



The Ghost Navigator: Revisiting the Hidden Vulnerability of Localization in Autonomous Driving

Junqi Zhang¹, Shaoyin Cheng^{1,7}, Linqing Hu¹, Jie Zhang², Chengyu Shi³, Xingshuo Han⁴,
Tianwei Zhang⁴, Yueqiang Cheng⁵, and Weiming Zhang^{1,6}

¹University of Science and Technology of China, ²CFAR and IHPC, A*STAR, ³DeepBlue College,
⁴Nanyang Technological University, ⁵MediaTek, ⁶Anhui Province Key Laboratory of Digital Security,
⁷Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation

Abstract

Localization is crucial for Autonomous Driving (AD), which serves as a critical foundation impacting the performance of downstream modules. While Multi-Sensor Fusion (MSF) techniques enhance localization accuracy and reliability, the security of fusion-based localization systems has emerged as a major concern. Although existing studies have extensively investigated security aspects of these systems, the impact of vehicle dynamics on the effectiveness of Global Positioning System (GPS) spoofing attacks is persistently overlooked.

Bridging this research gap, we propose the Motion-Sensitive Analysis Framework (MSAF), which focuses on analyzing previously underestimated dynamic behaviors of vehicles. Our investigation demonstrates that two dynamic scenarios, acceleration and high-speed cruising, significantly influence the success rates of GPS spoofing attacks. These scenarios, commonly encountered across driving conditions, exhibit heightened vulnerabilities under MSAF analysis. Building on these insights, we design two dynamics-targeted attack strategies and evaluate them across three testbeds: our simulated framework (*MSAF_MSF*) and two real-world MSF-based autonomous driving systems (*Apollo_MSF* and *Shenlan_MSF*). The results demonstrate a significant attack efficiency improvement by our method: MSAF requires substantially less time to complete attacks compared to the baseline while achieving higher success rates. Code and attack demos are available at <https://sites.google.com/view/msaf-attack>.

1 Introduction

Autonomous vehicles are leading a reimagining of our modes of mobility, marking a significant advancement in automotive technology. Vehicle localization emerges as a fundamental task in autonomous driving (AD), particularly in vehicles equipped with high-level autonomous driving systems [1, 2]. The localization module, essential in determining the vehicle’s position and orientation, serves as the primary data source

for the entire process. Its accuracy and reliability are crucial, directly influencing the efficacy of downstream modules such as perception, planning, and control [3, 4].

As a crucial tool for acquiring broad global positioning in traditional localization systems, GPS is vulnerable to signal spoofing threats [5–7]. A more robust solution is Multi-Sensor Fusion (MSF) based localization, which leverages the combined strengths of various sensors to improve accuracy and resilience. By integrating observations from GPS, Inertial Measurement Units (IMUs), and the Light Detection and Ranging (LiDAR) locator, MSF localization achieves a more accurate and robust localization system [2, 8, 9]. Despite these enhancements, MSF localization still shows vulnerabilities to spoofing attacks under certain conditions, leading to substantial deviations in vehicle localization [10–13]. These vulnerabilities can induce *takeover effects*, wherein GPS data dominates and inputs from the LiDAR locator are disregarded as outliers, exposing challenges in the design of MSF systems.

Prior studies [10, 11] mainly attribute the cause of *takeover effects* to factors like sensor noise and sensor update frequency, while ignoring the impact of the vehicle’s dynamic state. Our empirical evaluations reveal that under the combined conditions of turning and acceleration, the *takeover effects* could still be triggered even with minimal changes in sensor noise and sensor update frequency. This indicates that previous analysis tends to underestimate the influence of vehicle motion states on triggering the *takeover effects*. In other words, it is insufficient to only consider scenarios where the vehicle is assumed to be in a stable motion state.

To bridge the identified gap, we introduce a novel Motion-Sensitive Analysis Framework (MSAF) to investigate security vulnerabilities in localization under dynamic motion states. This framework consists of two principal components: offline vulnerability analysis and online exploitation. The offline component assesses how varying motion states influence GPS spoofing effectiveness, focusing on acceleration and high-speed cruising scenarios (Sec. 2.2). The online component leverages these insights to execute context-aware attacks in simulated and real-world environments.

 Shaoyin Cheng and Weiming Zhang are the corresponding authors.

Despite these methodological advantages, three key challenges emerge in constructing MSAF: 1) The absence of aligned motion data (IMU, GPS, and LiDAR), 2) limited availability of open-source implementations for production-grade multi-sensor fusion architectures (particularly error correction modules), and 3) the requirement for high-dimensional state space decomposition encompassing 15 distinct dimensions of position, velocity, orientation, and sensor bias parameters. More details can be seen in Sec. 3.2.

To address the above challenges, as shown in Figure 4, we develop a Motion Data Generator (Sec. 4.1) in the offline vulnerability analysis phase, capable of generating simulated datasets that include a variety of vehicle motion states and sensor configurations. Following this, a Sensor Fusion Engine (Sec. 4.2) is designed to emulate the integration process of an IMU+GPS+LiDAR fusion structure, performing essential Error State Kalman Filtering (ESKF). This process allows us to assess the effects of GPS spoofing under different motion states. Additionally, a State Dependency Analyzer (Sec. 4.3) is introduced, which utilizes noise-free simulated data to first analyze the stability of system matrices through condition number evaluation, then evaluate the observability ranking of critical states, and finally quantify Kalman gain variations affecting GPS position measurements in sensor fusion — systematically disentangling dependencies among 15-dimensional states. Based on these offline analysis results, we propose an Injector (Sec. 4.4), which adjusts attack strategies by analyzing the vehicle’s real-time motion state (e.g., yaw and speed) and adapting the spoofing intensity to simulate precise and dynamic GPS spoofing attacks. MSAF exposes vulnerabilities within the specific fusion structure and illustrates how to strategically exploit these weaknesses to enhance GPS spoofing attack effectiveness.

To demonstrate the effectiveness of the proposed MSAF, we test it with three LiDAR-based fusion systems: Apollo_MSF, Shenlan_MSF, and our MSAF_MSF. We further conduct end-to-end attack validations on actual autonomous vehicles in the real world. The experimental results indicate that the conclusions drawn from MSAF are highly applicable and effective within practical autonomous driving fusion systems.

The main contributions can be summarized below:

- **Unveiling motion state impacts on MSF security analysis.** We identify a critical but underexplored vulnerability in MSF localization: *dynamic states, especially acceleration and high-speed cruising, significantly impact the GPS spoofing success rates.* This challenges the previously held belief about the minimal impact of varying vehicle speeds and shifts the focus of traditional security paradigms to the importance of vehicle motion states in MSF systems.
- **Design and implementation of MSAF: a Motion-Sensitive Analysis Framework for MSF security analysis.** To explore the overlooked dimension of motion state changes, we propose and develop a prototype of MSAF, focusing on the security analysis of fusion localization sys-

tems in autonomous driving affected by subtle variations in motion states. Implemented on a noise-free dataset, MSAF is designed to enhance the understanding of how different motion states impact the GPS spoofing success rates. The prototype and the dataset are open-sourced to support further research in this area.

- **Evaluating MSAF on the real-world vehicle.** Through comprehensive evaluations on datasets and two leading fusion localization systems (Apollo_MSF and Shenlan_MSF), we have comprehensively evaluated the effectiveness of MSAF. The results show that MSAF significantly improves the attack efficiency. Specifically, the success rates in the *off-road* attack scenario increased from 59.5% to 82%, while the *wrong-way* attacks rose from 45.5% to 73.5%. Furthermore, MSAF drastically reduces attack durations: the *off-road* attack completes in an average of 16.6 seconds (± 3.6 seconds), compared to FusionRipper’s 20.2 seconds (± 12.3 seconds), and the *wrong-way* scenario shortens from 24.5 seconds (± 13.2 seconds) to 21.8 seconds (± 2.9 seconds). Additionally, MSAF demonstrates the ability to conduct GPS spoofing *without an additional vehicle physically tailing the victim in real time*, simplifying the overall attack mechanism and enhancing feasibility in practical contexts.

2 Background and Threat Model

2.1 Background

AD Localization and Multi-Sensor Fusion. Autonomous driving systems critically depend on Multi-Sensor Fusion (MSF) algorithms to achieve the precise localization required for reliable navigation. By integrating data from LiDAR, GPS, and IMUs, MSF algorithms compensate for individual sensor limitations while enhancing overall accuracy [14]. LiDAR sensors generate high-resolution 3D environmental maps crucial for path planning, though their effectiveness decreases in adverse weather and geometrically uniform areas [15, 16]. GPS provides absolute positioning but becomes unreliable in signal-deprived environments like urban canyons [17]. IMUs track continuous motion but suffer from error accumulation over time [18]. This integration ensures autonomous vehicles to navigate safely and efficiently, adapting to diverse and challenging conditions.

The Kalman Filter (KF) and its variant, the Error-State Kalman Filter (ESKF), are widely recognized for their applicability in both academic and industry settings due to their ability to estimate the state of dynamic systems with high accuracy [10, 19–21]. Apollo_MSF [4], employed from Apollo 2.0 to Apollo 10.0, serves as the industry’s benchmark for robust fusion algorithms but operates as a black-box system with proprietary strategies. In contrast, Shenlan_MSF [22] is open source with different fusion methods. Both systems are based on the ESKF and show similar accuracy. The ESKF operates by separating the state vector into a nominal state

and an error state, making it particularly well-suited for systems with Gaussian noise in linear systems [23]. At time k , the error state $\delta\mathbf{x}_k$ is defined by the following vehicle states:

$$\delta\mathbf{x}_k = [\delta\mathbf{p}_k \quad \delta\mathbf{v}_k \quad \delta\mathbf{q}_k \quad \delta\mathbf{b}_{a_k} \quad \delta\mathbf{b}_{w_k}]^T \quad (1)$$

where $\delta\mathbf{p}_k$, $\delta\mathbf{v}_k$, and $\delta\mathbf{q}_k$ represent the position, velocity, and orientation error states, respectively, and $\delta\mathbf{b}_{a_k}$ and $\delta\mathbf{b}_{w_k}$ are the accelerometer and gyroscope bias error states. Each of these is a three-dimensional vector, making $\delta\mathbf{x}_k$ a 15-dimensional error state vector. The ESKF predicts the current error state $\delta\mathbf{x}_k$ from the previous error state $\delta\mathbf{x}_{k-1}$ and updates it with new observational data. During the prediction phase, system dynamics are used to estimate the prior error state $\delta\tilde{\mathbf{x}}_k$:

$$\begin{aligned} \delta\tilde{\mathbf{x}}_k &= \mathbf{F}_{k-1}\delta\tilde{\mathbf{x}}_{k-1} + \mathbf{B}_{k-1}\mathbf{w}_k, \\ \check{\mathbf{P}}_k &= \mathbf{F}_{k-1}\check{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{B}_{k-1}\mathbf{Q}_k\mathbf{B}_{k-1}^T, \end{aligned} \quad (2)$$

Here, the state transition matrix \mathbf{F}_{k-1} characterizes how the state evolves over time, while the input matrix \mathbf{B}_{k-1} relates process noise \mathbf{w}_k to system dynamics. The covariance matrix \mathbf{Q}_k captures uncertainties in system dynamics. This prediction, alongside its covariance $\check{\mathbf{P}}_k$, reflects the anticipated system accuracy. The subsequent correction phase adjusts these estimates using the latest measurements \mathbf{y}_k with the Kalman gain \mathbf{K}_k , leading to an updated $\delta\tilde{\mathbf{x}}_k$ and $\check{\mathbf{P}}_k$:

$$\begin{aligned} \mathbf{K}_k &= \check{\mathbf{P}}_k\mathbf{G}_k^T (\mathbf{G}_k\check{\mathbf{P}}_k\mathbf{G}_k^T + \mathbf{C}_k\mathbf{R}_k\mathbf{C}_k^T)^{-1}, \\ \delta\hat{\mathbf{x}}_k &= \delta\tilde{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{G}_k\delta\tilde{\mathbf{x}}_k), \\ \hat{\mathbf{P}}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{G}_k)\check{\mathbf{P}}_k. \end{aligned} \quad (3)$$

Here, \mathbf{G}_k represents the observation matrix that maps the state space into the measurement space, and \mathbf{C}_k is a transformation matrix within the measurement model. The measurement noise covariance matrix \mathbf{R}_k quantifies the expected accuracy of the measurements, and \mathbf{I} is the identity matrix. The ESKF broadens the reach of the KF by estimating error states in nonlinear systems, thereby overcoming its limitations and expanding its applicability.

Security Analysis of MSF Algorithms. Analyzing the security of MSF algorithms involves considering a fundamental threat model where attackers send GPS spoofing signals with the intent to divert the vehicle from the lane centerline. However, the high-frequency and high-accuracy localization provided by the LiDAR locator can mitigate the deception attempts. Thus, attackers must exploit specific vulnerabilities within the MSF model, specifically those model properties that can facilitate GPS spoofing efforts. Prior studies [10, 11] have demonstrated that attackers can successfully launch GPS spoofing when the uncertainty associated with the LiDAR locator is high, or the uncertainty of the KF's previous state is significant. Attackers often begin by closely following the target vehicle, transmitting a constant spoofing signal to subtly influence the vehicle's trajectory. This phase aims to incrementally deviate the vehicle from the lane's centerline without

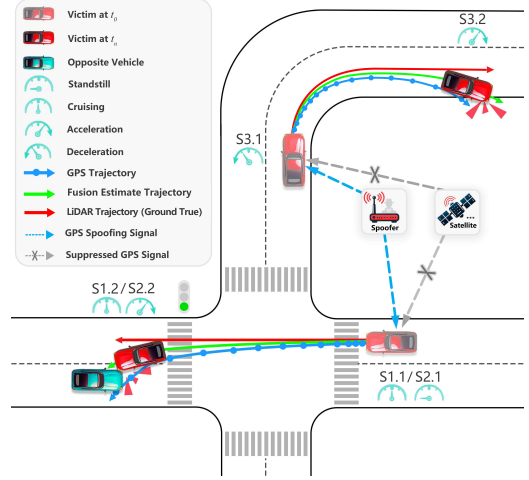


Figure 1: Illustration of two attack scenarios target three motion states: high-speed cruising on straight paths (S1.1->S1.2), accelerating from a standstill on straight paths (S2.1->S2.2), and transitioning from deceleration to acceleration in turns (S3.1->S3.2).

triggering immediate detection by the system's anomaly detectors. Once the deviation exceeds a predefined threshold, specifically the distance of the vehicle from the lane's centerline, indicating the vehicle is in a vulnerable state, attackers then escalate their efforts to exponential spoofing.

2.2 Threat Model

Attack Goals. The attacker attempts to exploit the subtleties of vehicular dynamics by performing GPS spoofing during specific dynamic scenarios, aiming to deviate the vehicle towards the curb or into oncoming traffic. Figure 1 showcases two attack scenarios target three critical motion states:

1. **Cruising attack.** Targets vehicles during high-speed cruising on straight paths (S1.1->S1.2).
2. **Acceleration attack.** Targets vehicles from standstill to acceleration on straight paths (S2.1->S2.2) and from deceleration to acceleration within turns (S3.1->S3.2).

These two scenarios are designed to exploit specific motion states, with the attack built around selecting the most vulnerable motion states for effective GPS spoofing.

Attacker's Capability. We assume that the attacker can perform GPS spoofing while maintaining normally distributed signal quality, thereby evading potential detection mechanisms. Additionally, the attacker can assess the target vehicle's motion state by utilizing advanced object-tracking techniques (e.g., sensor fusion of camera and LiDAR data with Kalman filtering), enabling continuous monitoring of yaw orientation and velocity patterns. Furthermore, the attacker can exploit the motion-sensitive vulnerabilities by either passively awaiting or actively creating conditions when the target vehicle is

most vulnerable, such as during a standstill to an acceleration state (S2.1->S2.2). These scenarios can be anticipated to occur naturally, or the attacker can deliberately provoke them, for instance, by suddenly decelerating or stopping abruptly in front of the victim’s vehicle to force it back into an acceleration state.

3 Motivation and Challenges

We introduce a motivational example to demonstrate the inherent vulnerabilities in MSF algorithms and the challenges of analyzing these vulnerabilities across different motion states.

3.1 Motivation

Previous security analysis on MSF concluded that the impact of IMU dynamics on the take-over effect is negligible. However, our motivation experiment challenges this conclusion. Specifically, in our experiments conducted under turning scenarios, characterized by simultaneous changes in both acceleration and angular velocity from the IMU, we observed a 16.7% chance of triggering the take-over effect, where GPS becomes dominant and LiDAR measurements are discarded as outliers, as illustrated in Figure 2.

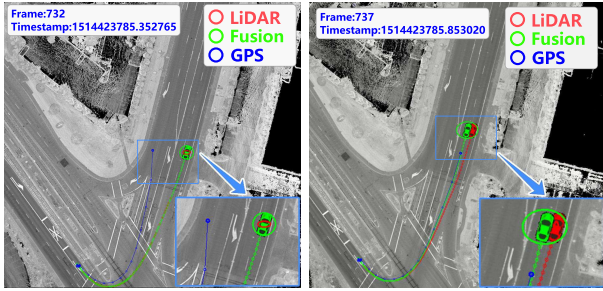


Figure 2: Turning scenario with simultaneous changes in both acceleration and angular velocity, showing a failed take-over effect (left) and a successful take-over effect (right).

Our experiment builds on the two-stage attack framework and parameters proposed in [10], where $d = 0.5$ and $f = 1.6$ were identified as the optimal attack parameters for the baidu-64 dataset. We modified the second stage by setting the trigger to the vehicle speed rather than lateral deviation from the centerline. Figure 3 illustrates the dynamic interactions between vehicle speed, spoofing offset, and lateral deviation. The experimental results suggest that vehicle motion states, such as acceleration, may contribute to the success rate of GPS spoofing attacks.

Results Explanation. Previous security analysis focused solely on positional discrepancies, thus proving insufficient as they overlooked the impact of dynamic states on the MSF

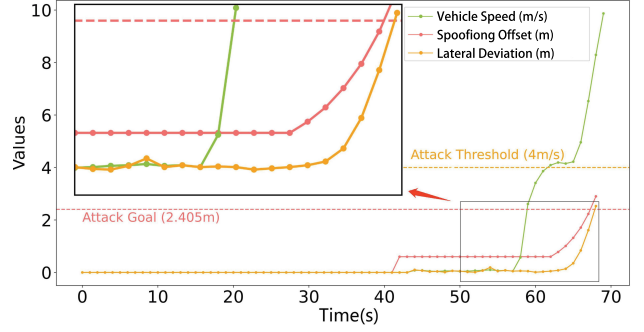


Figure 3: Trigger GPS spoofing attacks based on the speed threshold. Green, red, and yellow curves denote vehicle speed, spoofing signal, and vehicle lateral deviation, respectively.

model. These dynamic motion states, however, directly impact the F matrix [2, 24], a key component in the prediction step of state estimation:

$$F = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{F}_{23} & \mathbf{0}_{3 \times 3} & \mathbf{C}_b^n \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{F}_{33} & -\mathbf{C}_b^n & \mathbf{0}_{3 \times 3} \\ & & \mathbf{0}_{3 \times 15} & & \\ & & \mathbf{0}_{3 \times 15} & & \end{bmatrix}, \quad (4)$$

This F matrix incorporates key vehicle motion parameters, such as acceleration and angular velocity. While certain elements of the F matrix can be approximated as constant over short intervals, elements like F_{23} and F_{33} become critical during significant changes in vehicle motion states and should not be overlooked. Specifically, F_{23} captures the interaction between Earth’s rotation and the vehicle’s acceleration, showing how vertical velocity (f_U) influences northward (f_N) and eastward (f_E) velocities, and vice versa.

$$F_{23} = \begin{bmatrix} 0 & -f_U & f_N \\ f_U & 0 & -f_E \\ -f_N & f_E & 0 \end{bmatrix}, \quad (5)$$

F_{33} reflects Earth’s rotation and the vehicle’s angular velocity on its orientation. The matrix shows how angular velocity (ω) and latitude (L) affect the vehicle’s heading.

$$F_{33} = \begin{bmatrix} 0 & \omega \sin L & -\omega \cos L \\ -\omega \sin L & 0 & 0 \\ \omega \cos L & 0 & 0 \end{bmatrix}. \quad (6)$$

By affecting critical parameters like acceleration and angular velocity, these states alter the elements of the F matrix.

In summary, the F matrix captures the interaction between vehicle dynamics, which are crucial for accurate state prediction. Analyzing F_{23} and F_{33} is key to understanding the MSF model’s vulnerability to GPS spoofing. To further explore these dependencies, we will introduce an analytical framework in Sec. 4 that examines the relationships between 15-dimensional states under different dynamics.

3.2 Challenges

Key challenges in building our framework include:

Challenge 1: *How to mitigate sensor noise interference in dynamic condition analysis?*

Existing security analyses typically attribute the take-over effect primarily to environmental noise. Our key challenge lies in generating high-fidelity noise-free datasets that preserve essential dynamic characteristics while eliminating sensor noise contamination. This requires addressing two critical requirements: First, the dataset must maintain precise temporal synchronization and physical consistency between IMU (acceleration/angular velocity), GPS measurements (position/velocity), and LiDAR measurements (position/attitudes). Second, it should comprehensively cover diverse motion patterns including static, constant velocity, acceleration/deceleration, and turning scenarios to enable systematic analysis.

Challenge 2: *How to emulate the black-box fusion structure for assessing the potential importance of velocity?*

Given that real-world MSF algorithms, like those used by Apollo, are often black-box implementations, it becomes challenging to reverse-engineer comparable fusion structure without access to the internal mechanisms. Even for systems like Shenlan_MSF, which are more open-source and support various fusion architectures, there is still no design that explicitly incorporates velocity in the IMU+GPS+LiDAR fusion process. Our goal is to replicate the fusion strategy of the target system to construct an IMU+GPS+LiDAR fusion structure, despite lacking detailed knowledge of the algorithm. This challenge encompasses two main aspects: (1) understanding and emulating the target’s fusion structure, especially supporting fusion structures both with and without velocity integration; and (2) ensuring that the designed fusion strategy can effectively process the data generated in **Challenge 1**.

Challenge 3: *How to establish quantitative metrics for state information capacity across dynamic scenarios?*

Previous studies have often downplayed the role of vehicle speed in influencing the *take-over effect*, with analysis typically constrained to singular trajectories and minor variations in motion states. To gain a nuanced understanding of how different trajectories impact the information capacity of vehicle dynamics, it is imperative to develop a methodology for quantifying the information capacity of various vehicle states (e.g., position, velocity, orientation, gyroscope bias, accelerometer bias) within an IMU+GPS+LiDAR fusion framework. The methodology must enable comparative analysis of state information entropy across different motion patterns, particularly examining velocity’s role in spoofing vulnerability during transitional states like acceleration and turning.

4 Motion Sensitive Analysis Framework

We introduce MSAF (see Figure 4) to address the three challenges identified in Sec. 3.2 through two phases: *Offline Vulnerability Profiling* and *Online Exploitation*. In the *Offline Vulnerability Profiling* phase, the **Motion Data Generator** resolves Challenge 1 by creating noise-free sensor data across diverse motion states, enabling robust simulation. The **Sensor Fusion Engine** tackles Challenge 2 by replicating a black-box fusion structure with velocity integration. The **State Dependency Analyzer** addresses Challenge 3 by quantitatively analyzing state observability and matrix stability under varying dynamics. In the *Online Exploitation* phase, the **Injector** handles Challenge 3 by dynamically identifying motion states and simulate precise GPS spoofing attacks in real-time.

4.1 Motion Data Generator

The Motion Data Generator is designed to meticulously manage and integrate raw sensor data across a spectrum of motion states, facilitating comprehensive simulations through precise data integration and data synchronization.

Data Integration. In this step, it is challenging to simulate the pose data for the LiDAR locator, as `gnss_ins_sim` [25] primarily supports IMU and GPS data simulation. To address this, positional and attitudinal noise is introduced to mimic real-world inaccuracies. Position $\mathbf{p}_{\text{lidar}}$ is derived by adding Gaussian noise \mathbf{n}_{pos} with zero mean and standard deviation σ_{pos} to the ground truth \mathbf{p}_{gt} , formulated as:

$$\mathbf{p}_{\text{lidar}} = \mathbf{p}_{\text{gt}} + \mathbf{n}_{\text{pos}}, \quad \mathbf{n}_{\text{pos}} \sim \mathcal{N}(0, \sigma_{\text{pos}}^2 \mathbf{I}). \quad (7)$$

where $\mathbf{p}_{\text{lidar}}$ is the position of the lidar locator, \mathbf{p}_{gt} is the ground truth position, \mathbf{n}_{pos} is the noise vector with zero mean and standard deviation σ_{pos} , and \mathbf{I} is the identity matrix.

Orientation $\mathbf{q}_{\text{lidar}}$ is simulated by adding rotational noise \mathbf{q}_{rot} to the ground truth orientation \mathbf{q}_{gt} , represented by:

$$\mathbf{q}_{\text{lidar}} = \mathbf{q}_{\text{gt}} \otimes \exp\left(\frac{1}{2} \sigma_{\text{rot}} \boldsymbol{\eta}\right), \quad (8)$$

where $\mathbf{q}_{\text{lidar}}$ is the quaternion representing the lidar locator’s orientation, \mathbf{q}_{gt} is the ground truth quaternion, σ_{rot} is the standard deviation of the rotational noise, $\boldsymbol{\eta}$ is a vector following a Gaussian distribution $\mathcal{N}(0, \mathbf{I})$, and \otimes represents quaternion multiplication.

To support the generation of both benign and malicious signals, *Direct Injection* is applied by adding predefined deviations to a vehicle’s GPS data, independent of dynamic state assessments. For straight driving, a fixed deviation $\boldsymbol{\delta}_{\text{straight}}$ is added to the GPS position:

$$\mathbf{p}_{\text{gps, spf}} = \mathbf{p}_{\text{gps, org}} + \boldsymbol{\delta}_{\text{straight}}. \quad (9)$$

In turning scenarios, deviation $\boldsymbol{\delta}_{\text{turning}}$ with the vehicle’s heading θ is used to modify the position \mathbf{p}_{gps} to simulate a turn:

$$\mathbf{p}_{\text{gps, spf}} = \mathbf{p}_{\text{gps, org}} + \boldsymbol{\delta}_{\text{turning}} \cos(\theta), \quad (10)$$

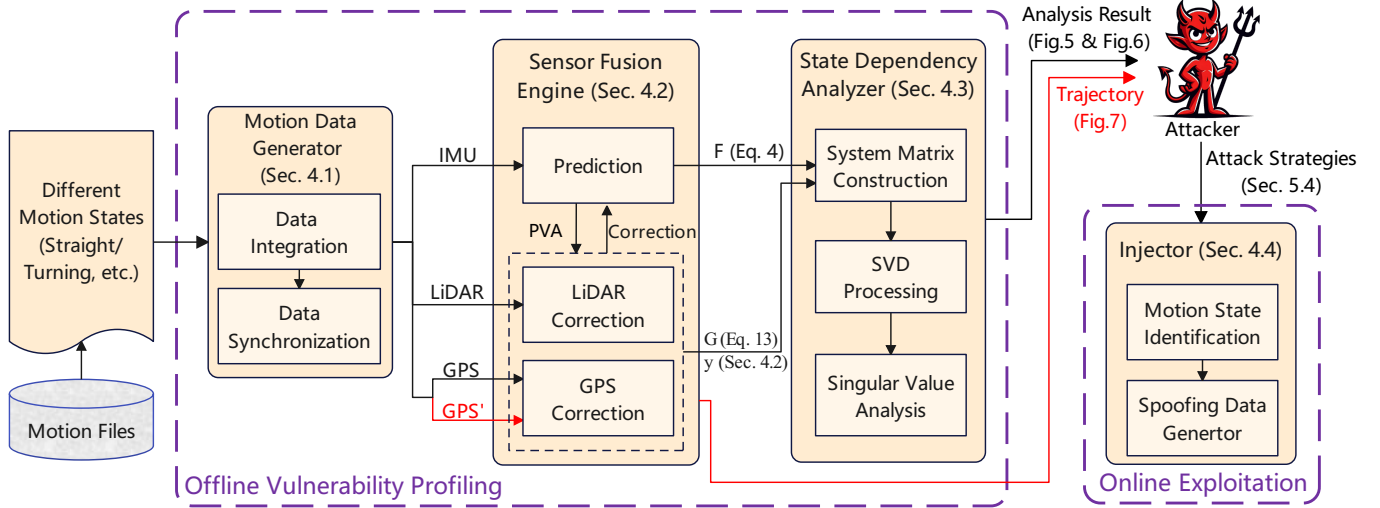


Figure 4: Overview of the proposed Motion Sensitive Analysis Framework (MSAF).

$$\mathbf{p}_{\text{gps, spf}} = \mathbf{p}_{\text{gps, org}} + \delta_{\text{turning}} \sin(\theta).$$

where $\mathbf{p}_{\text{gps, spf}}$ is the spoofed (spf) gps position vector, $\mathbf{p}_{\text{gps, org}}$ is the original (org) gps position vector, δ_{straight} is the fixed deviation for straight driving, δ_{turning} is the deviation for turning, and θ is the vehicle's heading.

Direct Injection facilitates the simulation of specific movements and disruptions by adjusting GPS data, which is crucial for detailed motion state simulations and accurate sensor data generation under varied motion states.

Data Synchronization. This module ensures accurate timing and alignment of sensor data, focusing on synchronizing data from IMU (accelerometer and gyroscope), GPS (position and velocity), and LiDAR locator (position and attitudes). By utilizing GPS timestamps as the reference, this module aligns timestamps across these sensor outputs for coherence. By employing linear interpolation for timing alignment, this synchronization process markedly increases the system's precision in handling and amalgamating data from diverse sensors.

4.2 Sensor Fusion Engine

The Sensor Fusion Engine employs the ESKF model to integrate data from IMU, GPS, and LiDAR locator, creating a fusion structure that combines various sensor inputs for precise state estimation. The process involves initializing the state, predicting using IMU data, and updating the state with observations from GPS and LiDAR locator. Error correction is then applied to refine the state estimates, ensuring they align with actual observations.

Prediction. We incorporate the Earth's model to enhance the vehicle's state updates, a methodology widely adopted within high-precision integrated navigation systems to significantly improve state estimation and control under various navigational conditions [26, 27]. The model accounts for the Earth's

rotation (ω_{ie}^T) and curvature (R_N and R_M), which are integral factors in refining the vehicle's state estimates:

$$\omega_{ie}^T = [0, \omega \cos L, \omega \sin L], \quad (11)$$

$$\omega_{en}^T = \left[-\frac{v_N}{R_M + h}, \frac{v_E}{R_N + h}, \frac{v_E \tan L}{R_N + h} \right], \quad (12)$$

where R_N and R_M , the prime vertical and meridian radii of curvature respectively, are pivotal in calculating the effects of Earth's geometry on the vehicle's motion. When integrated into the system dynamics matrix \mathbf{F}_t in the prediction equation, they enable precise anticipation of the vehicle's state for accurate navigation in both linear and rotational movements.

Correction. In the correction phase, MSAF uses GPS and LiDAR measurements to enhance the predicted states of the vehicle. GPS provides crucial positional and velocity information, while the LiDAR locator offers detailed insights into position and attitude. These inputs are synthesized into the observation matrix \mathbf{G} and the observation vector \mathbf{y} , expressed as $\mathbf{y}^T = [\mathbf{d}\mathbf{p}_{\text{lidar}}^T \ \mathbf{d}\mathbf{v}_{\text{gps}}^T \ \mathbf{d}\mathbf{q}_{\text{lidar}}^T \ \mathbf{d}\mathbf{p}_{\text{gps}}^T]$. Here, $\mathbf{d}\mathbf{p}_{\text{lidar}}$, $\mathbf{d}\mathbf{v}_{\text{gps}}$, $\mathbf{d}\mathbf{q}_{\text{lidar}}$, and $\mathbf{d}\mathbf{p}_{\text{gps}}$ represent errors in LiDAR position, GPS velocity, LiDAR orientation, and GPS position, respectively. Subsequently, the Kalman Gain \mathbf{K} is determined based on current state estimates and observation data. This gain, derived from the predicted error covariance \mathbf{P} and accounting for both process and observation noise, is essential for updating the error state \mathbf{X} . Utilizing \mathbf{G} and \mathbf{y} , the system identifies and corrects discrepancies between observed and estimated values, thereby refining the vehicle's position, velocity, and orientation estimates. The observation matrix \mathbf{G} is defined as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 6} \\ \mathbf{0}_{3 \times 3} & \mathbf{C}_n^b & -\mathbf{C}_n^b \mathbf{V} \times & \mathbf{0}_{3 \times 6} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 6} \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 6} \end{bmatrix}, \quad (13)$$

where \mathbf{C}_n^b is the transformation matrix from navigation to body coordinates, and \mathbf{V} denotes velocity. \mathbf{G} converts GPS and LiDAR observations into refined state error estimations. The error state vector is then reset and accumulated discrepancies are eliminated to maintain system state estimation integrity.

4.3 State Dependency Analyzer

The State Dependency Analyzer identifies vulnerabilities in the sensor fusion process by analyzing the numerical properties of system matrices. This includes both stability analysis and observability analysis via singular value evaluation. Stability analysis examines the sensitivity of system matrices to numerical perturbations through the condition number, while observability analysis determines how well the system states, such as position and velocity, can be inferred from sensor data. These factors directly influence the stability and accuracy of the state estimation process [28–30], enabling the identification of both ill-conditioned matrices that are sensitive to perturbations and weakly observable states that could be exploited by attackers.

System Matrix Construction. The system matrix construction process begins with the collection and integration of process data, forming the foundation for a comprehensive analysis. This involves accumulating system matrices \mathbf{G}_i and corresponding vectors \mathbf{y}_i over time. Following this, the system matrices \mathbf{G}_i and vectors \mathbf{y}_i are accumulated into a system matrix \mathbf{Q}_{som} and vector \mathbf{y}_{som} , which are essential for subsequent analysis steps. The following equations define this process:

$$\mathbf{Q}_{\text{som}} = \sum_{i=1}^n \mathbf{G}_i \cdot \mathbf{F}_{\text{accumulate}}^{(i)}, \quad \mathbf{y}_{\text{som}} = \sum_{i=1}^n \mathbf{y}_i. \quad (14)$$

In this formulation, \mathbf{Q}_{som} and \mathbf{y}_{som} encapsulate the overall system's dynamics by integrating the effects of state transitions over time through $\mathbf{F}_{\text{accumulate}}^{(i)}$, preparing the data for in-depth analysis of system stability and observability.

Singular Value Decomposition (SVD) Processing. SVD is crucial for revealing the internal structure of the system matrix \mathbf{Q}_{som} . By decomposing it into $\mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$, where \mathbf{U} and \mathbf{V} represent the singular vector matrices and \mathbf{S} is the diagonal matrix of singular values, we gain insight into the system's stability and observability through the singular values in \mathbf{S} .

Singular Value Evaluation. After obtaining the singular values, they are used for both stability and observability analysis.

Stability Analysis. The condition number $\kappa = \sigma_{\text{max}}/\sigma_{\text{min}}$, calculated from the maximum and minimum singular values, reflects the system's overall sensitivity to perturbations [31, 32]. A higher κ indicates that the system matrix, as a whole, is more vulnerable to numerical instability, making it potentially exploitable by attackers. This evaluation helps determine whether the entire system can maintain stability in the presence of disturbances or errors.

Observability Analysis. In observability analysis, the focus shifts to the individual states of the system, such as the

vehicle's position, velocity, and orientation. Higher singular values correspond to states that are more easily inferred from sensor inputs, while lower values indicate states that are weakly observable [33, 34], potentially sensitive to external disruptions. To support this analysis, the Piecewise Constant Systems (PWCS) method [35], widely used in dynamic systems like autonomous driving, links singular values to specific system states, showing how the observability of each state evolves over time. To quantify the observability of each state, an observation matrix \mathbf{X} is constructed as follows:

$$\mathbf{X} = \mathbf{V} \cdot \mathbf{S}^{-1} \cdot \mathbf{U}^T \cdot \mathbf{y}_{\text{som}}, \quad (15)$$

where \mathbf{U} , \mathbf{V} , and \mathbf{S} are derived from the SVD of \mathbf{Q}_{som} , and \mathbf{y}_{som} represents system observations. The observability profile is formed by identifying the maximum indices in \mathbf{X} and mapping the corresponding singular values to these states.

4.4 Injector

To trigger GPS spoofing attacks, we first identify the ego vehicle's motion states to simulate the identification of the victim vehicle's motion states, and then generate spoofing data accordingly.

Motion State Identification. Identifying the vehicle's motion state involves assessing the yaw and speed, critical for understanding orientation and movement to execute GPS spoofing attacks effectively.

Yaw Identification. Accurate determination of the yaw angle from quaternion data is critical for GPS spoofing to introduce lateral deviations. The yaw angle reflects the vehicle's orientation on the horizontal plane, vital for the alignment of spoofed GPS signals. With a normalized quaternion $\text{normalized_q} = (q_w, q_x, q_y, q_z)$, the calculation of the yaw angle ψ incorporates trigonometric equations directly: the yaw angle is derived from $\sin(\psi) = 2 \times (q_w \times q_z + q_x \times q_y)$ and $\cos(\psi) = 1 - 2 \times (q_y^2 + q_z^2)$, leading to $\psi = \text{atan2}(\sin(\psi), \cos(\psi))$. Such precise calculations enable accurate lateral adjustments in GPS spoofing, aligning the vehicle's perceived orientation with the intended direction effectively.

Speed Identification. Vehicle speed is crucial for launching GPS spoofing attacks. It is determined by analyzing the vehicle's velocity data, which is derived from real-time motion captured by the IMU. The overall speed of the vehicle (**vel**) is calculated by taking the square root of the sum of the squares of the vehicle's x and y velocity components: $\text{vel} = \sqrt{x_{\text{vel}}^2 + y_{\text{vel}}^2}$, where x_{vel} and y_{vel} represent the vehicle's velocity components in the horizontal plane. This method accurately reflects the vehicle's speed, which is essential for timing GPS spoofing attacks to match specific vehicle speeds for effective manipulation.

Understanding both the yaw and the vehicle's speed provides a comprehensive view of the vehicle's motion state,

aiding attackers in optimizing the timing and execution of GPS spoofing. This ensures that the spoofed signals closely align with the vehicle’s actual state, increasing the effectiveness and subtlety of the attack.

Spoofing Data Generation. The underlying principle of the injector model is designed to exploit the motion state of a vehicle, dynamically initiating GPS spoofing when it is either accelerating or moving at a specific speed. This approach leverages the dynamics of the vehicle’s movement, enabling more effective and precisely timed spoofing attacks. The revised target function of the injector, which is dependent on the vehicle’s motion state, is formulated as:

$$A(t) = \begin{cases} (d \cdot f^i) & \text{under certain conditions,} \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

Here, $A(t)$ denotes the injection sequence at time t , with d and f as foundational parameters akin to those in Fusion-Ripper, and i indicating the iteration number. The specific condition for initiating the spoofing process is determined by factors such as vehicle speed and acceleration.

For core concepts of the ESKF framework and the notation used in MSAF design, please refer to Appendix A.5 and A.6, which detail the error-state formulation and the mathematical symbols employed in the MSAF architecture

5 Evaluation on Offline Vulnerability Profiling

In this section, we first identify potential attack scenarios, we then evaluate how different motion states impact the fusion system and influence GPS spoofing attacks. Based on these observations, we propose two targeted attack strategies.

5.1 Identifying Potential Attack Scenarios

To systematically identify potential attack scenarios, we began by analyzing the key variables that govern the dynamics of autonomous vehicles. The initial assessment included velocity, acceleration, and the three attitude angles (Roll, Pitch, Yaw), leading to $2^5 = 32$ possible combinations. To refine the model, we excluded roll, as it is not typically considered in practical autonomous driving applications [2], and omitted uphill/downhill scenarios to reduce the complexity. This adjustment narrowed the combinations to $2^3 = 8$. Subsequently, we removed implausible cases, such as scenarios involving acceleration changes without the corresponding velocity changes, and disregarded highly complex cases where all variables vary simultaneously. Through this systematic reduction, five representative scenarios were identified, as summarized in Table 1, with "straight_vel" encompassing both stationary and constant velocity conditions.

Table 1: Overview of experimental synthetic scenarios.

Scenarios	Vel (mps)	Acc (mpss)	AngVel (degps)
<i>straight_vel</i>	0, 1, 5, 15	0	0
<i>straight_acc</i>	2	0, ± 0.2 , 1, 2	0
<i>turning_yaw</i>	2	0	3, 6, 9, 12
<i>turning_yaw_vel</i>	1, 3, 4, 6	0	3, 9, 12, 18

Note: " ± 0.2 " indicates acceleration at 0.2 m/s^2 followed by deceleration at 0.2 m/s^2 .

5.2 The Impact on the Fusion System

Here, we assess the stability of the system matrix, the observability of vehicle states, and Kalman gain variations across different motion states to understand their impact on the system performance. In a nutshell, we find that *velocity plays a pivotal impact on the fusion system*.

Experimental Setup. We utilized noise-free data to isolate the impact of sensor noise on the system performance. The ESKF noise parameters, including initial, prediction, and observation noises, were set to 1.0×10^{-6} . The vehicle was aligned with the y-axis, with each scenario lasting 20 seconds. Sensor frequencies were set to 100Hz for the IMU and 10Hz for GPS and LiDAR. Prior to the experiment, we conducted a preliminary evaluation of MSAF’s fusion localization accuracy to establish a reliable baseline. Detailed results of this evaluation are provided in Appendix A.1.

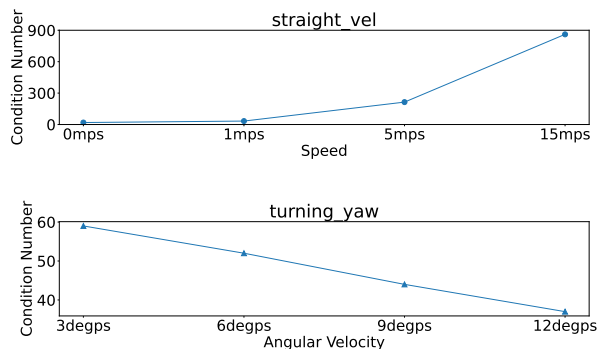


Figure 5: Stability analysis of the system matrix.

Stability Analysis. The condition number serves as a key metric for evaluating the system’s sensitivity to external disturbances. As shown in Figure 5, in the *straight_vel* scenario, the condition number of the system matrix \mathbf{Q}_{som} remains near 0 at very low speeds but rises dramatically as the speed increases, reaching close to 900 at 15 mps. This sharp rise indicates that the system becomes increasingly unstable at higher speeds. In contrast, the *turning_yaw* scenario shows a more gradual decline in the condition number, from around 60 at three deg/s to approximately 40 at 12 deg/s. Compared to

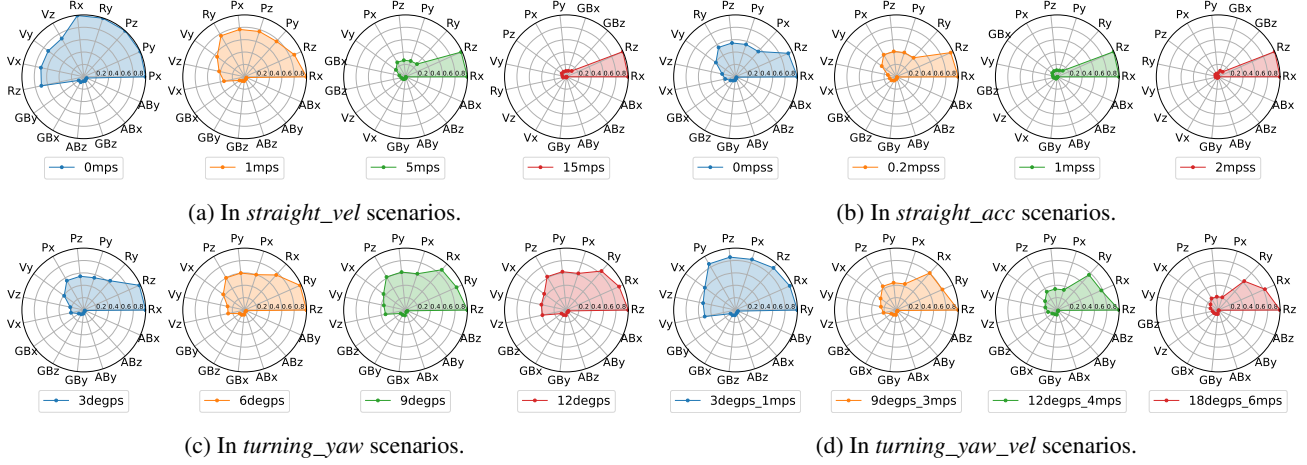


Figure 6: Observability ranking in four scenarios (refer to Table 1 for details). The top1-ranked observable motion state corresponds to the horizontal starting point for counterclockwise rotation.

the significant changes observed in the speed-based scenario, the variation in turning is much smaller, suggesting that the system’s stability remains relatively consistent during turning.

Observability Analysis. Although the stability analysis shows greater instability at higher speeds, it does not specify which vehicle states are most affected. Therefore, we conduct an observability analysis to evaluate each vehicle state’s performance under different motion states. Figure 6 shows the observability analysis for \mathbf{Q}_{som} across different motion states. Radar charts represent observability in 15 state dimensions, including position (P_x, P_y, P_z), velocity (V_x, V_y, V_z), attitude (R_x, R_y, R_z), and biases in gyroscope (GB_x, GB_y, GB_z) and accelerometer (AB_x, AB_y, AB_z). Key findings are as follows:

- In the *straight_vel* scenario, as the vehicle’s velocity increases, a decrease occurs in the observability of the position dimensions (e.g., P_x and P_y) as well as the velocity dimensions (V_x, V_y, V_z). In contrast, the attitude dimensions (e.g., R_x and R_z) exhibit increased observability. This suggests that higher speeds may lead to reduced position and velocity observability but enhanced attitude observability.
- In the *straight_acc* scenario, higher accelerations lead to a similar trend of decreased position observability, with P_x and P_y being the most affected. This suggests that higher accelerations lead to reduced observability of the position dimensions, similar to the effect of increasing velocity.
- In the *turning_yaw* scenario, where the vehicle maintains a steady velocity while turning, the position dimensions (P_x, P_y) show only minor variations, reflecting stable positional observability. The attitude dimensions (R_x, R_y, R_z) also show slight variations, indicating minimal changes in orientation observability.
- In the *turning_yaw_vel* scenario, where the vehicle experiences changes in both the turning rate and speed, we notice a more complex interplay between the speed and observability.

As the vehicle’s speed increases, the observability for attitude dimensions, notably R_x and R_z , demonstrates an inverse correlation, with higher speeds leading to decreased attitude observability.

Overall, significant variations are observed in the position, velocity, and attitude dimensions, where a decrease in the position’s singular values at higher speeds leads to reduced observability. In contrast, the six bias dimensions (gyroscope and accelerometer biases) exhibit relatively minor variations.

Observation 1: The stability of the system matrix and the positional observability decrease significantly with high-speed cruising and acceleration.

Kalman Gain Analysis. Here, we mainly focus on *straight_acc* and *turning_yaw_vel* scenarios due to their diverse motion states, ideal for studying Kalman gain trends for GPS positioning ($K_{(1,10)}$) in our MSFA model. During the prediction phase, the error covariance increases due to system dynamics, causing higher uncertainty in the state estimate. As a result, the Kalman filter relies more on external measurements. In the correction phase, incorporating GPS and LiDAR measurements reduces this covariance, adjusting the Kalman gain to reflect the system’s reliance on these measurements. For more details, please refer to Sec. A.2.

- In the *straight_acc* scenario (Figure 7-top), starting from an initial velocity of 2.0m/s and no acceleration, the Kalman gain for position stabilizes, indicating a balanced trust in inertial and GPS data. As acceleration increases, there is a notable upward trend in the Kalman gain, which signifies that the system begins to place a greater emphasis on GPS.
- The *turning_yaw_vel* scenario (Figure 7-bottom) captures how the Kalman gain $K_{(1,10)}$ responds to changes in vehicle speed alone, ranging from 1m/s to 6m/s. Notably, the gain initially decreases and then subsequently increases. This

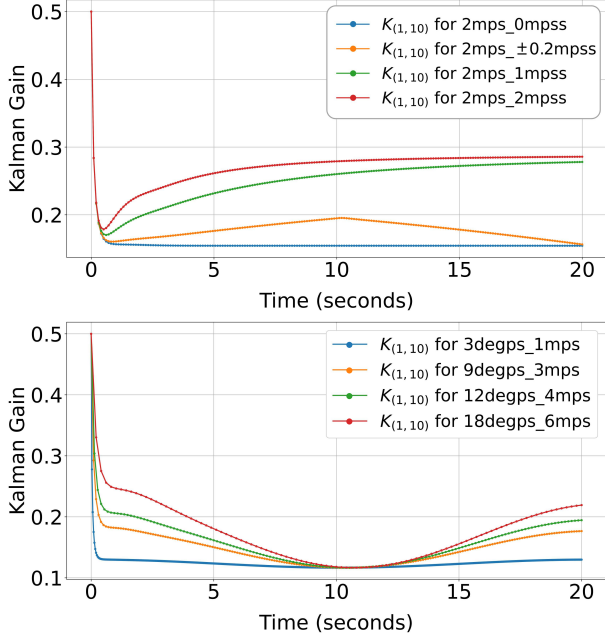


Figure 7: Kalman gain at $K_{(1,10)}$ in *straight_acc* (top) and *turning_yaw_vel* (bottom) scenarios.

pattern also indicates that the system’s reliance on GPS data adjusts in correlation to the vehicle’s speed.

Observation 2: As vehicle acceleration intensifies, a corresponding increase in the Kalman gain is observed, indicating a heightened dependency on GPS data.

5.3 The Impact on Constant Spoofing Attacks

To further validate Observations 1 and 2, we conducted constant spoofing attacks on the synthetic dataset to observe whether the attack outcomes vary across motion states.

Experimental Setup. We follow the setting in Sec. 5.2 to minimize influences from sensor noise, sensor frequency, and ESKF model noise. We perform GPS spoofing injections for the motion states detailed in Table 1, aligning the vehicle forward along the y-axis. By determining the yaw angle as described in Sec. 4.4, we inject lateral deviations with five constant offset points δ_a (2m) perpendicular to the yaw direction, ensuring consistent lateral injection and robust impact assessment. In *straight_acc* scenarios, a single spoofing instance per trajectory is injected, using a horizontal line at 0 as the ground truth. For *turning_yaw* scenarios, three spoofing instances per trajectory are introduced to explore repeated spoofing effects at varying angular velocities.

Results. We observe two obvious effects of the velocity on constant spoofing attacks as follows:

Deviation Amplification Effect. In *straight_acc* scenarios (Figure 8-top), the deviation is stable without acceleration but

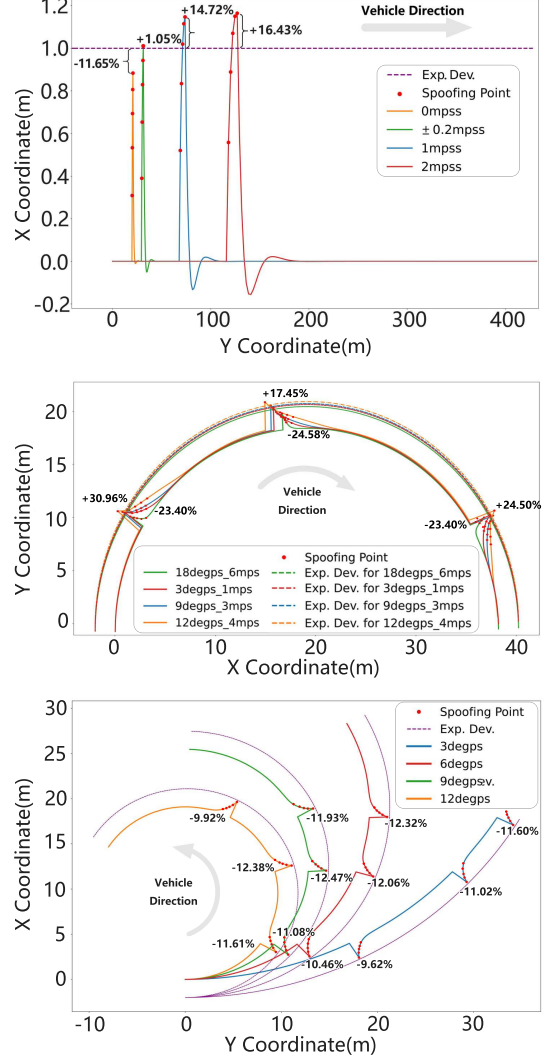


Figure 8: Injection results in *straight_acc* (top), *turning_yaw_vel* (middle), and *turning_yaw* (bottom) scenarios. Exp.Dev. denotes Expected Deviation.

increases with the vehicle acceleration. At the acceleration of 2 m/s^2 , the deviation can exceed the expected values by 16.43%, showing that acceleration amplifies GPS spoofing effects. In *turning_yaw_vel* scenarios (Figure 8-middle), as the yaw rate and speed increase, the maximum deviation from the expected offsets also rises, with up to 30.96% greater deviation at higher speeds and sharper turns, indicating that speed and turn sharpness amplify the deviation.

Deviation Stability Effect. In *turning_yaw* scenarios with a constant speed of 2 m/s and varying angular velocities, offset changes remain consistent, as shown in Figure 8-bottom. The average offset change rate is below 2%, indicating the deviation stability despite different angular velocities. This suggests that at steady speeds, angular velocity variations minimally influence the GPS spoofing effects.

For a detailed analysis of the velocity convergence proper-

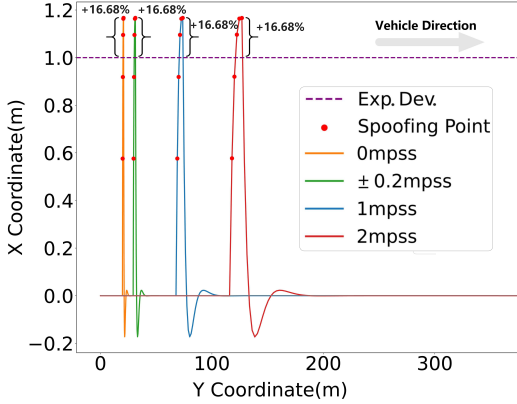


Figure 9: Injection results in *straight_acc* scenarios without velocity fusion.

ties and deviation results in the *straight_vel* scenarios, please refer to Appendix A.3.

Observation 3: As the cruising speed and acceleration increase, the lateral deviation becomes larger, while variations in the angular velocity have minimal effect.

Ablation Study. An ablation study was designed to investigate the role of velocity in assessing vulnerabilities. By excluding the velocity observation from the fusion structure, the study aimed to uncover the extent to which velocity data impacts the Kalman filter’s susceptibility to spoofing. The results in Figure 9 show that removing the velocity observation lead to a uniform offset increase in *straight_acc* scenarios, with deviations consistently 16.68% above the expected. This uniformity in deviations, in contrast to the varied deviations observed when the velocity is included, highlights the crucial role velocity plays in the fusion model.

5.4 Attack Strategies Design

In Sec. 5.2, the results show significant drops in system stability and positional observability during high-speed cruising and acceleration phases, as indicated by *Observation 1*, and a corresponding increase in the Kalman gain with vehicle acceleration, indicating a heightened dependency on GPS data, as described in *Observation 2*. Based on these insights, we further validated the vulnerabilities through constant spoofing attacks in Sec. 5.3, leading to *Observation 3*, which demonstrated the system’s susceptibility to increased lateral deviation under these conditions. Building on these findings, we propose two exponential spoofing-based attack strategies targeting these vulnerable states. Both strategies aim to manipulate the vehicle’s trajectory, causing deviations towards the curb (*off-road*) or into oncoming traffic (*wrong-way*).

1. **Cruising attack.** This strategy targets the vehicle during high-speed cruising on straight paths, where the increased

velocity amplifies the impact of GPS spoofing. A constant speed helps stabilize Kalman gain dynamics, minimizing the disturbances and enabling attackers to precisely manipulate the spoofing effect.

2. **Acceleration attack.** This strategy targets two dynamic phases: (1) when the vehicle accelerates from a standstill on straight paths and (2) during the transition from deceleration to acceleration within turns. These phases are particularly vulnerable due to the system’s transition from stability to dynamic changes, making it more susceptible to lateral deviations as the Kalman gain adjusts gradually.

6 Evaluation on Online Exploitation

6.1 Simulation Accuracy

We compare the attack results of MSAF under simulated data with those of Apollo_MSF under real-world data. This comparison aims to verify whether the black-box Apollo_MSF exhibits the same property identified in Observation 3, specifically that higher cruising speeds result in larger lateral deviations under the same level of GPS spoofing input. Furthermore, it aims to evaluate MSAF’s capability of predicting actual lateral deviations.

Experimental Setup. In the real-world scenario, the vehicle cruises at constant speeds of 0.5, 1.5, 2.5, 3.5, and 4.5 m/s along a fixed straight path. For the simulation, we generate data at the same speeds using MSAF. We set the IMU and GPS velocity sampling rates as 100 Hz, and GPS position and LiDAR rates as 5 Hz. The GPS and LiDAR measurement uncertainty is fixed to minimize the dynamic noise. We feed the real data into Apollo_MSF, and simulated data into MSAF. The attack is triggered at a specific coordinate using exponential spoofing ($d=0.05$, $f=1.1$) lasting 10 seconds, with each scenario repeated 50 times for consistency. LiDAR is used as ground truth, and lateral deviation is calculated by comparing MSF and LiDAR outputs.

Results. The results are shown in Figure 10, confirming that Apollo_MSF demonstrates the expected trend, where higher cruising speeds lead to larger lateral deviations. MSAF effectively predicts these deviations, with an average prediction error of 1.33%, as shown in Table 3. These results indicate that MSAF’s prediction error is highly consistent with that of the black-box Apollo_MSF across different speeds.

6.2 Attack Effectiveness

We systematically evaluate the proposed attack strategies across multiple LiDAR-based fusion models and datasets, as shown in Table 2. Our primary goal is to verify the success rates of both acceleration and cruising attack under various motion states, confirming that the vulnerabilities identified earlier are indeed exploitable in practice.

Table 2: Success rate of two attack strategies under different attack parameters. The number in parentheses following each scenario indicates the total instances of that scenario within the dataset.

Attacked MSF	Dataset	Scenario	Attack Param		Acceleration attack		Cruising attack	
			d	f	Off-Road	Wrong-Way	Off-Road	Wrong-Way
Apollo_MSFF	Baidu-64	acceleration(3)	0.2	1.2	98%	92.7%	-	-
		constant(3)	0.1	1.2	-	-	91%	85%
	Baidu-128	acceleration(2)	0.1	1.2	100%	100%	-	-
		constant(1)	0.1	1.2	-	-	100%	100%
	MSAF-32	acceleration(5)	0.1	1.2	100%	100%	-	-
		constant(4)	0.1	1.2	-	-	100%	100%
Shenlan_MSFF	KITTI-64	acceleration(8)	0.2	1.2	100%	100%	-	-
		constant(1)	0.2	1.2	-	-	100%	100%
	MSAF-32	acceleration(5)	0.2	1.2	100%	100%	-	-
		constant(4)	0.2	1.2	-	-	100%	100%
MSAF_MSFF	MSAF-Sim	acceleration(5)	0.1	1.01	100%	100%	-	-
		constant(5)	0.1	1.01	-	-	100%	100%

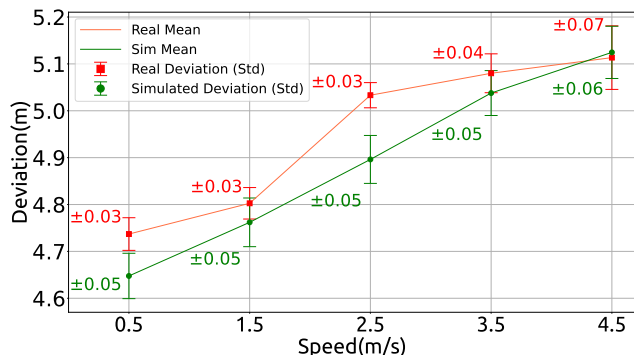


Figure 10: Lateral deviations for real-world (Apollo_MSFF) and simulated (MSAF) conditions at different speeds.

Table 3: Prediction error of MSAF at different velocities.

Velocity (m/s)	0.5	1.5	2.5	3.5	4.5	Average
Error (%)	1.92	0.85	2.80	0.84	0.22	1.33

Experimental Setup. For all datasets, data beyond ± 2 standard deviations were filtered out to obtain a more stable spoofing signal. The minimum value within the range of mean minus 2 standard deviations was then selected as the position uncertainty. Given variations in LiDAR uncertainty across scenarios, we applied basic exponential spoofing in both acceleration and constant-speed cases to find the optimal spoofing point. We tested spoofing points starting from $d = 0.1$, $f = 1.1$, with an increment of 0.1. Each point was spoofed 10 times, and the one with the highest success rate was chosen as the optimal point and parameters. Finally, 100 spoofing attempts were performed at the optimal point with the best parameters, and the success rate was recorded. The off-road and wrong-way attack goals were set to 0.895m and 2.405m, respectively, consistent with [10].

Results. In the experiments, our proposed attack strategies showed high success rates. For the Apollo_MSFF model on the Baidu-64 dataset, the acceleration strategy achieved a 98% success rate for off-road attacks and 92.7% for wrong-way attacks, while the constant-speed strategy achieved 91% and 85% success rates, respectively. On the Baidu-128 dataset, both strategies reached 100% for all attack types. For Shenlan_MSFF and MSAF_MSFF, the success rates were consistently 100% across all datasets and scenarios, demonstrating the effectiveness of our proposed strategies.

6.3 Attack Robustness

GPS spoofing attacks require exploiting vulnerabilities during critical phases, such as when the lateral offset exceeds a threshold [10], or during acceleration and high-speed cruising identified by MSAF. Delays caused by sensor noise and the attacker’s reaction time can significantly affect both the timing and overall effectiveness of the attack.

Error Sources and Modeling. We refer to the work of [10] to model localization and uncertainty errors, identifying three main sources: 1) localization error σ_1 from the attacker’s self-localization, 2) distance measurement error σ_2 from LiDAR sensors, and 3) GPS receiver error σ_3 , representing the deviation between intended and actual GPS positions. These errors follow a combined normal distribution $N(0, 0.058^2)$ for the total position error σ_{pos} . In addition, measurement uncertainty σ_{var} is set to 0.008 based on real-world data. Beyond these sources of inaccuracy, we also account for the attacker’s reaction time, which introduces further timing uncertainty. According to the U.S. Federal Highway Administration (FHWA), typical driver reaction times range from 0.75 to 1.5 seconds [36], while research by the Visual Expert indicates a range of 0.7 to 1.5 seconds [37]. Based on these findings, we derive an average reaction time of 1.1 seconds with a stan-

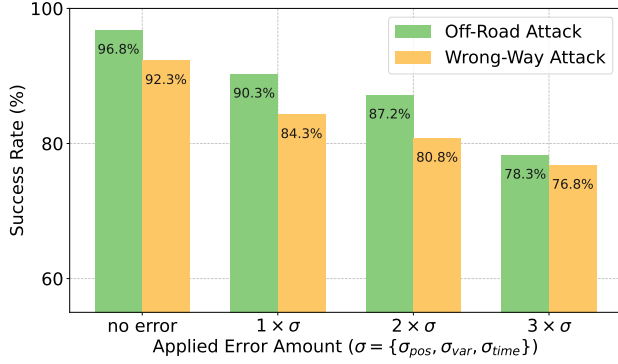


Figure 11: Attack success rates on the Baidu-64 dataset under varying spoofing inaccuracies and reaction time errors.

ard deviation of 0.2 seconds. This models potential delays in triggering the attack, incorporating more realistic timing uncertainties into the spoofing process through σ_{time} .

Experimental Setup. We apply the aforementioned error distributions to evaluate the robustness of MSAF, incorporating localization errors from [10] and timing uncertainties introduced by MSAF. For each GPS input, localization errors are sampled from $N(0, \sigma_{pos}^2)$, with directions uniformly distributed (0-360 degrees), and measurement uncertainties are drawn from $N(0, \sigma_{var}^2)$. The attack is then triggered with a delay sampled from $N(1.1, 0.2)$ to represent real-world reaction times. We also evaluate $2\times$ and $3\times$ error amounts to test robustness. Each scenario is repeated 200 times for reliability.

Results. Figure 11 shows that the off-road and wrong-way attack success rates remain high even under increased error amounts. Without error injection, the success rates are 96.8% and 92.3%, respectively. When normal errors ($1 \times \sigma$) are applied, the rates decrease slightly to 90.3% and 84.3%. Even with tripled errors ($3 \times \sigma$), the attacks maintain robust success, with the rates of 78.3% and 76.8%.

6.4 Attack Comparison

We compare MSAF with the existing method FusionRipper [10] on the Baidu-64 dataset, focusing on both attack success rates and durations. This comparison allows us to clarify the respective strengths of these two approaches and further evaluate how MSAF improves upon prior work.

Experimental Setup. We adopted the optimal parameters reported for FusionRipper, $d = 0.6$ and $f = 1.5$, while for MSAF, we used $d = 0.1$ and $f = 1.2$. MSAF triggered the attack at random points during one of three acceleration or three cruising phases on the map, using exponential spoofing. In contrast, FusionRipper initiated its two-phase spoofing attack from random points on the map. Since both methods were tested using simulated GPS signals, we incorporated the maximum error values (3σ) derived from Sec. 6.3 to better approximate real-world conditions.

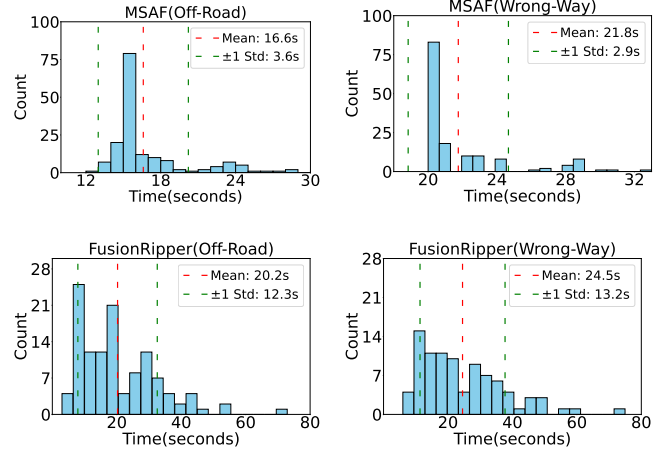


Figure 12: Comparison of attack durations.

Comparison of Attack Success Rates. Table 4 summarizes the success rates for off-way and wrong-way attack scenarios. MSAF demonstrated higher effectiveness, achieving success rates of 82.0% for off-way attacks and 73.5% for wrong-way attacks. In comparison, FusionRipper achieved lower success rates, with 59.5% in the off-way scenario and 45.5% in the wrong-way scenario. These results highlight MSAF’s improved capability to induce substantial vehicle deviations.

Table 4: Comparison of attack success rates

Method	Off-way (%)	Wrong-way (%)
MSAF	82.0	73.5
FusionRipper	59.5	45.5

Comparison of Attack Durations. Figure 12 highlights the clear advantage of MSAF compared to FusionRipper in terms of both efficiency and consistency in attack durations. The attack durations for MSAF are not only shorter but also more concentrated, with no instance exceeding 32 seconds. Specifically, MSAF achieved mean durations of 16.6 seconds for off-road attacks and 21.8 seconds for wrong-way attacks, with relatively small standard deviations of 3.6 and 2.9 seconds, respectively. In contrast, FusionRipper displayed significantly longer and more variable attack time, with mean durations of 20.2 seconds for off-road attacks and 24.5 seconds for wrong-way attacks, coupled with much higher standard deviations of 12.3 and 13.2 seconds. FusionRipper’s attack durations occasionally extended to nearly 80 seconds, underscoring the method’s unpredictability and reliance on more extended spoofing phases, whereas MSAF consistently maintained more efficient and reliable attack durations.

6.5 End-to-End Vehicle Evaluation

Prior experiments focused on the impact on the localization module, uncovering and exploiting vulnerabilities under different motion states. However, they did not fully account for how the vehicle’s dynamic responses and control strategies could affect the success of GPS spoofing attacks. To address this gap, this section extends the scope of evaluation to include the entire vehicle system, encompassing perception, positioning, planning, and control modules. By conducting experiments on actual autonomous vehicles, we aim to confirm the practical effectiveness of our attack methods on real-world autonomous vehicles.

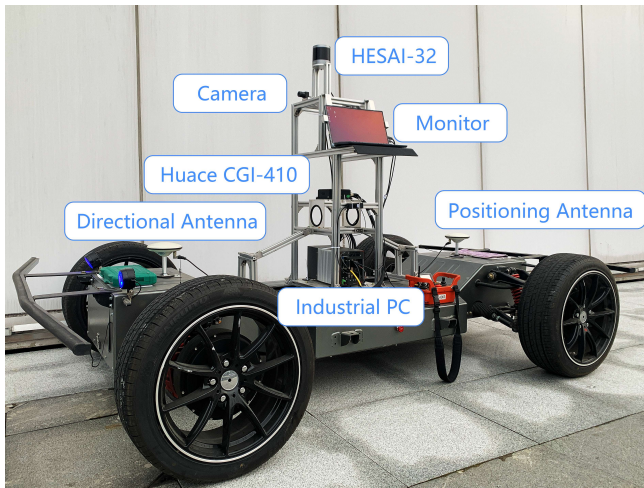


Figure 13: Pix hooke chassis with Apollo 6.0 Edu platform.

Experimental Setup. As depicted in Figure 13, our autonomous vehicle is equipped with a 32-line LiDAR, Huace CGI-410 INS, and a Nuvo-8111 industrial PC with an Intel Core i9-9900K CPU, NVIDIA RTX 3060 GPU, 32GB RAM, and 1TB SSD, integrated with Pix Hooke Chassis and Apollo 6.0 Edu Platform. We evaluate the autonomous driving system’s response to GPS spoofing at the speeds ranging from 1 m/s to 4 m/s across various scenarios, including straight-line driving, turns, and start-up acceleration.

Results. As shown in Figure 14, the end-to-end evaluation, encompassing startup and turning scenarios, demonstrated the successful execution of lateral GPS spoofing attacks, compelling the vehicle to collide with obstacles on either side of the road. These findings unequivocally show that MSAF can effectively compromise the security of autonomous vehicles by exploiting motion-sensitive vulnerabilities in the localization module.

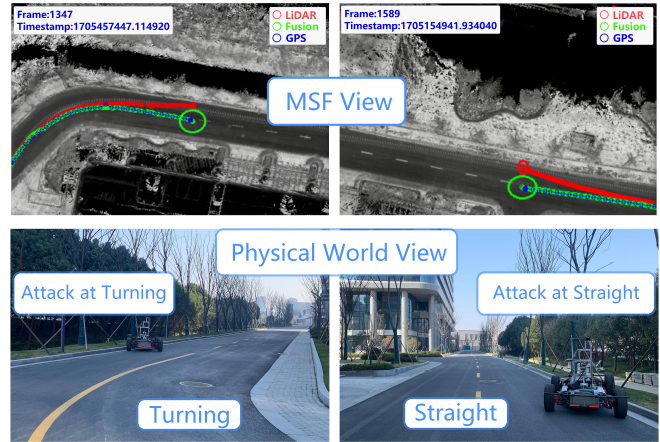


Figure 14: The vehicle strikes the curb after a GPS spoofing.

7 Limitations and Defense Discussions

7.1 Limitations

Simulation Constraints. We adopted a white-box approach to analyze vulnerabilities within the ESKF fusion structure, allowing us to examine both the design parameters and the internal state updates in detail. However, due to legal constraints, we did not conduct genuine GPS spoofing. Instead, we introduced perturbations and delays into the spoofing signals to more closely simulate real-world conditions.

Speed Testing Limitations. Although our simulations tested speeds ranging from 0 to 60 m/s, which exceed Apollo’s default city road limit of 15.67 m/s, the Pix chassis used for physical experiments is limited to a maximum of 4.5 m/s. As a result, higher speed tests were not feasible in our current setup. Therefore, exploring tests at higher speeds remains a valuable direction for future research.

7.2 Defense Discussions

Algorithm-Level Defenses. Algorithm-level state monitoring could be used to track changes in MSF’s internal states over time, detecting anomalies indicative of an attack. However, such defenses may exhibit limited effectiveness against attacks employing gradual accumulation of subtle perturbations. The persistent nature of such low-magnitude manipulations poses detection challenges through state monitoring alone, particularly when compared to more aggressive approaches characterized by sudden parametric mutations over short periods.

System-Level Defenses. By detecting position discrepancies between LiDAR and GPS, the system can flag inconsistencies in localization, providing an additional layer of

verification. System-level defenses enhance the resilience because errors or manipulations in one sensor type (e.g., GPS spoofing) can be detected through cross-validation with other independent sensors, such as the LiDAR locator. This redundancy may improve the overall security and reliability of the autonomous system against GPS spoofing attacks.

8 Related Work

Sensor Spoofing Targeting LiDAR. Cao *et al.* [38] developed a method for attackers to synchronize a photodiode with a LiDAR, creating deceptive points in the point cloud. Tu *et al.* [39] explored the creation of adversarial 3D objects to mislead LiDAR systems. These objects, however, are noticeable due to their unique shapes and placements. Zhu *et al.* [40] focused on identifying crucial adversarial positions in physical space, aiming to deceive LiDAR systems more efficiently. Jin *et al.* [41] designed a physical laser attack against LiDAR-based 3D object detection. These studies primarily concentrate on single-sensor deception strategies targeting LiDAR in autonomous driving systems, overlooking the complexities involved in multi-sensor fusion positioning tasks that incorporate LiDAR.

Sensor Spoofing Targeting IMU. In the realm of IMU spoofing, two main types of attacks are identified. Trippel *et al.* [42] exposed the susceptibility of MEMS accelerometers to malicious acoustic interference, leading to compromised linear and angular velocity data. Ji *et al.* [43] used acoustic waves to affect the gyroscope sensors in cameras, causing motion blurs and thus disabling object detection. Similar to the studies on LiDAR deception, research on IMU spoofing predominantly focuses on attacks against individual sensors and does not address the challenges in scenarios involving the fusion of multiple sensors.

Security Analysis on Sensor Fusion Model. Nashimoto *et al.* [44] explored the vulnerabilities of an Attitude and Heading Reference System (AHRS) under signal injection attacks, demonstrating significant security risks in systems that fuse data from multiple sensors, notably in inclination measurements. This work suggests new directions for bolstering the security of sensor fusion systems. Shen *et al.* [10] developed FusionRipper, a technique for identifying and exploiting vulnerabilities in LiDAR-based ESKF systems, combining theoretical analysis with simulation experiments to pinpoint critical weaknesses, such as LiDAR locator uncertainty and ESKF initial state uncertainty. Chang *et al.* [11] found that the sensor update frequency significantly affects the success of GPS spoofing attacks, corroborating FusionRipper’s premises. However, vulnerabilities were deemed more critical in steady states, indicating the IMU’s limited role in initiating takeover effects. Kim *et al.* [45] systematically analyzed the prereq-

uisites and quantified the real-world hardness of conducting various sensor attacks against robotic vehicles, revealing previously unknown root causes stemming from design flaws in the fail-safe logic.

9 Conclusion

This study reveals a critical vulnerability in autonomous vehicle localization systems: the significant correlation between vehicular motion states and GPS spoofing effectiveness. Our proposed Motion-Sensitive Analysis Framework (MSAF) establishes a new paradigm for analyzing security risks in multi-sensor fusion systems, demonstrating how transitional states like acceleration and high-speed cruising create attack surfaces overlooked by conventional security analysis models. Experimental validation through comprehensive testing on real-world systems demonstrates MSAF’s capability to significantly reduce attack success duration and increase attack success rates in operational scenarios. These findings necessitate the integration of dynamic state analysis into security evaluation frameworks, particularly during motion state transitions where sensor fusion vulnerabilities become exploitable in autonomous driving systems.

Ethical Considerations

Ethical Conduct and Disclosure. This work utilized publicly accessible autonomous driving platforms and related academic papers to conduct a comprehensive security analysis. Detailed information on design vulnerabilities, attack methodologies, and experimental findings was responsibly communicated to the developers of the affected autonomous driving system prior to the public release of our results. This proactive disclosure aimed to facilitate the mitigation of potential security issues. All datasets employed in our real-world experiments were obtained through open and legitimate channels. While assessing the effectiveness of the attacks on actual vehicles, we abstained from performing real GPS spoofing. Instead, we used simulated spoofing signals injected into the original data within a controlled environment, ensuring that there was no interference with surrounding satellite signals, a risk typically associated with traditional GPS spoofing.

Safety Protocols and Risk Mitigation. During the data collection phase in real-world settings, we strictly adhered to established safety protocols to mitigate potential risks. The attack tests were conducted on newly constructed roads within our research institute, which were not yet open to the public, thereby avoiding any impact on the general population. To prevent unauthorized vehicle entry into the experimental area, conspicuous traffic cones and warning signs were placed 100 meters after the test section. All experimental activities were

overseen and executed by trained personnel, ensuring the highest standards of safety and procedural reliability.

Compliance with Open Science Principles. We uphold the principles of open science, emphasizing transparency, reproducibility, and collaborative research. To support further research and allow for independent validation by the scientific community, we have open-sourced the complete MSAF codebase along with the generated simulated datasets. These resources, including the code and experimental videos, are available at <https://sites.google.com/view/msaf-demo>. By providing these materials, we aim to contribute valuable tools for advancing the understanding and mitigation of security challenges in autonomous driving systems.

Acknowledgments

We sincerely thank the anonymous reviewers for their constructive feedback. Special gratitude is extended to Xiaojian Li and Liyu Zhu for their help with data analysis, and to Baijun Chen and Wen Wang for their dedicated efforts in vehicular system debugging. This research was supported in part by the National Natural Science Foundation of China under Grants U20B2047, 62072421, 62002334, 62102386, 62121002, and U2336206, by the Open Fund of Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation (Grant CSSAE-2021-007), and by the National Research Foundation, Singapore, and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-GC-2023-008).

References

- [1] Tong Qin, Yuxin Zheng, Tongqing Chen, Yilun Chen, and Qing Su. A light-weight semantic map for visual localization towards autonomous driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11248–11254. IEEE, 2021.
- [2] Guowei Wan, Xiaolong Yang, Renlan Cai, Hao Li, Yao Zhou, Hao Wang, and Shiyu Song. Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4670–4677. IEEE, 2018.
- [3] Athanasios Chalvatzaras, Ioannis Pratikakis, and Angelos A Amanatiadis. A survey on map-based localization techniques for autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 8(2):1574–1596, 2022.
- [4] Baidu apollo autonomous driving platform. <https://github.com/ApolloAuto/apollo>.
- [5] Harshad Sathaye, Martin Strohmeier, Vincent Lenders, and Aanjhan Ranganathan. An experimental study of gps spoofing and takeover attacks on uavs. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3503–3520, 2022.
- [6] Harshad Sathaye, Gerald LaMountain, Pau Closas, and Aanjhan Ranganathan. Semperfi: Anti-spoofing gps receiver for uavs. In *Network and Distributed Systems Security (NDSS) Symposium 2022*, 2022.
- [7] Tesla model s and model 3 vulnerable to gnss spoofing attacks. <https://tinyurl.com/3fxv9hpa>.
- [8] Chao Qin, Haoyang Ye, Christian E Pranata, Jun Han, Shuyang Zhang, and Ming Liu. Lins: A lidar-inertial state estimator for robust and efficient navigation. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 8899–8906. IEEE, 2020.
- [9] Wendong Ding, Shenhua Hou, Hang Gao, Guowei Wan, and Shiyu Song. Lidar inertial odometry aided robust lidar localization system in changing city scenes. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4322–4328. IEEE, 2020.
- [10] Junjie Shen, Jun Yeon Won, Zeyuan Chen, and Qi Alfred Chen. Drift with devil: Security of {Multi-Sensor} fusion based localization in {High-Level} autonomous driving under {GPS} spoofing. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 931–948, 2020.
- [11] Jiachong Chang, Liang Zhang, Li-Ta Hsu, Bing Xu, Feng Huang, and Dingjie Xu. Analytic models of a loosely coupled gnss/ins/lidar kalman filter considering update frequency under a spoofing attack. *IEEE Sensors Journal*, 22(23):23341–23355, 2022.
- [12] Jiachong Chang, Feng Huang, Liang Zhang, Dingjie Xu, and Li-Ta Hsu. Selection of areas for effective gnss spoofing attacks to a vehicle-mounted msf system based on scenario classification models. *IEEE Transactions on Vehicular Technology*, 2023.
- [13] Junjie Shen, Yunpeng Luo, Ziwen Wan, and Qi Alfred Chen. Lateral-direction localization attack in high-level autonomous driving: Domain-specific defense opportunity via lane detection. *arXiv preprint arXiv:2307.14540*, 2023.
- [14] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [15] You Li and Javier Ibanez-Guzman. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020.

- [16] Yuxiao Zhang, Alexander Carballo, Hanting Yang, and Kazuya Takeda. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:146–177, 2023.
- [17] Xue-Bo Jin, Wei Chen, Hui-Jun Ma, Jian-Lei Kong, Ting-Li Su, and Yu-Ting Bai. Parameter-free state estimation based on kalman filter with attention learning for gps tracking in autonomous driving system. *Sensors*, 23(20):8650, 2023.
- [18] Zhenbo Liu, Leijie Wang, Feng Wen, and Hongbo Zhang. Imu/vehicle calibration and integrated localization for autonomous driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4013–4019. IEEE, 2021.
- [19] Jun Zhu, Hongyi Li, and Tao Zhang. Camera, lidar, and imu based multi-sensor fusion slam: A survey. *Tsinghua Science and Technology*, 29(2):415–429, 2023.
- [20] Yanbin Gao, Shifei Liu, Mohamed M Atia, and Aboelmagd Noureldin. Ins/gps/lidar integrated navigation system for urban and indoor environments using hybrid scan matching algorithm. *Sensors*, 15(9):23286–23302, 2015.
- [21] Zui Tao, Ph Bonnifait, Vincent Fremont, and Javier Ibanez-Guzman. Mapping and localization using gps, lane markings and proprioceptive sensors. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 406–412. IEEE, 2013.
- [22] Shenlan msf. https://github.com/shenlan2017/Apollo_ShenLan.
- [23] Joan Sola. Quaternion kinematics for the error-state kalman filter. *arXiv preprint arXiv:1711.02508*, 2017.
- [24] Aboelmagd Noureldin, Tashfeen B Karamat, and Jacques Georgy. *Fundamentals of inertial navigation, satellite-based positioning and their integration*. Springer Science & Business Media, 2012.
- [25] Aceinna. GNSS-INS-SIM: An Open-source GNSS/INS Simulation Platform. <https://github.com/Aceinna/gnss-ins-sim>.
- [26] Junxiang Jiang, Xiaoji Niu, and Jingnan Liu. Improved imu preintegration with gravity change and earth rotation for optimization-based gnss/vins. *Remote Sensing*, 12(18):3048, 2020.
- [27] Feng Sun, Haiyu Lan, Chunyang Yu, Naser El-Sheimy, Guangtao Zhou, Tong Cao, and Hang Liu. A robust self-alignment method for ship’s strapdown ins under mooring conditions. *Sensors*, 13(7):8103–8139, 2013.
- [28] Arvo Kaldmäe and Ülle Kotta. A note on observability of nonlinear discrete-time systems. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 7483–7488. IEEE, 2023.
- [29] Yifeng Li and Jiandong Zhu. Observability decomposition of boolean control networks. *IEEE Transactions on Automatic Control*, 68(2):1267–1274, 2022.
- [30] DRORA Goshen-Meskin and IY Bar-Itzhack. Observability analysis of piece-wise constant systems. ii. application to inertial navigation in-flight alignment (military applications). *IEEE Transactions on Aerospace and Electronic systems*, 28(4):1068–1075, 1992.
- [31] Stephen M Tanny and Michael Zuker. The sensitivity of eigenvalues under elementary matrix perturbations. *Linear Algebra and its Applications*, 86:123–143, 1987.
- [32] Paulo Manrique-Mirón. Condition number of random tridiagonal toeplitz matrix. *arXiv preprint arXiv:2305.11971*, 2023.
- [33] Chris J Dafis and Chika O Nwankpa. Characteristics of degree of observability measure for nonlinear power systems. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 68b–68b. IEEE, 2005.
- [34] Fredric M Ham and R Grover Brown. Observability, eigenvalues, and kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, (2):269–273, 1983.
- [35] D Goshen-Meskin and IY Bar-Itzhack. Observability analysis of piece-wise constant systems with application to inertial navigation. In *29th IEEE Conference on Decision and Control*, pages 821–826. IEEE, 1990.
- [36] Federal Highway Administration. Speed concepts: Informational guide, 2010. Accessed: [2024-08-06].
- [37] Marc Green. Driver reaction time, 2000. Accessed: 2024-09-12.
- [38] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 2267–2281, 2019.
- [39] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13716–13725, 2020.

- [40] Yi Zhu, Chenglin Miao, Tianhang Zheng, Foad Hajjaghajani, Lu Su, and Chunming Qiao. Can we use arbitrary objects to attack lidar perception in autonomous driving? In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1945–1960, 2021.
- [41] Zizhi Jin, Xiaoyu Ji, Yushi Cheng, Bo Yang, Chen Yan, and Wenyuan Xu. Pla-lidar: Physical laser attacks against lidar-based 3d object detection in autonomous vehicle. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1822–1839. IEEE, 2023.
- [42] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks. In *2017 IEEE European symposium on security and privacy (EuroS&P)*, pages 3–18. IEEE, 2017.
- [43] Xiaoyu Ji, Yushi Cheng, Yuepeng Zhang, Kai Wang, Chen Yan, Wenyuan Xu, and Kevin Fu. Poltergeist: Acoustic adversarial machine learning against cameras and computer vision. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 160–175. IEEE, 2021.
- [44] Shoei Nashimoto, Daisuke Suzuki, Takeshi Sugawara, and Kazuo Sakiyama. Sensor con-fusion: Defeating kalman filter in signal injection attack. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 511–524, 2018.
- [45] Hyungsub Kim, Rwitam Bandyopadhyay, Muslim Ozgur Ozmen, Z Berkay Celik, Antonio Bianchi, Yongdae Kim, and Dongyan Xu. A systematic study of physical sensor attack hardness. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 143–143. IEEE Computer Society, 2024.
- [46] Michael Grupp. evo: Python package for the evaluation of odometry and slam, 2017. Accessed: 2024-07-06.

A Appendix

A.1 Fusion Precision Evaluation

For vulnerability analysis, it is crucial to have a robust fusion localization simulation framework that can reflect the real-world conditions. This section examines the precision of MSAF’s fusion localization, comparing its accuracy across various simulated noise levels and real-world conditions.

Experimental setup. We evaluate MSAF’s localization accuracy using both simulated and real-world data. The sensor frequencies are configured as 100 Hz for IMU and GPS, and 10 Hz for LiDAR, while GPS position operates at 5 Hz.

The simulated scenarios include straight and turning trajectories across three noise levels: noise-free, high-accuracy, and middle-accuracy. For the middle-accuracy configuration, the IMU and GPS noise parameters are calibrated to replicate the stochastic characteristics of the integrated navigation system employed in the experimental vehicle, as described in Section 6.5. For straight scenarios, the system maintains a constant speed of 3 m/s, while in turning scenarios, it operates at 3 m/s with an angular speed of 3 deg/s. Each experiment lasts 40 seconds, with the initial 3 seconds excluded to ensure data reliability and account for the Kalman-filter convergence period. For both simulated and real-world datasets, we use the open-source evaluation framework evo [46] to assess the Relative Pose Error (RPE). We calculate the Root Mean Square Error (RMSE) by comparing our results to the ground truth data formatted in the style of the KITTI dataset. The results can be found in Table 5.

Table 5: MSAF localization accuracy across different noise levels in two conditions

Case	Noise-free(m)	High(m)	Mid(m)	Real-World(m)
Straight	4.0×10^{-6}	8.8×10^{-3}	3.7×10^{-2}	4.4×10^{-2}
Turning	7.2×10^{-5}	8.4×10^{-3}	3.7×10^{-2}	2.0×10^{-2}

Results. In both simulated and real-world cases, MSAF demonstrates similar localization performance. The accuracy in the simulated environment with the middle-accuracy noise setting closely aligns with the results from the real-world test. These findings emphasize the consistency and reliability of MSAF across both simulated and real-world conditions.

A.2 Dynamics of Error Covariance

Figure 15 illustrates the error covariance P_{00} at 10 seconds for straight_vel scenarios. Here, P_{00} is the first element of the 15x15 error covariance matrix P , representing the uncertainty in the x -axis position estimate. Notably, in the correction phase (red squares), the error covariance decreases compared to the prediction phase (blue circles), demonstrating the Kalman filter’s effectiveness in reducing uncertainty by incorporating GPS and LiDAR measurements. However, despite this reduction, P_{00} still shows an increasing trend with higher velocities, highlighting the persistent need for sensor measurements to counteract the growing internal uncertainty at higher speeds.

A.3 Dynamics of Velocity-Offset

When the vehicle cruises at a uniform speed, a higher velocity correlates with a more substantial offset. Specifically, exceeding speeds of 15 m/s results in offsets exceeding the expected value by 15.46%, as demonstrated in Figure 17. Following

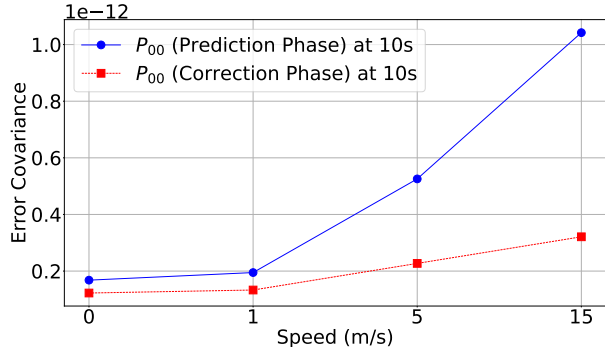


Figure 15: Error covariance P_{00} in prediction (blue) and correction (red) phases at 10 seconds in straight_vel scenarios.

this observation, as the vehicle speed steadily increases and exceeds 15 m/s, the offset growth rate decelerates, eventually stabilizing around 16% above the expected value, as demonstrated in Figure 16.

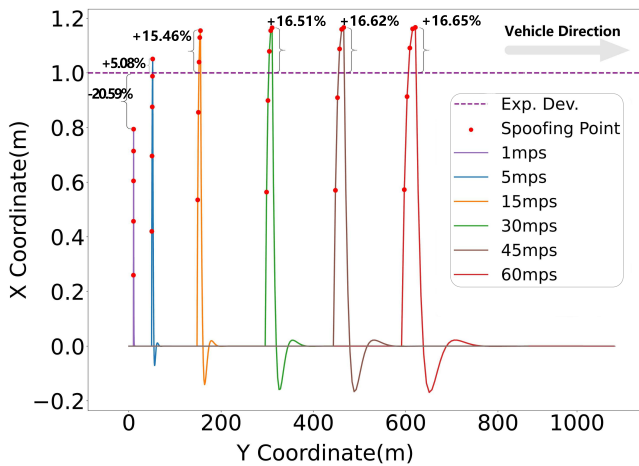


Figure 16: The offset initially increases with speed but begins to converge around 15 m/s, stabilizing at approximately 16% above the expected value.

A.4 Ablation Study of Parameter Effects.

Ablation experiments were performed to evaluate the impact of attack parameters relative to attack strategies on the efficacy of GPS spoofing. Parameters were strategically chosen to include FusionRipper’s three optimal sets [10] and our best-performing parameters. Additionally, an intermediate set with $d = 0.2$ and $f = 1.3$ was evaluated to bridge the gap between the two extremes and observe its effect on attack success. These selections aimed to explore the range of positional offsets an attacker might attempt to inject. The parameters were tested in real-time against two distinct scenarios, with results presented in Figure 18.

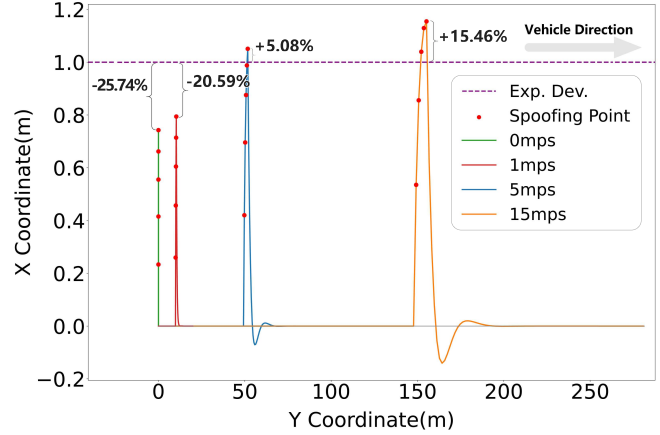


Figure 17: Injection results in straight_vel scenarios

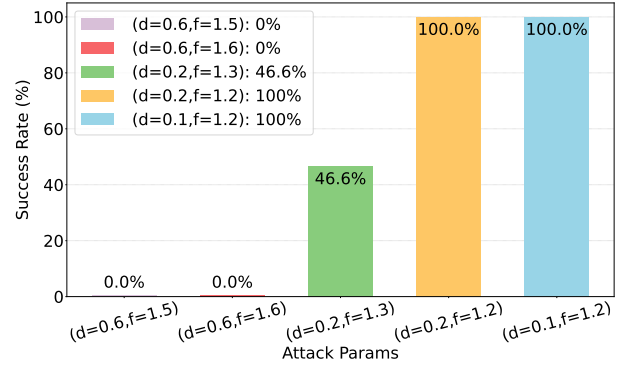


Figure 18: Success rate under different attack parameters.

The results distinctly show that FusionRipper’s optimally selected parameter sets did not achieve any success, recording a 0% success rate across both strategies. In contrast, our optimally selected parameters accomplished a 100% success rate in each scenario. The aforementioned intermediate parameter set achieved a success rate of 46.6%, underscoring the nuanced influence of parameter adjustments. These findings highlight the importance of selecting a minimal initial offset to enable the ESKF to smoothly adapt to GPS data deviations, which can lead to more effective and stealthy spoofing attacks.

A.5 ESKF process

Under the Error-State Kalman Filter (ESKF) framework, data from the IMU, GPS, and LiDAR locator are integrated into a unified estimation process to improve the inferred accuracy of vehicle pose and velocity. Unlike traditional Kalman filters, the ESKF explicitly models and tracks the error terms of key states, such as position, velocity, and orientation. This design better manages drift and accumulated errors when fusing both high-frequency and low-frequency sensor measurements. As illustrated in Fig. 19, the ESKF simultaneously uses high-frequency accelerations and angular velocities from the IMU

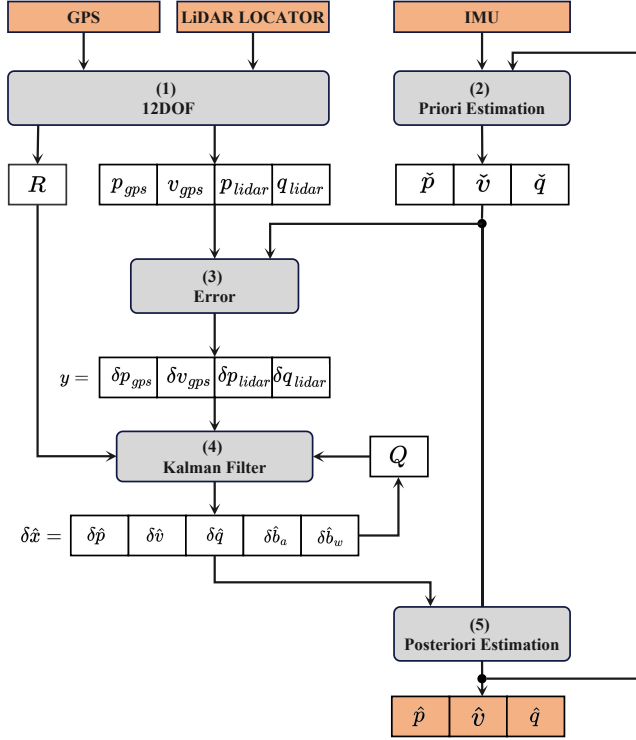


Figure 19: Dataflow of ESKF-based sensor fusion engine

for prior state prediction, and less frequent but more precise observations from GPS and LiDAR for state correction, ultimately achieving a smooth yet reliable state estimation.

In practical execution, the ESKF proceeds in five main steps: (1) GPS and LiDAR sensors provide 12 degrees of freedom (DOF) observations—including position p , velocity v , and orientation q —along with the corresponding observation noise R , serving as a high-accuracy reference for subsequent corrections; (2) the IMU outputs accelerations and angular velocities, which drive the prior state prediction step based on the previous state and system motion model; (3) the predicted state is compared against sensor measurements, yielding the residual y ; (4) the Kalman filter incorporates this residual, as well as process noise Q and observation noise R , to update the error state $\delta\hat{x}$, thereby correcting any drift and uncertainty accumulated during the prediction phase; (5) finally, the corrected error state $\delta\hat{x}$ is fed back into the prior estimate to obtain the posterior estimates of position p , velocity v , and orientation q . By iterating this predict–correct cycle, the ESKF suppresses accumulative error and achieves real-time, high-precision estimation of vehicle motion in dynamic environments.

A.6 List of Symbols

Table 6 provides an overview of the primary notation employed in the ESKF model and the corresponding vulnerabil-

Table 6: Notations in ESKF Model and Vulnerability Analysis

Stage	Notation	Description
ESKF	$\delta\mathbf{x}$	Error state (errors in $\mathbf{p}, \mathbf{v}, \mathbf{q}, \mathbf{b}_a, \mathbf{b}_w$)
	\mathbf{B}	Control input matrix
	\mathbf{F}	State transition matrix
	\mathbf{Q}	Covariance matrix of process noise
	\mathbf{R}	Covariance matrix of observation noise
	\mathbf{P}	Error covariance
	\mathbf{K}	Kalman gain
	\mathbf{G}	Observation matrix
Analysis	$\boldsymbol{\omega}$	Earth’s angular velocity
	$\boldsymbol{\delta}$	Spoofing offset
	\mathbf{Q}_{som}	Accumulated observation matrix
	\mathbf{y}_{som}	Accumulated observation vector
	\mathbf{U}, \mathbf{V}	SVD singular vector matrices
	\mathbf{S}	SVD singular value matrix
	\mathbf{X}	Observation matrix

ity analysis. It details both the variables central to the filtering process and the key parameters used to evaluate adversarial spoofing scenarios, offering a concise reference for understanding the mathematical framework of MSAF.