Problem 1:Autonomous Passenger Drone: Cabin Design

This document outlines the design of a lightweight, safe, and comfortable passenger cabin for an autonomous drone. It prioritizes safety, comfort, and optimized weight-to-strength ratio.

by ASHISH KUMAR

Introduction to Project

1. **Concept**

   Develop a safe and comfortable passenger cabin for an autonomous drone.

2. **Innovation**

   Focus on lightweight materials and innovative design for optimal performance.

3. **Passenger Experience**

   Prioritize passenger comfort and a positive user experience.

Design Objectives

Safety

Meet or exceed all safety standards.

Lightweight

Minimize weight while maintaining structural integrity.

Comfort

Provide a comfortable and enjoyable passenger experience.

Passenger Safety Considerations

Emergency Systems

Integrated parachute system for controlled descent.

Impact Protection

Energy-absorbing materials strategically placed within the cabin.

Redundancy

Backup systems for critical flight components.

Egress

Easy and quick exit strategy in case of emergencies.

Cabin Structural Design

1

Frame

Lightweight carbon fiber composite structure.

2

Reinforcements

Strategic placement of reinforcing members.

3

Stress Tests

Rigorous testing under various flight conditions.

Cabin Aerodynamics and Weight Optimization

1

Aerodynamic Design

Minimize drag for efficient flight.

2

Material Selection

Lightweight materials like carbon fiber composites.

3

Optimization Analysis

Computational fluid dynamics for optimal shape.

Cabin Interior Layout and Ergonomics

Seating

Ergonomic design for maximum comfort.

Controls

Simple and intuitive controls.

Passenger Comfort Features

Climate Control

Individualized temperature settings.

Entertainment

Integrated infotainment system.

Noise Reduction

Active noise cancellation for a quiet ride.

Cabin Materials and Manufacturing

Material	Carbon Fiber Composite
Manufacturing	Advanced composite manufacturing techniques
Benefits	Lightweight, high strength, durability

Regulatory Compliance and Certification

1 **Compliance**

Adherence to all relevant safety regulations.

2 **Certification**

Obtaining necessary certifications before operation.

3 **Testing**

Rigorous testing to meet regulatory standards.

Key Design Features

Ergonomic Seat Design

Adjustable seating designed for comfort during flight, with lumbar support and shock absorption. Made from high-density foam wrapped in synthetic leather for durability.

Lightweight Frame

The cabin frame uses advanced composites, ensuring optimal weight reduction without sacrificing strength. The frame can endure extreme pressure and vibrations during flight.

Advanced Entry & Exit Mechanism

Side-hinged doors allow easy access while ensuring secure closure. The doors include emergency release mechanisms accessible from inside and outside the cabin.

Integrated Digital Display Panel

Onboard display for navigation updates, flight duration, safety instructions, and environmental controls. Passengers can monitor real-time information and adjust the climate settings.

Safety & Comfort Features

Passenger Restraint System

The cabin includes a three-point harness designed to securely restrain the passenger. An ergonomic seat layout ensures minimized strain and maximized safety during flight maneuvers.

Emergency Exit System

Doors are equipped with an emergency release system. In case of an emergency, the passenger can activate an easily accessible lever to unlock and exit the cabin quickly.

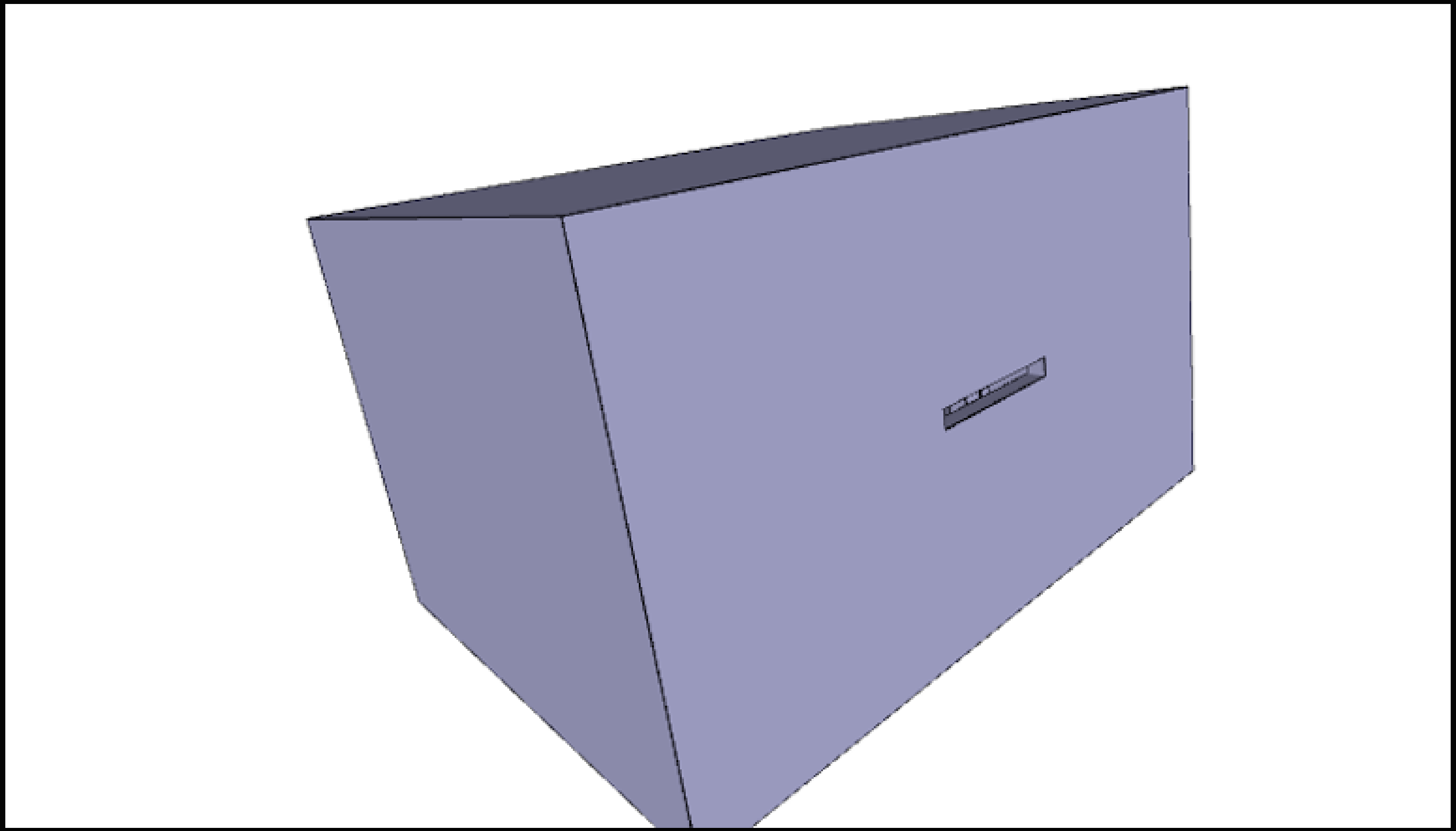
Noise-Reduction Insulation

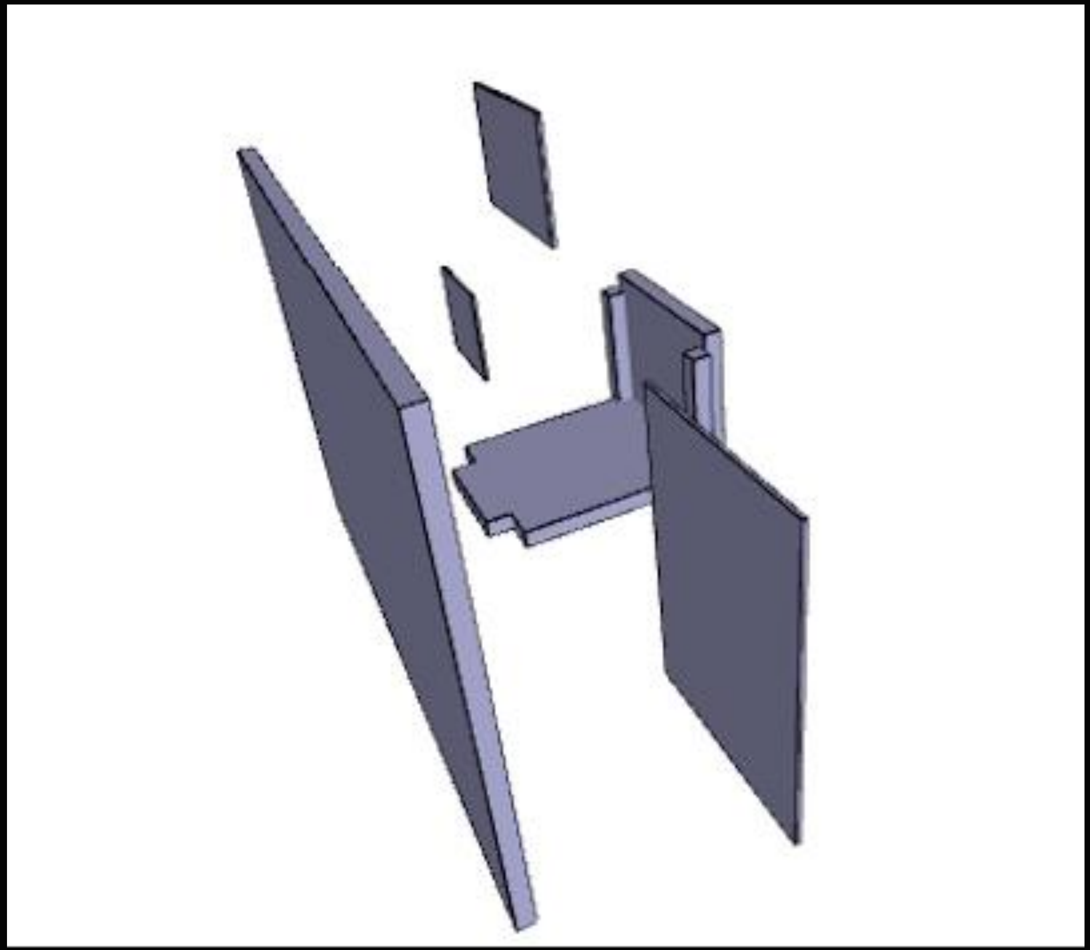
The cabin walls include a layer of noise-reduction material, reducing engine and wind noise to ensure a quieter, more comfortable passenger experience.

Energy Efficiency & Sustainability

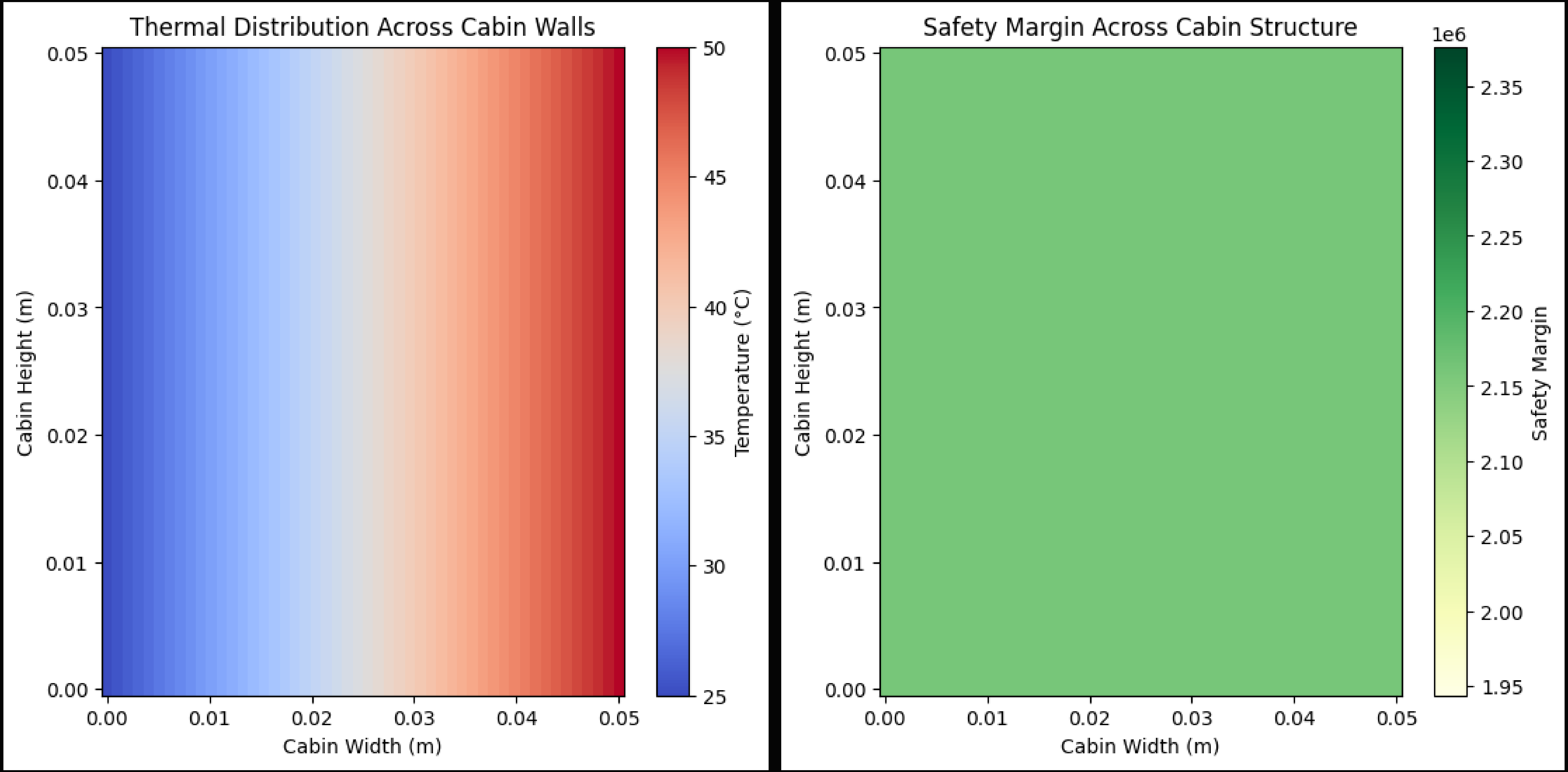
Environmental considerations are integrated into the cabin design. By using recycled materials, energy-efficient manufacturing processes, and low-weight components, the design minimizes environmental impact.

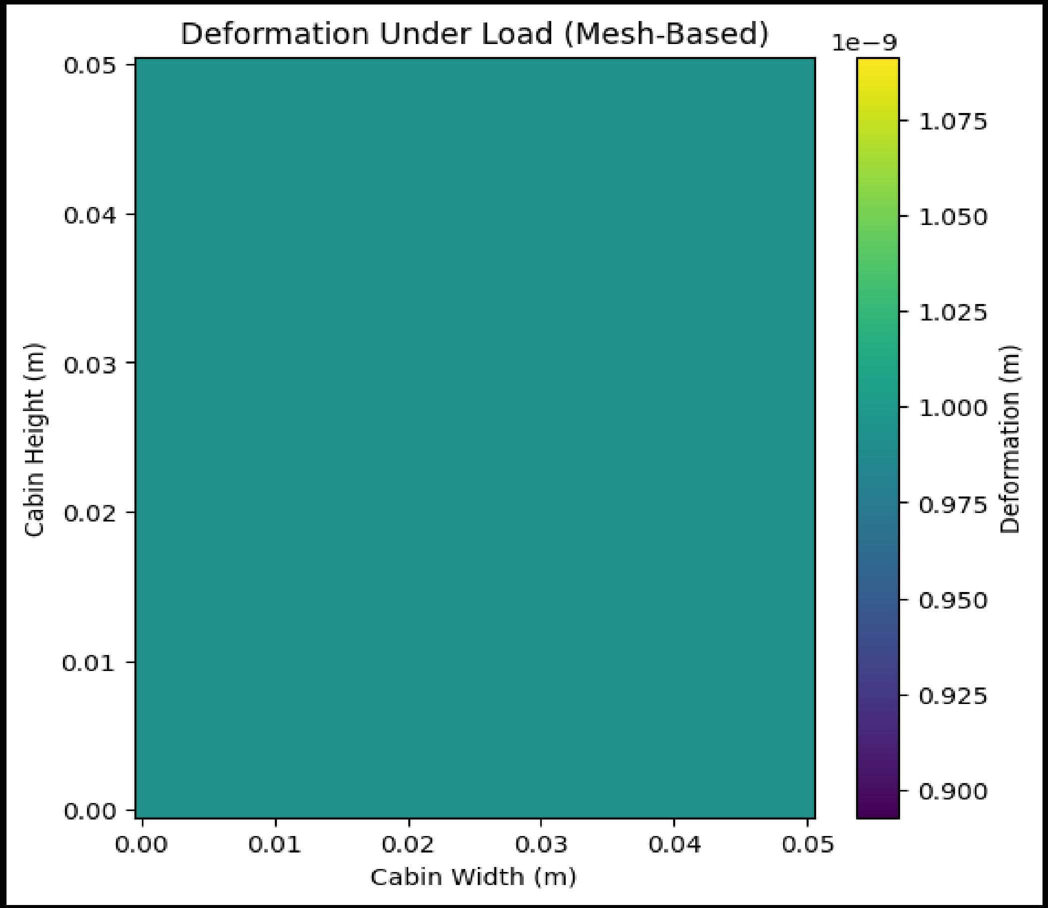
- **Recyclable Materials:** Aluminum and polymer parts are fully recyclable.
- **Efficient Aerodynamics:** The cabin shape minimizes drag, optimizing fuel efficiency and reducing emissions.
- **Eco-Friendly Coatings:** Non-toxic, eco-friendly coatings are used on all interior surfaces.

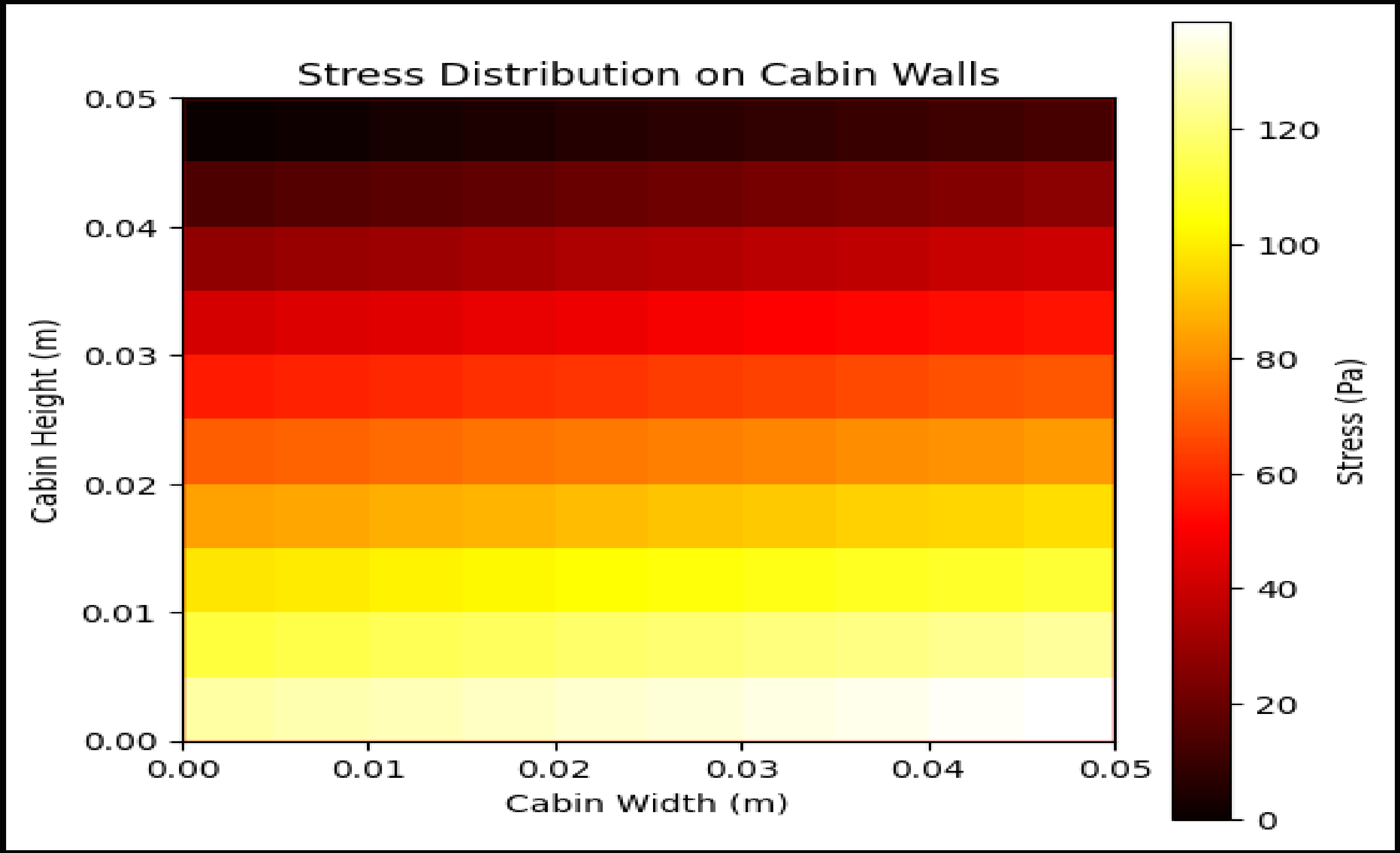




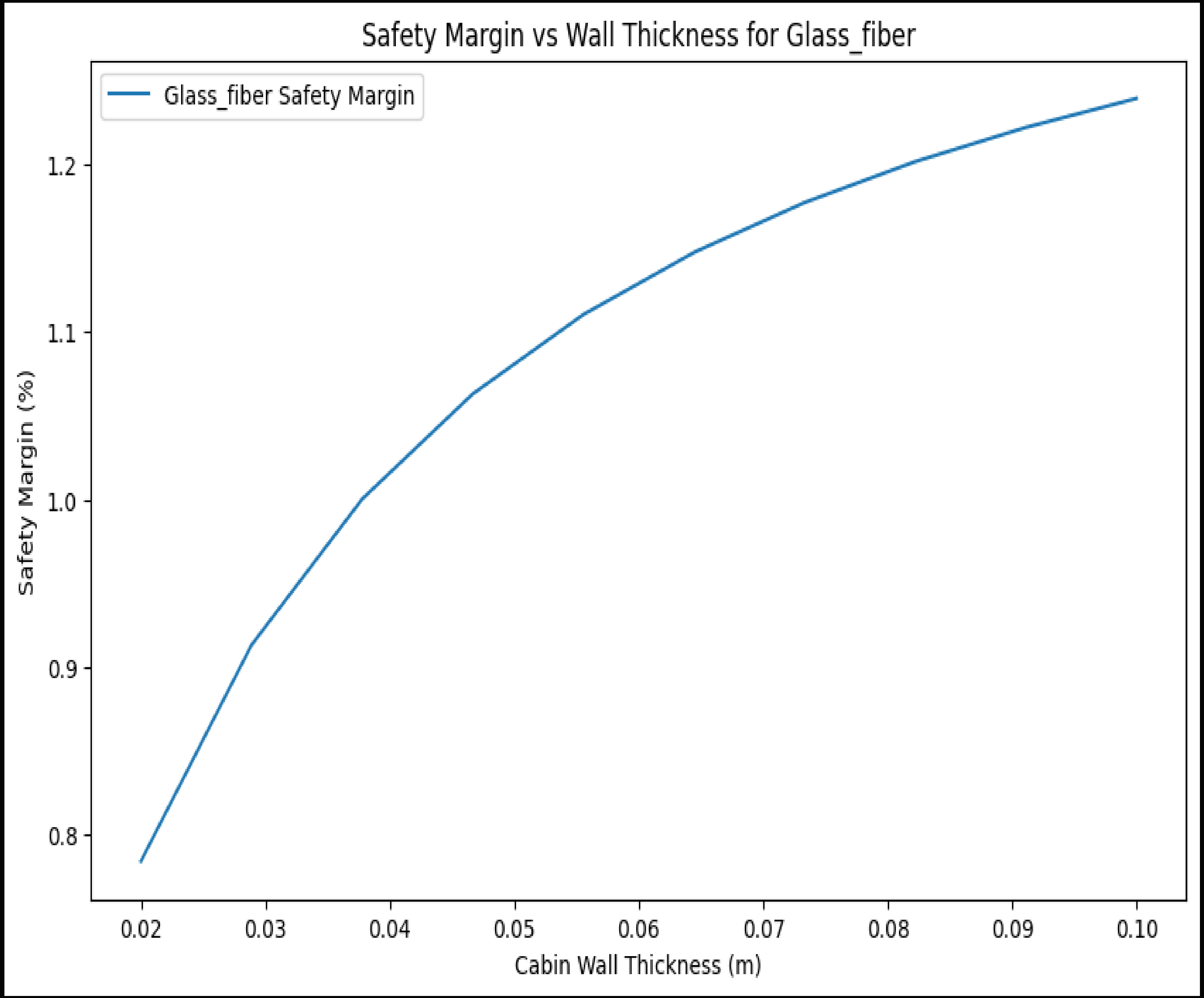
```
Material Properties:
{'Density': '1200 kg/m^3', 'Tensile Strength': '300 MPa', 'Thermal Resistance': '0.5 K/W', 'Elasticity Modulus': '7 GPa'}

Cabin Weight: 4320.00 kg

Stress Analysis Results:
Stress (Pa): 138.88888888888889
Strain: 1.984126984126984e-08
Deformation (m): 9.920634920634920e-10
Safety Margin: 2160000.0
Passes Safety Check: True

Thermal Analysis Results:
Heat Transfer Rate (W): 50.0
Thermally Comfortable: True

Safety Factor: 2160000.00
```

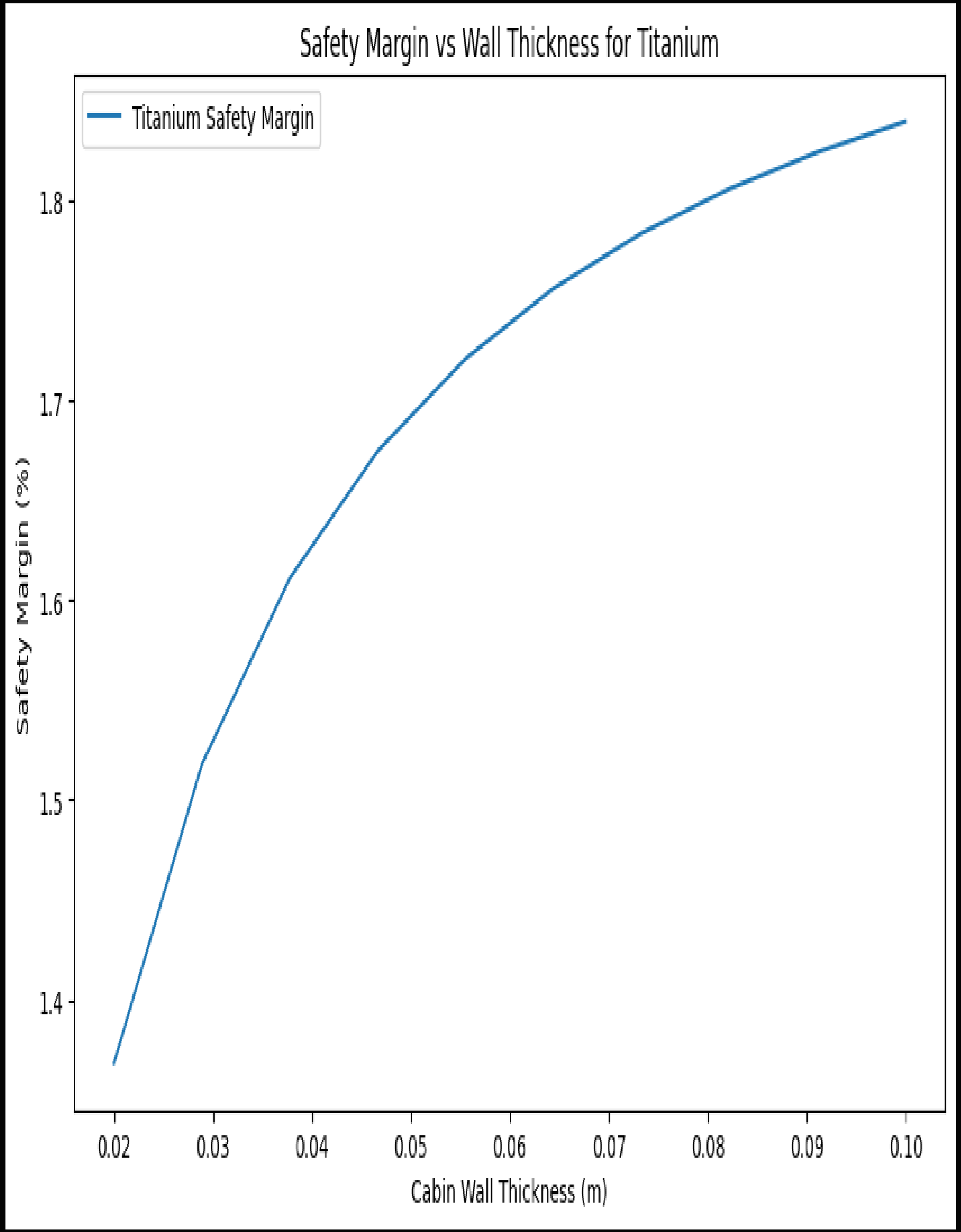

Thermal Distribution Across Cabin Walls
Cabin Height (m)
0.00
0.01
0.02
0.03
0.04
0.05
Cabin Width (m)
0.00
0.01
0.02
0.03
0.04
0.05
Temperature (°C)
25
30
35
40
45
50
Safety Margin Across Cabin Structure
Cabin Height (m)
0.00
0.01
0.02
0.03
0.04
0.05
Cabin Width (m)
0.00
0.01
0.02
0.03
0.04
0.05
1e6
Safety Margin
1.95
2.00
2.05
2.10
2.15
2.20
2.25
2.30
2.35

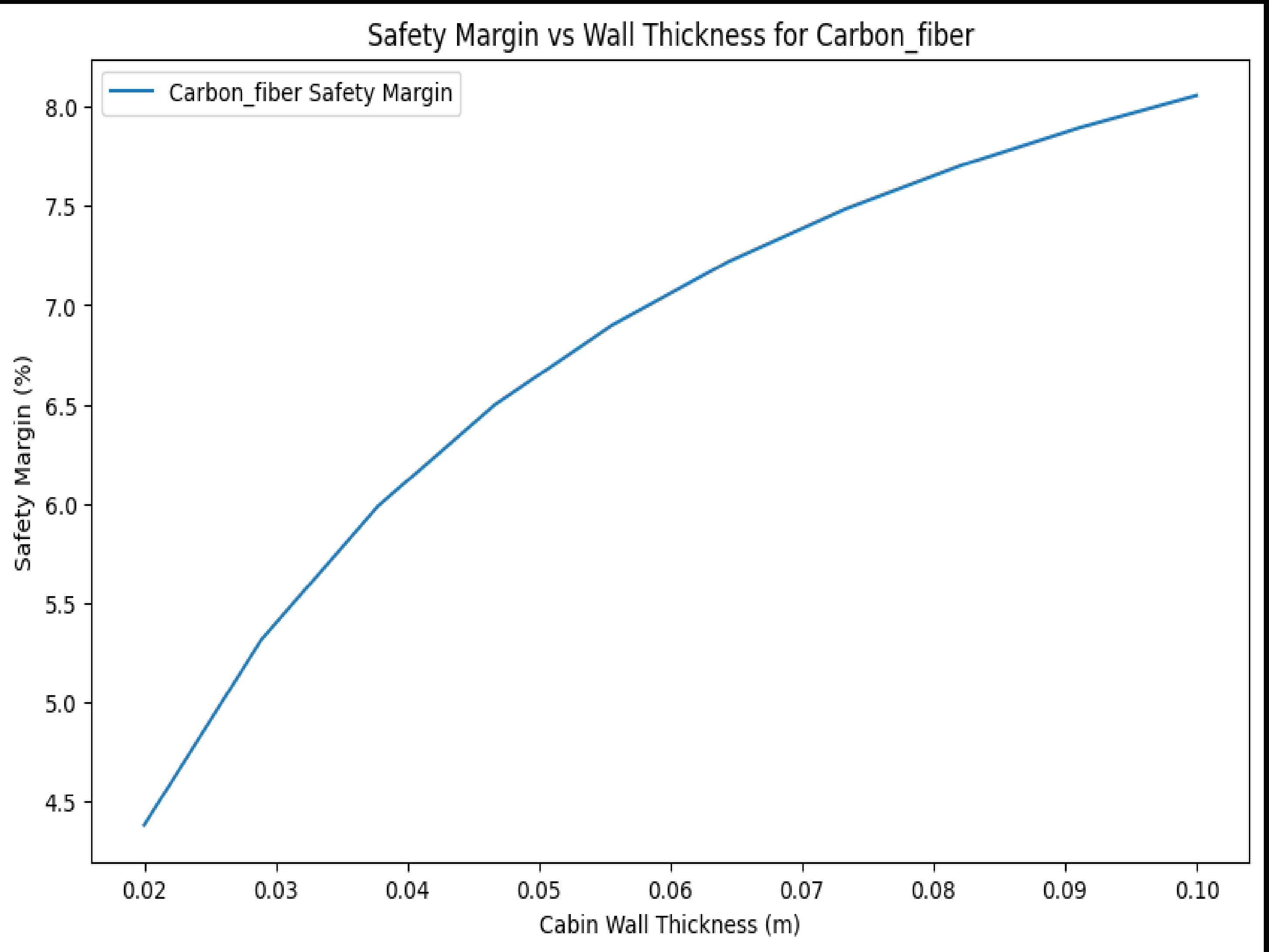


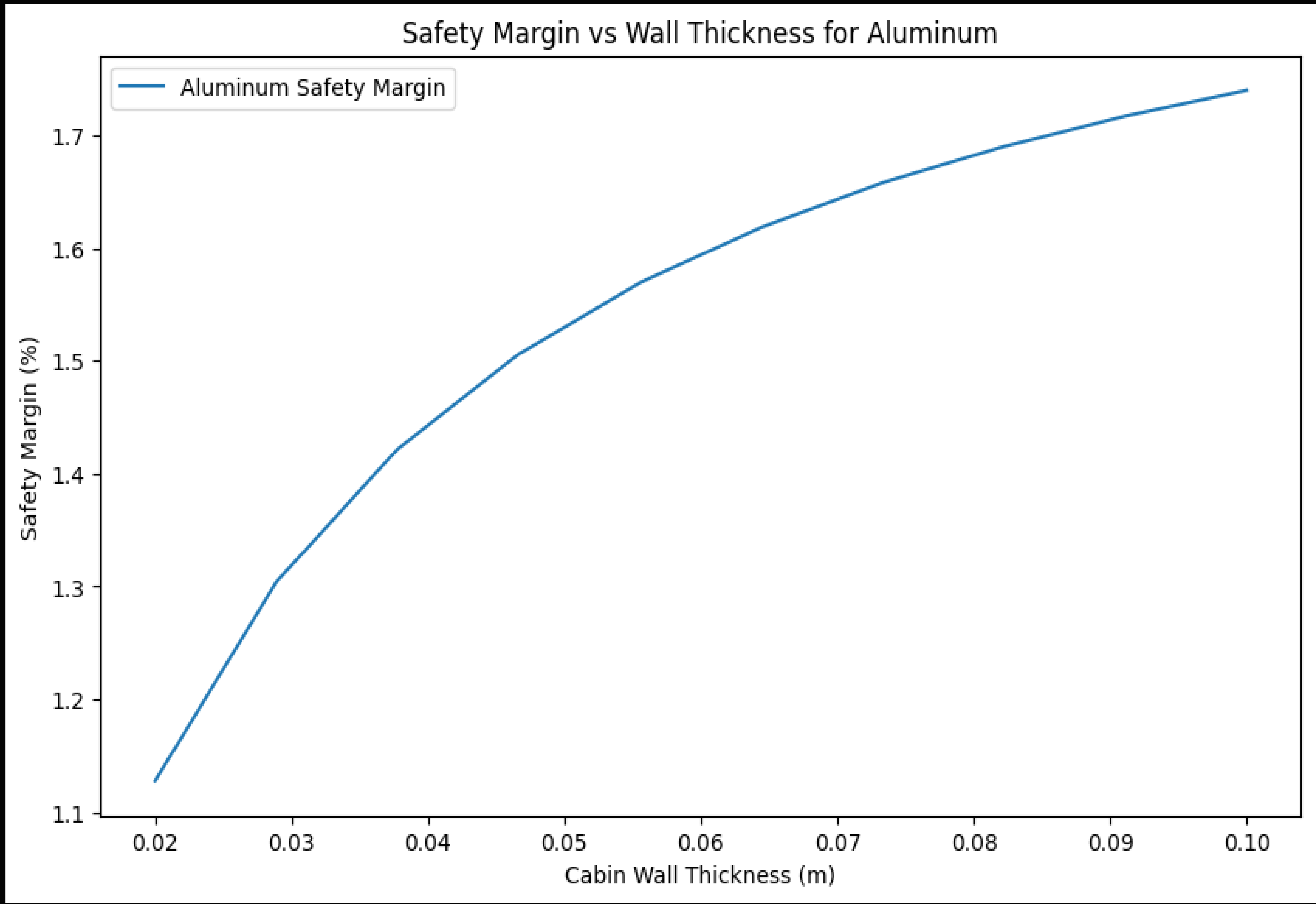



```
Cabin Volume: 0.67 cubic meters
Cabin Weight: 808.80 kg
Weight-to-Strength Ratio: 2.70
Safety Check: Safe under nominal load

Ergonomic Layout (Initial): {'Seat Type': 'reclining', 'Legroom': 0.9, 'Headroom': 1.0}

Safety Features (Initial): {'Airbags': True, 'Seat Belts': True, 'Emergency Exit': True, 'Padding Material': 'foam'}
```


Safety Margin vs Wall Thickness for Glass_fiber
Glass_fiber Safety Margin
Safety Margin (%)
1.2
1.1
1.0
0.9
0.8
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
Cabin Wall Thickness (m)


Safety Margin vs Wall Thickness for Titanium
Titanium Safety Margin
Safety Margin (%)
1.8
1.7
1.6
1.5
1.4
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
Cabin Wall Thickness (m)






```
=== Comfort and Environmental Controls ===
Air Change Time: 4.00 seconds
Temperature Control Range: 21.80°C - 22.20°C
```

Cabin Structural and Safety Report

Material Properties

Density: 1200 kg/m^3

Tensile Strength: 300 MPa

Thermal Resistance: 0.5 K/W

Elasticity Modulus: 7 GPa

Stress and Deformation Analysis

Stress (Pa): 138.88888888888889

Strain: 1.984126984126984e-08

Deformation (m): 9.92063492063492e-10

Safety Margin: 2160000.0

Passes Safety Check: True

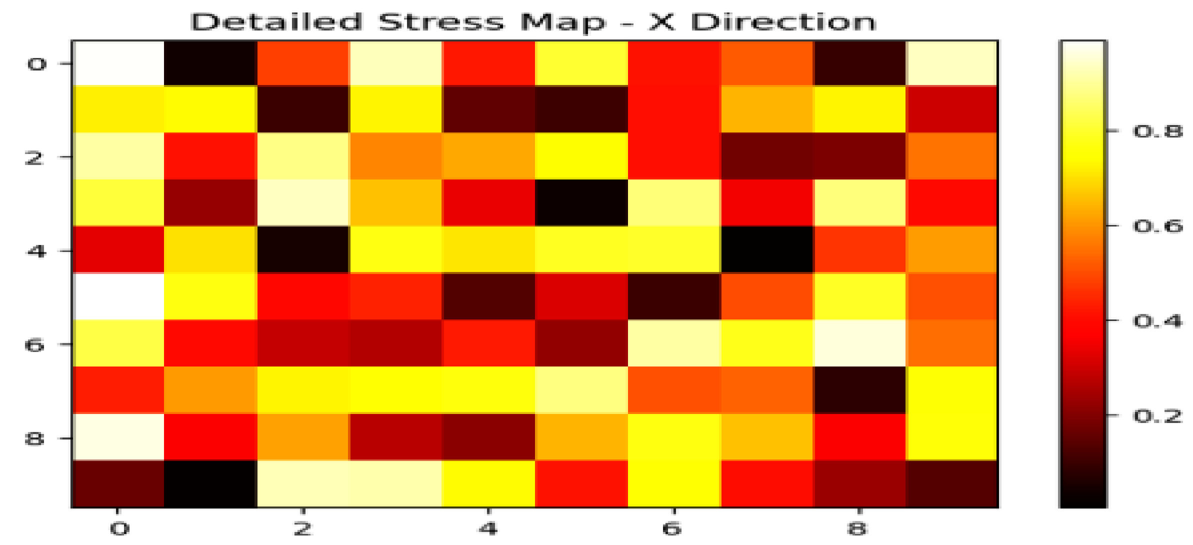
Safety Factor: 2160000.00

Thermal Analysis

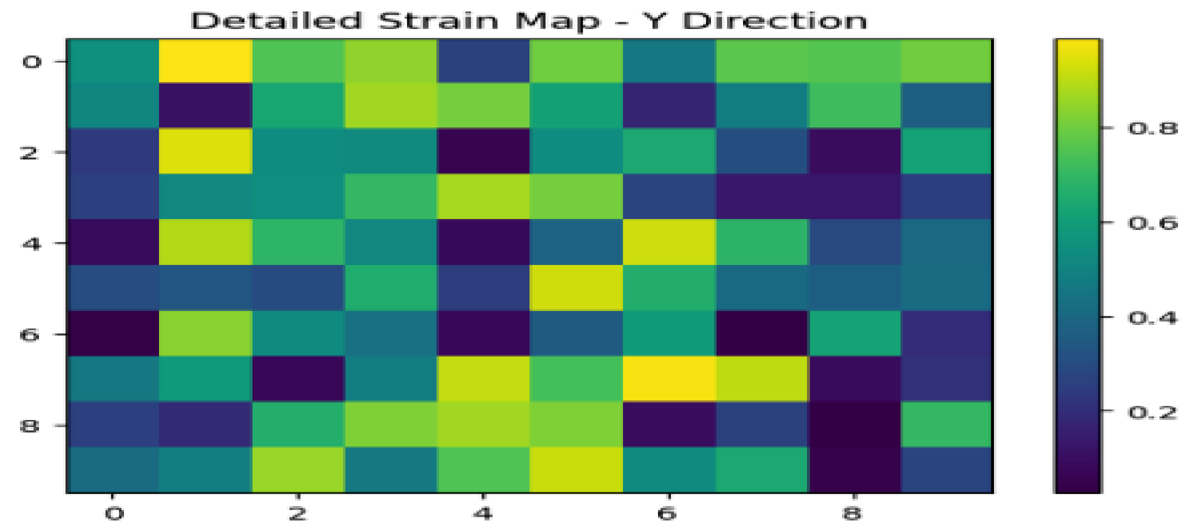
Heat Transfer Rate (W): 50.0

Thermally Comfortable: True

Safety Margin Visualization



Detailed Stress Map - X Direction



Detailed Strain Map - Y Direction

Problem 2:Custom Firmware for Drone Quadcopter

Team LUMEN presents innovative custom firmware for drone quadcopters.

Our solution enables advanced flight modes, enhancing stability and control.

This presentation outlines our approach, development process, and results.

ASHISH KUMAR

Introduction to Team LUMEN

Innovation

Pushing boundaries in drone technology with cutting-edge solutions.

Expertise

Team of skilled engineers specializing in robotics and firmware development.

Collaboration

Fostering teamwork to tackle complex challenges in drone engineering.

Understanding Drone Flight Modes

Stabilize Mode

Maintains level flight, allowing manual control of altitude and direction.

Position Hold

Keeps the drone in a fixed position, resisting wind and external forces.

Altitude Hold

Maintains a constant height while allowing horizontal movement.

Stabilize Mode

1. **Gyroscope Integration**

   Utilizes gyroscopic data to detect and correct angular deviations.

2. **PID Control**

   Implements PID algorithms for smooth and responsive attitude corrections.

3. **User Input Mapping**

   Translates user controls into precise motor adjustments for intuitive flight.

Position Hold Mode

1

GPS Acquisition

Obtains accurate GPS coordinates to establish the drone's initial position.

2

Drift Compensation

Continuously adjusts thrust to counteract wind and maintain the fixed position.

3

Obstacle Avoidance

Integrates sensor data to prevent collisions while holding position.

Altitude Hold Mode

Barometric Sensing

Uses barometric pressure sensors to accurately measure and maintain altitude.

Thrust Modulation

Dynamically adjusts motor output to compensate for air density changes.

Vertical Stability

Implements advanced algorithms to minimize vertical oscillations during flight.

Firmware Development Approach

1

Requirements Analysis

Defined technical specifications and performance goals for each flight mode.

2

Algorithm Design

Developed robust control algorithms tailored for quadcopter dynamics.

3

Implementation

Coded firmware in C++ for optimal performance and resource efficiency.

4

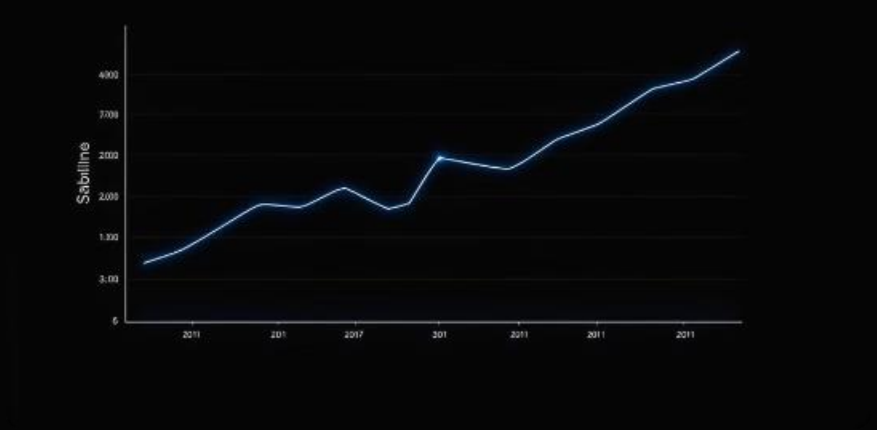
Testing & Optimization

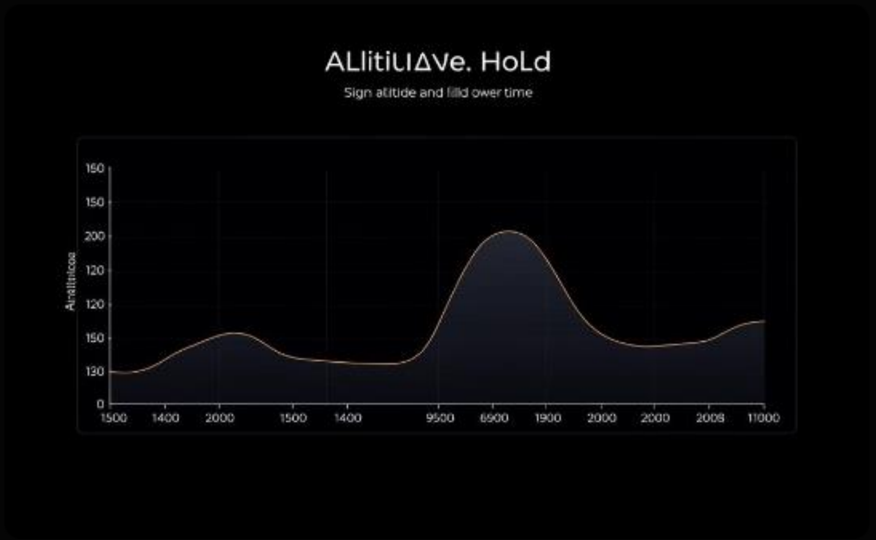
Rigorous testing in simulated environments, followed by real-world flight tests.

C++ vs Python: Tradeoffs

Aspect	C++	Python
Performance	Higher	Lower
Development Speed	Slower	Faster
Memory Management	Manual	Automatic
Real-time Capabilities	Better	Limited

Simulation Results and





Stabilize Mode Performance

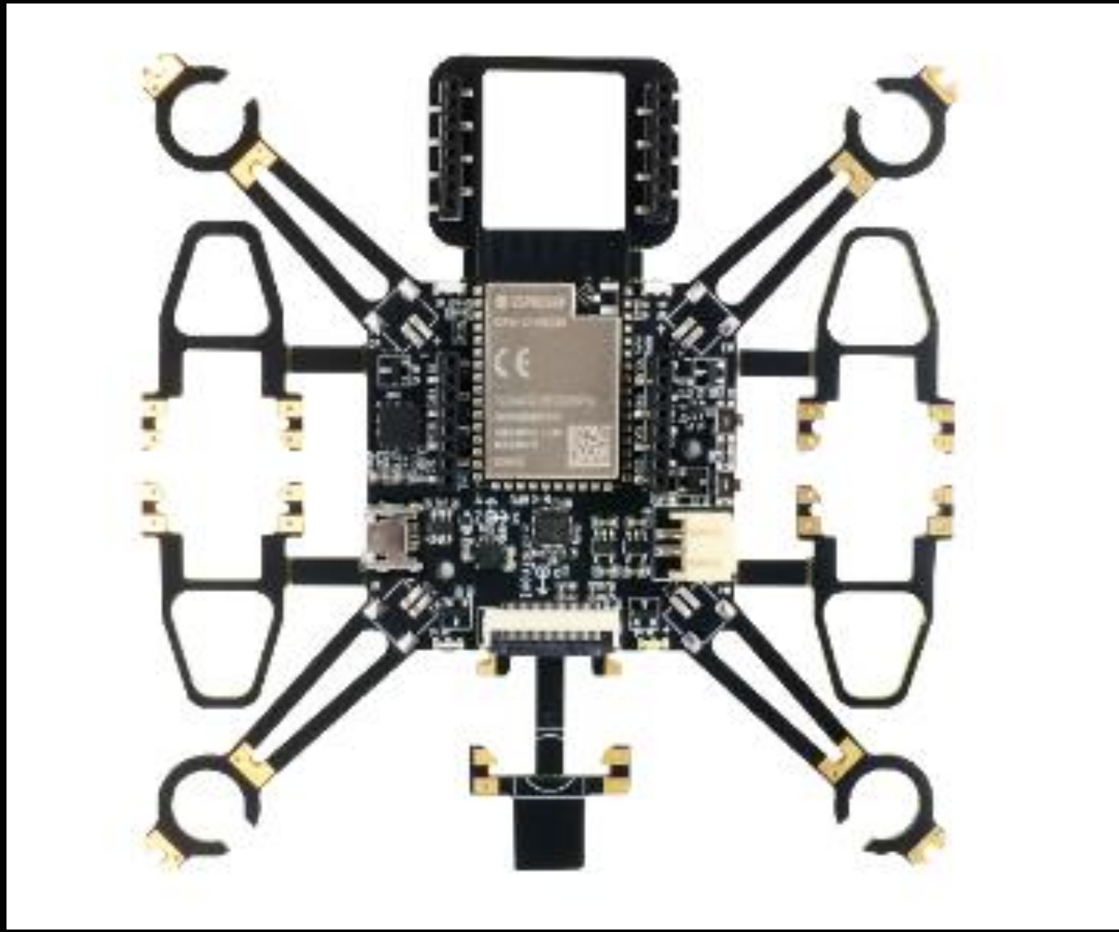
Simulation demonstrates rock-solid stability with minimal drift over time.

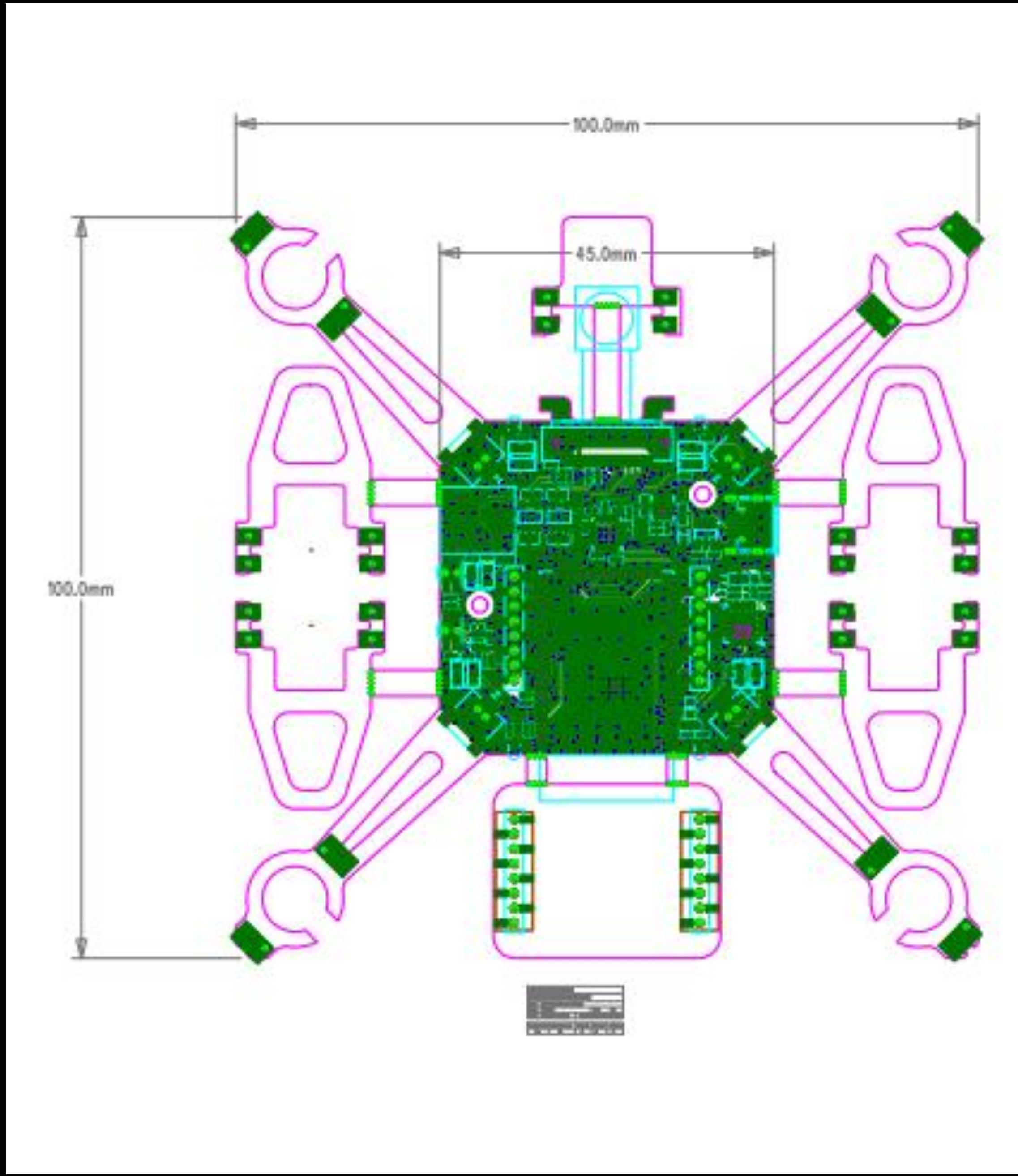
Position Hold Accuracy

Results show sub-meter position accuracy even in simulated wind conditions.

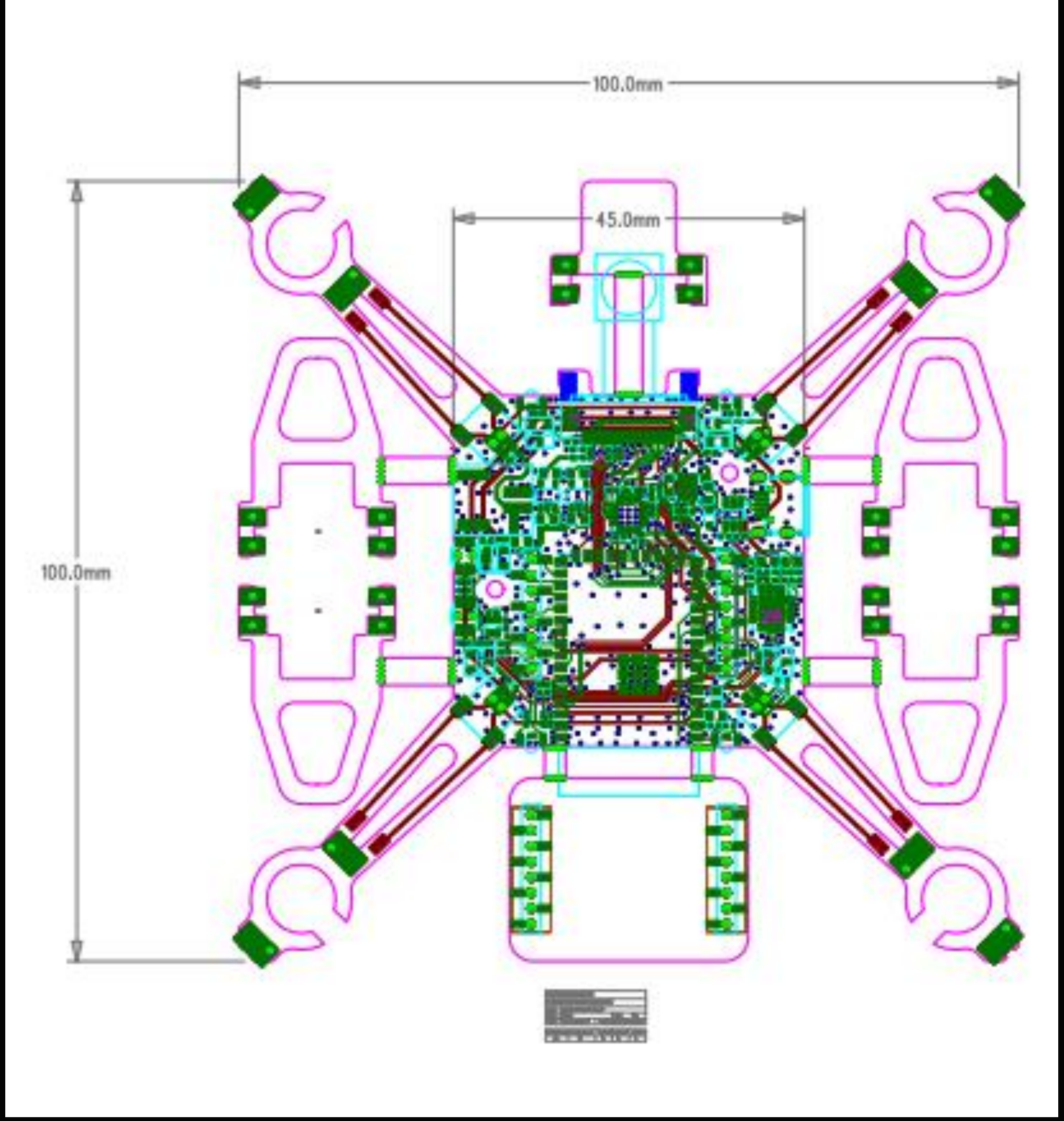
Altitude Hold Precision

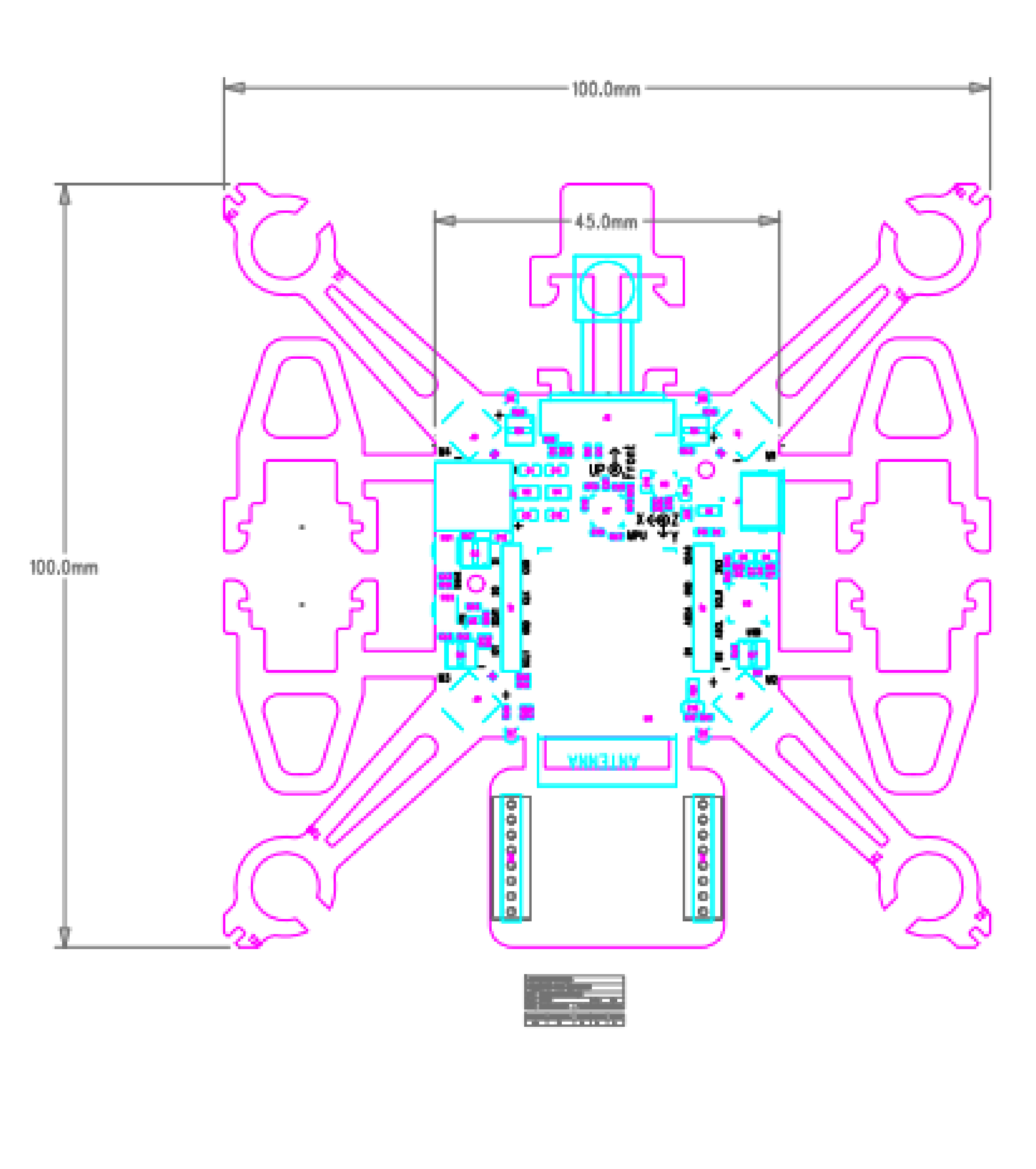
Altitude maintained within ±0.5m range across various simulated environmental conditions.

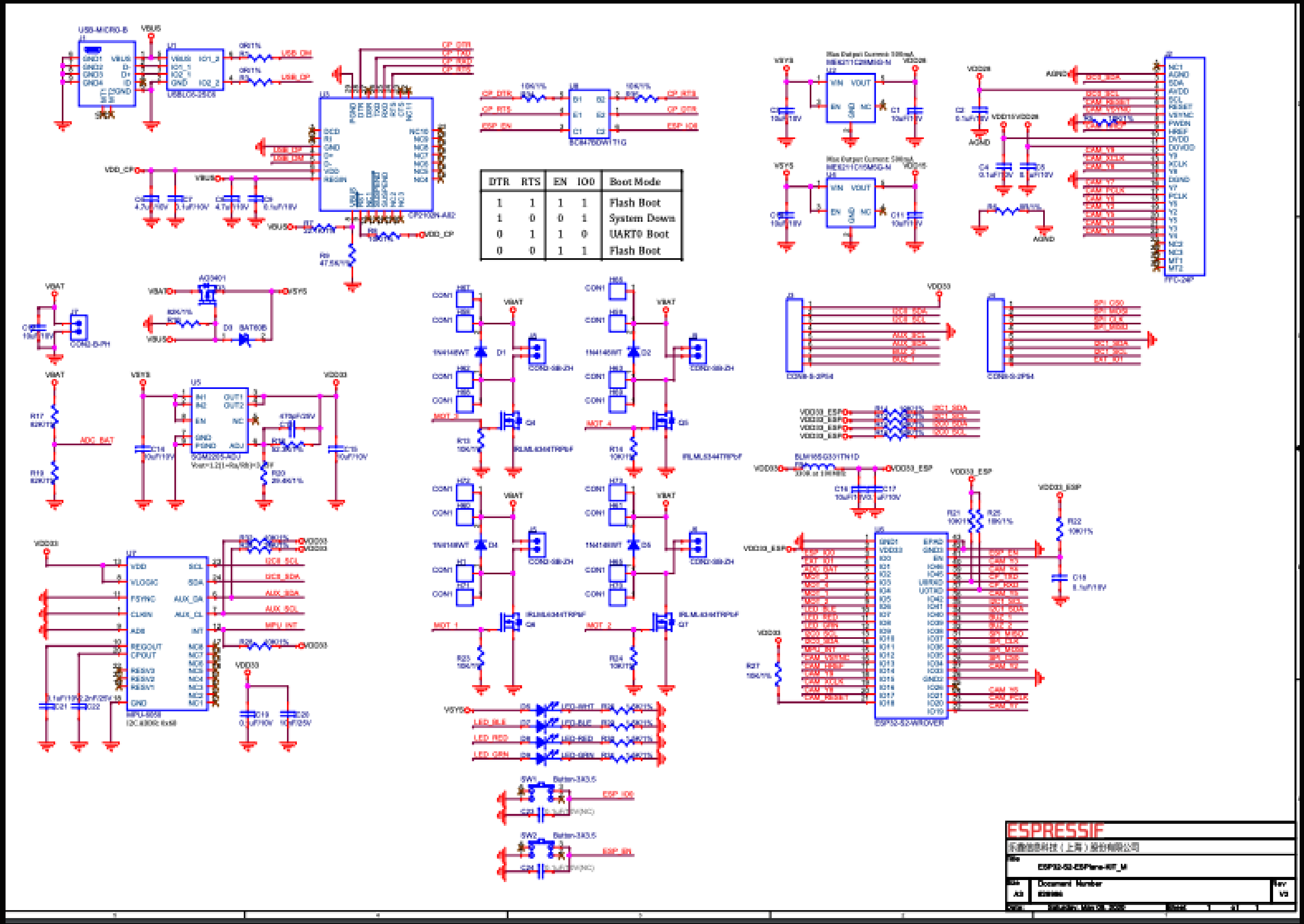




100.0mm
45.0mm
100.0mm


100.0mm
45.0mm
100.0mm

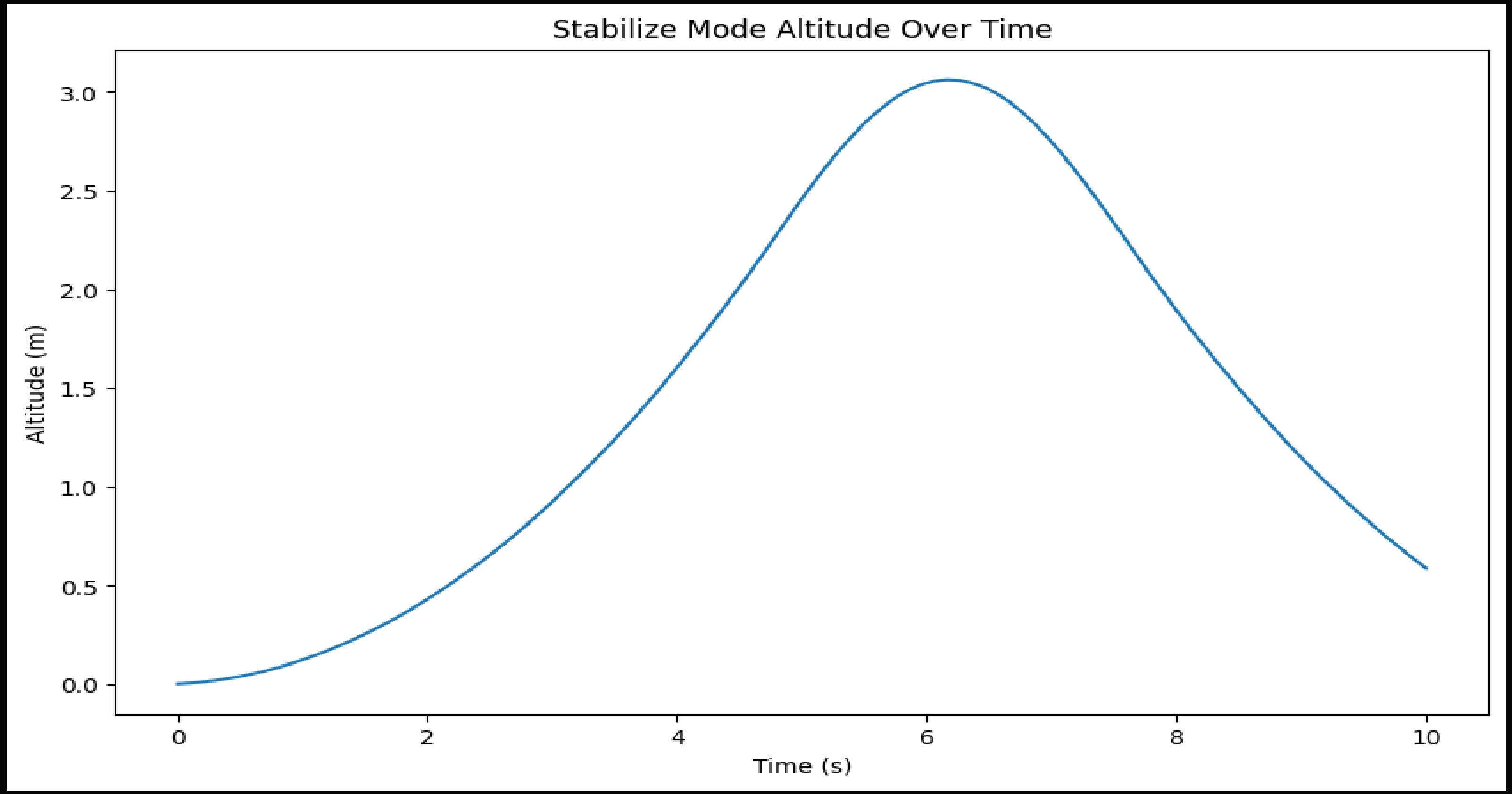

100.0mm
45.0mm
100.0mm
ANTENNA

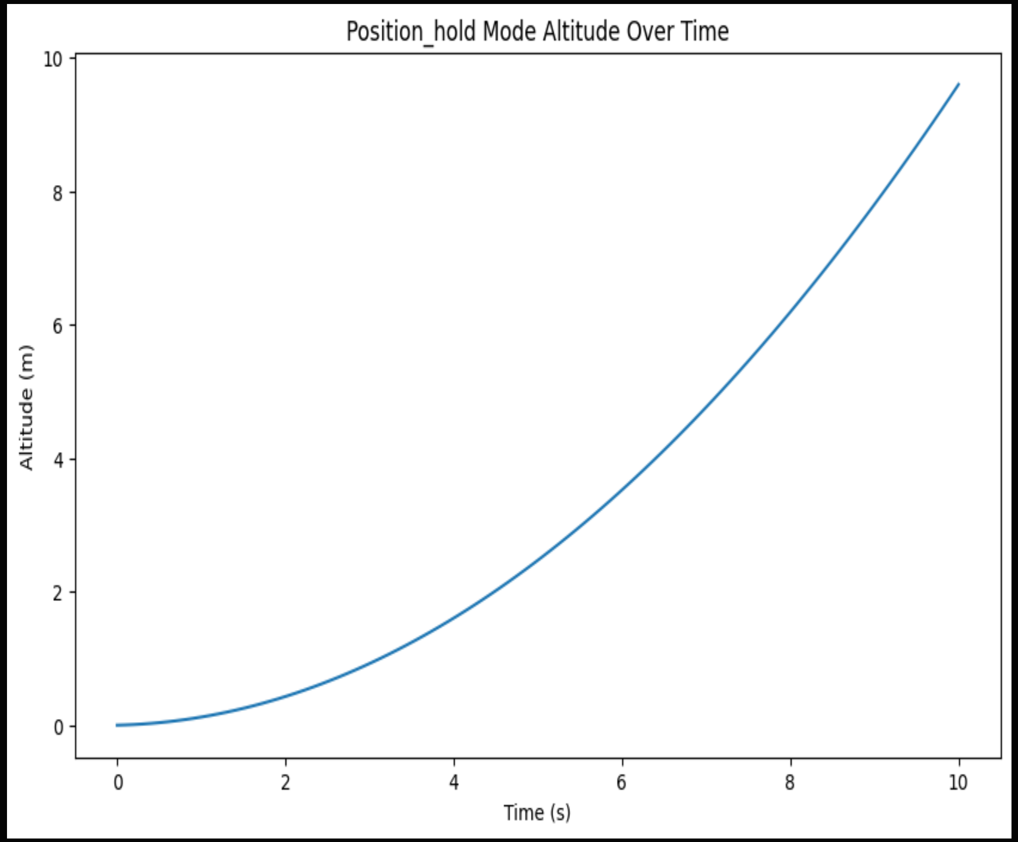


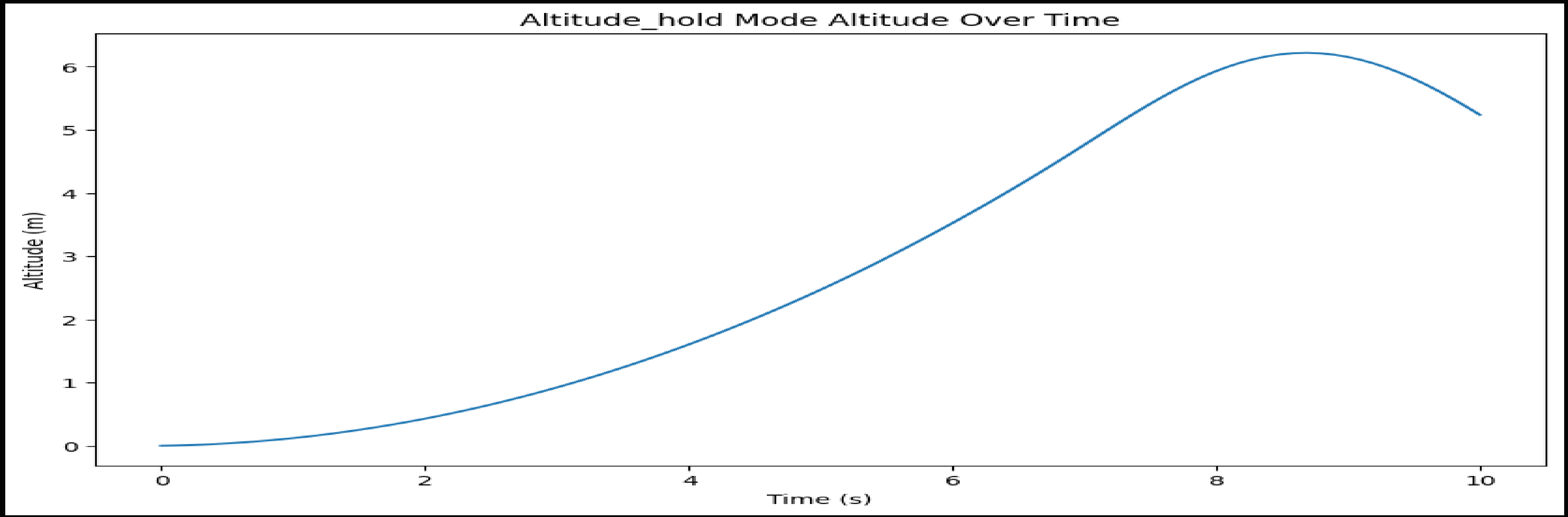
DTR	RTS	EN	IO0	Boot Mode
1	1	1	1	Flash Boot
1	0	0	1	System Down
0	1	1	0	UART0 Boot
0	0	1	1	Flash Boot

```
Flight mode set to: HOVER
Stabilize: Orientation=[-0.00050488 -0.00031308  0.0006271 ]
Altitude hold: Target Altitude=10.0, Current Altitude=0.00
Hover Mode: Altitude=0.00, Orientation=[-0.00050488 -0.00031308  0.0006271 ]
Telemetry | Position: [0.005      0.005      1.52433596], Velocity: [ 0.05       0.05       15.24335965], Orientation: [-0.00050488 -0.00031308  0.0006271 ], Battery: 9
Stabilize: Orientation=[ 0.00024429 -0.00065798  0.00056914]
Altitude hold: Target Altitude=10.0, Current Altitude=1.52
Hover Mode: Altitude=1.52, Orientation=[ 0.00024429 -0.00065798  0.00056914]
Telemetry | Position: [0.015      0.015      2.32463389], Velocity: [0.1        0.1        8.00297927], Orientation: [ 0.00024429 -0.00065798  0.00056914], Battery: 99.9
Stabilize: Orientation=[ 0.00230772 -0.00124343  0.00014221]
Altitude hold: Target Altitude=10.0, Current Altitude=2.32
Hover Mode: Altitude=2.32, Orientation=[ 0.00230772 -0.00124343  0.00014221]
Telemetry | Position: [0.03       0.03       3.07725403], Velocity: [0.15       0.15       7.52620135], Orientation: [ 0.00230772 -0.00124343  0.00014221], Battery: 99.9
Stabilize: Orientation=[ 0.00087556 -0.00058201 -0.00058179]
Altitude hold: Target Altitude=10.0, Current Altitude=3.08
Hover Mode: Altitude=3.08, Orientation=[ 0.00087556 -0.00058201 -0.00058179]
Telemetry | Position: [0.05       0.05       3.76465026], Velocity: [0.2        0.2        6.87396235], Orientation: [ 0.00087556 -0.00058201 -0.00058179], Battery: 99.9
Stabilize: Orientation=[-0.00013955  0.00018022 -0.00026891]
Altitude hold: Target Altitude=10.0, Current Altitude=3.76
Hover Mode: Altitude=3.76, Orientation=[-0.00013955  0.00018022 -0.00026891]
Telemetry | Position: [0.075      0.075      4.38695114], Velocity: [0.25       0.25       6.22300877], Orientation: [-0.00013955  0.00018022 -0.00026891], Battery: 99.9
Stabilize: Orientation=[ 0.00388034 -0.00032061  0.00112654]
Altitude hold: Target Altitude=10.0, Current Altitude=4.39
Hover Mode: Altitude=4.39, Orientation=[ 0.00388034 -0.00032061  0.00112654]
Telemetry | Position: [0.105      0.105      4.95404891], Velocity: [0.3        0.3        5.67097767], Orientation: [ 0.00388034 -0.00032061  0.00112654], Battery: 99.9
Stabilize: Orientation=[ 0.00247865 -0.00121346 -0.00022284]
Altitude hold: Target Altitude=10.0, Current Altitude=4.95
Hover Mode: Altitude=4.95, Orientation=[ 0.00247865 -0.00121346 -0.00022284]
Telemetry | Position: [0.14       0.14       5.47779361], Velocity: [0.35       0.35       5.23744707], Orientation: [ 0.00247865 -0.00121346 -0.00022284], Battery: 99.9
Stabilize: Orientation=[ 0.00216083 -0.00156299  0.00131783]
Altitude hold: Target Altitude=10.0, Current Altitude=5.48
Hover Mode: Altitude=5.48, Orientation=[ 0.00216083 -0.00156299  0.00131783]
Telemetry | Position: [0.18       0.18       5.97205347], Velocity: [0.4        0.4        4.94259856], Orientation: [ 0.00216083 -0.00156299  0.00131783], Battery: 99.9
Stabilize: Orientation=[ 1.96809686e-03 -3.12347334e-03  4.32151479e-05]
Altitude hold: Target Altitude=10.0, Current Altitude=5.97
```

```
Flight mode set to: HOVER
Stabilize: Orientation=[0. 0. 0.]
Altitude hold: Target Altitude=10.0, Current Altitude=0.00
Hover Mode: Altitude=0.00, Orientation=[0. 0. 0.]
Position: [0.005 0.005 1.51 ], Velocity: [ 0.05  0.05 15.1 ], Battery: 99.99%
Stabilize: Orientation=[0. 0. 0.]
Altitude hold: Target Altitude=10.0, Current Altitude=1.51
Hover Mode: Altitude=1.51, Orientation=[0. 0. 0.]
Position: [0.015   0.015   2.30199], Velocity: [0.1    0.1    7.9199], Battery: 99.98%
Stabilize: Orientation=[0. 0. 0.]
Altitude hold: Target Altitude=10.0, Current Altitude=2.30
Hover Mode: Altitude=2.30, Orientation=[0. 0. 0.]
Position: [0.03       0.03       3.05837951], Velocity: [0.15       0.15       7.5638951], Battery: 99.97%
Stabilize: Orientation=[0. 0. 0.]
Altitude hold: Target Altitude=10.0, Current Altitude=3.06
Hover Mode: Altitude=3.06, Orientation=[0. 0. 0.]
Position: [0.05       0.05       3.74785171], Velocity: [0.2        0.2        6.89472204], Battery: 99.96%
Stabilize: Orientation=[0. 0. 0.]
Altitude hold: Target Altitude=10.0, Current Altitude=3.75
Hover Mode: Altitude=3.75, Orientation=[0. 0. 0.]
Position: [0.075      0.075      4.37797471], Velocity: [0.25       0.25       6.30122997], Battery: 99.95%
Stabilize: Orientation=[0. 0. 0.]
Altitude hold: Target Altitude=10.0, Current Altitude=4.38
Hover Mode: Altitude=4.38, Orientation=[0. 0. 0.]
Position: [0.105      0.105      4.95367489], Velocity: [0.3        0.3        5.75700183], Battery: 99.94%
Stabilize: Orientation=[0. 0. 0.]
```


Stabilize Mode Altitude Over Time
Altitude (m)
Time (s)


Position_hold Mode Altitude Over Time
Altitude (m)
Time (s)


Altitude_hold Mode Altitude Over Time
Altitude (m)
Time (s)

Real-time Testing and Performance Validation

1. **Field Testing**

   Conducted extensive outdoor flight tests across various weather conditions.

2. **Data Collection**

   Gathered telemetry data using onboard sensors and external tracking systems.

3. **Performance Analysis**

   Analyzed flight data to validate firmware performance against design specifications.

4. **Iterative Refinement**

   Implemented improvements based on real-world test results and pilot feedback.

Prob 3- Team Lumen: Localization in GPS-Denied Areas

Team tackles the challenge of reliable autonomous navigation in GPS-denied environments. Our system uses computer vision, sensor fusion, and map-matching for precise vehicle positioning.

Introduction: The Challenge of Autonomous Driving Without GPS

GPS Limitations

Traditional GPS systems fail in urban canyons, tunnels, and under dense tree cover.

Alternative Methods

Computer vision, sensor fusion, and map-matching are crucial for reliable localization.

System Requirements

High-accuracy positioning is vital for safe and efficient autonomous navigation.

Limitations of Traditional GPS-Based Localization

1 **Signal Blockage**

Tall buildings and dense foliage weaken GPS signals, causing inaccuracy.

2 **Multipath Errors**

Reflected signals create errors in position estimations, leading to drift.

3 **Low Accuracy**

In challenging environments, GPS accuracy can be insufficient for autonomous driving.

Computer Vision for Robust Localization

Visual Features

Cameras identify landmarks, road markings, and other visual cues.

Feature Matching

Algorithms match features in camera images with a pre-existing map.

Position Estimation

The vehicle's location is estimated based on the matched features.

Sensor Fusion Algorithms for Enhanced Positioning

1

Sensor Data

Multiple sensors (LiDAR, radar, IMU) provide complementary data.

2

Data Fusion

Algorithms combine sensor data to improve accuracy and robustness.

3

Position Refinement

Combined data refines the vehicle's position estimate, minimizing errors.

Leveraging Map-Matching Techniques

1

Sensor Data

Raw sensor data provides initial position estimates.

2

Map Matching

Algorithms match vehicle trajectory to a digital map of the road network.

3

Position Correction

Mismatches are resolved, leading to a corrected, more accurate position.

Handling Dynamic Obstacles in GPS-Denied Environments

Obstacle Detection

Sensors detect dynamic obstacles in real time.

Path Planning

Algorithms replan the vehicle's path to avoid collisions.

Navigation Updates

The localization system adapts to dynamic changes.

Precision Localization in Urban Canyons and Tunnels

Environment	Challenges	Solutions
Urban Canyons	Signal blockage, multipath	Sensor fusion, visual odometry
Tunnels	No GPS, limited visibility	Inertial navigation, map matching

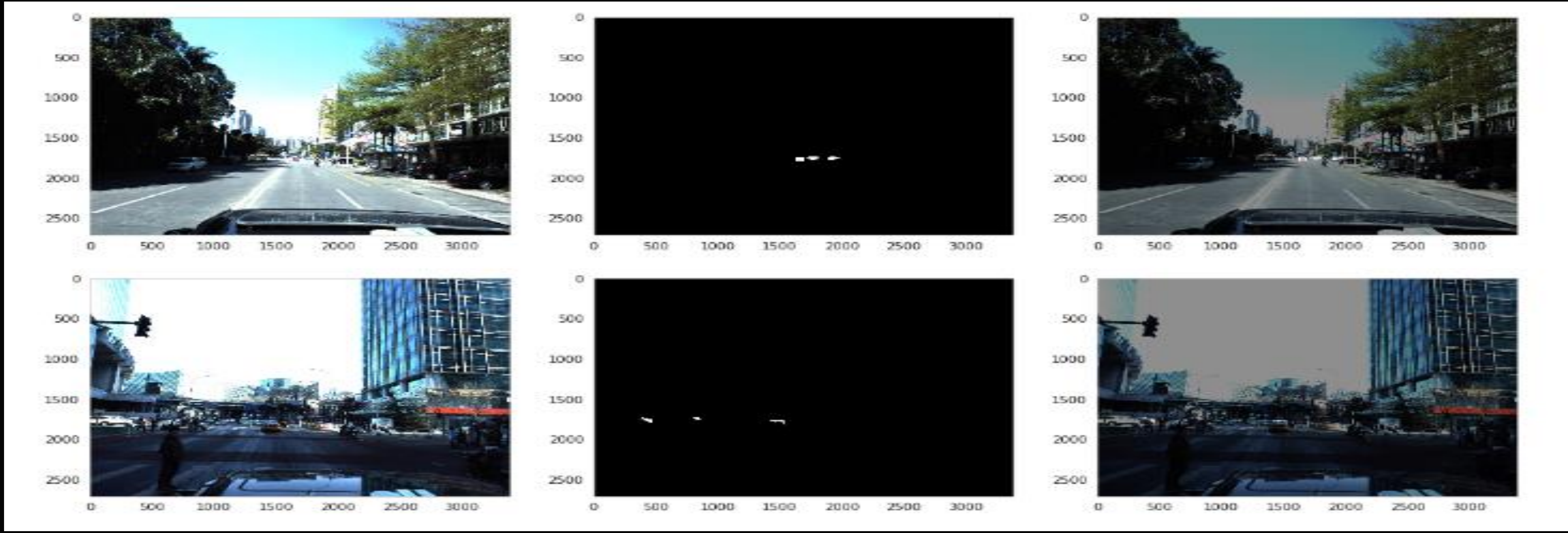
Field Testing and Validation of the Localization System

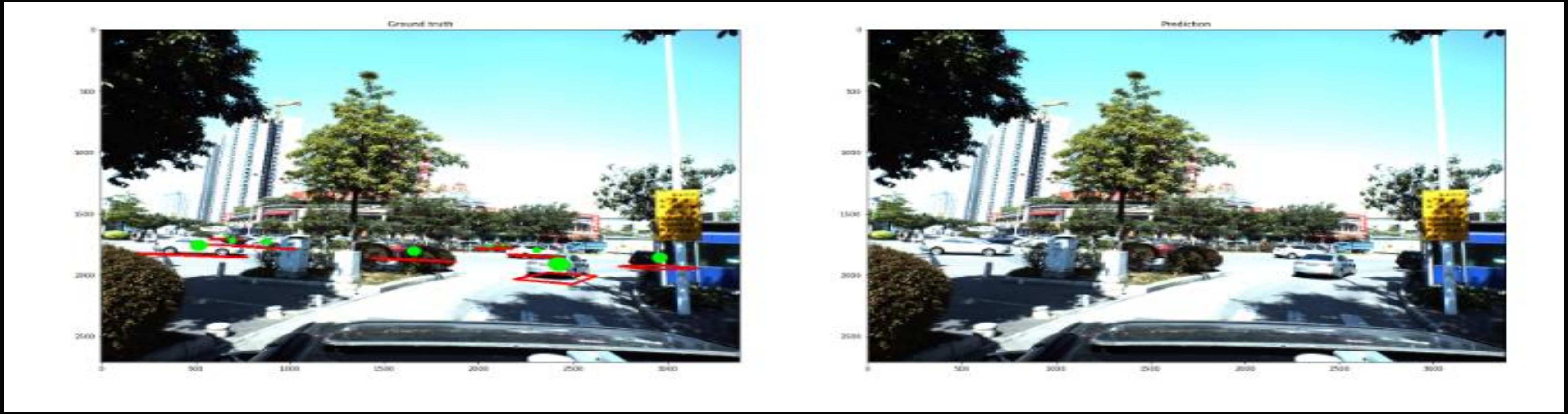
Urban Testing

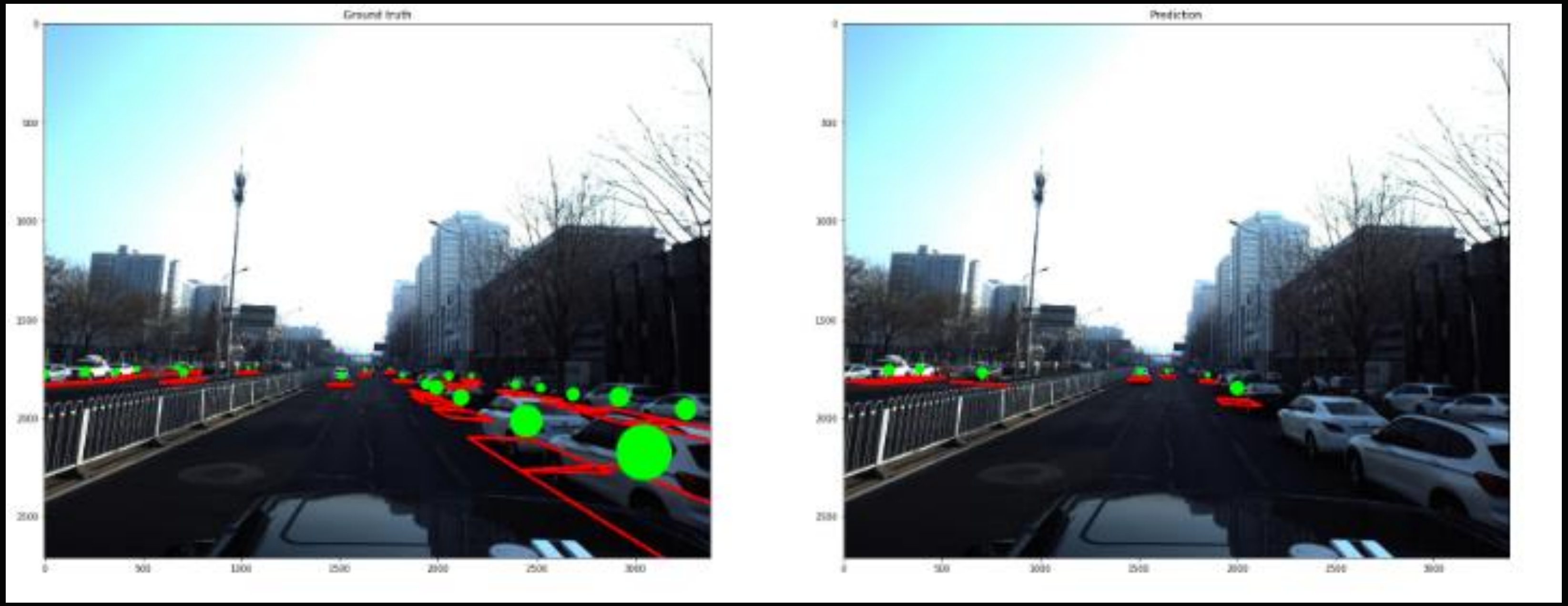
Testing in realistic urban conditions validates system performance.

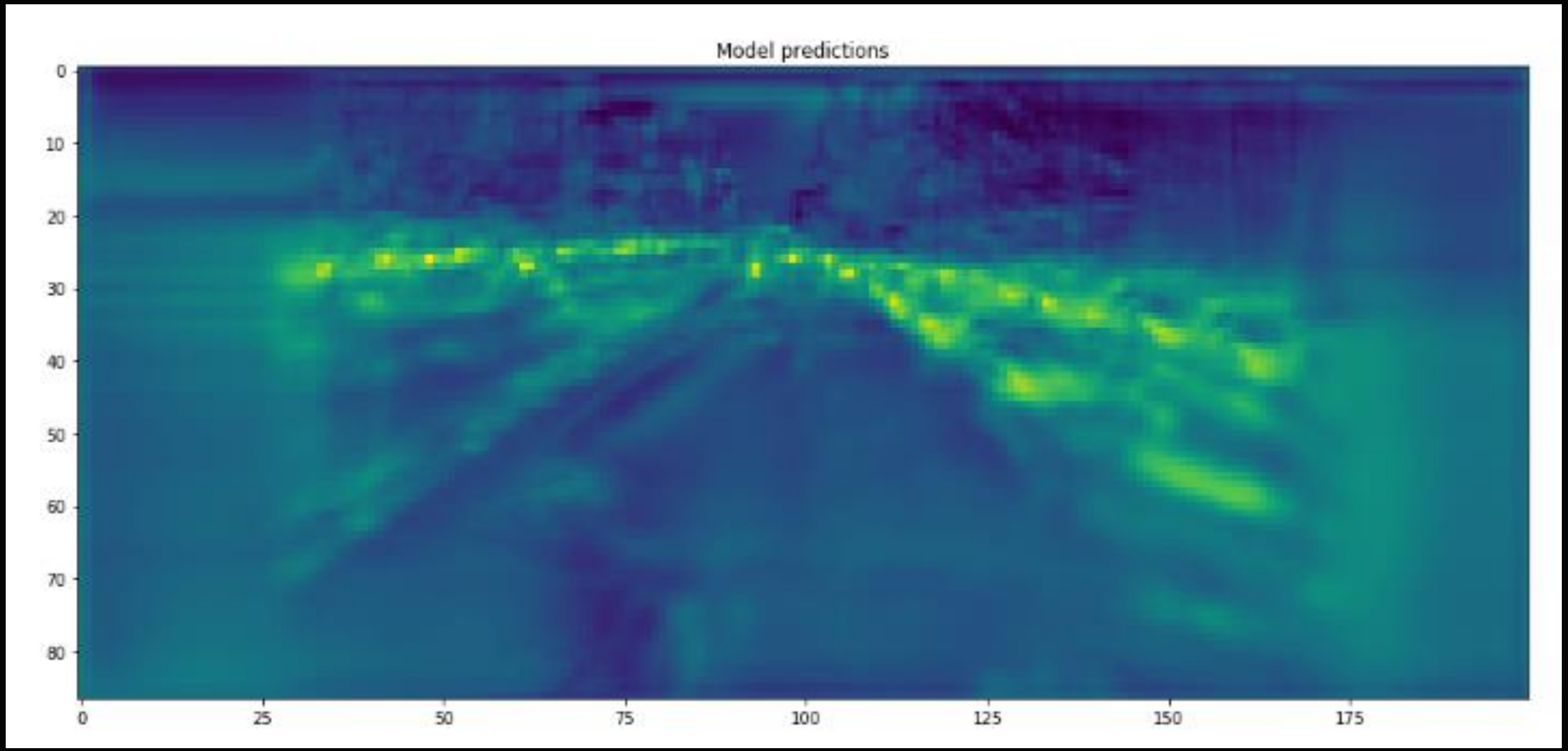
Tunnel Testing

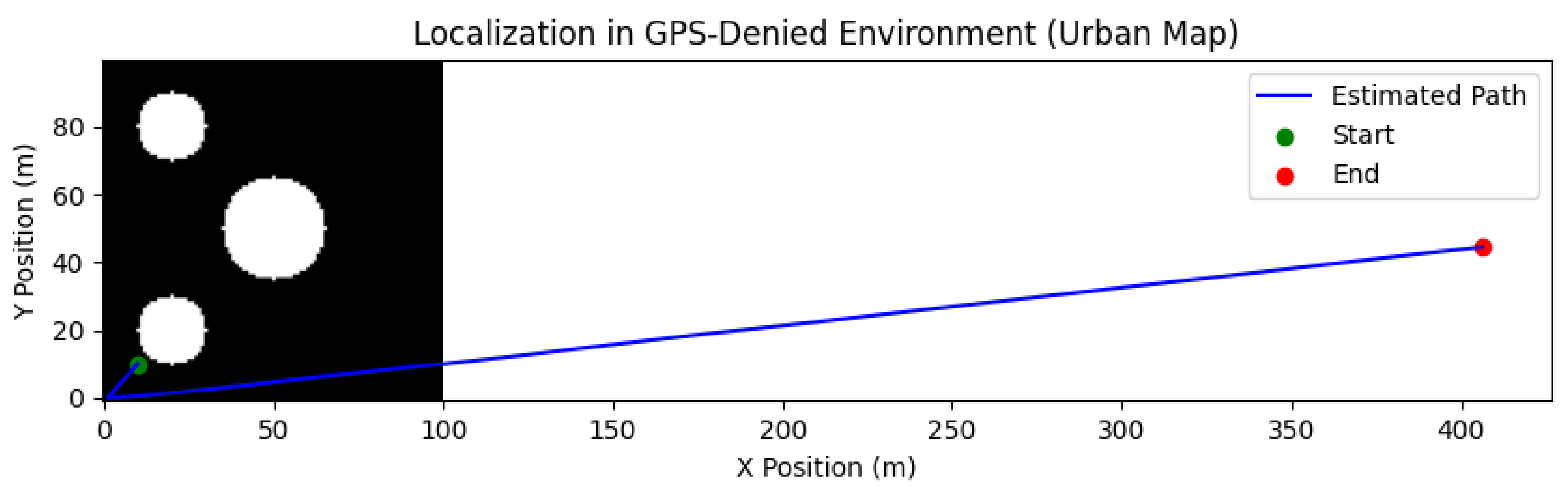
Tunnel tests assess the system's accuracy and robustness.

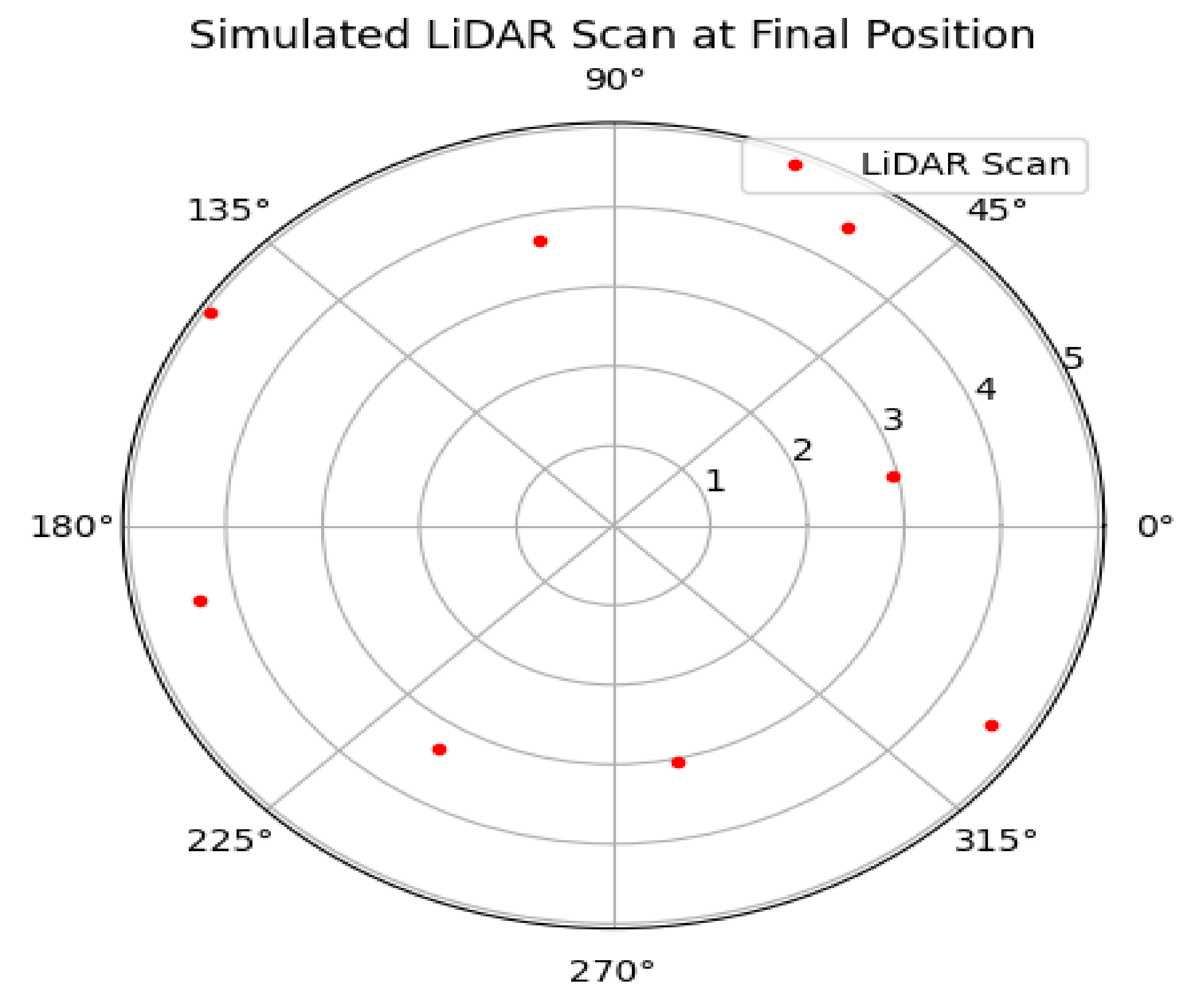


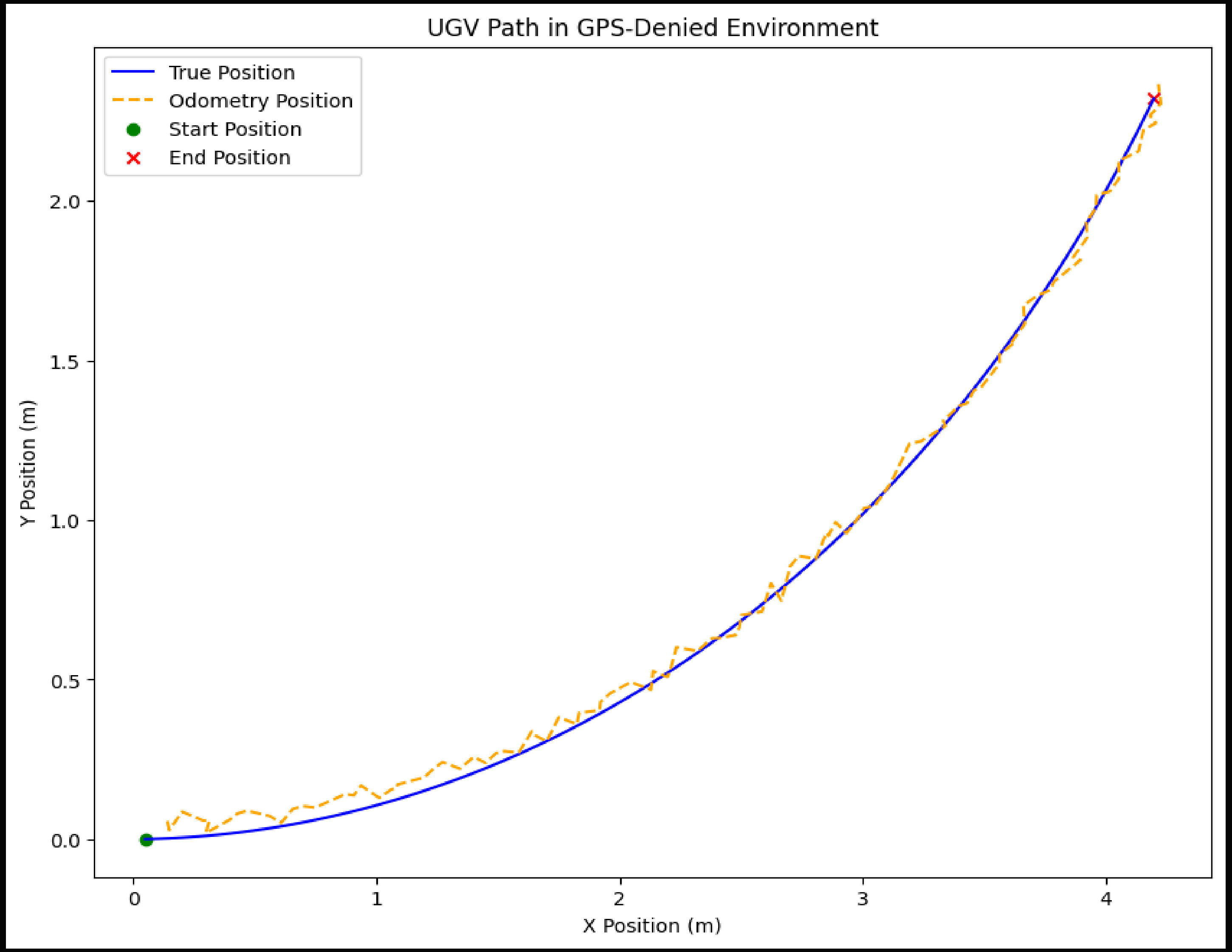

Ground truth
Prediction

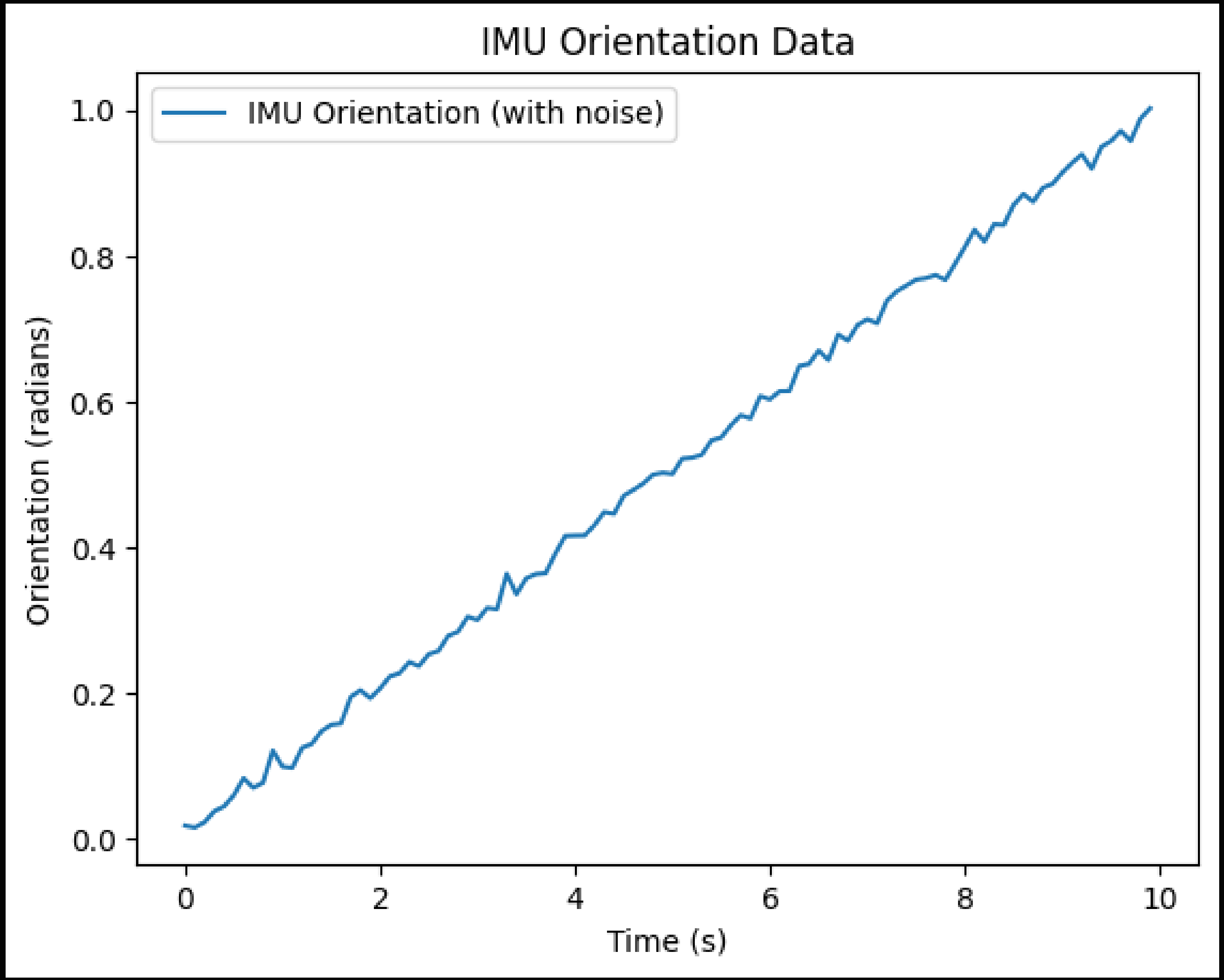

Ground truth
Prediction

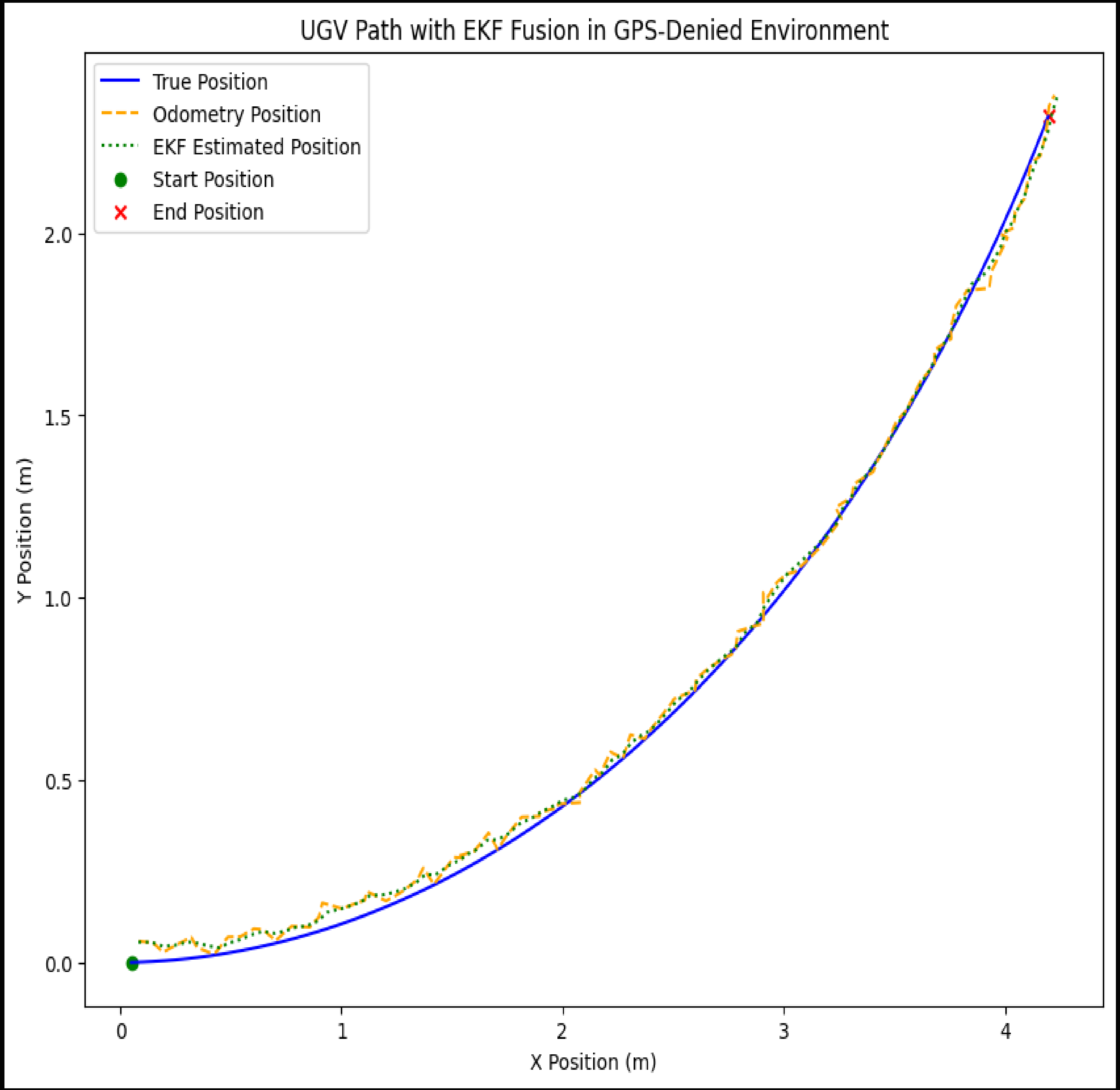

Model predictions
0
10
20
30
40
50
60
70
80
0
25
50
75
100
125
150
175


Localization in GPS-Denied Environment (Urban Map)
Estimated Path
Start
End
Y Position (m)
X Position (m)
0
20
40
60
80
0
50
100
150
200
250
300
350
400


Simulated LiDAR Scan at Final Position
LiDAR Scan
90°
135°
45°
180°
0°
225°
315°
270°
1
2
3
4
5


UGV Path in GPS-Denied Environment
True Position
Odometry Position
Start Position
End Position
Y Position (m)
X Position (m)
0.0
0.5
1.0
1.5
2.0
0
1
2
3
4

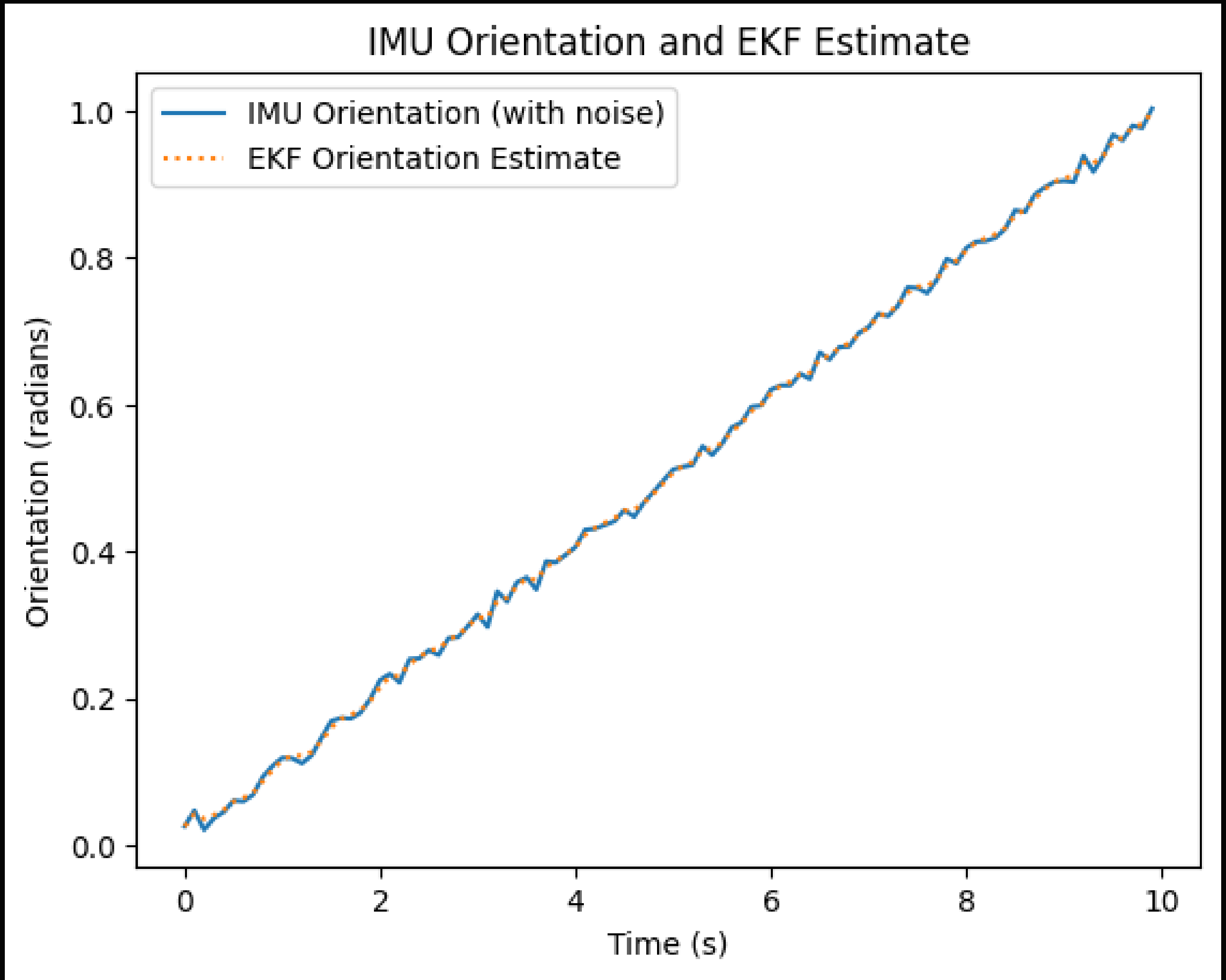

IMU Orientation Data
IMU Orientation (with noise)
Orientation (radians)
Time (s)
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10



Extended Kalman Filter (EKF) based on sensor fusion. In this context, the EKF combines noisy data from different sensors (such as IMU and odometry) to produce a more accurate estimate of the UGV's position and orientation over time. The goal of using the EKF is to correct for the individual errors and noise in each sensor reading, resulting in a more reliable path estimate than relying on any single sensor alone.

Why the EKF-Estimated Path is Useful

In GPS-denied environments, accurate localization is challenging because you cannot rely on GPS signals. By fusing multiple sensors, the EKF provides a robust way to estimate the UGV’s position and orientation, helping it to navigate more accurately in these environments.



IMU Orientation refers to the angular position or orientation of an object (in this case, the UGV) in space, measured relative to a starting position. The IMU (Inertial Measurement Unit) is a sensor that provides information about the UGV's orientation by measuring angular velocities and accelerations. However, the IMU data tends to have noise and can drift over time, especially in GPS-denied environments where there's no external correction (like GPS) to recalibrate it.

IMU Orientation in the Context of UGV Localization

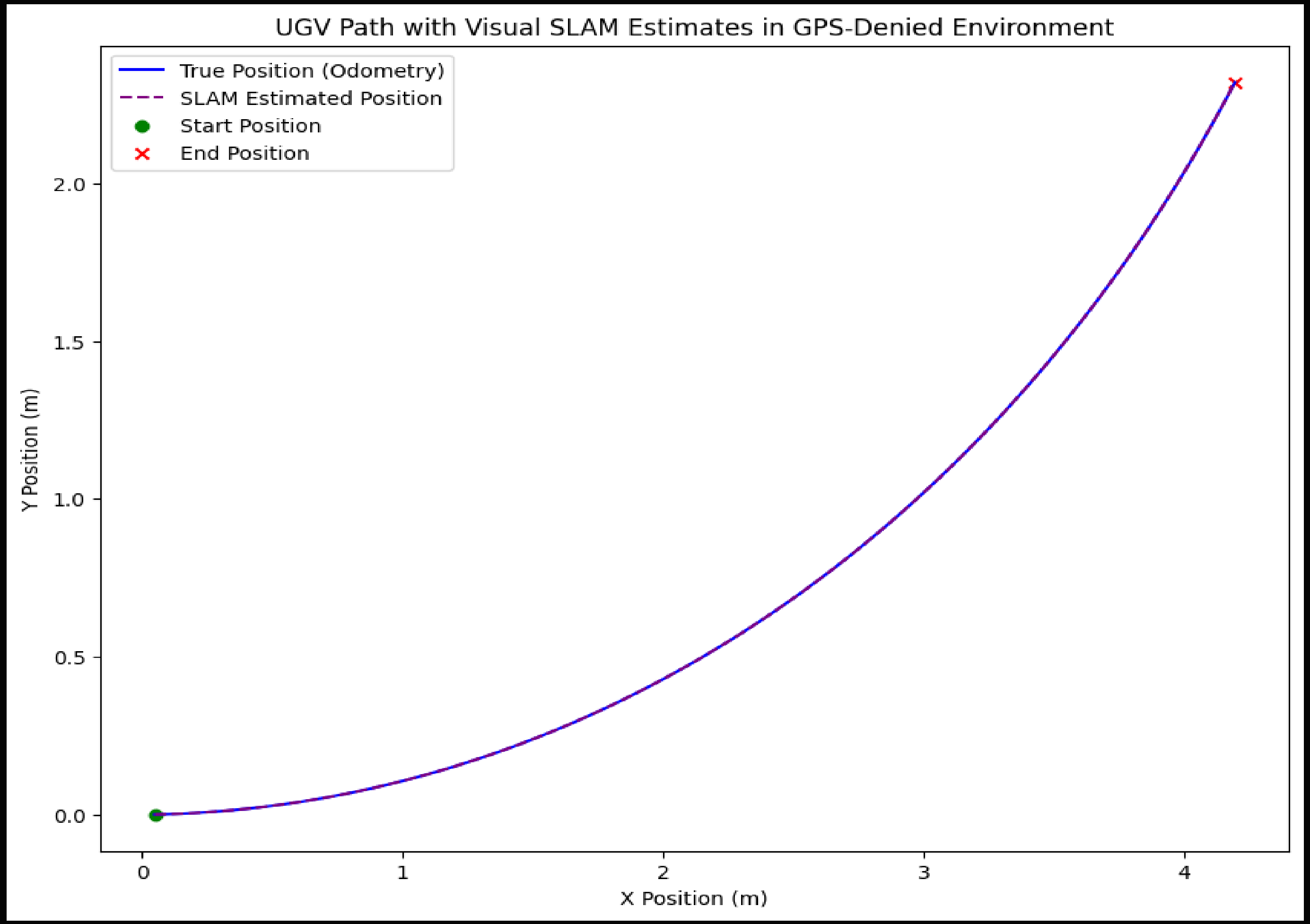
Orientation (Angle): The IMU measures the angular position (orientation) of the UGV. This is often given in radians or degrees relative to the starting orientation.

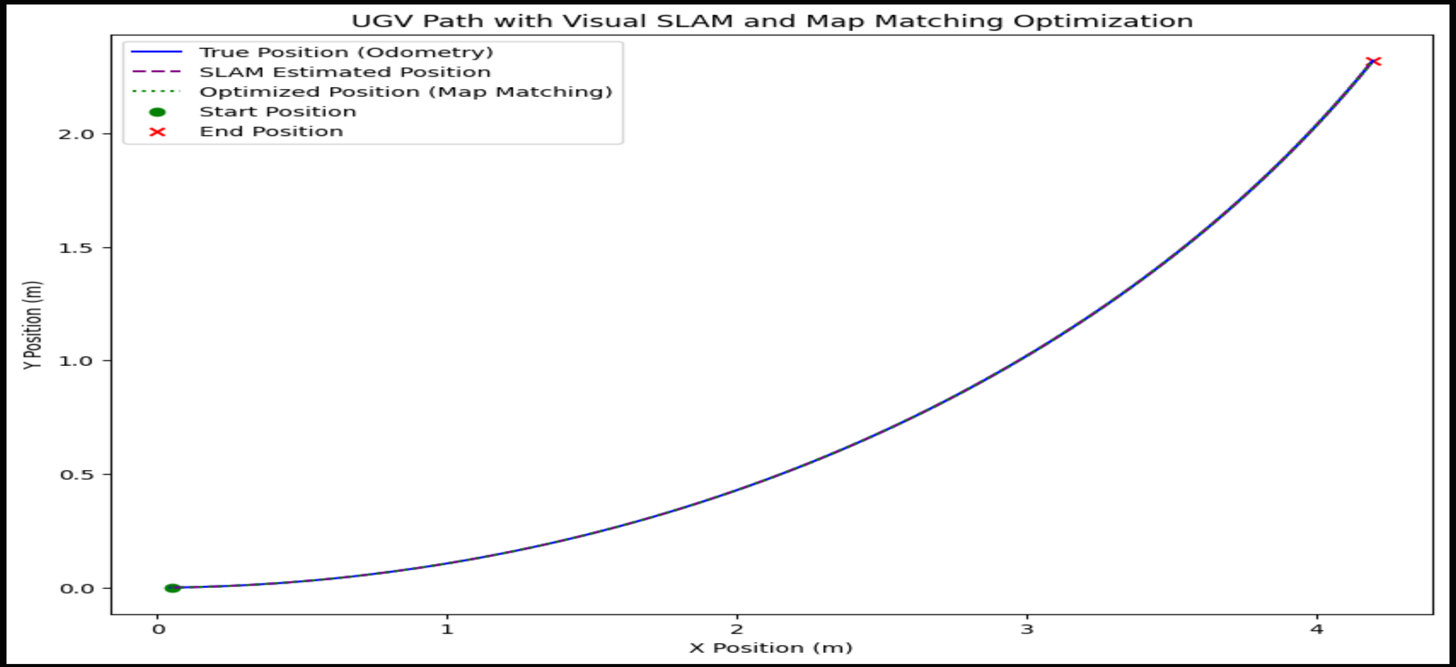
Drift and Noise: IMUs can be noisy and may experience drift, especially over long durations, due to accumulated errors in measurements.

Limitations: While an IMU can give a good estimate of orientation over short time periods, relying on it alone for accurate orientation can be problematic because the accumulated error can cause the orientation estimate to "drift" away from the actual value over time.

Graph of IMU Orientation vs. EKF Orientation Estimate

In the code, we visualize both the IMU Orientation and the EKF Orientation Estimate over time to demonstrate how the EKF reduces noise and drift, producing a more accurate orientation





Visual SLAM (Simultaneous Localization and Mapping) is a method that uses visual information (from a camera or other visual sensors) to help a robot or autonomous vehicle map its surroundings while simultaneously estimating its position within that map. In essence, Visual SLAM is a solution to the problem of "Where am I?" and "What does the surrounding environment look like?" especially in scenarios where traditional localization methods (like GPS) are unavailable or unreliable.

Interpretation of the Graph

Alignment: If the SLAM Estimated Position closely aligns with the True Position (Odometry), it indicates that Visual SLAM is successfully enhancing the UGV's localization by correcting for odometry drift.

Discrepancies: Any significant deviation between the True Position and SLAM Estimated Position can occur if the feature matching is not accurate enough or if there aren't enough distinctive landmarks for SLAM to recognize, which could result in estimation errors.

Path Correction: Visual SLAM helps "anchor" the UGV to a consistent map of the environment, allowing it to correct its path and reduce localization error.

Conclusion: Towards Reliable Autonomous Navigation Anywhere

Team Lumen's localization system enables reliable autonomous navigation in GPS-denied areas. This system advances autonomous vehicle technology, improving safety and efficiency.

Problem 4-Autonomous Navigation in Dense Urban Traffic

pedestrians

Team Lumen has developed an AI-driven solution that enables autonomous ground vehicles to navigate through dense urban traffic. This technology leverages advanced sensors and machine learning algorithms to predict and adapt to the unpredictable behaviors of other vehicles, , and two-wheelers commonly found in Indian cities.

Challenges of Urban Driving in India

1 **Congested Roads**

Indian cities often experience heavy traffic congestion, making it difficult for autonomous vehicles to navigate efficiently.

2 **Irregular Driving Patterns**

Drivers in India often exhibit less predictable driving patterns, which can be challenging for AI systems to interpret.

3 **Frequent Stops**

Frequent stops and starts due to traffic signals and other obstacles can disrupt the flow of autonomous vehicles.

4 **Minimal Lane Discipline**

Lane discipline is often relaxed in Indian cities, requiring the autonomous system to adapt to unpredictable lane changes and maneuvers.

Sensor Suite for Perception

LiDAR

LiDAR provides accurate 3D point cloud data, enabling the vehicle to perceive its surroundings with high precision.

Cameras

Cameras provide visual information about the environment, including lane markings, traffic signs, and other vehicles.

Radar

Radar sensors detect the presence and movement of objects, even in low-visibility conditions like fog or rain.

Ultrasonic Sensors

Ultrasonic sensors measure distances to nearby objects, providing critical information for parking and obstacle avoidance.

Mapping and Localization

High-Definition Maps

Accurate and detailed maps are crucial for autonomous navigation, providing information about road geometry, lane configurations, and traffic signals.

Localization

The vehicle must accurately determine its position and orientation in the world using sensor data and map information.

Dynamic Map Updates

The system needs to adapt to changing road conditions, such as construction zones, accidents, and temporary road closures.

Prediction of Vehicle Behavior

1

Trajectory Estimation

Predicting the future paths of other vehicles based on their current movement and surrounding context.

2

Intention Recognition

Determining the goals and intentions of other vehicles, such as turning, lane changes, or stopping.

3

Risk Assessment

Evaluating the potential for collisions based on predicted trajectories and intentions of other vehicles.

Adapting to Pedestrians and Two-Wheelers

Pedestrian Detection

The system must reliably detect and track pedestrians, even in crowded environments with limited visibility.

Two-Wheeler Detection

Two-wheelers are a common sight in Indian cities, requiring the autonomous system to adapt to their unique behavior.

Behavior Prediction

Predicting the actions of pedestrians and two-wheelers, such as crossing the road, turning, or stopping.

Safety Measures

The system must prioritize safety by yielding to pedestrians and two-wheelers, ensuring smooth and safe interactions.

Navigational Decision Making

1. **Route Planning**

   The system selects the most efficient route based on current traffic conditions, road closures, and destination.

2. **Lane Keeping**

   Maintaining the vehicle's position within the lane, avoiding lane drifts and unexpected maneuvers.

3. **Traffic Light Recognition**

   Identifying and responding to traffic signals, ensuring the vehicle stops at red lights and proceeds at green lights.

4. **Obstacle Avoidance**

   Detecting and avoiding obstacles in the path, including stationary objects and moving vehicles.

Path Planning and Trajectory Generation

Smooth and Efficient Trajectories	Generate trajectories that are safe, smooth, and minimize travel time.
Adaptive Path Planning	The system must adapt to changing road conditions, such as lane closures, traffic jams, and unexpected obstacles.
Collision Avoidance	Planning trajectories that avoid collisions with other vehicles, pedestrians, and obstacles.
Driver Comfort	Generate trajectories that are comfortable for passengers, avoiding sudden accelerations, decelerations, and turns.

Real-World Testing and

Simulation

Testing the system in virtual environments to validate its performance under various conditions.

Real-World Trials

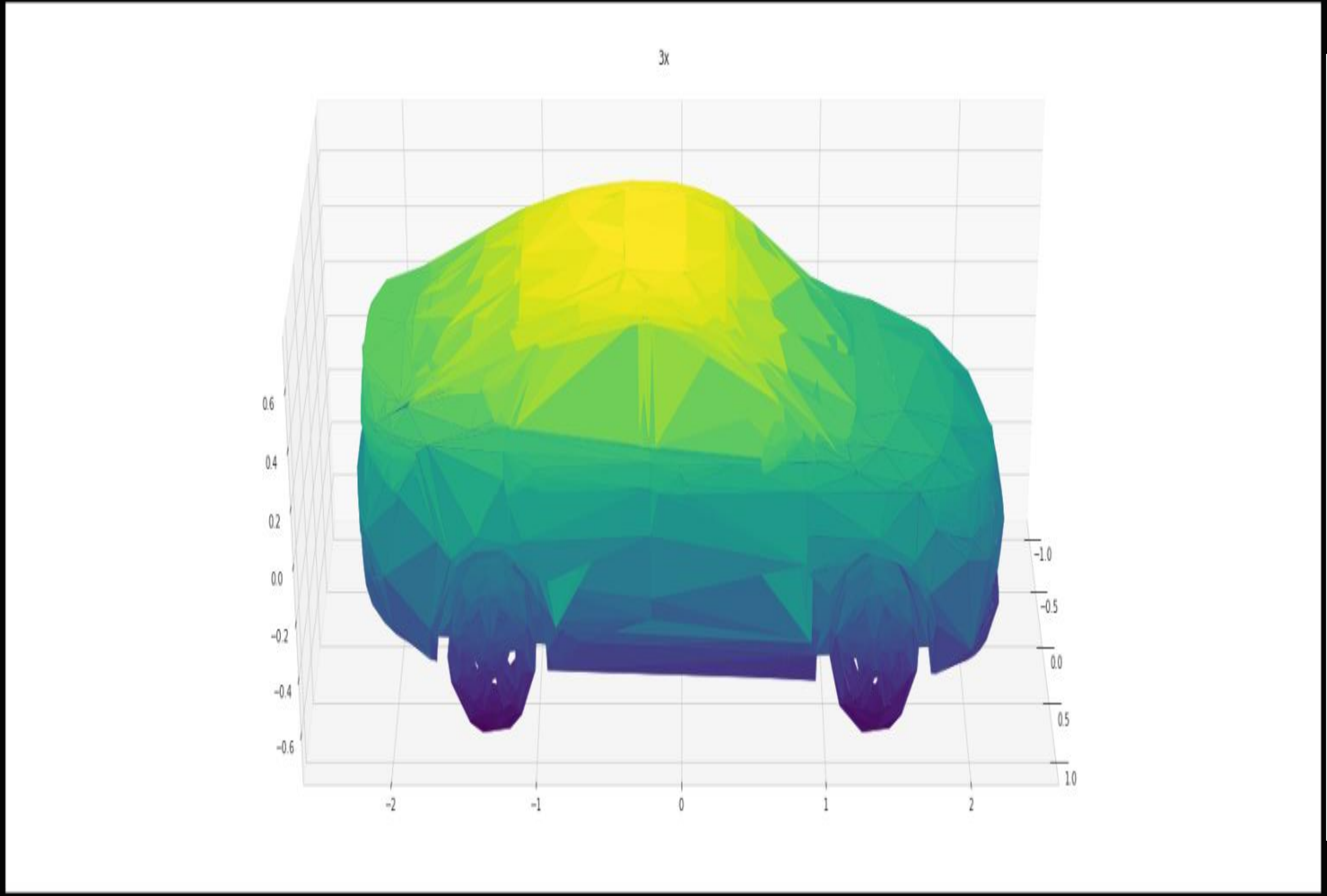
Conducting tests on public roads with human oversight to evaluate the system's performance in real-world scenarios.

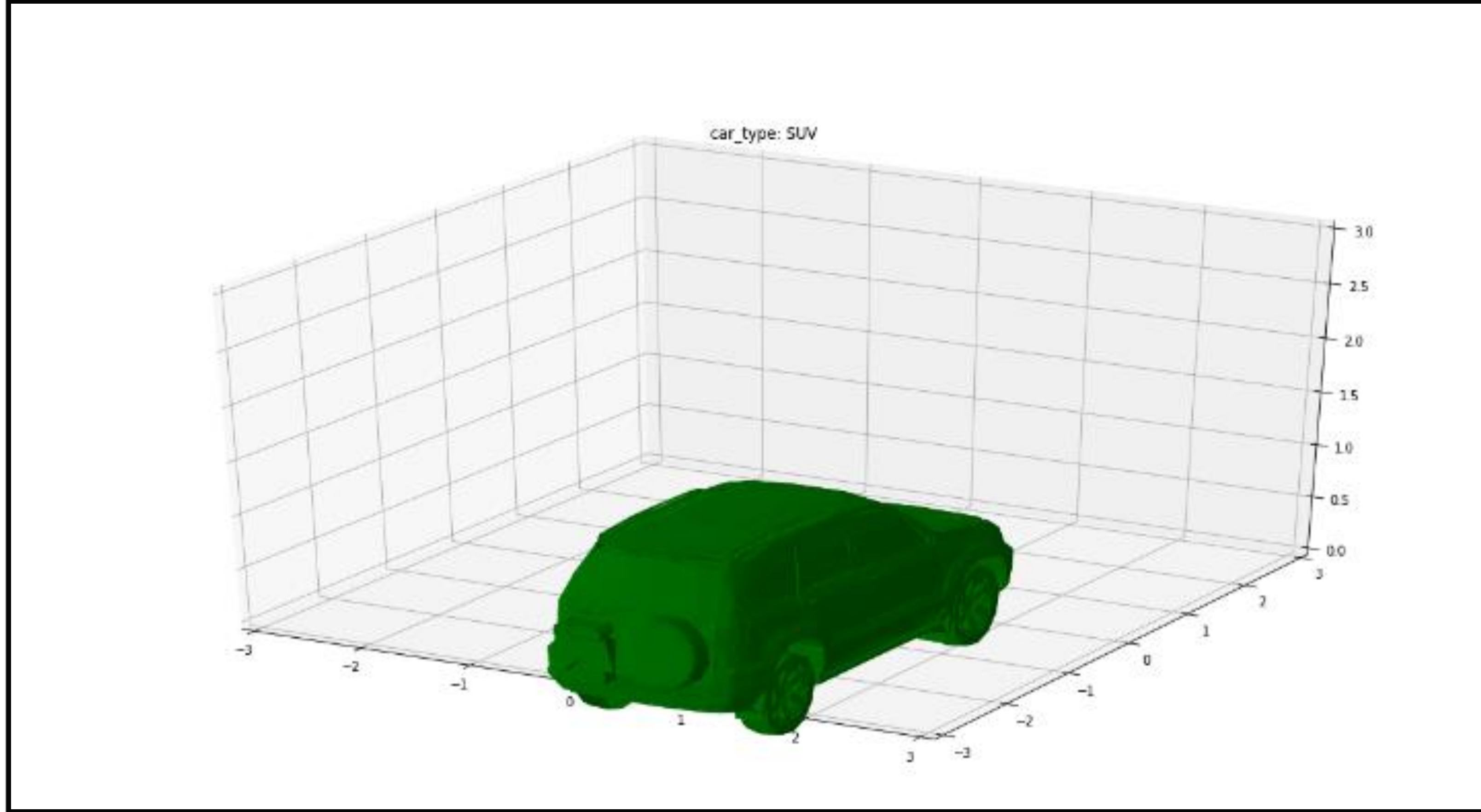
Data Collection

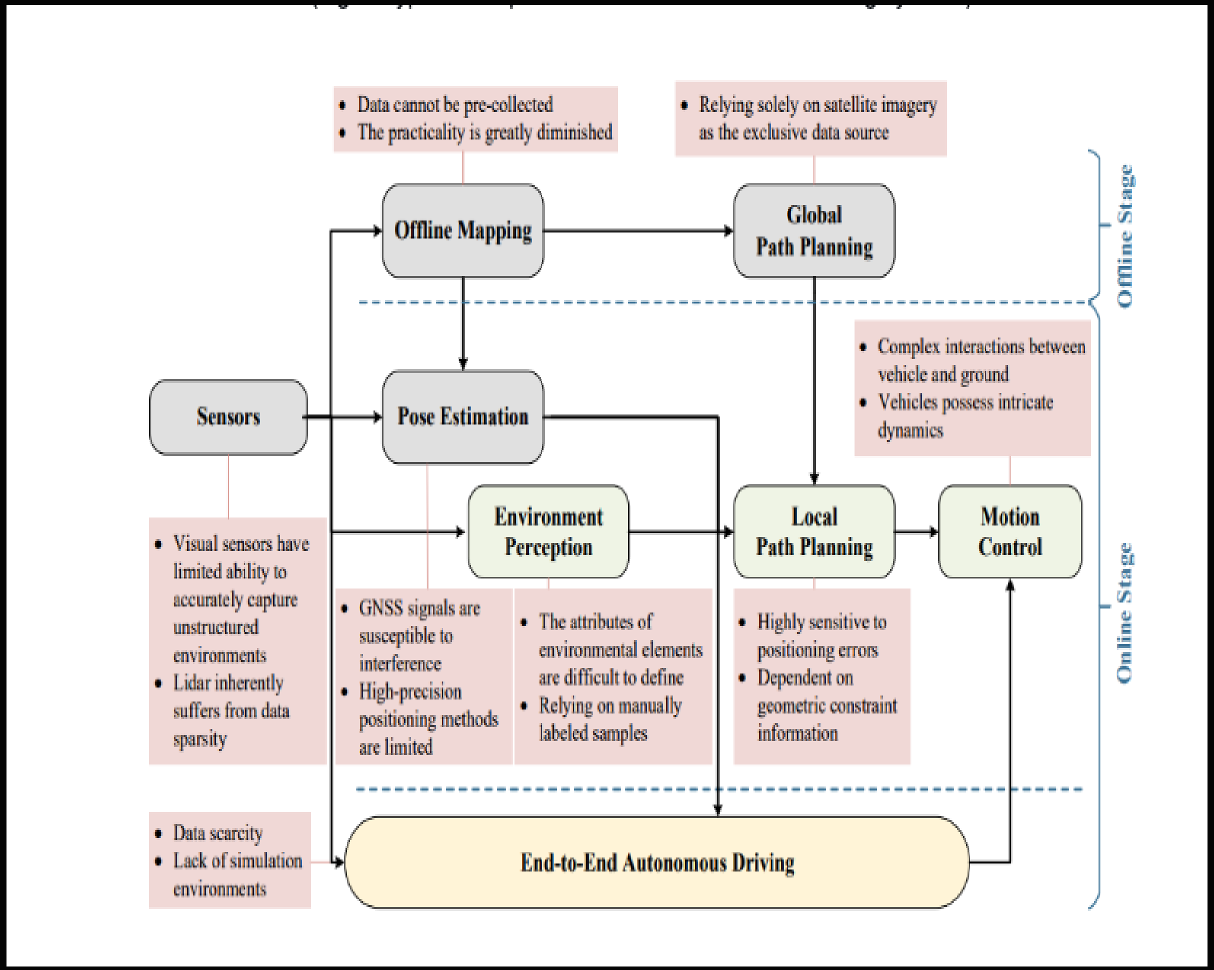
Gathering data from real-world trials to improve the system's algorithms and enhance its performance.

Continuous Improvement

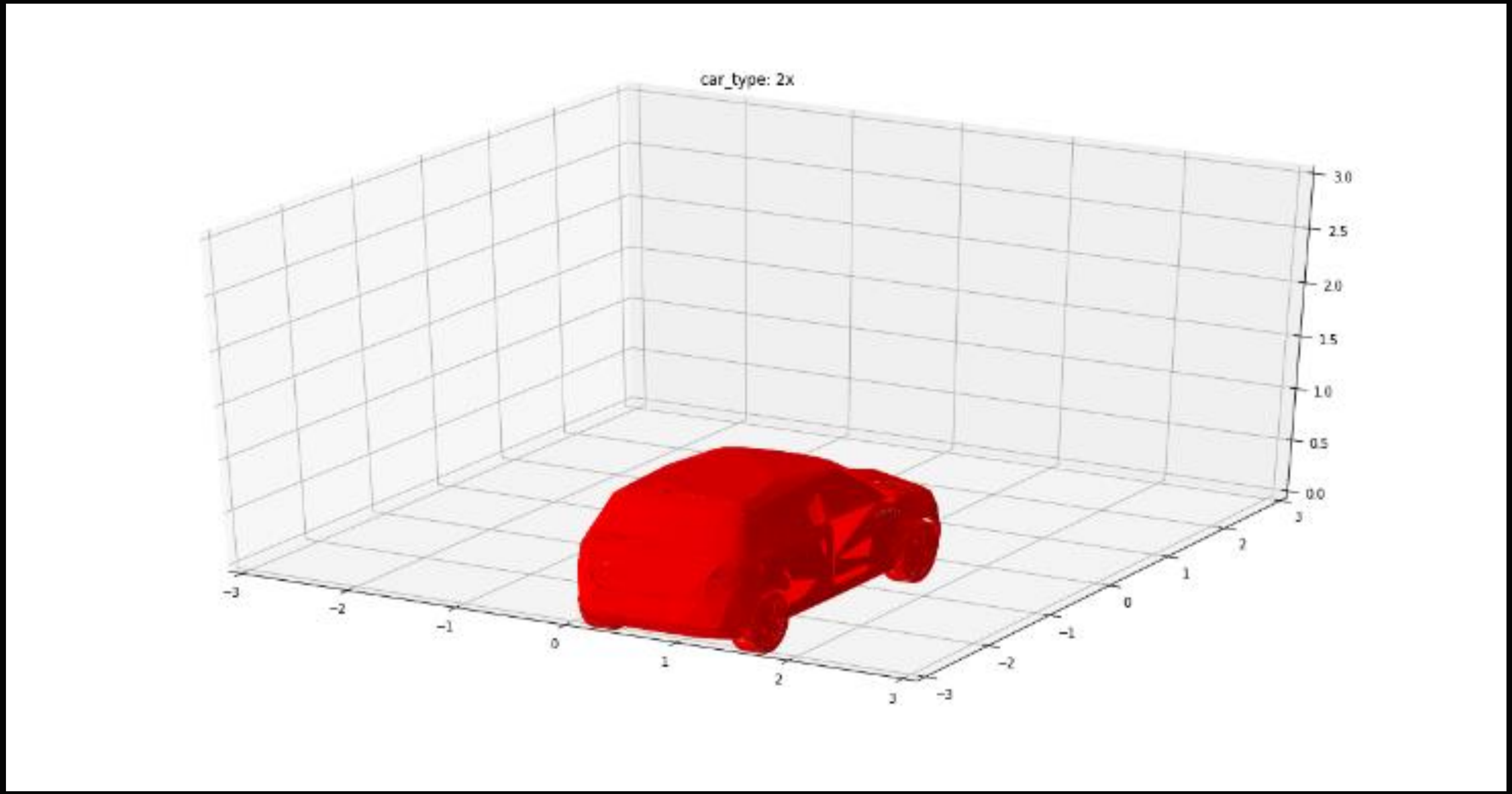
Analyzing test results and identifying areas for improvement, leading to ongoing refinement and optimization of the autonomous navigation system.


3x
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-2
-1
0
1
2
-1.0
-0.5
0.0
0.5
1.0

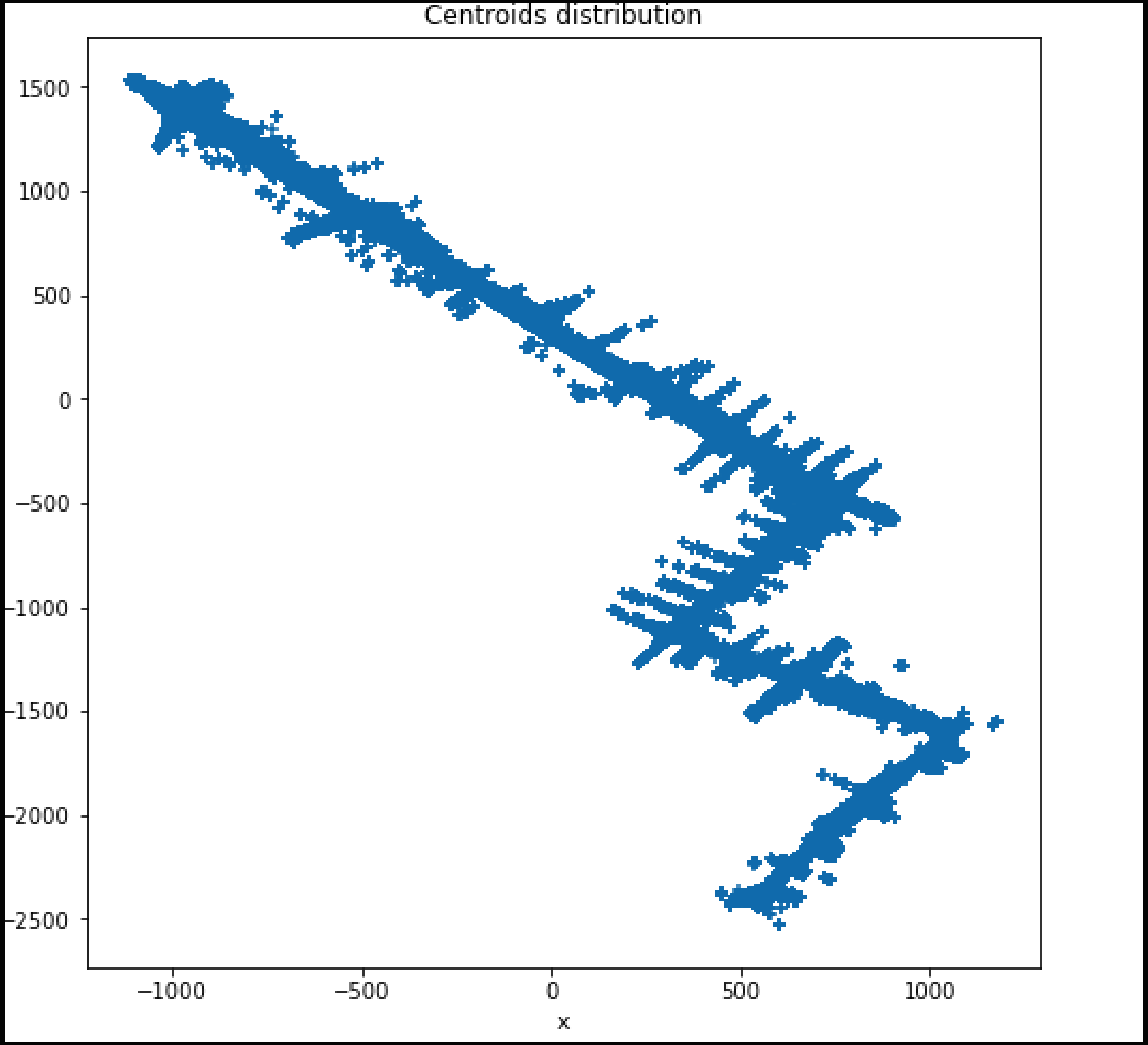

car_type: SUV
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-3
-2
-1
0
1
2
3
3
2
1
0
-1
-2
-3



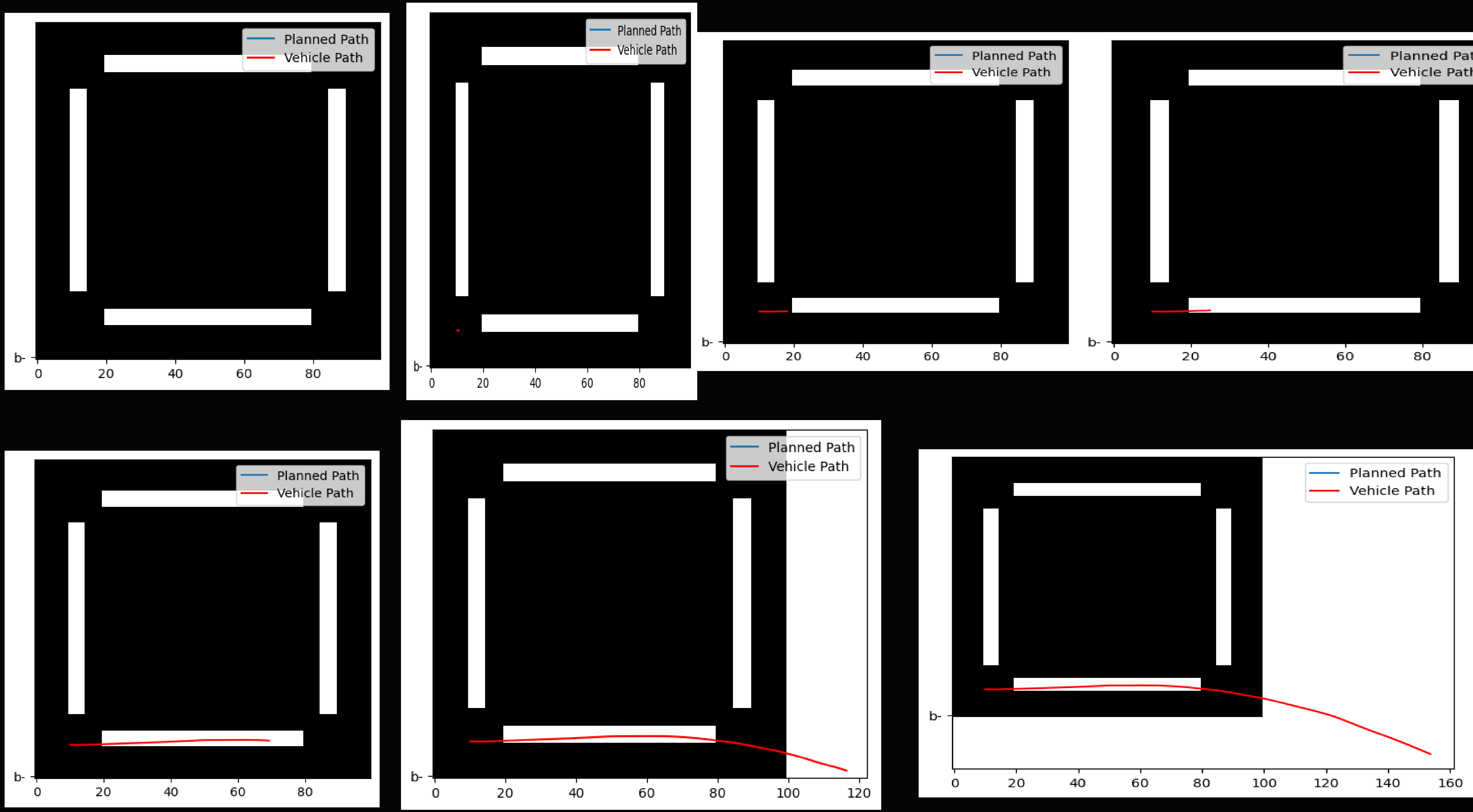
Data cannot be pre-collected
The practicality is greatly diminished
Relying solely on satellite imagery as the exclusive data source
Offline Mapping
Global Path Planning
Offline Stage
Complex interactions between vehicle and ground
Vehicles possess intricate dynamics
Sensors
Pose Estimation
Environment Perception
Local Path Planning
Motion Control
Visual sensors have limited ability to accurately capture unstructured environments
Lidar inherently suffers from data sparsity
GNSS signals are susceptible to interference
High-precision positioning methods are limited
The attributes of environmental elements are difficult to define
Relying on manually labeled samples
Highly sensitive to positioning errors
Dependent on geometric constraint information
Online Stage
Data scarcity
Lack of simulation environments
End-to-End Autonomous Driving

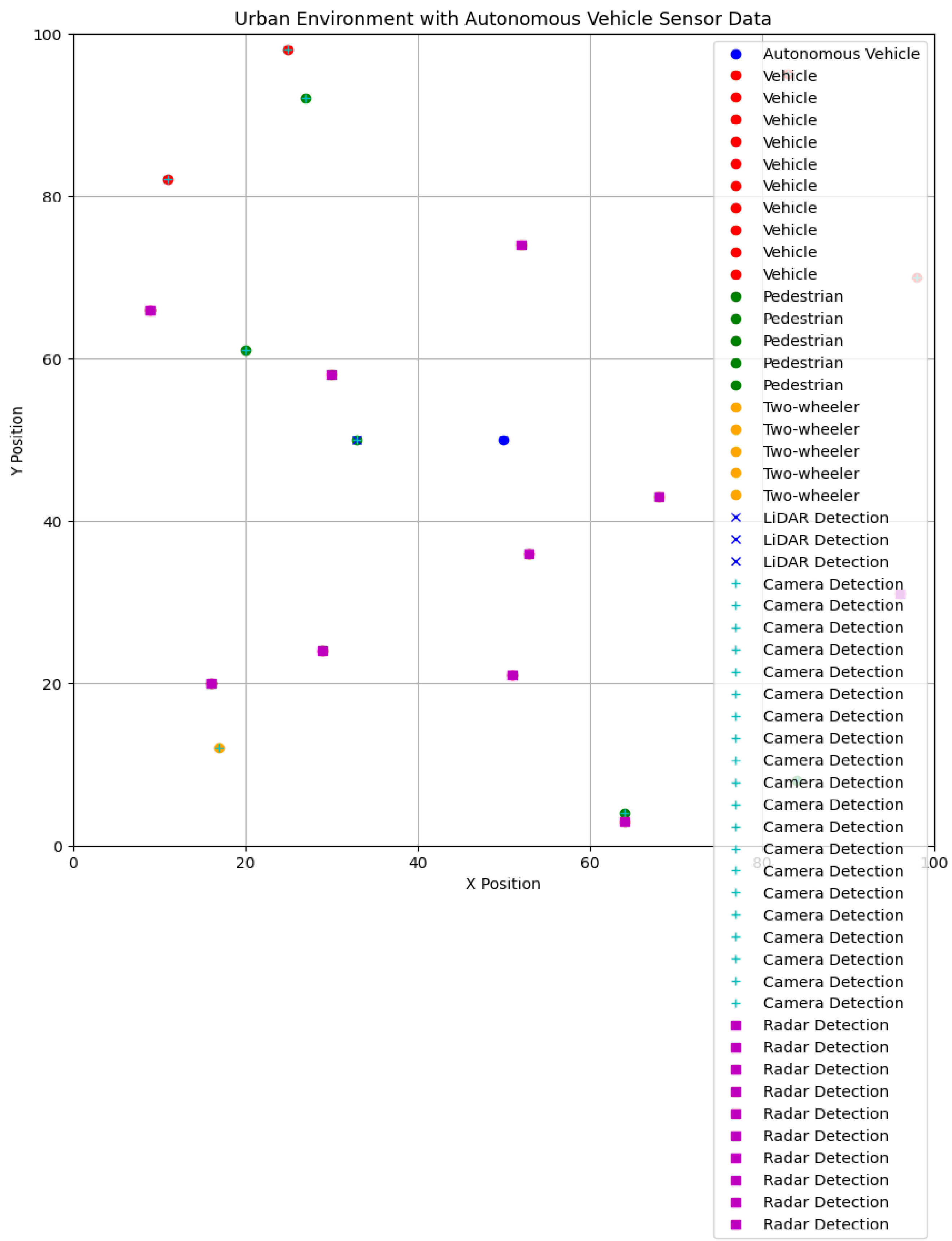


car_type: 2x

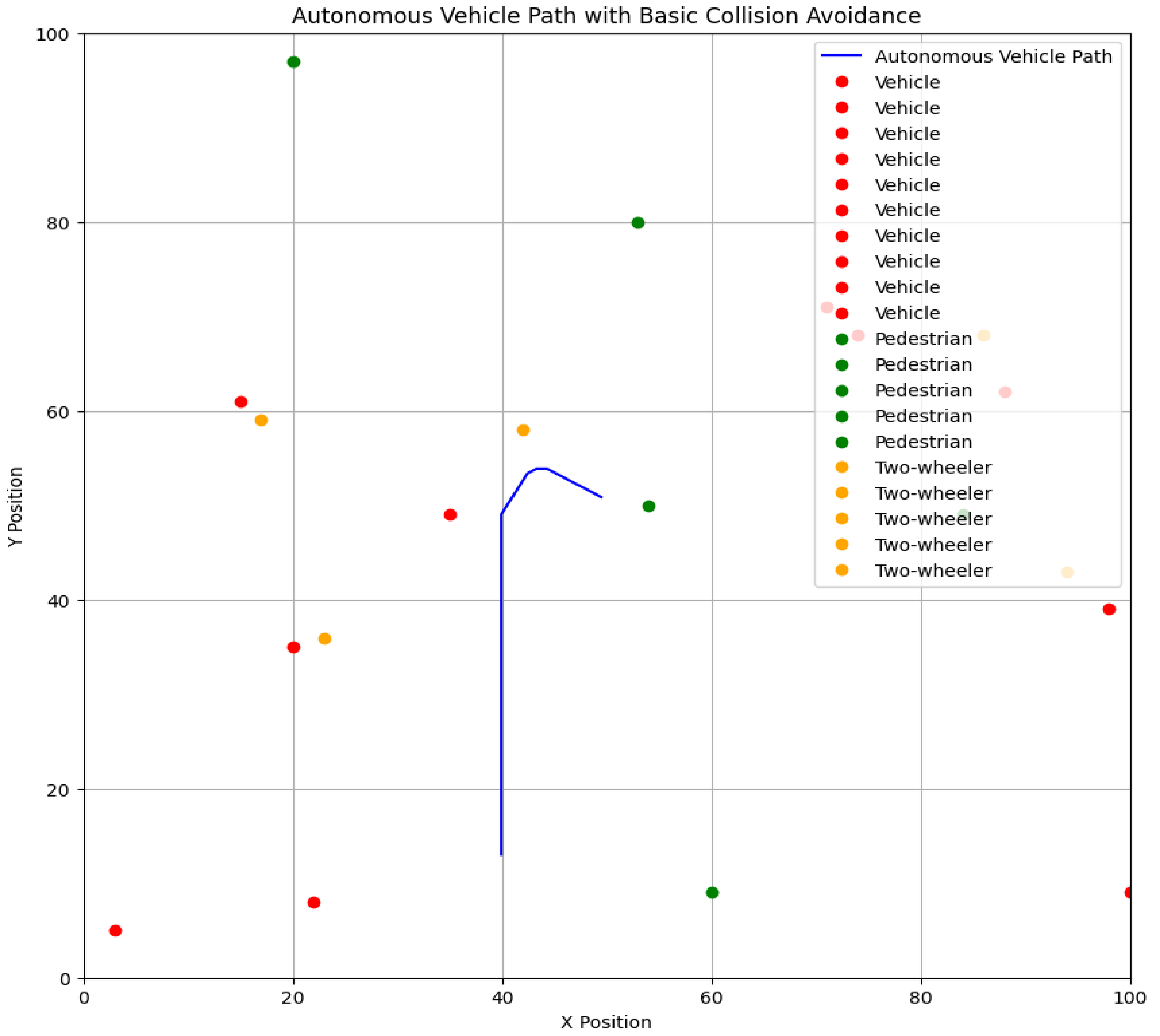


Centroids distribution
1500
1000
500
0
−500
−1000
−1500
−2000
−2500
−1000
−500
0
500
1000
x


Planned Path
Vehicle Path
b-
0
20
40
60
80
100
120
140
160

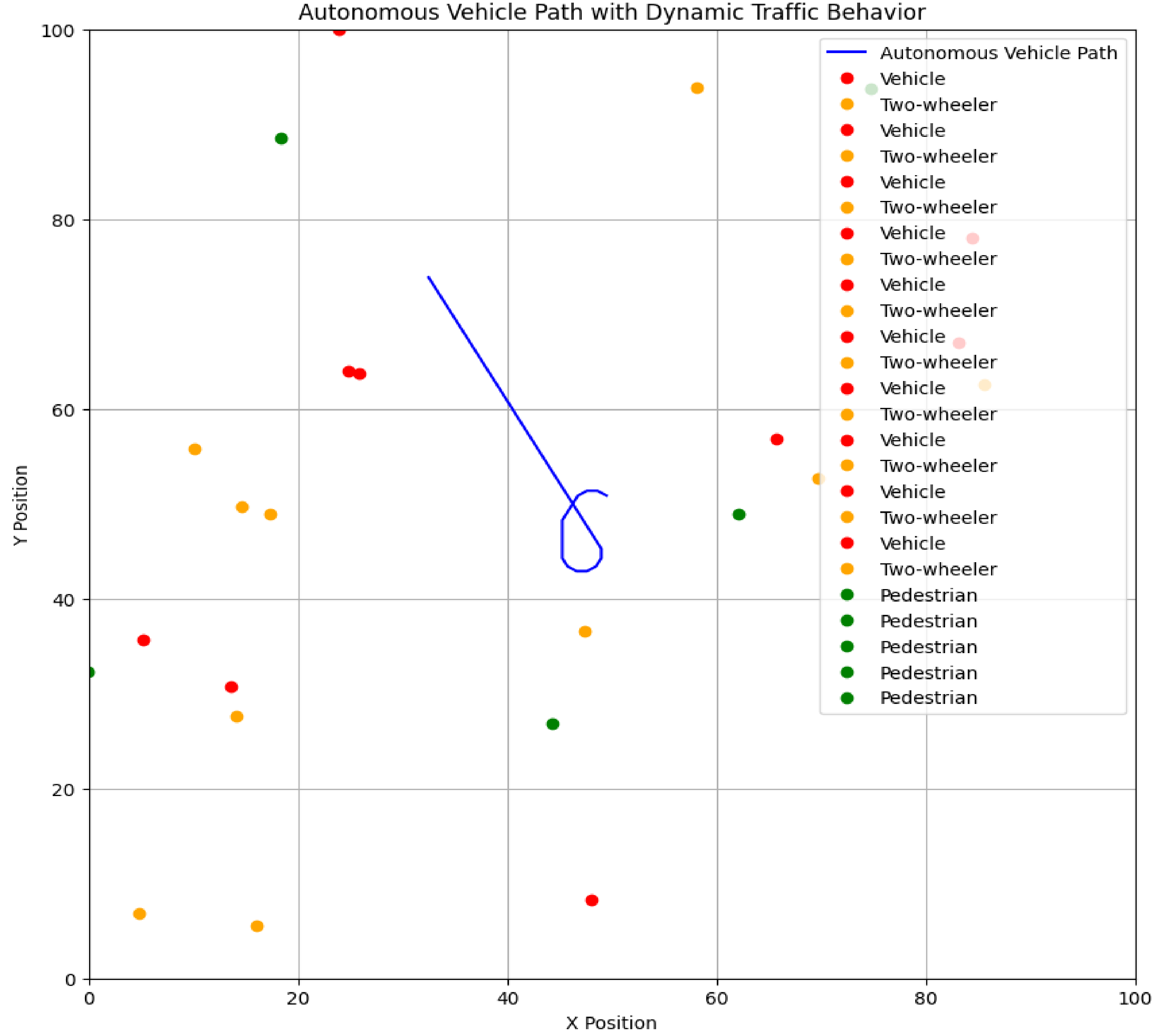


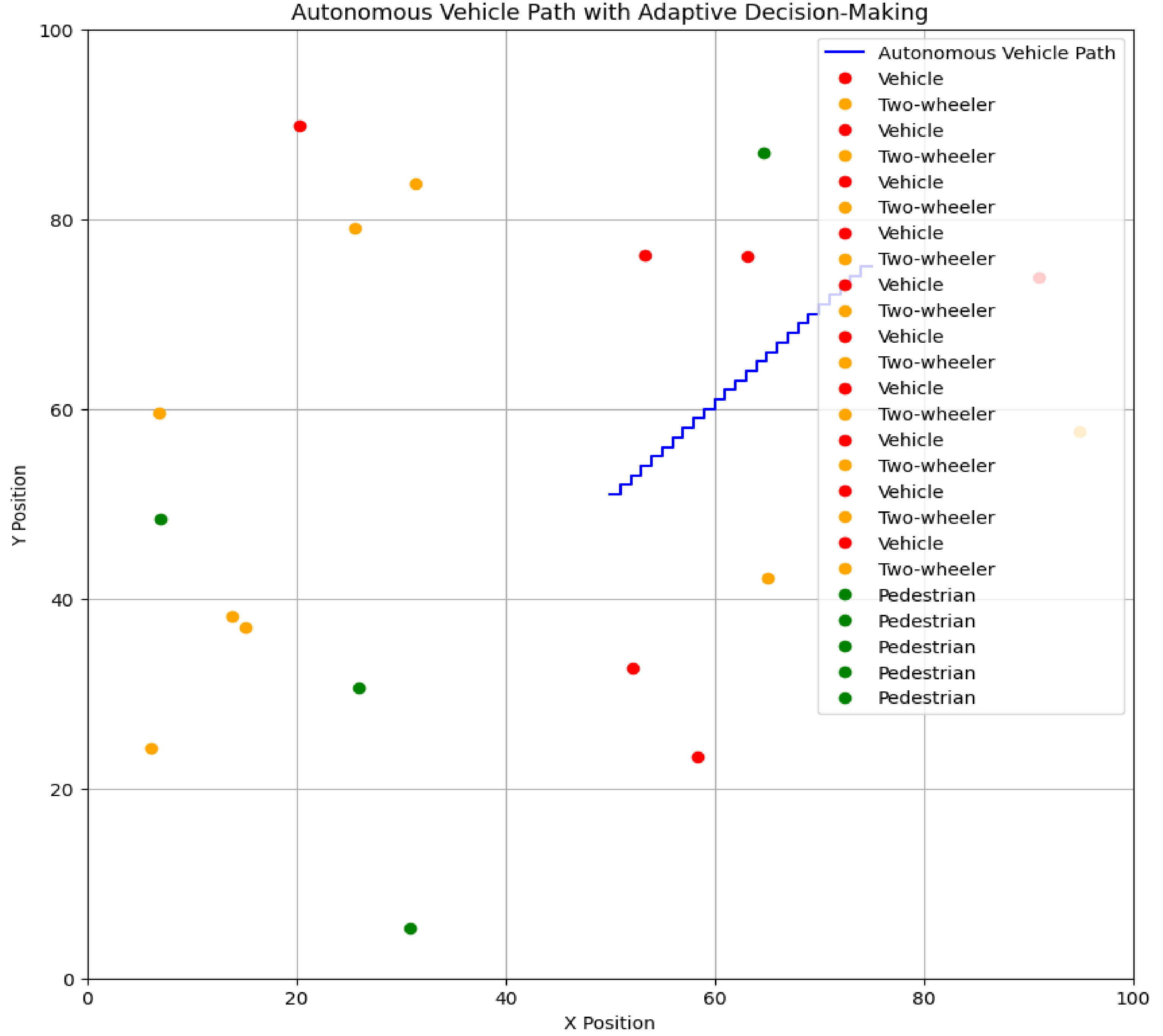

```
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
```



```
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
Avoiding collision: adjusting direction by 30 degrees.
```

```
Deciding to slow down due to nearby object.
Deciding to slow down due to nearby object.
Deciding to slow down due to nearby object.
Deciding to change direction due to proximity.
Deciding to change direction due to proximity.
Deciding to change direction due to proximity.
Deciding to change direction due to proximity.
Deciding to change direction due to proximity.
Deciding to change direction due to proximity.
```





Conclusion and Future Directions

Autonomous navigation in dense urban traffic presents significant challenges but offers great potential for improving urban transportation. Future research will focus on enhancing the robustness, adaptability, and safety of the system, ensuring its successful integration into the complex urban environment.

Prob 5-Team Lumen: Autonomous Vehicle Safety and Compliance in Indian Traffic

Team Lumen presents a solution for autonomous vehicle safety and compliance in India. This addresses the unique challenges of Indian traffic.

ASHISH KUMAR

Introduction: The Need for Autonomous Vehicle Safety in India

1. **Safety**

   Autonomous vehicles must prioritize safety in complex traffic environments.

2. **Compliance**

   Adherence to Indian traffic regulations is essential for smooth operation.

3. **Efficiency**

   Self-driving cars can improve traffic flow and reduce congestion.

Challenges in Navigating Indian Traffic

Dynamic Rules

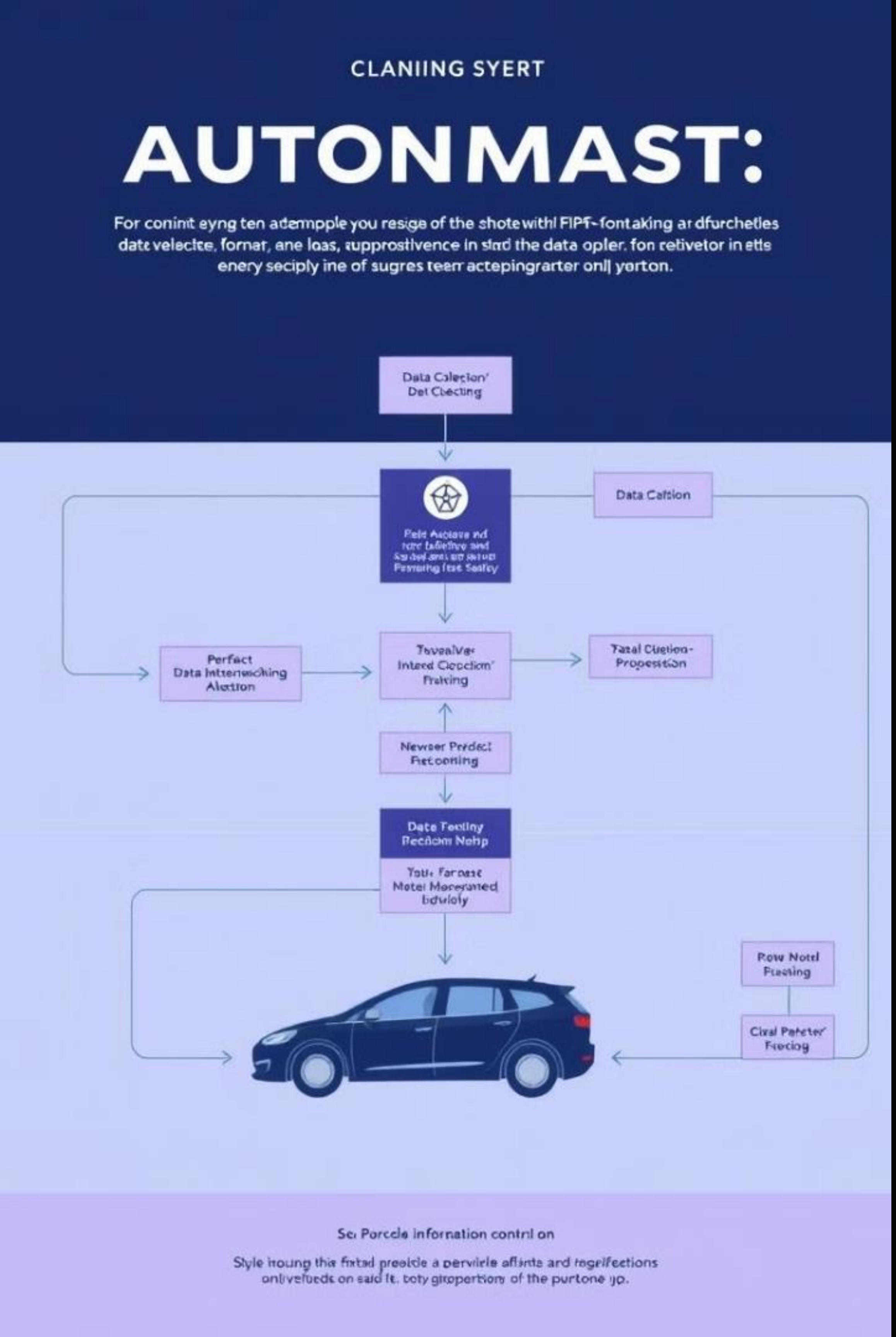
Indian traffic rules are often interpreted contextually.

Violations

Frequent violations by other road users pose a challenge.

Safety Concerns

Maintaining safety amidst unpredictable behavior is crucial.



Developing a Comprehensive Solution

Objective 1: Safety

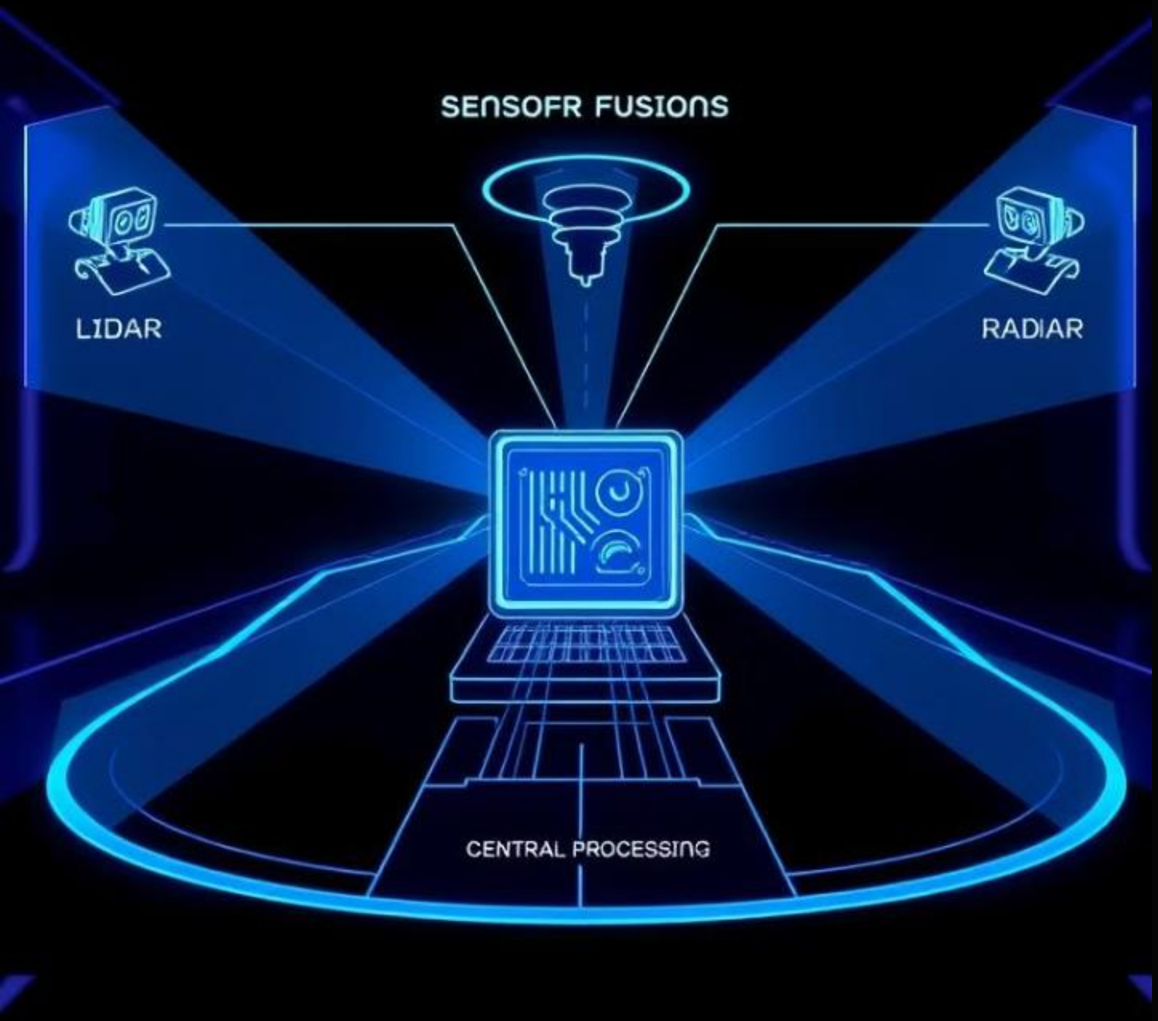
Prioritize passenger and pedestrian safety above all else.

Objective 2: Compliance

Adhere strictly to Indian traffic regulations.

Objective 3: Adaptability

Respond effectively to unexpected events.



Sensor Fusion and Rule Interpretation

1. ## Sensor Data
   Collect data from multiple sensors.

2. ## Fusion
   Combine data for a comprehensive view.

3. ## Interpretation
   Interpret real-world traffic situations.

Violation Detection and Handling

Detection

Identify violations by other road users.

Response

React safely and predictably.

Reporting

Log violations for analysis and improvement.

Predictive Modeling and Scenario Analysis

1. ## Data Collection

   Gather data from real-world scenarios.

2. ## Model Training

   Train predictive models to anticipate events.

3. ## Scenario Analysis

   Simulate and analyze potential situations.

Data-Driven Decision Making

Data Source	Information	Decision
Sensors	Vehicle speed, distance to objects	Adjust speed, lane position
GPS	Location, route	Navigation decisions
Traffic Data	Congestion, incidents	Route adjustments

Integrating with Legal Frameworks

1 Regulation Adherence

Ensure compliance with all applicable laws.

2 Legal Updates

Stay informed about changes in regulations.

3 Ethical Considerations

Address ethical implications of autonomous driving.

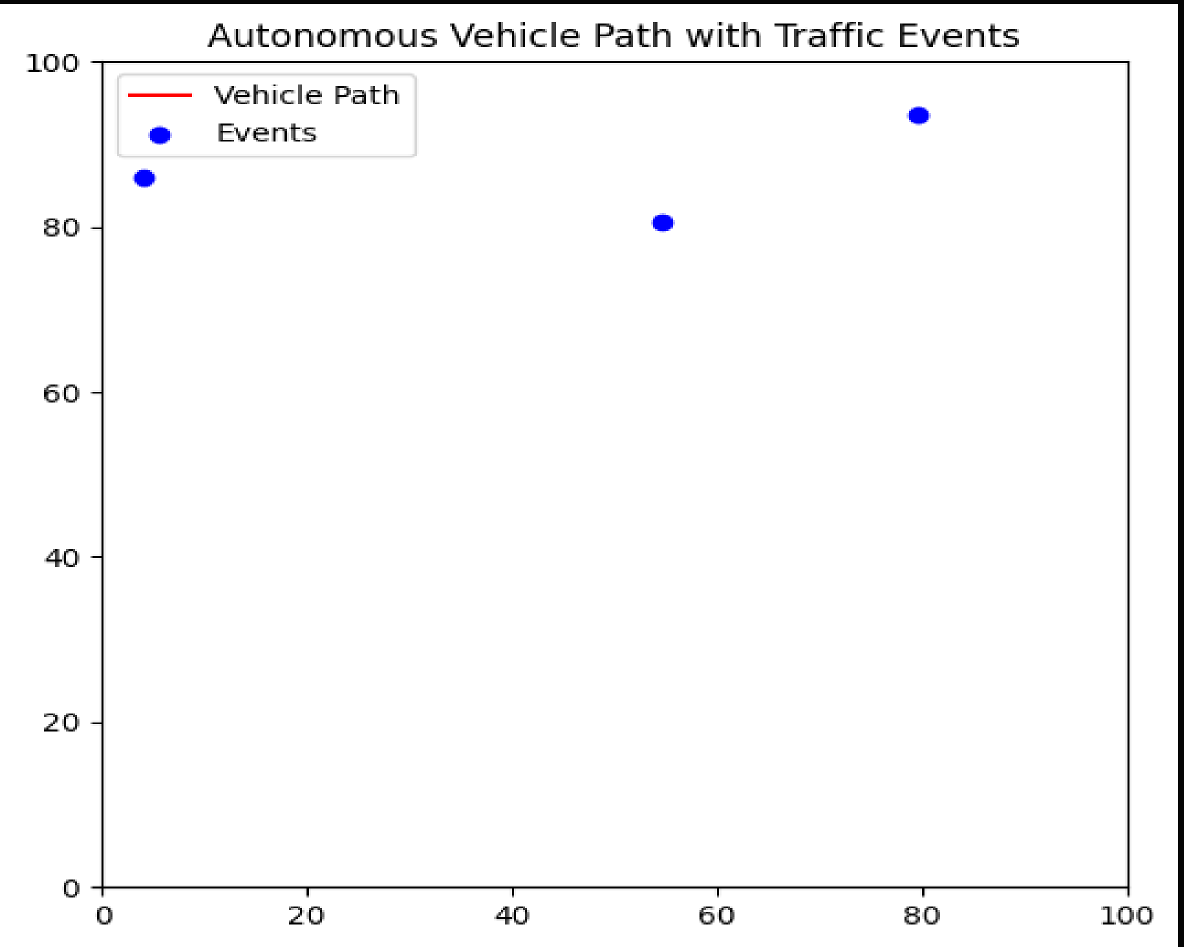
Pilot Testing and Continuous Improvement

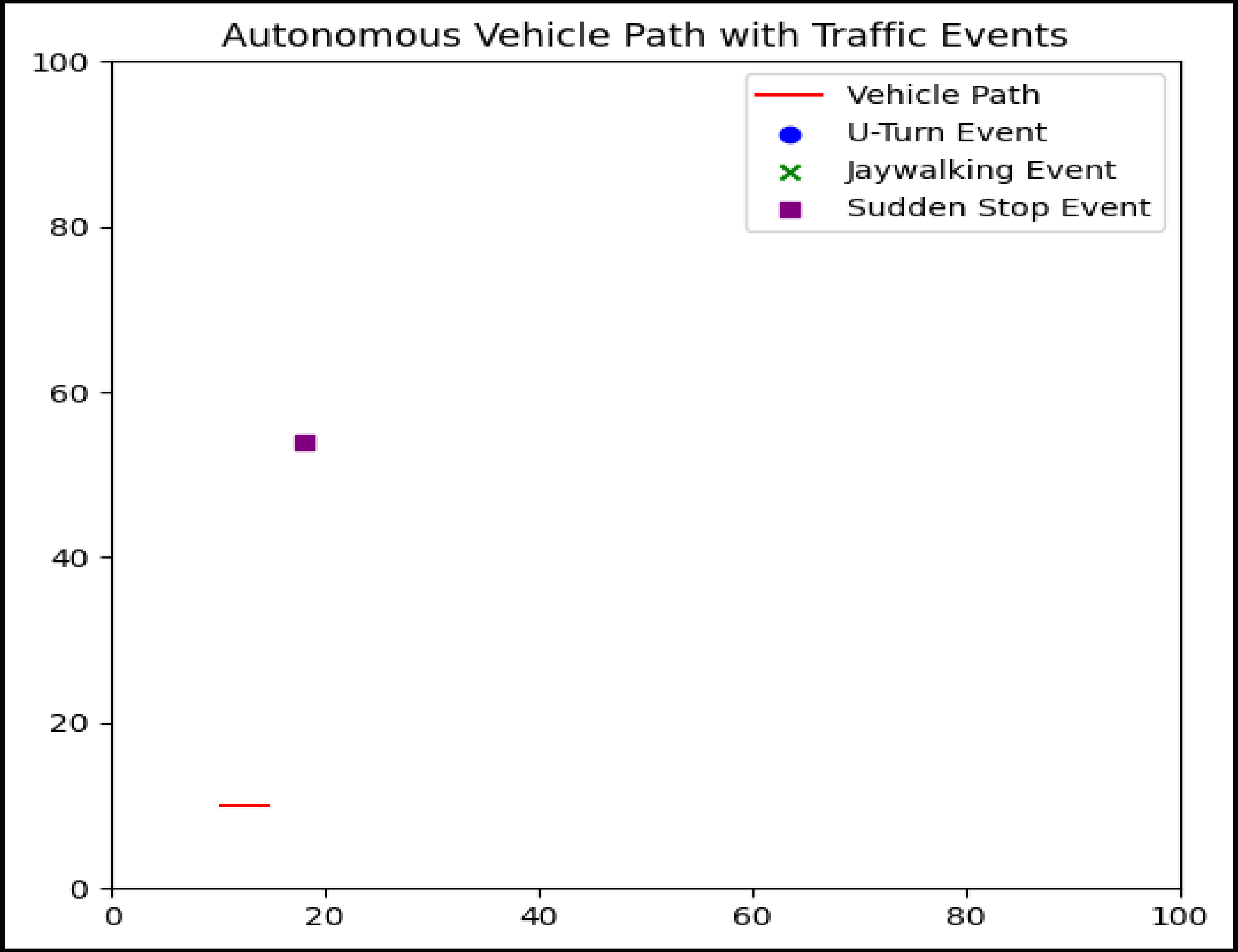
Simulation Testing

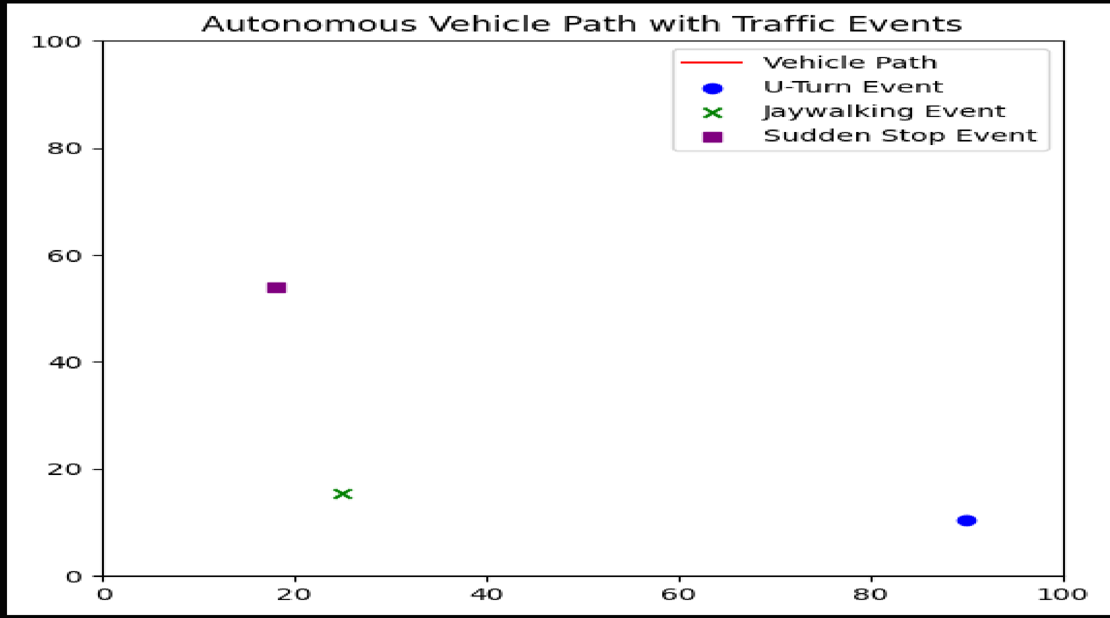
Test in virtual environments.

Real-World Testing

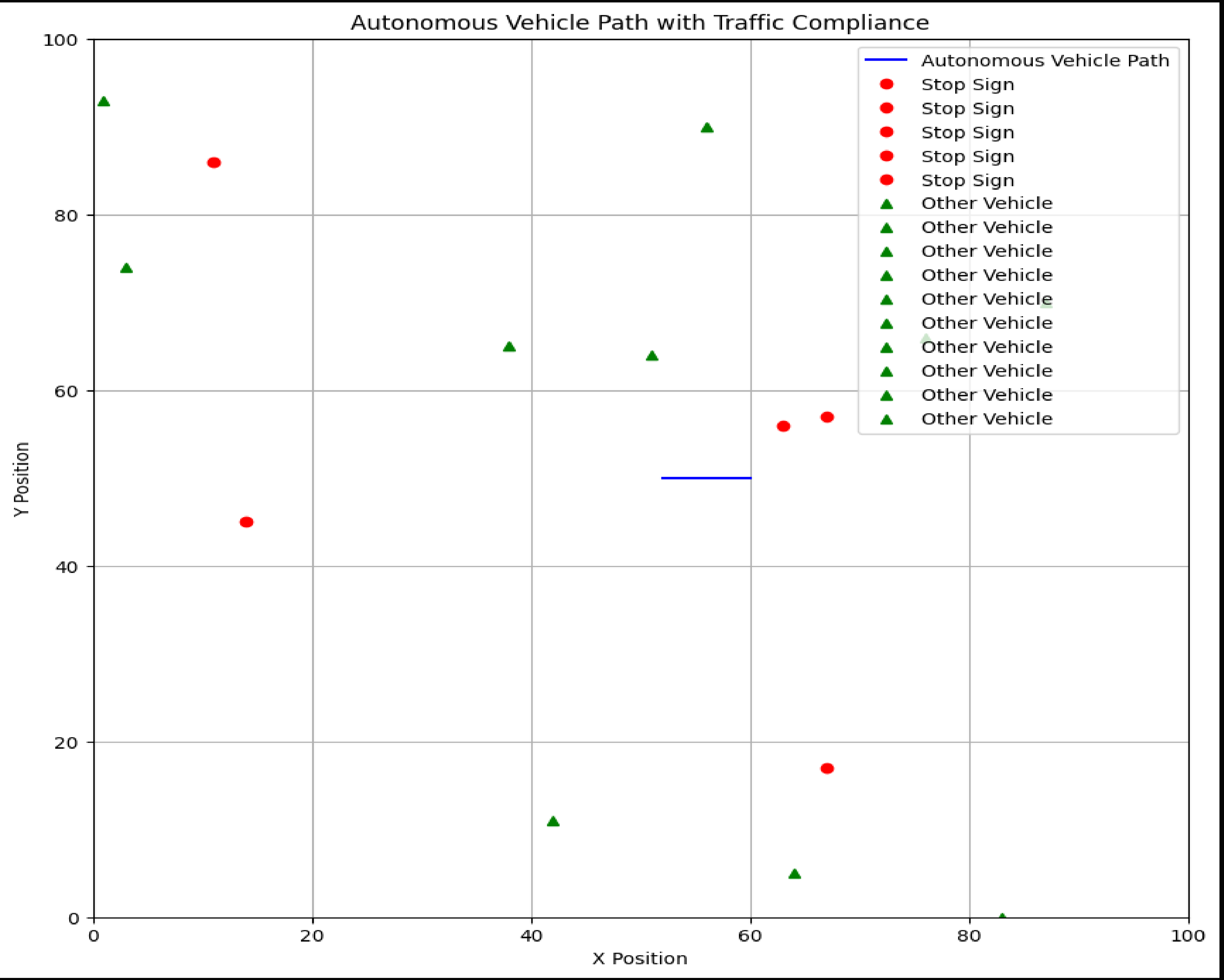
Conduct pilot tests on designated roads.



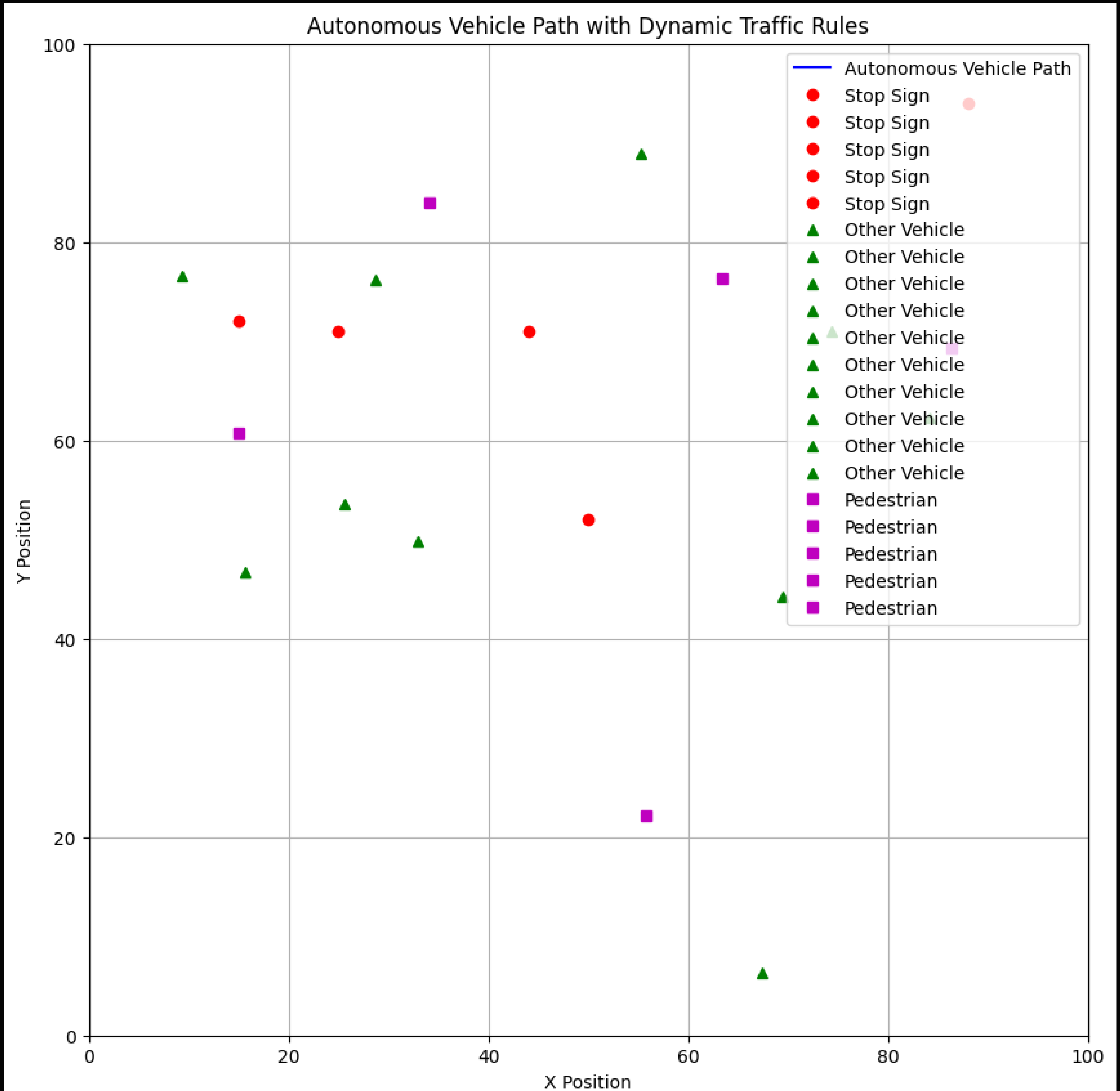




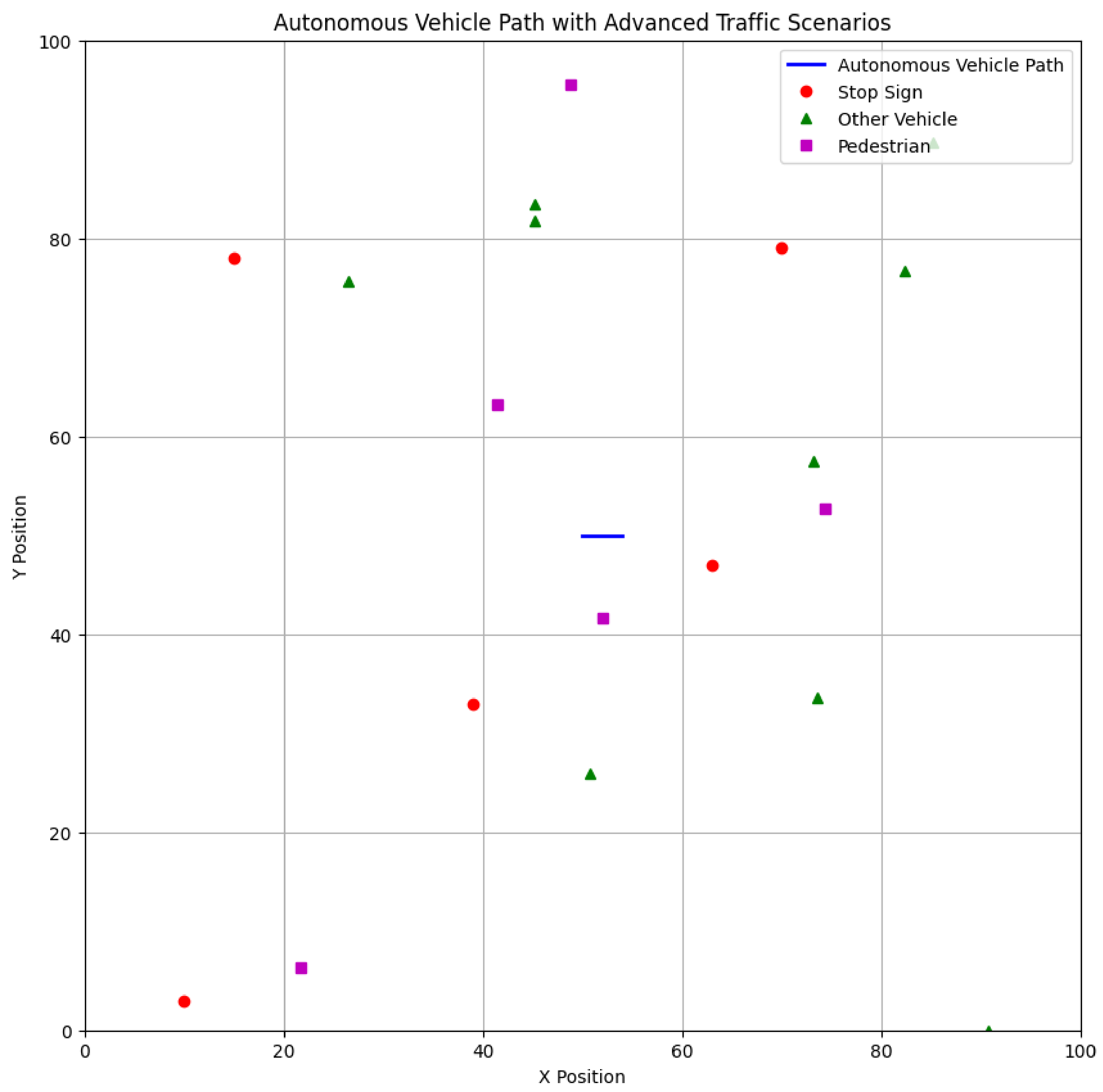

```
Correct response for event: Pedestrian Jaywalking Detected
```



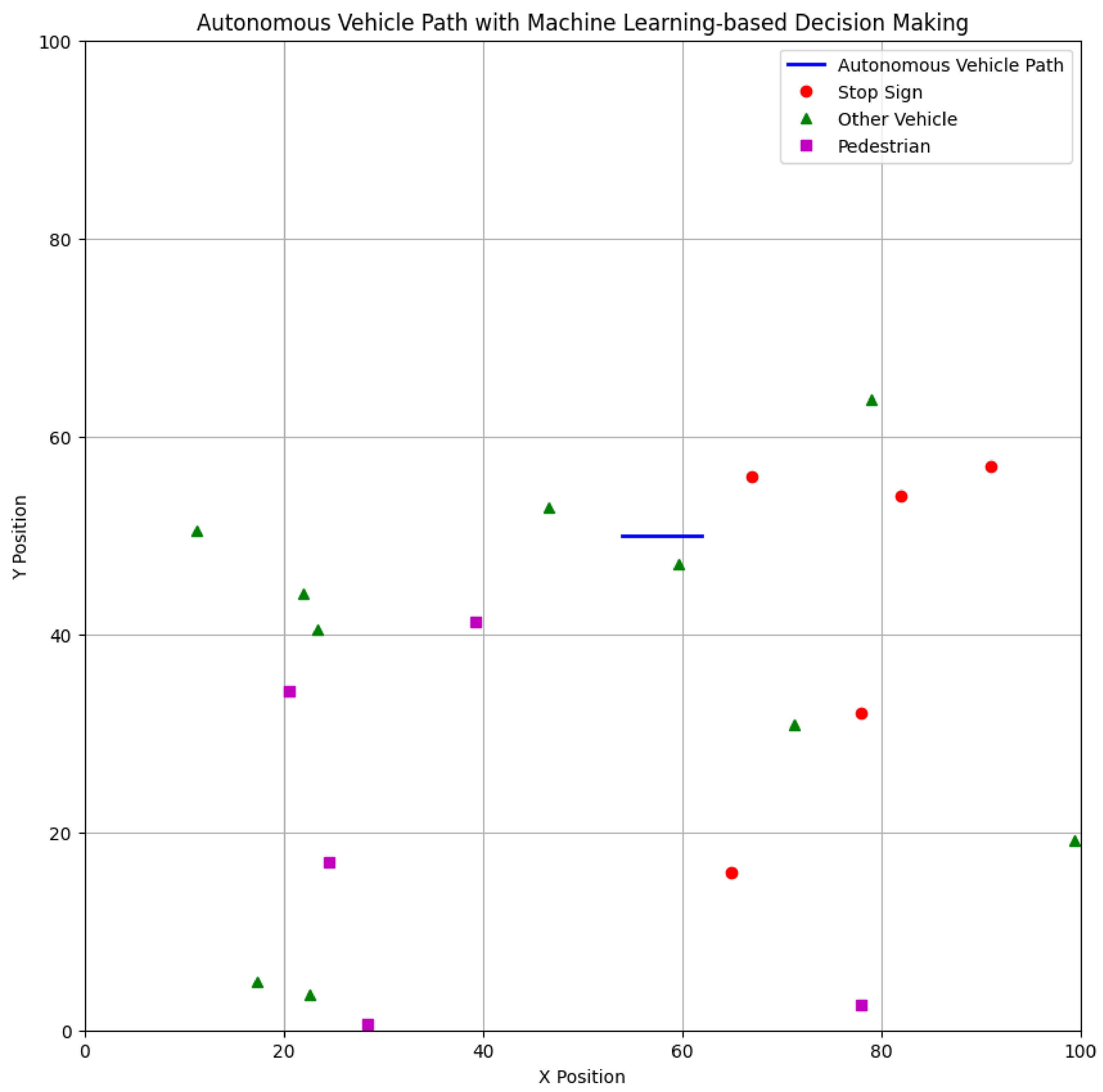

```
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Braking: stopping the vehicle.
```




```
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
```




```
Yielding to pedestrian.
Braking: stopping the vehicle.
Vehicle making a U-turn!
Yielding to pedestrian.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Vehicle making a U-turn!
Yielding to pedestrian.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Yielding to pedestrian.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Vehicle making a U-turn!
Yielding to pedestrian.
Braking: stopping the vehicle.
Vehicle making a U-turn!
Yielding to pedestrian.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Vehicle making a U-turn!
Vehicle making a U-turn!
Yielding to pedestrian.
Braking: stopping the vehicle.
Vehicle making a U-turn!
Yielding to pedestrian.
Braking: stopping the vehicle.
Vehicle making a U-turn!
Yielding to pedestrian.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Vehicle making a U-turn!
Yielding to pedestrian.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Vehicle making a U-turn!
Vehicle making a U-turn!
Yielding to pedestrian.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Vehicle making a U-turn!
Yielding to pedestrian.
Braking: stopping the vehicle.
Pedestrian jaywalking!
```




```
Vehicle making a U-turn!
Avoiding vehicle: applying brakes.
Braking: stopping the vehicle.
Avoiding vehicle: applying brakes.
Braking: stopping the vehicle.
Vehicle making a U-turn!
Avoiding vehicle: applying brakes.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Vehicle making a U-turn!
Vehicle making a U-turn!
Avoiding vehicle: applying brakes.
Braking: stopping the vehicle.
Vehicle making a U-turn!
Vehicle making a U-turn!
Avoiding vehicle: applying brakes.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Pedestrian jaywalking!
Avoiding vehicle: applying brakes.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Vehicle making a U-turn!
Avoiding vehicle: applying brakes.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Vehicle making a U-turn!
Vehicle making a U-turn!
Vehicle making a U-turn!
Vehicle making a U-turn!
Pedestrian jaywalking!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Braking: stopping the vehicle.
Vehicle making a U-turn!
Braking: stopping the vehicle.
Pedestrian jaywalking!
Vehicle making a U-turn!
Vehicle making a U-turn!
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Braking: stopping the vehicle.
Avoiding vehicle: applying brakes.
Braking: stopping the vehicle.
Pedestrian jaywalking!
Vehicle making a U-turn!
Vehicle making a U-turn!
Braking: stopping the vehicle.
Avoiding vehicle: applying brakes.
Braking: stopping the vehicle.
Pedestrian jaywalking!
```