.

4.0.1 Energy Modeling Approach

We intend to model UAV energy consumption by dividing it into three main components:

$$E_{total} = E_{flight} + E_{hover} + E_{compute} + E_{communication}$$

- **Flight Energy** (E_{flight}): Calculated based on UAV travel distance and speed.
- **Hover Energy** (E_{hover}): Account for energy consumed while UAVs are stationary in the air.
- **Computation Energy** ($E_{compute}$): Linked to the amount of processing load a UAV handles for tasks.
- **Communication Energy** ($E_{communication}$): Linked to the amount of energy consumed during communicational operations.

Each UAV will start with a defined battery capacity, and their energy level will decrease dynamically based on activity during the simulation.

4.0.2 Energy-Aware Behavior States

We plan to implement a state-based behavior system, where UAV operations vary depending on current energy levels. The proposed states are as follows:

- **High Energy Mode**: UAVs will operate at full capacity, moving freely and actively searching for offloading opportunities.
- **Mid Energy Mode**: UAVs will prioritize task completion with moderate movement, staying relatively close to recharge zones.
- **Low Energy Mode**: UAVs will shift to a conservative strategy, minimizing motion and avoiding new task assignments unless urgent.
- **Critical Energy Mode**: UAVs will stop all operations and return to the nearest charging station or safe landing location.

This layered model is expected to help simulate realistic mission planning and endurance-limited behavior.

4.0.3 Simulator Integration Plan

To integrate energy into the simulator, we plan to:

- Modify the UAV class to include attributes for battery capacity, current energy level, and energy tracking functions.
- Update the event loop to compute energy consumption at every movement, task execution, and idle interval.
- Introduce behavior transition logic that triggers state changes based on UAV energy thresholds.
- Extend logging functionality to record per-UAV energy usage and behavior states throughout the simulation.

We also aim to design the system in a modular way to allow different energy models or thresholds to be tested in the future.

4.0.4 Planned Evaluation Strategy

Once the energy features are implemented, we plan to evaluate our enhancements using multiple simulation scenarios if the time permits. These will vary based on:

- Number of users and their mobility patterns
- Number of UAVs and their initial energy levels
- Placement and availability of charging stations
- Task intensity and distribution frequency

We intend to compare the performance of the energy-aware version with the original AirCompSim by measuring metrics such as task success rate, average energy consumption, number of completed tasks before depletion, and UAV downtime.

These evaluations are expected to demonstrate the impact of integrating energy as a constraint and provide insight into more sustainable aerial computing designs.

Chapter 5

REQUIREMENTS SPECIFICATION

This chapter outlines the functional and non-functional requirements of our proposed energy-aware extension of AirCompSim. The goal is to ensure the simulator accurately reflects energy constraints while maintaining modularity, scalability, and user interpretability.

5.1 Functional Requirements

(i) The simulator shall calculate UAV energy consumption in three categories: flight, hover, and computation.

(ii) The UAV object shall store and update its energy level throughout simulation runs.

(iii) The simulator shall classify UAVs into four energy behavior states based on current energy level:

- High Energy Mode
- Mid Energy Mode
- Low Energy Mode
- Critical Energy Mode

(iv) UAV behavior (movement and task processing) shall be restricted in each mode accordingly.

(v) The simulator shall trigger UAV return-to-charge behavior when entering Critical Energy Mode.

(vi) The simulator shall support dynamic task offloading decisions that consider both task urgency and remaining energy.

(vii) The system shall log per-UAV energy data and transitions between energy states for post-simulation analysis.

5.2 Non-Functional Requirements

- **Modularity:** Energy modeling must be integrated in a way that allows for future extension (e.g., solar recharge, energy prediction).
- **Usability:** Users should be able to configure energy thresholds, initial battery capacities, and energy cost models.
- **Performance:** Energy tracking computations should not significantly increase simulation time.
- **Maintainability:** New components must conform to existing AirCompSim architecture and follow clean code practices.
- **Extensibility:** The design should allow integration of additional aerial nodes such as HAPs and satellites in future versions.

5.3 Use Case Diagram

The use case diagram in Figure 5.1 depicts the primary interactions between the user, the simulator, and the UAV behavior modules. It highlights how the energy extension enables new simulation scenarios and analysis options.

5.4 Assumptions and Constraints

- All UAVs are assumed to start with a full battery unless configured otherwise.
- Energy consumption values are based on abstracted models and not tied to specific drone hardware.
- Charging behavior is assumed to be instantaneous or takes place off-simulation for the current implementation.
- UAVs do not currently cooperate for energy redistribution or energy-based task handoff.

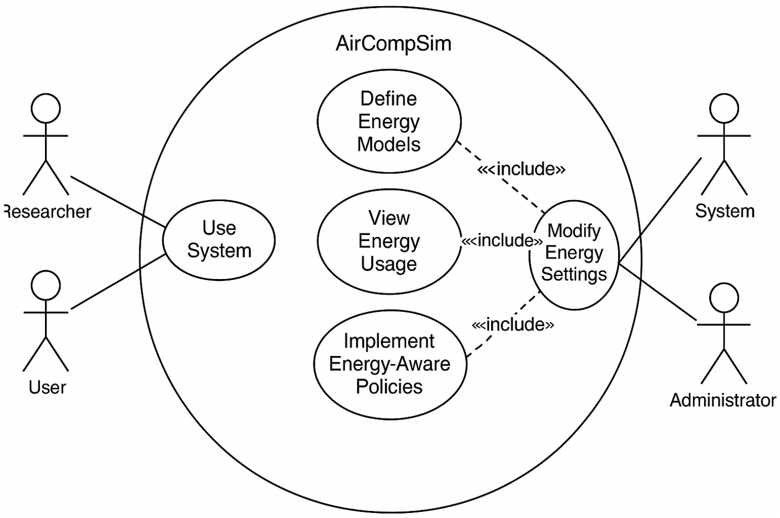

Figure 5.1: Use Case Diagram for Energy-Aware AirCompSim

Chapter 6

DESIGN

6.1 Information Structure

The information structure of our extended AirCompSim system is organized around several core entities and their relationships. The key data elements include UAVs, tasks, charging stations, and user nodes. Each of these entities maintains structured attributes that define their behavior and state throughout the simulation.

Entity-Relationship Diagram (ERD) is used to illustrate the connections between major data structures:

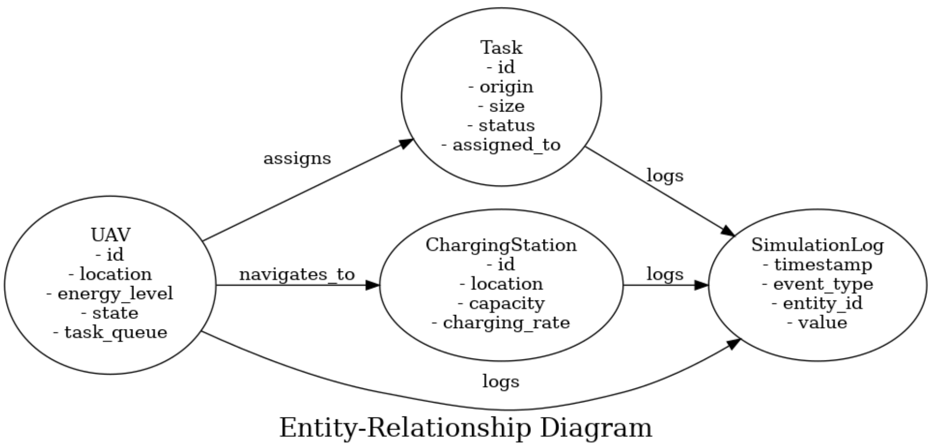

Figure 6.1: Entity-Relationship Diagram of UAV-based Energy-Aware Simulation

This structure allows for efficient simulation of real-world UAV behavior while incorporating battery limitations and mobility constraints.

6.2 Information Flow

The dynamic behavior of UAVs is modeled using a task lifecycle that spans from task creation to completion, with energy-aware state transitions. This flow is best described through both an activity diagram and a sequence diagram.

The **Activity Diagram** below describes the decision-making and operational steps of a UAV when a task is assigned. It captures evaluation of candidate UAVs, energy checks, execution of the task, and behavioral adjustments based on battery levels.

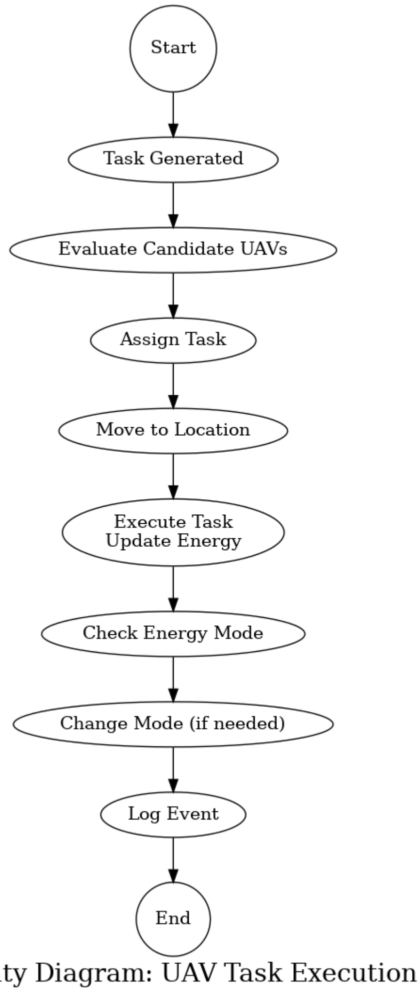

Figure 6.2: Activity Diagram: UAV Task Execution and Energy Monitoring

The **Sequence Diagram** complements the activity diagram by show-ing interactions between core modules in the simulation, including users, the task manager, UAVs, and the energy monitoring system. This diagram emphasizes how information moves between components during assignment, execution, and logging.

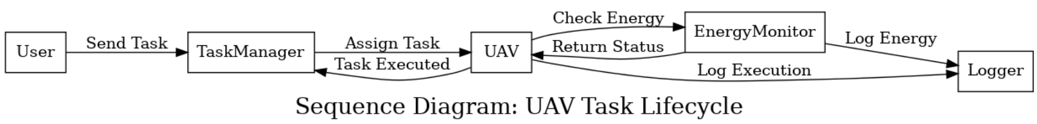

Figure 6.3: Sequence Diagram: Lifecycle of a Task in an Energy-Constrained UAV Simulation

These diagrams together provide a clear overview of the temporal and procedural aspects of simulation behavior under energy constraints.

6.3 System Design

The extended system design builds upon the modular architecture of AirCompSim by introducing energy-specific modules while maintaining compatibility with the existing framework.

Key modules include:

- `EnergyModel` – Calculates energy usage for flight, hovering, processing and communication.
- `EnergyAwareUAV` – A UAV class extension that incorporates energy levels and state-based decision logic.
- `ChargingStation` – Handles charging logic and re-entry of UAVs into the task pool.
- `EnergyLogger` – Tracks energy usage and state transitions for analysis.

These modules interact through clearly defined interfaces and can be independently tested and extended. In future iterations, additional modules such as cooperative UAV behavior and predictive recharge scheduling may be incorporated.

The class and module diagrams for this section are under development and will be finalized in the second half of the project timeline.

6.4 User Interface Design (if applicable)

Chapter 7

IMPLEMENTATION AND TESTING

7.1 Implementation

First, an energy modeling infrastructure was implemented for each UAV. This infrastructure enables the UAVs to track energy consumption from four distinct activities: flight, hovering, computation, and communication. Each UAV now contains new attributes such as:

Battery capacity and current energy level Energy mode (High, Mid, Low, Critical) Behavioral transitions based on energy levels To support this, the UAV class was refactored. UAVs now make autonomous decisions based on their battery status. For instance, when the energy drops below a threshold, the UAV switches to "Low Energy Mode" and navigates to the nearest charging station before accepting new tasks. This behavior is controlled by a finite state machine logic with clearly defined transitions.

Additionally, the event loop of the simulator was modified to include real-time energy consumption updates. Each movement, computation, or idle time during simulation deducts the appropriate amount of energy from the UAV's battery, depending on the activity.

Lastly, an EnergyLogger class was developed to continuously log battery levels, energy transitions, and charging events, enabling both analysis and debugging.

7.2 Testing

Several types of testing were conducted to validate the new energy-aware components.

- Behavioral Tests: Verified that UAVs correctly transitioned between energy modes. For example, a UAV with less than %15 battery should enter Low Energy Mode and head to a charging station.
- Scenario-Based Simulations: Ran simulations with varying user densities, UAV initial energy levels, and task intensities. Comparisons between energy-aware and non-energy-aware modes demonstrated expected behavioral differences.
- Observability Tests: Analyzed logs and plots generated at runtime to confirm that energy transitions and UAV movements aligned with the modeled behavior.

These tests collectively ensured the robustness and accuracy of the energy-awareness logic within the simulator.

7.3 Deployment

To support future use and experimentation, we structured the deployment process as follows: A detailed README file was created, documenting how to run simulations with energy settings enabled and how to interpret results. A curated requirements.txt file was generated to simplify environment setup using Python virtual environments.

Chapter 8

Results

To assess the impact of the introduced energy-aware UAV behavior in AirCompSim, multiple simulation scenarios were executed comparing baseline (non-energy-aware) and energy-integrated configurations under varying user loads and UAV counts.

8.1 Task Offloading Distribution

The energy-aware UAV model significantly altered the task distribution among edge servers, UAVs, and the cloud. In energy-unaware scenarios, UAVs continued task execution regardless of battery levels, often leading to unrealistic performance estimations. With energy-awareness, UAVs began to reject tasks or return to charging stations when battery levels reached critical thresholds, which increased cloud offloading rates and exposed infrastructure limitations more realistically.

8.2 Task Success Rate

Overall task success rates showed a minor decrease in the energy-aware simulations due to UAV task rejections in low or critical battery modes. However, this decrease is a direct consequence of simulating physical constraints and thus represents a more reliable performance metric for system design. Notably, the introduction of UAV waiting policies and conservative behavior in Mid/Low energy modes helped mitigate QoS degradation.

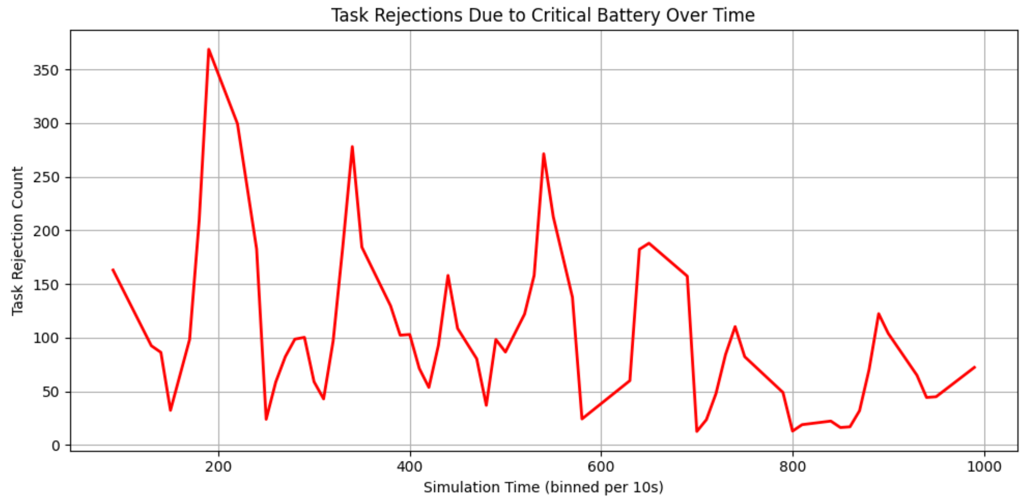

Figure 8.1: Task Rejections Over Time

8.3 Energy Mode Distribution

UAVs dynamically transitioned between High, Mid, Low, and Critical modes based on energy consumption profiles. Analysis showed that frequent transitions to Mid and Low energy modes prompted UAVs to limit mobility and hover time, extending mission longevity. Critical mode was observed predominantly in high-load scenarios, triggering return-to-charge behavior and avoiding unrealistic task completions.

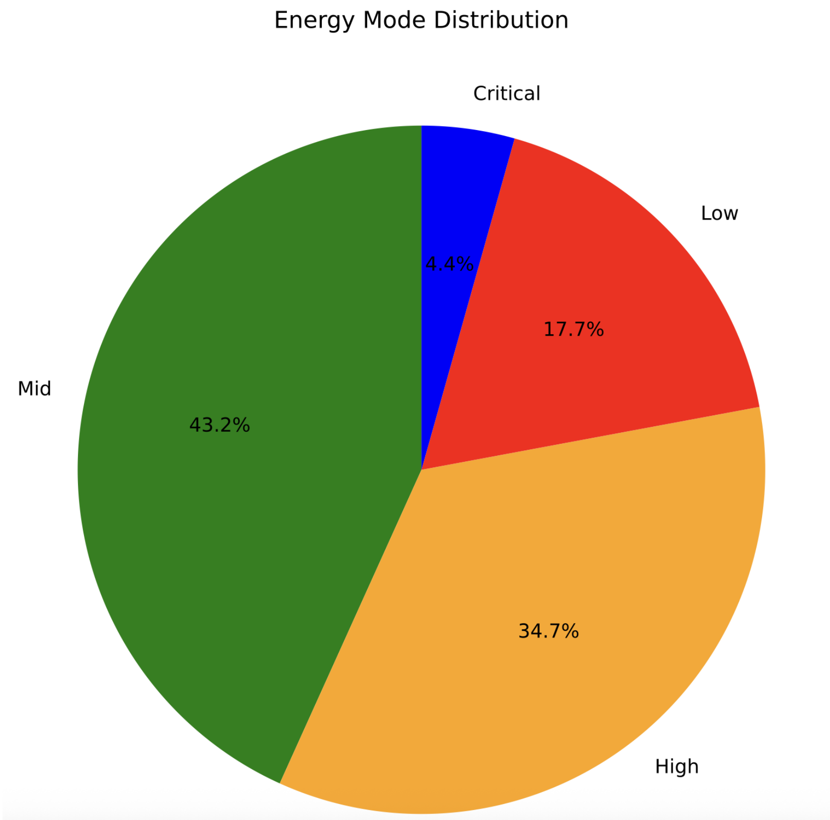

Figure 8.2: Energy Mode Distribution

8.4 UAV Utilization and Energy Consumption

With the integration of the energy model, UAV utilization profiles became more stable and correlated with realistic endurance limits. Total UAV energy consumption increased proportionally with user density and task complexity. The most energy-intensive component was flight energy, followed by computation and hover energy.

8.5 Visualization & Logging

Detailed simulation logs and result plots (e.g., task offloading ratios, average battery levels, energy-mode transitions) facilitated insightful comparisons. Visualizations validated that energy-aware UAVs avoided overcommitment and yielded more sustainable system behavior in dynamic environments.

8.6 Scenario-Based Evaluations

To rigorously evaluate the performance of our energy-aware UAV framework, we modeled three realistic scenarios inspired by real-world events. Each setup emphasizes different infrastructure constraints, UAV behavior, and application profiles.

8.6.1 AlpineHearts – Mountain Event

This scenario simulates a mountain expedition setting where 100 mobile users follow a Random Waypoint model in a terrain with sparse edge coverage.

- **Infrastructure**: UAV-only compute layer (no nearby edge/cloud fallback)
- **UAV Setup**: 10 UAVs, altitude-limited due to terrain
- **Applications**: *Image Classification*, *Rendering*
- **Objective**: Evaluate task success under short-range compute and limited energy capacity

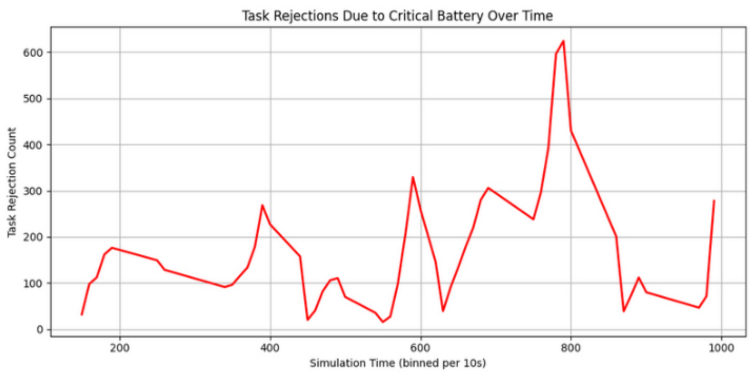

Figure 8.3: Task Rejections Over Time in AlpineHearts Scenario

The results show that as UAVs approach critical energy levels, task rejections increase. The energy-aware model successfully triggers UAV returns before full depletion, avoiding overcommitment.

8.6.2 BurningCloud – Urban Festival

This scenario reflects a crowded urban event (e.g., concert or street festival) with over 400+ users exhibiting low mobility (Nomadic behavior).

- **Infrastructure**: Edge servers exist but are **overloaded**
- **UAV Setup**: 10 UAVs dynamically rerouted based on server queue and battery level
- **Applications**: *Entertainment*, *Multimedia*
- **Objective**: Assess hybrid offloading and UAV coordination under heavy load

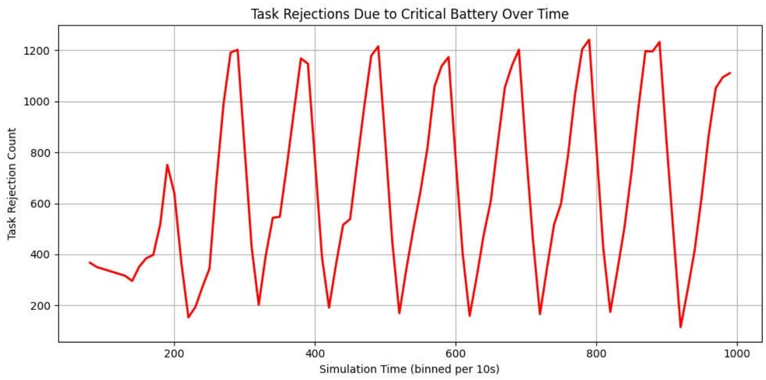

Figure 8.4: Task Rejections Over Time in BurningCloud Scenario

Despite available infrastructure, the overloaded edge layer increases reliance on UAVs. The task rejection pattern reflects UAV cycling between compute and charge behavior in real time.

8.6.3 İzmir Earthquake – Disaster Recovery

This emergency scenario simulates a post-earthquake environment with 200 mobile users scattered across multiple danger zones. All ground infrastructure is assumed unavailable.

- **Infrastructure**: UAVs as the **primary compute layer**
- **UAV Setup**: Equipped with **visual detection** (camera-enabled) for locating users
- **Applications**: *Real-Time Messaging*, *Image Classification*
- **Objective**: Analyze system behavior under urgency, mobility, and energy stress

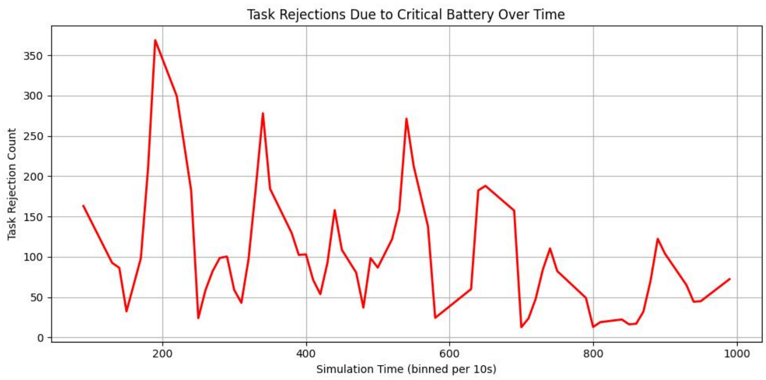

Figure 8.5: Task Rejections Over Time in İzmir Earthquake Scenario

The UAVs exhibit frequent transitions to Low and Critical modes due to extended hover/search behavior. The system's realism is validated by the repeating pattern of charging-induced UAV unavailability.

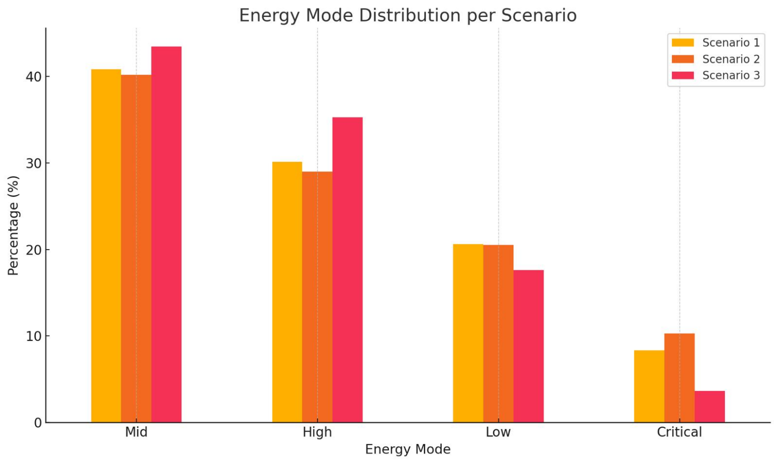

Figure 8.6: Energy Mode Distribution Comparison

An important factor contributing to this distribution is the number of UAVs used in each scenario. Scenario 3 utilizes 20 UAVs, whereas Scenarios 1 and 2 operate with only 10 UAVs. The increased UAV count in Scenario 3 enables better load distribution and more conservative energy usage per UAV, reducing the frequency of critical energy levels. This clearly highlights the scalability and robustness benefits of a denser UAV network.

Chapter 9

Conclusion

In this project, we enhanced the AirCompSim simulation framework by integrating a realistic energy consumption model and implementing energy-aware behavior policies for UAVs. This modification enabled simulations to more accurately reflect the operational constraints of UAV-assisted edge computing systems.

Through a series of comparative experiments, we observed that the introduction of energy-aware decision-making notably affected task distribution, UAV utilization, and overall system behavior. UAVs were able to autonomously manage their energy by adapting mobility and processing decisions based on their current battery status, improving the realism and sustainability of simulation results.

The energy mode transitions and task rejection mechanisms ensured that UAVs avoided overcommitting under low energy conditions, a behavior crucial for mission-critical applications such as disaster recovery and rural deployment. Although task success rates slightly declined due to energy-related task rejections, this trade-off was necessary to model physically feasible operations.

Our improvements provide a foundation for future research on energy-efficient air computing. They also pave the way for advanced features such as renewable energy integration, predictive task scheduling, and reinforcement learning-based energy optimization. Overall, this work contributes toward making air computing simulations more representative, modular, and applicable to real-world use cases.

Beyond the technical contributions, this project also allowed us to explore a relatively new and evolving area within networking. We are especially glad to have gained experience in reviewing and understanding up-to-date academic resources in this domain. Learning to follow cutting-edge research and applying it to a practical simulation setting was a valuable experience that broadened our perspective on research in next-generation computing systems.

Bibliography

[1] B. Yamansavascilar, A. Ozgovde, and C. Ersoy, "Aircompsim: A discrete event simulator for air computing," 2025. Accessed via project documentation.

[2] B. Yamansavascilar, A. Ozgovde, and C. Ersoy, "Air computing: A survey on a new generation computation paradigm," *Computer Networks*, vol. 251, p. 110653, 2024.

[3] B. Yamansavascilar, A. Ozgovde, and C. Ersoy, "Deepair: A multi-agent deep reinforcement learning-based scheme for an unknown user location problem," *IEEE Access*, vol. 12, pp. 192195–192197, 2024.

[4] B. Yamansavascilar, A. Ozgovde, and C. Ersoy, "Llms are everywhere: Ubiquitous utilization of ai models through air computing," *LLM Magazine*, 2025. Not published yet.

Appendix A

SAMPLE APPENDIX

Contents of the appendix.