Case Study: Spoofing Attack on Autonomous Vehicles.

Shihab Ud Doula | Matriculation No. 2190679.

Advisor: Prof. Dr.-Ing. João Paulo Javidi da Costa

HOCHSCHULE
HAMM-LIPPSTADT

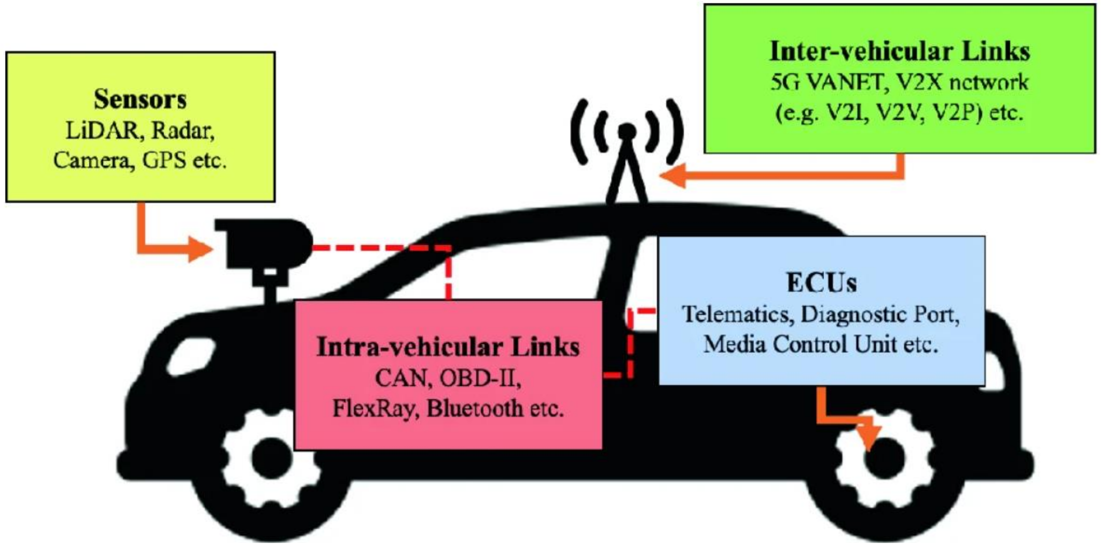
Understanding Spoofing in Autonomous Vehicles

Background: Autonomous vehicles (AVs) rely on GPS, time synchronization, and sensors for safe navigation. However, these dependencies make them vulnerable to spoofing attacks.

Problem Statement: Spoofing attacks can misguide navigation, disrupt system synchronization, and falsify environmental data, posing significant risks to safety.

Objectives:

- Simulate spoofing attacks (GPS, time, and LiDAR)
- Assess their impacts.
- Propose mitigation strategies.

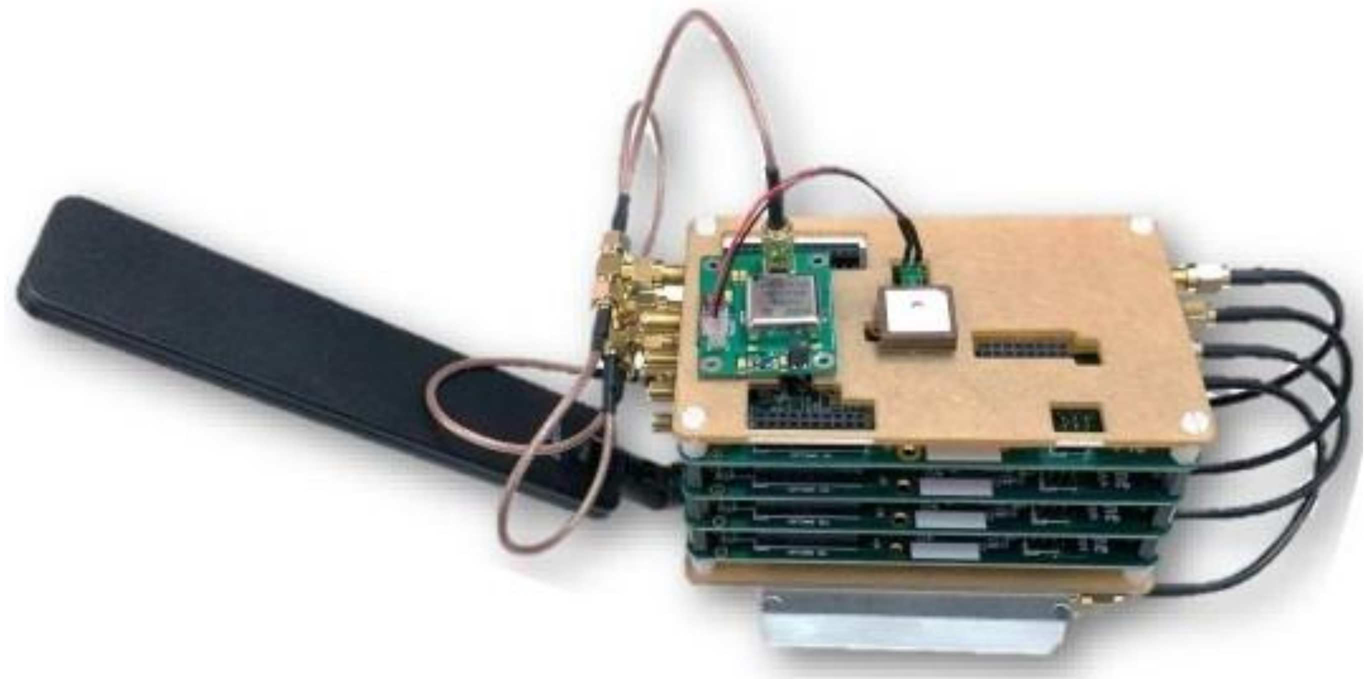


Research on Spoofing Attacks

- **GPS Spoofing (Regulus Cyber, 2019):** Misguided navigation with fake GPS signals.

- **Time Spoofing (Keen Security Lab, 2019):** System desynchronization causing critical errors.

- **LiDAR Spoofing (Cao et al., 2024):** False obstacle detection via manipulated sensor inputs.



Spoofing Signals from 4 Constellations at Once – Regulus Cyber

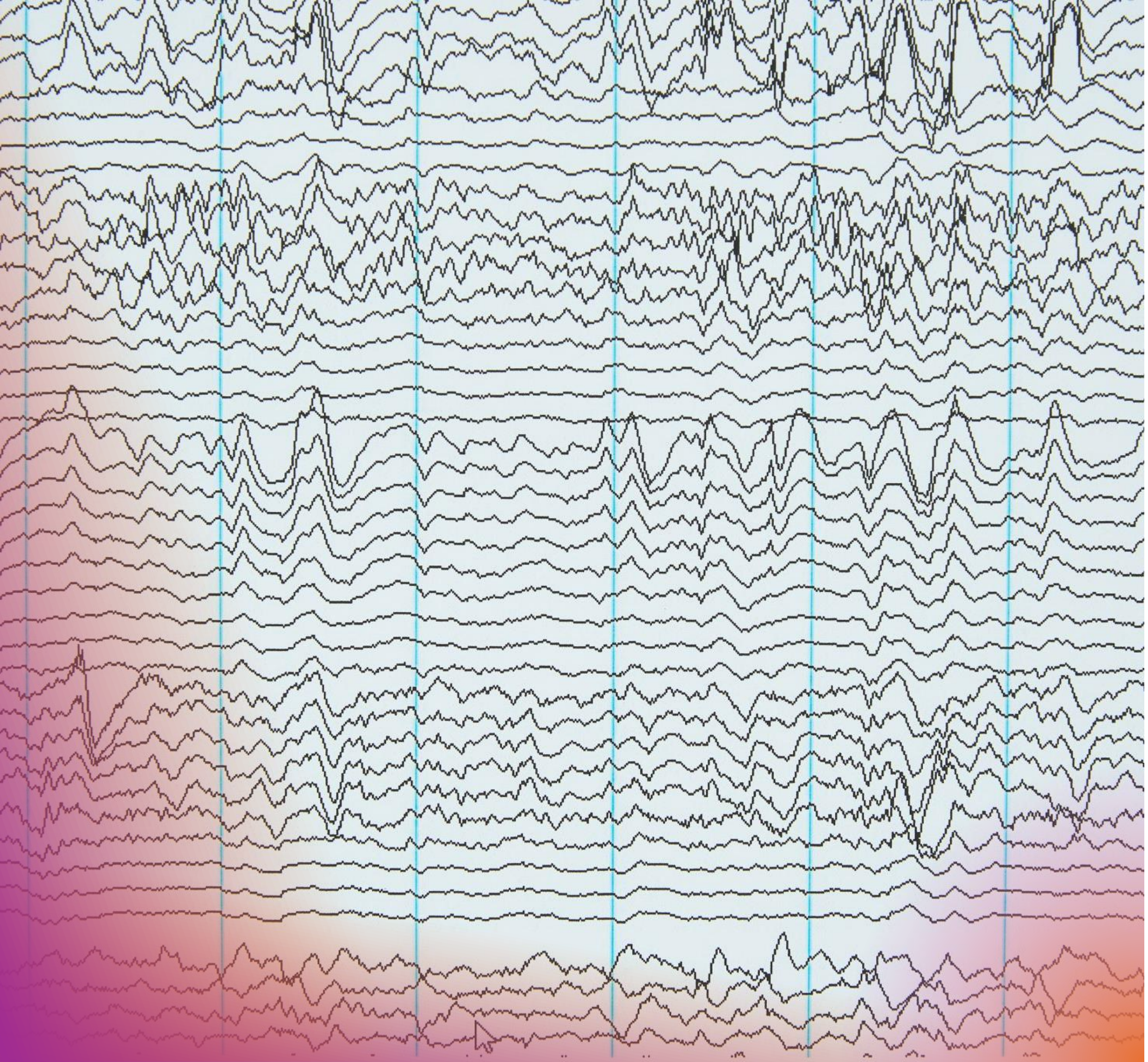
Simulation Design and Tools

Tools:

- MATLAB Live Script: Interactive computation and visualization.
- System: Windows 11, Intel i5, 16GB RAM.

Setup:

- **Simulation duration**: 10 seconds with 1000 time points.
- **Initialization**: Original signal defined as position (GPS), time (Time), or distance (LiDAR).
- **Spoofing start points**: 20–30% of the timeline, depending on the simulation.



Key Parameters:

GPS Spoofing:

- Straight-line path with amplitude, frequency, and noise variations.

Time Spoofing:

- Linear time signal with deviations

LiDAR Spoofing:

- Constant distance of 20m with added linear drift and sinusoidal noise.

Code Setup for All simulation

```
%Setup for All the Simulations

% Time Vector (Shared Setup)
t = linspace(0, 10, 1000);  % Time vector for 10 seconds

% GPS Spoofing Setup
original_position = t;      % Original GPS signal path
spoofing_start_index = round(length(t) * 0.2); % Start of spoofing

% Time Spoofing Setup
original_time = t;          % Original time signal (linear progression)

% LiDAR Spoofing Setup
true_obstacle_distance = 20; % Fixed obstacle distance (meters)
spoofing_start_index = round(length(t) * 0.3); % Start of spoofing

```

A GPS live script where parameter are defined

```
% GPS Spoofing Simulation with Real-Time Animation

% Parameters and Setup
t = linspace(0, 10, 1000);  % Time vector for 10 seconds
original_position = t;  % Original GPS signal path (straight line)
spoofed_position = original_position;  % Initialize spoofed position

% Spoofing starts at 20% of the timeline with increased noise and deviation
spoofing_start_index = round(length(t) * 0.2);
amplitude_variation = 1.0 + 0.3 * sin(0.1 * t(spoofing_start_index:end));  % Increased amplitude
frequency_variation = 1.0 + 0.1 * cos(0.05 * t(spoofing_start_index:end)); % Increased frequency
random_walk = cumsum(0.05 * randn(1, length(t) - spoofing_start_index + 1)); % Random walk
gaussian_noise = 0.2 * randn(1, length(t) - spoofing_start_index + 1); % Noise

% Spoofed position calculation
spoofed_position(spoofing_start_index:end) = original_position(spoofing_start_index:end) + ...
    amplitude_variation .* sin(frequency_variation .* t(spoofing_start_index:end)) + ...
    random_walk + gaussian_noise;
```

GPS Spoofing Simulation and Results

- **Overview:**
- Simulated GPS spoofing to observe its impact on position, velocity, and acceleration.
- Spoofing starts at 2 seconds, introducing deviations through amplitude variations, frequency changes, random walk, and Gaussian noise.
- Results highlight how spoofing disrupts the vehicle's navigation system by creating noticeable deviations in the trajectory.

Key Components

- Original Path: Straight-line trajectory (original_position = t).

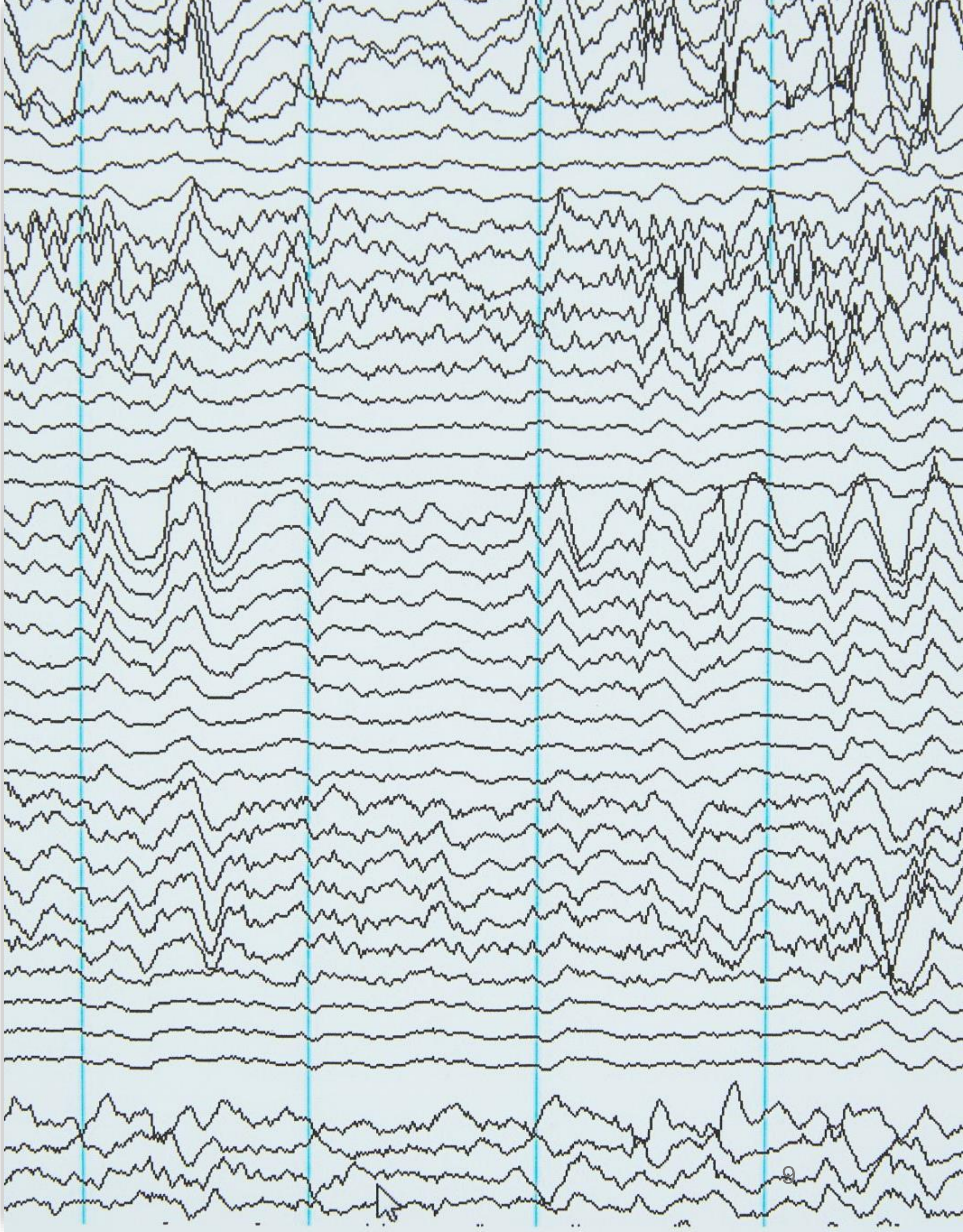
Spoofing Elements:

- Amplitude variation (1.0 + 0.3 * sin(0.1 * t)): Adds smooth oscillations.
- Frequency variation (1.0 + 0.1 * cos(0.05 * t)): Adds periodic deviations.
- Random walk (cumsum(0.05 * randn(...))): Introduces gradual drift.
- Gaussian noise (0.2 * randn(...)): Simulates real-world randomness.

Analysed Metrics:

Position: Deviation of spoofed path from original.

Velocity and Acceleration: Effects on motion dynamics.



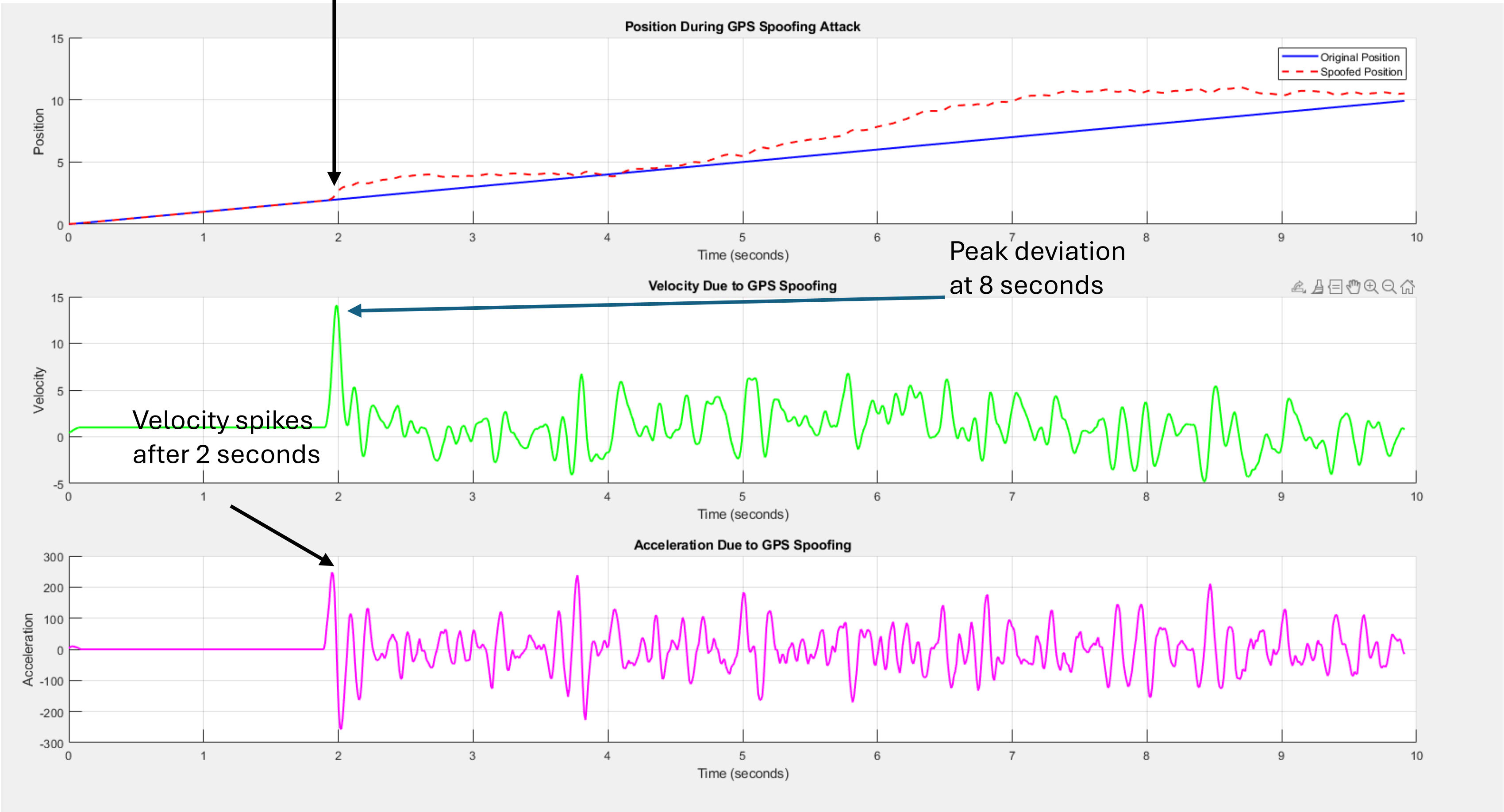
Main script for calculation

```
% Spoofed position calculation with enhanced dynamic components
spoofed_position(spoofing_start_index:end) = original_position(spoofing_start_index:end) + ...
    amplitude_variation .* sin(frequency_variation .* t(spoofing_start_index:end)) + ...
    random_walk + gaussian_noise;
```

Key Logic for Spoofed Path Generation

Results for GPS spoofing

Spoofing starts
at 2 seconds



Peak deviation
at 8 seconds

Velocity spikes
after 2 seconds

Time Spoofing Simulation and Results

Overview:

- Simulated **time spoofing attack** to observe its impact on GPS signals.
- Spoofing starts at **20% of the timeline**, introducing:
  - **Linear drift**: Mimics gradual timing errors.
  - **Sinusoidal oscillations**: Adds periodic disturbances.
  - **Random noise**: Simulates real-world signal disruptions.
- Results highlight deviations in time signal, measurable offsets, and disturbances in signal stability.

Spoofing elements:

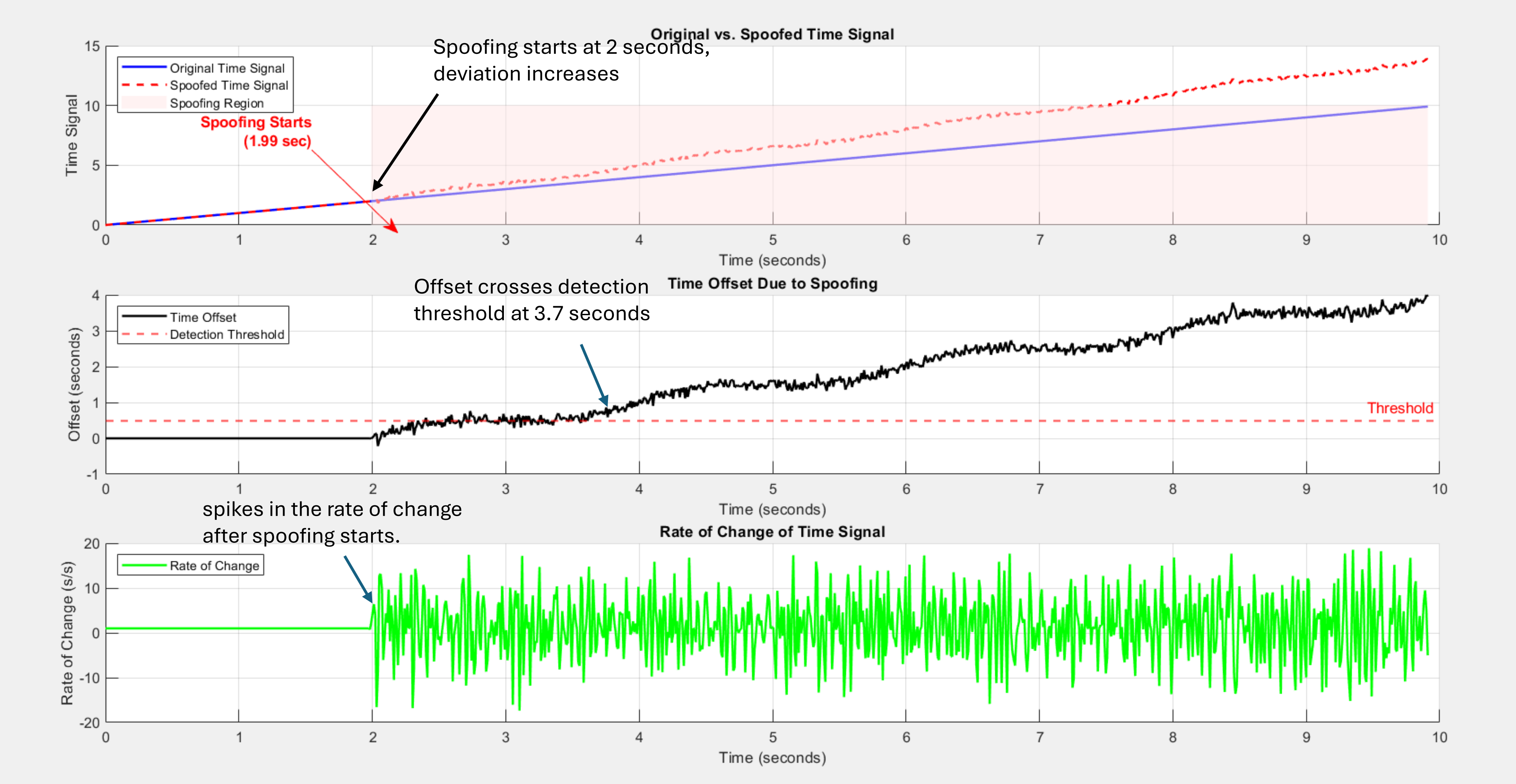
- Linear Drift: Gradual time shifts using:0.5 * (t - t_start)
- **Sinusoidal Disturbance**: Periodic variations modeled as: 0.2 * sin(2 * pi * 0.5 * t)
- Gaussian Noise: Adds random fluctuations using:0.1 * randn(...)

Main script for calculation

```
% Spoofed Time Signal
spoofed_time = original_time; % Initialize spoofed time
spoofed_time(spoofing_start_index:end) = original_time(spoofing_start_index:end) + ...
    time_deviation + sinusoidal_disturbance + gaussian_noise;
```

Demonstrates how drift, oscillations, and noise are combined to simulate the spoofed time signal.

Results for time spoofing

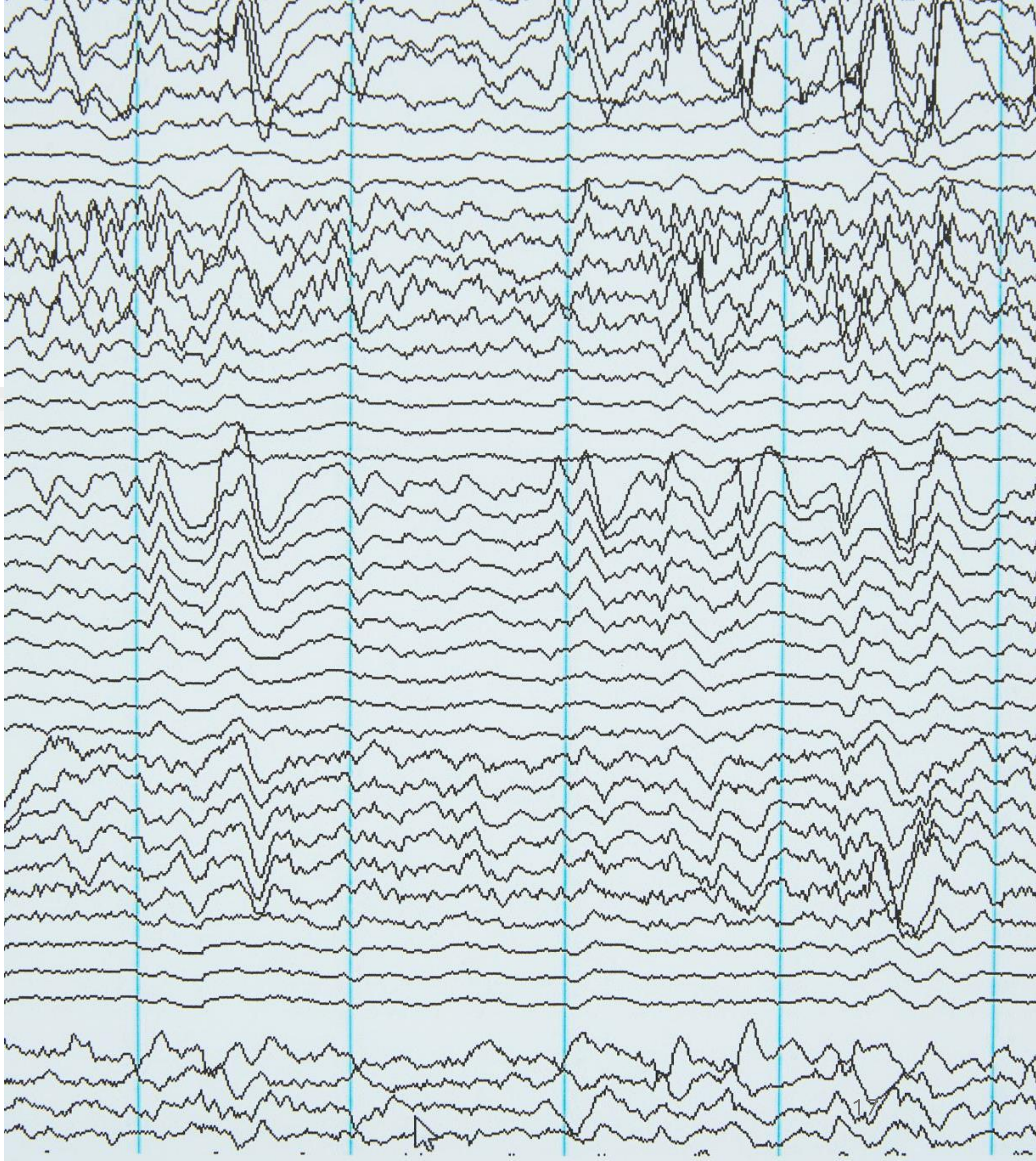


Sensor Spoofing Simulation and Results

- **Overview**:
- Simulated sensor spoofing attack on LiDAR distance measurements to analyze its impact.
- Spoofing begins at 30% of the timeline and ends at 60%, introducing:
  - **Linear drift**: Gradual increase in distance readings.
  - **Sinusoidal disturbances**: Mimics periodic fluctuations in distance.
  - **Gaussian noise**: Adds random variations to the data.
- Results highlight deviations in measured distances and significant impacts on sensor accuracy.

Sensor Spoofing Elemets:

- Linear Drift: Gradual shift using:0.5 * (t - t_start)
- Sinusoidal Disturbance: Periodic fluctuations modeled as:2 * sin(2 * pi * 0.2 * t)
- Gaussian Noise: Simulates randomness with:0.5 * randn(...)

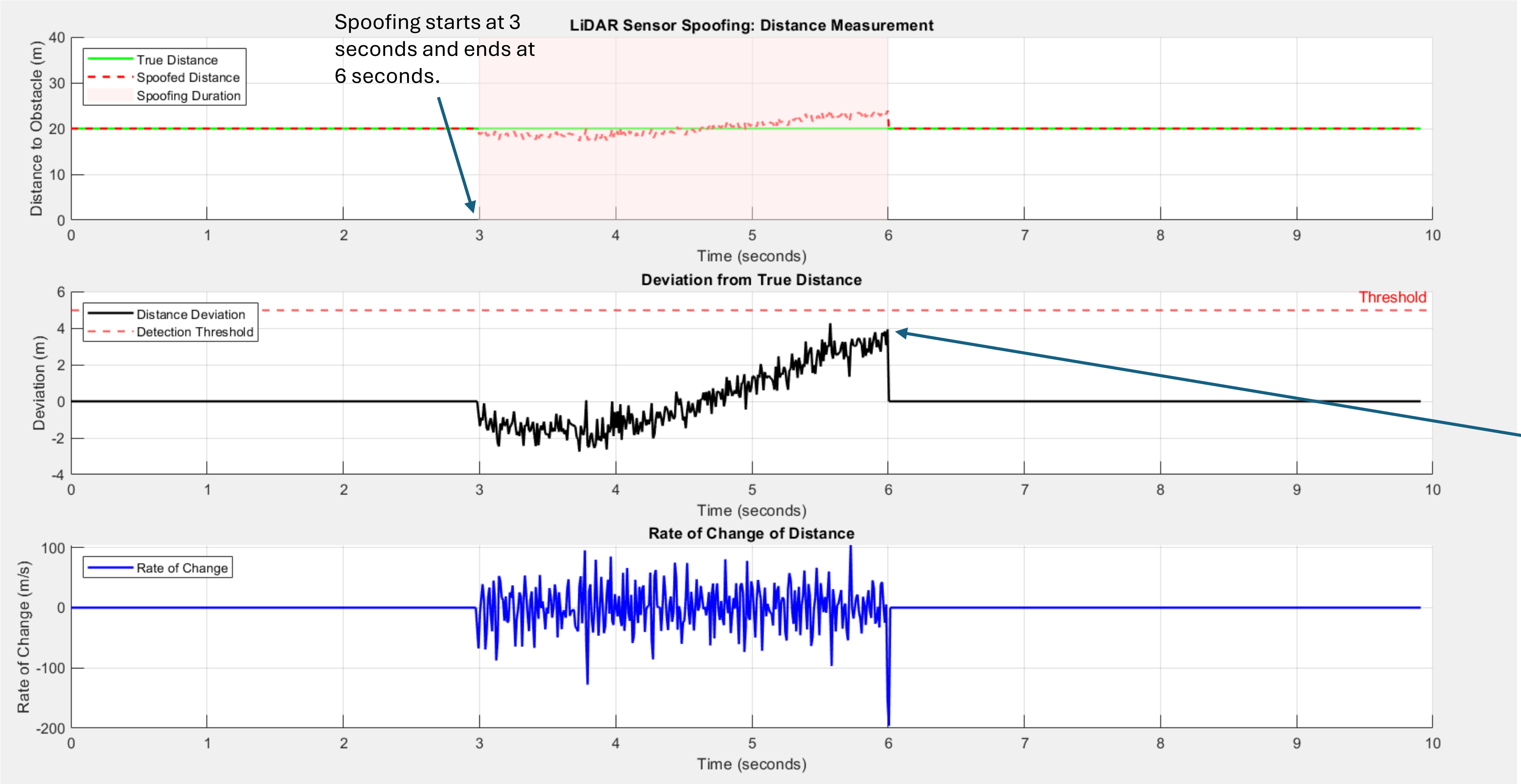


Main script for calculation

```
% Apply Spoofing
spoofed_distance(spoofing_start_index:spoofing_end_index) = true_obstacle_distance + ...
    linear_drift + sinusoidal_disturbance + gaussian_noise;
```

This code shows how the spoofing elements (drift, oscillations, noise) are combined to create a spoofed sensor reading.

Results for sensor spoofing



Although this does not change major effect above the threshold.

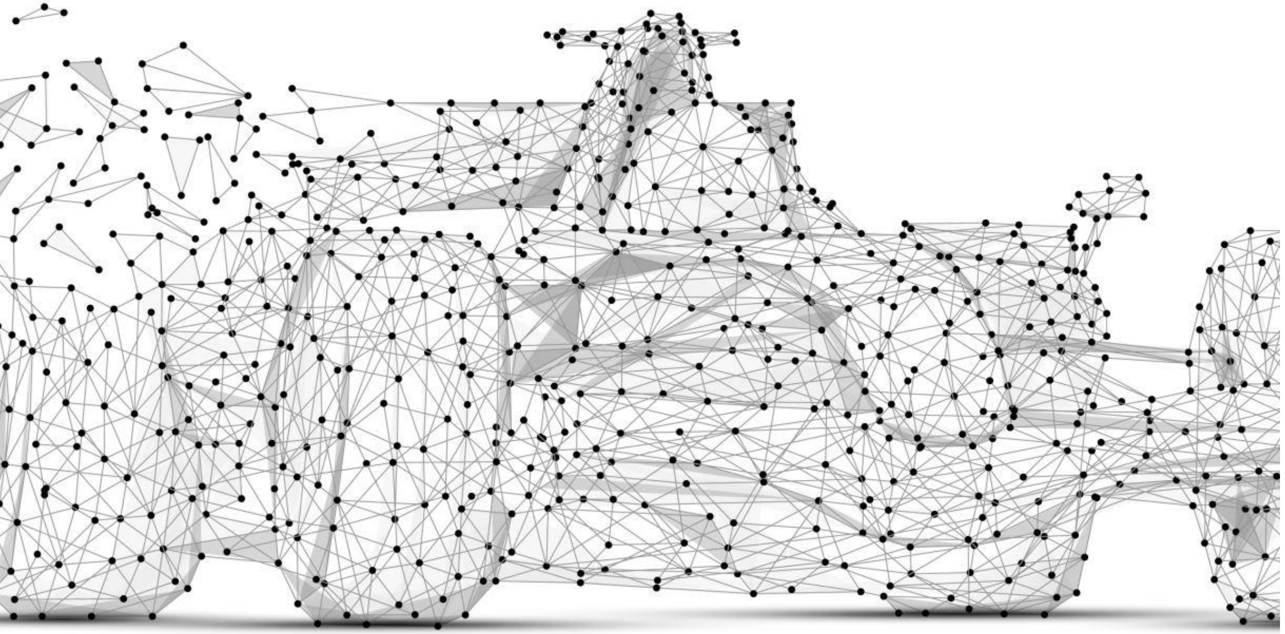
Comparison of Spoofing Types

- **Process:**
- Simulated each spoofing type using unique dynamics.
- Measured deviations, rate of change, and threshold crossing.

Metrics Evaluated:

- Maximum Deviation (quantifies error severity).
- Rate of Change (measures signal stability).
- Threshold Crossing (detectability).

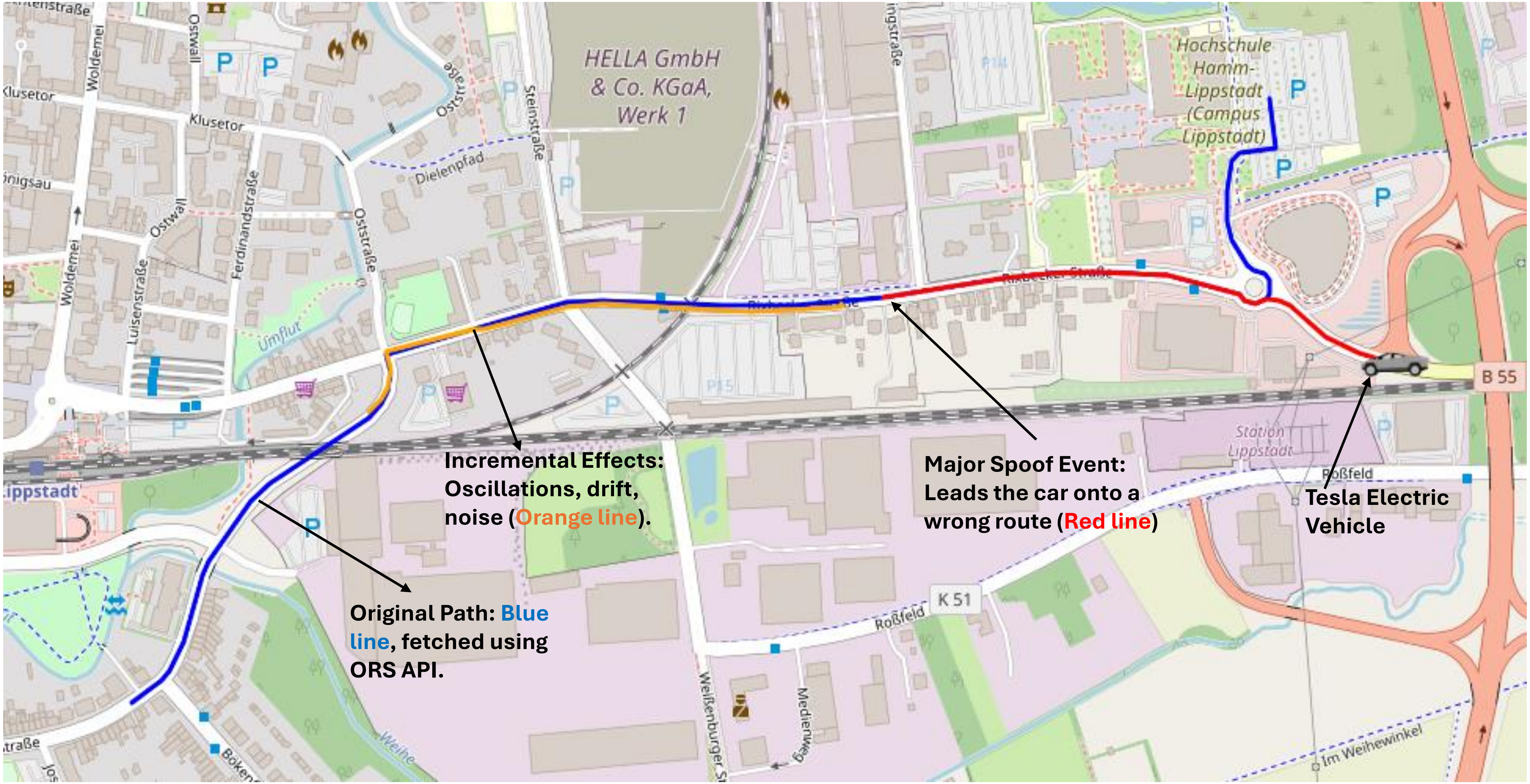
Spoofing Type	Max Deviation	Error Percentage	Noise Duration
GPS	3.5 m	35%	t =2 to 9 seconds
Time	1.5 s	15%	t =2 to 6 seconds
Sensor	4 m	20%	t =3 to 6 seconds



An Implementation of GPS Spoofing in Real World using API

Goal:

- Simulate GPS spoofing effects on a vehicle’s movement.
- Show incremental spoofing dynamics and their combined impact.


Incremental Effects: Oscillations, drift, noise (Orange line).
Major Spoof Event: Leads the car onto a wrong route (Red line)
Original Path: Blue line, fetched using ORS API.
Tesla Electric Vehicle
HELLA GmbH & Co. KGaA, Werk 1
Hochschule Hamm-Lippstadt (Campus Lippstadt)
Station Lippstadt
B 55
K 51
Roßfeld
Klusetor
Dielenpfad
Steinstraße
Oststraße
Ostwall
Ferdinandstraße
Luisenstraße
Woldemei
Umflut
Weißenburger St
Medienweg
Im Weihewinkel
Weihe

Step-by-Step Process

1. Original Path (Blue Line):

- Real GPS route from dorm to university (via ORS API).

2. Incremental Spoofing Dynamics (Orange Line):

- Amplitude Variation: Small oscillations.
- Frequency Variation: Varying motion.
- Random Walk: Gradual drift.
- Gaussian Noise: Random jitter.

3. Major Spoof Event (Red Line):

- All dynamics combined at 50% of the route.
- Redirects the car to a wrong route fetched using ORS.

Demo In Next Slide

Demo

- file:///C:/Users/shiha/OneDrive/Desktop/11th%20semester/Final%20Project/Javascript%20Visualisation%20-%20Attempt%20with%20Matlab%20-.html

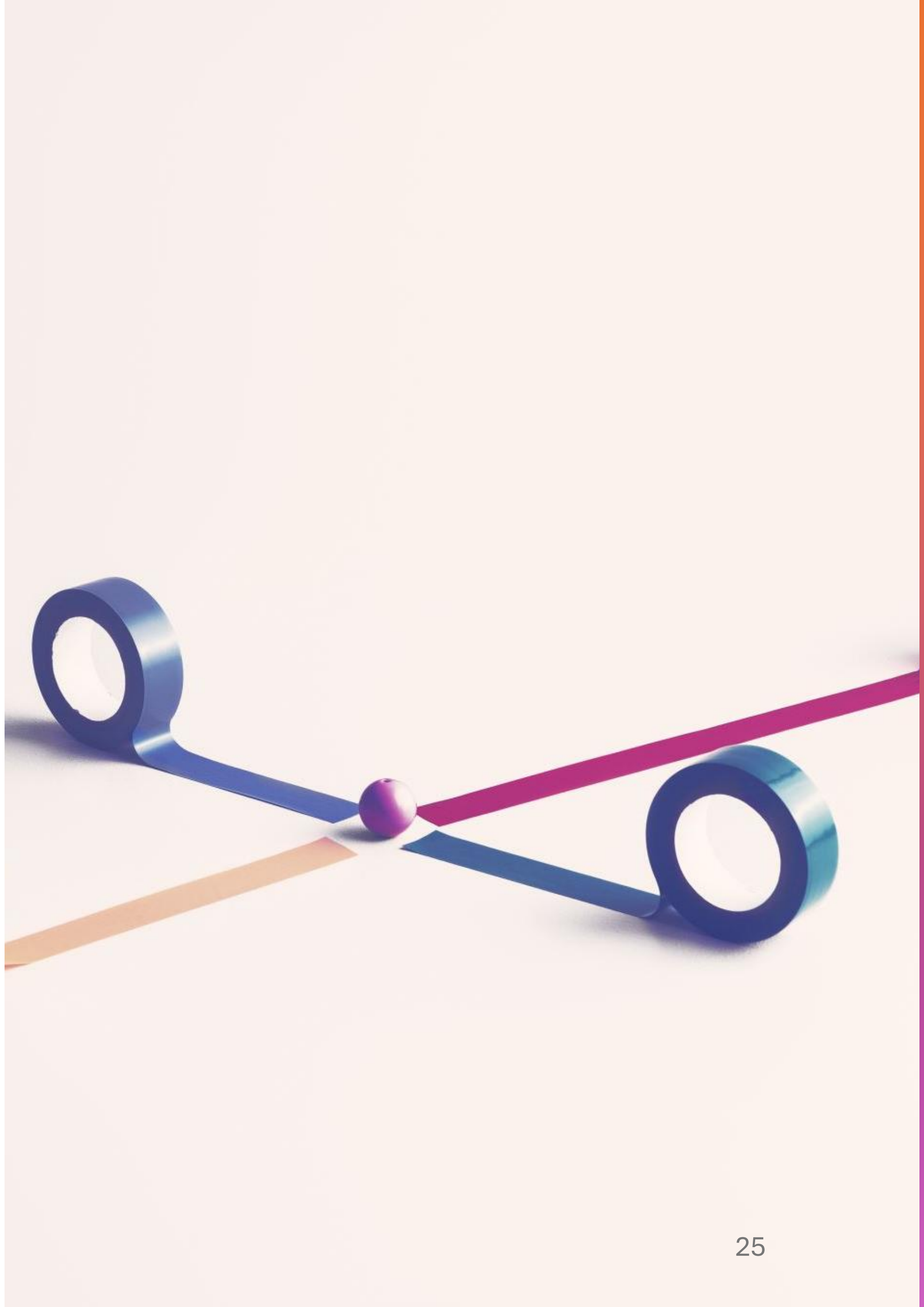
Discussion and Insights

Findings:

- GPS spoofing: Path deviations can misguide vehicles.
- Time spoofing: Sync delays impact system reliability.
- LiDAR spoofing: False readings compromise obstacle detection.

Challenges:

- Balancing simplicity and realism in simulations.
- Interpreting results in real-world contexts.



Conclusion and Future Work

Summary:

- Spoofing attacks significantly disrupt AV safety and navigation.
- GPS spoofing has the most direct impact on navigation.

Future Directions:

- Real-world testing to validate findings.
- Developing countermeasures for spoofing detection.
- Exploring combined spoofing scenarios for deeper insights.

References

1. Cao, X., Chen, L., Zhao, Y., & Wang, J. (2023). *Temporal consistency checks to detect LiDAR spoofing attacks on autonomous vehicles*. Retrieved from https://arxiv.org/pdf/2106.07833.pdf

2. Li, Z., Li, J., & Liu, X. (2023). *GPS-IDS: An anomaly-based GPS spoofing attack detection framework for autonomous vehicles*. Retrieved from https://arxiv.org/pdf/2405.08359v1.pdf

3. Wang, T., Zhang, P., & Liu, Y. (2023). *Unveiling the stealthy threat: Analyzing slow drift GPS spoofing attacks on autonomous vehicles*. Retrieved from https://arxiv.org/pdf/2401.01394.pdf

4. Zhou, H., Jiang, K., & Wu, Y. (2022). *Simulation of sensor spoofing attacks on unmanned aerial vehicles using ROS and Gazebo*. Retrieved from https://arxiv.org/pdf/2309.09648.pdf

5. Khan, A., Ahmed, R., & Bashir, U. (2023). *LiDAR spoofing attack detection in autonomous vehicles*. Retrieved from https://ieeexplore.ieee.org/document/9730540

6. Kim, Y., Chen, M., & Smith, J. (2023). *Adversarial sensor attack on LiDAR-based perception in autonomous driving*. Retrieved from https://arxiv.org/abs/1907.06826

7. Liu, H., Zhao, J., & Lin, S. (2023). *Anomaly detection against GPS spoofing attacks on connected and autonomous vehicles*. Retrieved from https://ieeexplore.ieee.org/document/10109166

8. Regulus Cyber. (2021). *Case study: Tesla autopilot spoofing attack*. Retrieved from https://www.regulus.com/case-study-tesla-spoofing-attack

9. Keen Security Lab. (2022). *Case study: Time spoofing vulnerabilities in autonomous driving systems*. Retrieved from [https://keenlab.tencent.com/en/autonomous-time-spoofing-case-study](https://keenlab.tencent.com/en/autonomous-time-spoofing-casestudy)

10. Park, J., & Lee, K. (2022). *Securing autonomous vehicles against GPS spoofing attacks: A deep learning approach*. Retrieved from https://ieeexplore.ieee.org/abstract/document/10264063

- Questions?