SPOOFING ATTACK TECHNOLOGY OF AUTONOMOUS VEHICLE MULTI-SENSOR NAVIGATION SYSTEM

JIACHONG CHANG

PhD

The Hong Kong Polytechnic University

This programme is jointly offered by The Hong Kong Polytechnic University and Harbin Institute of Technology

2025

The Hong Kong Polytechnic University
Department of Aeronautical and Aviation Engineering
Harbin Institute of Technology
School of Instrumentation Science and Engineering

Spoofing Attack Technology of Autonomous Vehicle Multi-Sensor Navigation System

Jiachong Chang

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

August 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

Jiachong CHANG (Name of student)

Abstract

In recent years, autonomous driving technology has been rapid advancements and has become a critical development in next-generation vehicle technology. Autonomous vehicles (AVs) require centimeter-level positioning accuracy for safe navigation. With the rapid advancement of autonomous driving technology, research on spoofing attacks against the multi-sensor fusion (MSF) systems of AVs has garnered widespread attention. Spoofing attacks refer to the behavior of transmitting false signals to satellite receivers. The navigation systems of AVs typically consist of multiple sensors, including light detection and ranging (LiDAR), inertial navigation systems (INS), and global navigation satellite systems (GNSS). MSF systems can suppress spoofing signals, thereby increasing the difficulty of successfully executing spoofing attacks on AVs. In-depth research on spoofing technology can effectively expose existing vulnerabilities in the field of AVs, which is of great significance for advancing anti-spoofing technology for AV navigation in complex environments.

Current research primarily focuses on generating GNSS spoofing signals and designing and defending spoofing algorithms based on GNSS/INS integrated navigation systems. However, prior studies have not provided a detailed analysis of spoofing MSF behavior, such as the development of analytical models and error mechanism analysis. Since the navigation systems of AVs are typically composed of multiple navigation sensors, traditional spoofing methods are easily detected by the system, resulting in failed deception attempts. AVs may operate in diverse geographical and weather environments. Conventional approaches have not fully evaluated the effectiveness of spoofing, and there is an issue of indiscriminately broadcasting spoofing signals. Therefore, in response to these unresolved challenges, this thesis conducts in-depth research. The main research content is as follows:

Targeting the issue of traditional error propagation models based on Kalman filters, which are complex in derivation due to differences in sensor update frequencies and numerous inversion operations, and where the mathematical relationship between state errors and system parameters is unclear under spoofing attacks, this study investigates a state error mechanism based on a lightweight information filter model. First, considering the impact of varying sensor update frequencies, an error state Kalman filter analytical model is established. The measurement update processes of GNSS and LiDAR, as well as the INS state recursion process, are derived. The primary factors leading to increased state errors are analyzed. In addition to the initial state covariance matrix and LiDAR uncertainty, the uncertainty of GNSS and the update frequency of measurement sensors are also key factors

affecting the spoofing success rate. To avoid the complex inversion operations in multiple LiDAR measurement updates within one GNSS update cycle, an analytical model based on a lightweight information filter is established. Finally, the state error vector update process is transformed into an information vector update process, and inversion operations in the information vector update process are avoided by disregarding the INS recursion update process. The state error propagation model has a clearer mathematical expression, intuitively reflecting the mathematical relationship between state errors and system parameters, thereby providing a theoretical foundation for research on spoofing technology in navigation systems across various environments.

Addressing the challenge that traditional spoofing methods struggle to adapt spoofing parameters according to spoofing effects, resulting in low spoofing success rates, a covert spoofing method based on a fuzzy inference model is proposed. This method involves monitoring the target AV, calculating real-time position error feedback adjustment factors, constructing fuzzy knowledge bases and fuzzy rules, and dynamically adjusting spoofing parameters using the multi-Zadeh method to improve spoofing success rates. By comparing position error feedback adjustment factors before and after real-time adjustments, the method determines whether the spoofing process has triggered the take-over effect. If the take-over effect is triggered, constraints on the maximum values of spoofing parameters are enforced. Real-world data test results demonstrate that the proposed method achieves a spoofing success rate 5% higher than traditional methods in typical test scenarios.

Regarding the issue that traditional spoofing technologies do not evaluate the effectiveness in complex geographical and adverse weather conditions and blindly launch spoofing attacks, this study investigates a spoofing effectiveness evaluation method based on sensor uncertainty estimation. For complex geographical scenarios, a three-dimensional building model of the target area is constructed. A sky visibility mask is generated based on the maximum elevation angle edge of the building model, and sky visibility and the number of visible satellites are estimated. A kernel partial least squares nonlinear regression model is established to estimate GNSS uncertainty, analyzing the relationship between sky visibility and spoofing success rates. For various weather scenarios, the impact of weather on LiDAR performance is evaluated through meteorological pulse response functions at different weather levels, including rain, snow, and fog. A LiDAR uncertainty estimation method based on a B-spline regression model is developed, and the relationship between different weather levels and spoofing success rates is quantitatively analyzed. The results indicate that the proposed method can evaluate whether the AV's environment is conducive to spoofing. Real-world data test results show that in different geographical scenarios,

when the proposed method determines the scenario is easy, the spoofing success rate exceeds 70%, outperforming traditional methods. In various weather scenarios, when the proposed method determines that the scenario is easy, the spoofing success rate exceeds 57%.

This study provides a theoretical foundation for designing defense algorithms in the event of potential malicious spoofing attacks, facilitating further research into GNSS anti-spoofing algorithms. Thereby, it enables vehicles to implement proactive measures, such as activating emergency plans, slowing down, or even stopping in the event of safety risks. Immediate restoration of MSF system performance ensures vehicle safety and reduces the risk of catastrophic traffic accidents. Ultimately, this study ensures the safety of the MSF system in various scenarios.

List of Publications

[1] **Chang Jiachong**, Zhang Liang, Hsu Li-Ta, Huang Feng, Xu Dingjie, et al., Analytic Models of a Loosely Coupled GNSS/INS/LiDAR Kalman Filter Considering Update Frequency Under a Spoofing Attack[J]. IEEE Sensors Journal, 2022. (SCI Index, IF=4.5)

[2] **Chang Jiachong**, Zhang Ya, Fan Shiwei, Xu Dingjie, Hsu Li-Ta, et al., An Anti-spoofing Model based on MVM and MCCM for a Loosely-coupled GNSS/INS/ LiDAR Kalman Filter[J]. IEEE Transactions on Intelligent Vehicles, 2023. (SCI Index, IF=14.3)

[3] **Chang Jiachong**, Huang Feng, Zhang Liang, Xu Dingjie, Hsu Li-Ta, et al., Selection of Areas for Effective GNSS Spoofing Attacks to a Vehicle-mounted MSF System based on Scenario Classification Models[J]. IEEE Transactions on Vehicular Technology, 2023. (SCI Index, IF=7.1)

[4] **Chang Jiachong**, Hu Runzhi, Huang Feng, Xu Dingjie, Hsu Li-Ta, et al., LiDAR-based NDT Matching Performance for Positioning in Adverse Weather Conditions[J]. IEEE Sensors Journal, 2023. (SCI Index, IF=4.5)

[5] **Chang Jiachong**, Fan Shiwei, Zhang Ya, Xu Dingjie, et al., A time asynchronous parameters calibration method of high-precision FOG-IMU based on a single-axis continuous rotation scheme[J]. Measurement Science and Technology, 2023. (SCI Index, IF=3.4)

[6] **Chang Jiachong**, Zhang Ya, Qian Meng,Li Guangmin, Hsu Li-Ta, A Covert Spoofing Attack Method based on the Fuzzy Inference Model for GNSS/INS/LiDAR-based Autonomous Vehicles[J]. lEEE Transactions on Intelligent Transportation Systems, 2025. (Under review)

[7] Li Jingchun, **Chang Jiachong**, Zhang Ya. Gradient descent optimization-based SINS self-alignment method and error analysis[J]. IEEE Access, 2021, 9: 8286-8298. (SCI Index, IF=3.6)

[8] **Chang Jiachong**, Zhang Ya, Wang Zhuo, et al., Optimal inner lever-arm parameters calibration method of high-precision FOG-IMU based on sinusoidal swing scheme[C]. 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, 2020: 734-739.

Acknowledgements

First and foremost, my deepest gratitude goes to my supervisor, Dr. HSU, for an experience that profoundly shaped my doctoral journey. Thank you for generously providing me with the invaluable opportunity to conduct research in Hong Kong. The year and a half I spent there stands as the most precious treasure of my PhD life. Beyond his immense guidance on my core research topic, Dr. HSU's mentorship extended far deeper. He personally guided me through the intricacies of academic writing, patiently taught me the fundamental principles and practices of rigorous scientific research, and consistently offered unwavering support and care that went well beyond the academic realm. His profound insights and exemplary approach have left an indelible mark on my personal values and outlook on life. I am eternally grateful for his exceptional kindness and dedication.

I would also like to express my sincere appreciation to my supervisor at Harbin Institute of Technology (HIT), Professor Gao Wei. His steadfast guidance and support have been instrumental throughout my entire graduate studies, from my Master's degree right through to the completion of my PhD. Professor Gao provided indispensable assistance at critical junctures, offering wisdom, resources, and encouragement whenever needed. His broad-minded character, generosity of spirit, and optimistic perspective are truly admirable and serve as a constant source of inspiration, qualities I shall always strive to emulate.

Special thanks are owed to Dr. Huang Feng for his crucial contributions to my research. His willingness to share significant datasets proved invaluable, saving me considerable time and effort by helping me avoid potential detours in my experimental work. I am also deeply thankful to Runtian Hu, Yihan Zhong, Liang Zhang, Liyuan Zhang, and Jiachen Zhang for their camaraderie and support, enriching both my academic pursuits and daily life with their friendship and assistance. Finally, I extend my gratitude to all the members of the Intelligent Positioning and Navigation Laboratory (IPNL) for fostering a collaborative and stimulating research environment.

Contents

List of Figures

List of Tables

List of Abbreviations

In this dissertation, bold characters are used to represent vector and matrix variables, and italic characters are used to represent scalar variables. The main abbreviations are as follows.

Table 1: List of main abbreviations

Abbreviation	Complete Definition
AVs	autonomous vehicles
MSF	multi-sensor fusion
LiDAR	light detection and ranging
INS	inertial navigation system
GNSS	global navigation satellite system
RF	radio frequency
UAV	unmanned aerial vehicle
SINS	strap-down inertial navigation system
USV	unmanned surface vehicle
GPS	global positioning system
LPH	LiDAR point-cloud height
IMU	inertial measurement unit
MMFIM	multiple multidimensional fuzzy inference model
FITA	first infer then aggregate
RMSE	root mean square error
RTK	real time kinematic
3DMA	3D map aided
KPLSR	kernel partial least squares regression
ENU	eastward northward upward
KF	Kalman filter
SSR	regression sum of Ssquares
SSE	error sum of squares

Abbreviation	Complete Definition
PLSR	partial least squares regression
SONR	second order nonlinear regression
LS-SVR	least squares support vector regression
NDT	normal distribution transformation

Chapter 1

Introduction

1.1 Research Background and Motivation

With the innovation of autonomous vehicles (AVs) and the continuous breakthroughs in key technologies, their role in future human life is gradually increasing. As AVs develop, ensuring their safety becomes increasingly important. In recent years, spoofing technology targeting navigation and localization systems has garnered significant attention from researchers as a critical threat to these systems.

Spoofing attacks refer to the broadcasting of false global navigation satellite system (GNSS) signals by a spoofing source to a target receiver, infiltrating the receiver's signal acquisition and tracking loop module and deceiving the target receiver into localizing to an erroneous location. Existing spoofing attack techniques inadequately investigate the error transfer mechanism of multi-sensor fusion (MSF) systems. Thus, analyzing the state error transfer mechanism under spoofing attacks in depth to identify key influencing factors for successful spoofing is of great significance.

The navigation system of AVs typically comprises multiple navigation sensors, and MSF systems can effectively detect and suppress spoofing signals through defense algorithms. Therefore, improving the covertness of spoofing technology is crucial for effective deception. Additionally, AVs can localize the orientation and position of spoofing sources using spoofing source localization technology. Broadcasting spoofing signals under unsuitable conditions may expose the spoofing source, threatening its security. Hence, selectively broadcasting spoofing signals to avoid detection by AVs is crucial for ensuring successful spoofing rates.

However, research on spoofing techniques for AV MSF systems remains inadequate. In terms of error transfer modeling under spoofing attacks, existing techniques lack sufficient study on MSF error transfer models. Regarding covert spoofing techniques, the key lies in dynamically adjusting spoofing parameters to enhance covertness and reduce detection risks. With respect to spoofing effectiveness evaluation, AVs operate in diverse geographical and weather environments. Evaluating the effectiveness of spoofing in these

scenarios to improve success rates remains a technical challenge. Therefore, the above issues remain technical challenges in the field of spoofing attacks.

This thesis begins with the demand to improve the spoofing success rate on AVs. We first research the error mechanism of spoofing attacks, considering the influence of different sensor update frequencies. We then establish the Kalman filter analysis model of the error state and the lightweight information filter analysis model and analyze the main factors leading to state error under spoofing attacks in detail. The state error transfer mechanism under a spoofing attack is studied in detail, and the main factors leading to state errors under a spoofing attack are analyzed. Then, we present a covert spoofing method based on a fuzzy inference model, dynamically adjusting the spoofing parameters according to the state change of the target AV to improve the spoofing success rate. Finally, a spoofing effectiveness assessment method based on sensor uncertainty estimation is proposed to evaluate the difficulty of spoofing in various geographic and weather scenarios by estimating the uncertainty of navigation sensors, thereby avoiding the implementation of spoofing attacks in high-spoofing-difficulty scenarios. In summary, this thesis presents additional strategies and foundations for researching spoofing technology for AVs in complex environments, thereby providing a stronger theoretical basis for AV anti-spoofing technology research.

1.2 Review of Related Research

Spoofing attack technology involves a spoofing source broadcasting false satellite signals to a target receiver, infiltrating the receiver's baseband signal processing module, and guiding the target to an erroneous location [1, 2, 3]. This thesis reviews the current research status of spoofing technology in the following aspects: spoofing attacks on AV MSF systems, covert spoofing techniques, and spoofing effectiveness evaluation.

1.2.1 Review of Spoofing Attacks on AV MSF Systems

Due to cost constraints and positional accuracy requirements, AV navigation and positioning systems commonly integrate GNSS, INS, and LiDAR [4, 5, 6]. AV systems establish high-accuracy and high-reliability MSF frameworks, leveraging the advantages of each sensor to achieve efficient fusion of multiple navigation sensors, thereby improving navigation accuracy and robustness [7]. As the core of AV navigation and positioning systems, MSF systems provide real-time high-accuracy and high-reliability positioning results. In

recent years, with the rapid development of military and civilian AV industries, MSF algorithms based on Kalman filter models have been widely adopted [8, 9]. These algorithms fuse navigation information from multiple sensors using recursive error formulas to achieve optimal estimation of navigation state parameters, ultimately yielding high-precision navigation results [10].

Although various MSF frameworks exist, including factor graph optimization [11, 12, 13], all-source navigation systems [14, 15, 16], and artificial intelligence-based navigation systems [17], their computational complexity is relatively high [18]. Due to limited computational resources within AVs, Kalman filter-based MSF algorithms remain widely used in practical applications. Examples include the Autoware and Apollo frameworks, which are widely used self-driving frameworks currently [19].

Shinpei Kato of Nagoya University developed the Autoware Autonomous Driving Platform, the world's first open-source AV software [20, 21]. Its first version was released in 2015, built on the ROS platform. The framework is primarily suited for urban environments but also covers other geographical areas such as highways and suburbs. The purpose of developing the Autoware platform is to provide a simulation environment based on navigation data for users without AVs, enabling them to formulate safety measures before on-site testing and assess potential risks. The architecture and interaction interface of Autoware are shown in Fig. 1.1 [22].

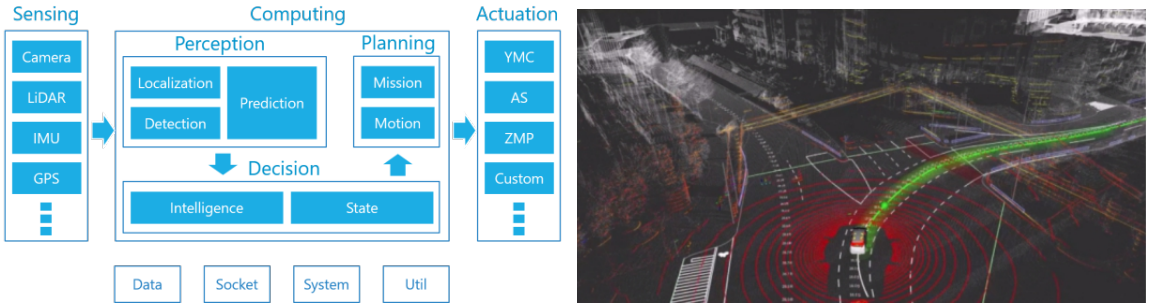

Figure 1.1: The architecture and interaction interface of Autoware

In 2017, Baidu released the Apollo platform to help automotive industry partners and autonomous driving developers rapidly build fully autonomous driving systems [23]. The platform provides access to sources of environmental perception, path planning, vehicle control, and vehicle operating systems, offering comprehensive development and testing tools [24].

AVs require MSF systems to achieve centimeter-level positioning accuracy. GNSS, as a critical source of absolute navigation information, plays an indispensable role in AV

navigation and positioning systems [25, 26]. Spoofing attacks aim to induce AVs to deviate from their intended lanes or even drive off course by broadcasting false GNSS signals, causing incorrect positioning estimates and potentially leading to traffic accidents [27, 28]. Most current research on spoofing attack techniques focuses on generating spoof signals and designing and defending spoofing algorithms based on combined GNSS/INS navigation systems [29]. Research on spoofing attack techniques for AV MSF systems remains in its infancy.

To achieve spoofing attacks on MSF systems, advanced spoofing techniques are typically employed. Radio frequency (RF)-based spoofing attacks are highly covert and complex, making them difficult for target receivers to detect and prevent [30]. First, transmitters deployed for spoofing attacks need to be considered, including the number of transmitters and their spatial locations. Attackers capture real GNSS signals, copy, modify, or delay them, and forward these modified signals to the target receiver, forcing it to lock onto the false signals [31]. Once captured, the attacker strategically manipulates the target receiver's navigation information, including position, velocity, and time. To achieve effective spoofing attacks, the forged RF signals must be highly synchronized in time with real GNSS signals, which typically requires attackers to possess precise time control capabilities and advanced signal processing techniques [32, 33]. However, many current spoofing attack algorithms may fail to spoof MSF systems, even after successfully spoofing GNSS, due to the presence of other sensors, such as INS and LiDAR. Industrial-grade MSF frameworks (e.g., Baidu Apollo) are inherently robust and highly resistant to GNSS spoofing attacks, representing a current technical challenge for spoofing attacks on MSF navigation systems [34, 32].

In 2020, researchers at the University of California, Irvine, conducted the first study on AV MSF system spoofing technology and designed a GNSS/SINS/LiDAR-based MSF spoofing algorithm 'Fusion-ripper' [35, 36]. The algorithm analyzes potential vulnerabilities of AV MSF systems, such as external attacks or interferences (e.g., hacker attacks, malware), which may disrupt system operation, leading to data leakage or system crashes. A spoofing model based on maximizing lateral deviation is designed, implementing spoofing attacks on AVs in two stages: vulnerability analysis and enhanced attacks, as shown in Fig. 1.2 . Security thresholds are set for each stage, with the technical means essentially involving constant-value and exponential-value spoofing on the MSF system. Real-world test data verifies that the Fusion-ripper spoofing algorithm can exploit vulnerabilities in MSF systems to achieve successful spoofing.

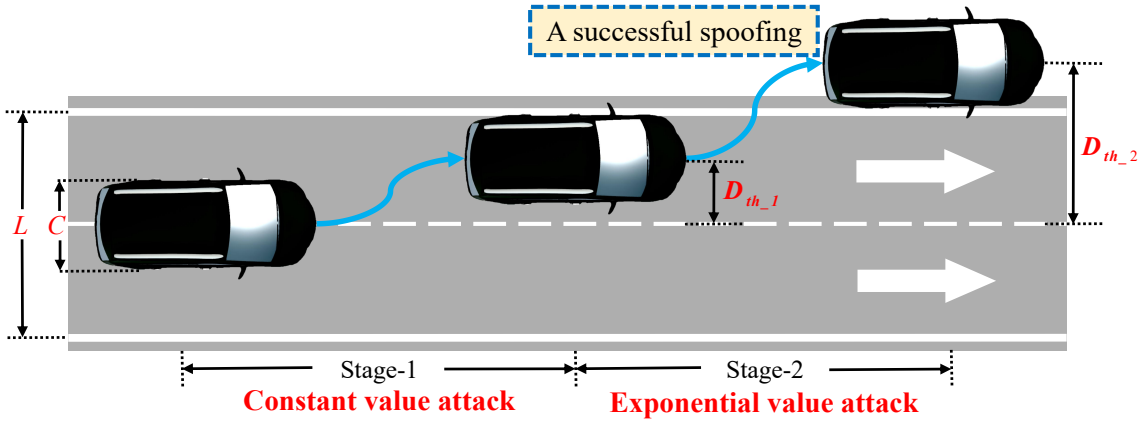

Figure 1.2: Fusion-ripper spoofing algorithm diagram

The analysis indicates that the key to the model's success lies in triggering the take-over effect. During spoofing attacks on GNSS/SINS/LiDAR-based MSF systems, GNSS input signals become the primary Kalman filter quantitative information update source for the navigation system. Meanwhile, LiDAR positioning results are mistakenly detected as outliers by the MSF system's defense algorithm and are no longer corrected due to GNSS spoofed signals. This is referred to as the take-over effect. The take-over effect fundamentally violates the design principle of AV MSF systems, which aim to fuse multiple navigation information sources to achieve higher robustness and accuracy. Using spoofing success rate as the evaluation metric, final simulation tests demonstrate that the method achieves a spoofing success rate exceeding 91% under generalized test conditions and over 74% under high GNSS uncertainty settings.

However, the Fusion-ripper spoofing attack algorithm has shortcomings in establishing the error transfer model. The scheme simplifies the analysis of error mechanism models. It neglects the differences in update frequencies of each sensor in actual MSF systems, leading to incomplete analysis of the state error transfer mechanism. Consequently, some key influencing factors are not reflected in the final results.

1.2.2 Review of Covert Spoofing Techniques

Covert spoofing techniques ensure that spoofing signals sent during attacks are difficult for the target system to detect. Improving covertness is critical for spoofing technology. If spoofing signals are detected, the target may increase vigilance, reduce the credibility of GNSS signals, or discard them entirely in subsequent navigation processes, thereby complicating subsequent spoofing attempts. Additionally, upon detecting spoofing interference, the target can localize the spoofing source using techniques such as direction of arrival

[37], time difference of arrival [38], signal propagation correction [39], and received signal strength[40], etc., potentially exposing the spoofing source's location.

Covert spoofing techniques require precise information about the spoofing target to avoid triggering alarms. However, in real-world scenarios, especially at long distances, these conditions are often unmet. In designing covert spoofing frameworks for GNSS receivers, a spoofing framework capable of real-time, covert GNSS spoofing is developed. This framework adjusts key parameters of the spoofed signal, including code phase, Doppler shift, signal strength, and navigation data bits [41]. Its observation module, comprising a LiDAR and an attitude heading reference system, provides complementary information to validate the effectiveness of spoofing. Real-world spoofing experiments on GNSS receivers demonstrate the actual threat of this covert spoofing framework to GNSS.

Different covert spoofing schemes have been designed for various application backgrounds. In the context of unmanned aerial vehicles (UAVs), covert spoofing algorithms for GNSS/SINS navigation systems have been proposed to mislead target UAVs to incorrect locations [42, 43]. The algorithm theoretically proves that when the acceleration component of the false GNSS signal is used as the difference between the UAV's current acceleration and the spoofing control input, the UAV can be spoofed. To avoid detection of spoofing signals or UAV crashes during spoofing, the algorithm requires the spoofing trajectory planned by the source to change slowly relative to the UAV's predefined reference trajectory. Simulation results validate the effectiveness of the UAV covert spoofing algorithm. However, this method focuses solely on UAV position without fully considering the effects of velocity and attitude on spoofing attacks. To address this, a covert spoofing method based on a steady-state gain matrix is proposed [44]. This method explores the mechanism of spoofing signals affecting the position output of the integrated GNSS/SINS navigation system, analyzing the impact of spoofing signals on the attitude and velocity outputs of the navigation system. It introduces velocity and attitude error-related constraints, determines the GNSS exponential spoofing signal model that meets covertness requirements and achieves covert directional spoofing of target unmanned platforms. Essentially, this method employs a two-parameter adjustable exponential spoofing signal for directional spoofing attacks. In 2024, researchers from the Beijing Institute of Technology proposed a new covert spoofing algorithm for UAV-integrated navigation systems, incorporating deep reinforcement learning to dynamically solve effective navigation spoofing positions in real time [45]. To ensure the generated navigation spoofing positions meet covertness requirements, the algorithm achieves a spoofing success rate exceeding 68% under noise levels below std6.

UAVs typically integrate both GNSS and SINS navigation sensors, making spoofing

more challenging. However, these studies provide a technical foundation for developing covert spoofing techniques against UAVs. In research on covert spoofing attacks on unmanned surface vehicles (USVs), scholars have proposed spoofing techniques for USV MSF systems [46]. USV navigation systems include GNSS, LiDAR, SINS, and other sensors. Researchers analyzed how spoofing affects the automatic guidance, navigation, and control systems of USVs. A comprehensive spoofing scheme for USVs was designed, establishing five spoofing modes: constant lateral deviation spoofing, slow-varying lateral deviation spoofing, constant latitude deviation spoofing, slow-varying latitude deviation spoofing, and replay spoofing. The study detailed how spoofing navigation estimates affect the closed-loop control loop of USVs. Comparing USV localization profiles in spoofing and safety scenarios revealed that spoofing signals were undetectable by the USV system. The spoofing scheme successfully deviated USVs from their intended trajectories, validating its effectiveness. Although this research differs in application background from AV MSF system spoofing, it offers valuable insights for algorithm design in AV MSF system spoofing attacks.

In the context of land vehicle spoofing, researchers have developed various spoofing trajectory design schemes to reduce the likelihood of detection by target systems during satellite signal spoofing. Some scholars have proposed spoofing algorithms demonstrating that spoofing sources can successfully mislead land vehicles to locations far from their intended destinations without triggering system alerts [47]. By leveraging urban road network rules, the 'ESCAPE' algorithm generates potential spoofed routes with given start and end points, constructing a graph model of the road network and conducting active spoofing attacks to deceive target vehicles into traveling to potential destinations [48]. This algorithm can generate highly plausible spoofed trajectories even when navigation information is provided by SINS sensors, making integrated GNSS/SINS navigation systems vulnerable to spoofing. Similarly, to ensure spoofing signals remain undetected by target vehicles, the spoofing algorithm in [49] builds a graph model of the road network, offering significant flexibility to the spoofing source. Even when target vehicles use SINS-assisted GNSS for trajectory tracking and navigation, the system does not raise alarms. Experimental validation and evaluation of spoofing attack impacts in over 10 cities worldwide show that spoofing sources can lure target vehicles up to 30 km away from their real destinations without triggering onboard security defense systems. In [50], the effect of satellite navigation spoofing attacks on land vehicle navigation system parameters is analyzed, and the maximum spoofing offset is calculated while maintaining spoofing covertness. This value serves as the maximum distance for target point searching and determines path searching and matching constraints. Finally, a breadth-first search algorithm is used to obtain spoofed

satellite signals that satisfy the constraints, achieving a spoofing success rate of over 83% in real-world tests.

For AVs, GNSS serves as a critical source of absolute navigation information, playing a significant role in AV navigation and localization systems. The goal of spoofing attacks is to induce AVs to deviate from their intended trajectories by broadcasting false GNSS signals, causing incorrect positioning estimates and potentially leading to traffic accidents [51]. Factors such as the number of transmitters and their spatial deployment locations need to be considered [52]. Spoofing sources capture real GNSS signals, duplicate, modify, or delay them, and relay them to the target receiver, forcing it to lock onto the false signals [53]. Once the spoofed signals are captured, the spoofing source's objectives dictate how it manipulates the target receiver's navigation information, including position, velocity, and time. To achieve effective spoofing attacks, the spoofing source must ensure that the forged signals are highly synchronized in time with real GNSS signals, requiring precise time control and advanced signal processing capabilities. However, many current spoofing and jamming algorithms may fail to spoof MSF systems, even after successfully spoofing GNSS, due to the presence of positioning sensors such as SINS and LiDAR. Industrial-grade MSF frameworks (e.g., Baidu Apollo) are inherently robust and highly resistant to GNSS spoofing attacks, posing a significant technical challenge for spoofing attacks on AV MSF systems. Thus, in-depth research on covert spoofing methods for MSF systems is necessary to enhance spoofing success rates and covertness.

In terms of covertness, the traditional Fusion-ripper spoofing attack algorithm for AV MSF systems has several shortcomings. First, the algorithm combines two spoofing modes to improve success rates but neglects the consideration of covertness. It cannot adaptively adjust spoofing parameters, making spoofing signals easily detectable by AVs under poor GNSS signal quality conditions. Second, the algorithm uses a traversal method to set spoofing parameters. While this method identifies optimal spoofing parameters for theoretical research, it is impractical for real-world applications as it cannot determine optimal parameters in actual scenarios. Therefore, the Fusion-ripper algorithm is not suitable for real-world spoofing situations.

In summary, since GNSS is integrated with other navigation systems, such as SINS and LiDAR, in MSF systems, AVs can detect spoofing signals through their defense algorithms, thereby ignoring the false positioning information provided by the spoofing source. This presents a technical challenge for improving the covertness of spoofing attacks on MSF systems. Additionally, spoofing parameter settings are critical for enhancing covertness. Existing covert spoofing techniques cannot adaptively adjust parameters, leading to easily

detectable spoofing signals by MSF systems, which fail to meet the requirements of covert spoofing. Thus, determining reasonable spoofing parameters for AV MSF systems based on GNSS/SINS/LiDAR integration is key to improving covertness.

1.2.3 Review of Spoofing Effectiveness Evaluation

Spoofing effectiveness evaluation determines the difficulty of implementing spoofing attacks in practical scenarios. Due to its various scenarios, intense confrontation, and complex game dynamics, spoofing attack evaluation is challenging in real-world applications. Spoofing effectiveness evaluation primarily includes theoretical assessment, simulation assessment, and dataset assessment.

1) Simulation and Dataset Evaluation of Spoofing Effectiveness

Spoofing evaluation and testing in real-world environments pose security risks. To address this, researchers have developed simulators for AV spoofing evaluation. Various autonomous driving systems have been deployed in actual AVs, such as Waymo [54], Autoware [21], Apollo [55], and openpilot [56]. To test the safety of autonomous driving systems, high-fidelity simulators like CARLA [57] and LGSVL [58] have been developed. Based on these simulators, the literature [59] developed ACERO, a simulation platform for autopilot spoofing. Using trajectory similarity metrics, ACERO classifies successful spoofing into different categories, enabling developers to analyze the root causes of spoofing and develop countermeasures. ACERO is evaluated on two open-source autopilot software platforms (openpilot and Autoware) and runs on the CARLA simulator.

The literature [60] proposes a method to generate baseband spoofing data using real-world signals by simultaneously recording GNSS signals using two separate receivers, one of which emulates the spoofed receiver and the other emulates the spoofing source to generate the spoofing signal. The emulator does not require costly hardware to generate intermediate spoofing signals, and the user can autonomously control the spoofing power or reproduce the same scenario with different parameters. In recent years, researchers have also developed a spoofing simulation framework 'Simutack' [61] for connected AVs, which provides a test environment for research related to spoofing technology. Unlike traditional simulators, Simutack supports users to simulate more specific spoofing modes, including spoofing for GNSS, blinding spoofing of cameras, etc.

In terms of research on simulation-based assessment of spoofing attacks at the receiver level, the literature [62] proposes a simulation-based evaluation method analyzing the probability of different initial parameters in loop spoofing. The method evaluates the

initial phase error and Doppler error in terms of area assessment. Tobias Bamberg evaluates spoofing effects based on parameter traversal methods [63]. The literature [64] analyzes the impact of loop bandwidth and discriminator on spoofing effectiveness. These simulators facilitate spoofing technique testing while ensuring safety.

Real-world dataset evaluation is another research hotspot. For instance, the Texas Spoofing Test Battery (TEXBAT) dataset developed by Prof. Todd Humphreys and his research group at the University of Texas at Austin combines real GPS signals with simulated spoofing signals [65]. It includes six datasets of spoofed GPS L1 signals (ds1-ds6) [66] and two other datasets (ds7 and ds8) [67]. The dataset encompasses various spoofing scenarios and types, including static and dynamic scenarios, as well as location and time deception. Based on TEXBAT, the U.S. Department of Energy's Oakridge National Laboratory launched the OAKBAT (Oak Ridge Spoofing and Interference Test Battery) dataset in 2020, featuring six spoofing scenarios similar to TEXBAT. The dataset also provides spoofing datasets for the GALILEO E1 signal, unlike the TEXBAT dataset, which only contains GPS L1 signals [68]. Additionally, the European Space Agency's Navigation Laboratory released the EWF (Evil Waveform) dataset[69], containing EWF signals for GPS L1, GPS L5, and GALILEO E1. These EWF signals result from failed GNSS satellite payloads but produce effects similar to spoofing attacks.

In 2022, Dr. Ghilas Aissou of the University of North Dakota, USA, made public a GPS spoofing dataset. The dataset comprises data extracted from real GPS signals collected from various locations to simulate both dynamic and static AVs using a generic software radio peripheral unit configured as a GPS receiver. In addition to real GPS signals, three types of GPS spoofing are simulated: simple, medium, and complex. The resulting dataset comprises 158,170 samples, including 55% normal samples and 45% spoofed samples. This dataset aids in studying the impact of GPS spoofing features on extraction and contributes to the development of spoofing detection techniques based on both supervised and unsupervised machine learning. In February 2024, the Finnish Geospatial Institute released the latest GNSS spoofing dataset repository, FGI-SpoofRepo [70]. The repository contains raw in-phase and orthogonal datasets with real-time satellite signals from GPS L1 C/A, Galileo E1, GPS L5, and Galileo E5a. The datasets cover three different types of spoofing signatures: synchronous, asynchronous, and spoofing interference.

2) Theoretical Evaluation of Spoofing Effectiveness

In terms of spoofing evaluation metrics, spoofing success rate and detection rate are primarily used. Other spoofing indicators are comprehensively considered in [71], which discusses the influence factors for comprehensive spoofing technology evaluation, analyzes

the results of commonly used spoofing technologies, and establishes a reference for each indicator through an expert system scoring model. These indicators reflect spoofing effectiveness to some extent but face challenges such as ambiguous and uncertain evaluation criteria, weak relationships between influencing factors, and limited applicability of evaluation methods. To address these issues, [72] proposes a gray correlation analysis and fuzzy comprehensive evaluation method. It evaluates 21 indicators across navigation signals, positioning results, hardware and software performance, and equipment combat capability. The hierarchical analysis method determines indicator weights by solving the eigenvector corresponding to the maximum eigenvalue of the judgment matrix. Gray correlation analysis examines the relationships between coefficients, and the fuzzy inference system provides comprehensive evaluation results for optimal program evaluation.

Regarding the impact of spoofing intensity, the literature [73] investigates the effects of three spoofing scenarios on UAV GPS receivers: no spoofing, low-intensity spoofing, and high-intensity spoofing. By monitoring UAV parameters, such as flight altitude, GPS noise level, number of GPS satellites, battery consumption, and CPU utilization, the study analyzes the impact of spoofing intensity on the physical parameters of the UAV network. Additionally, [74] evaluates GPS receiver behavior under spoofing in laboratory settings. Three commercial receivers subjected to spoofing signals at distances of 10 m, 50 m, and 100 m are monitored for position errors, and differences in spoofing behavior at various distances are discussed by comparing positioning results.

However, most studies on these evaluation methods are conducted under ideal single-condition simulations. In practical scenarios, spoofing attacks may be influenced by multiple external environmental factors. Therefore, some studies focus on evaluating the impact of the environment on spoofing attacks.

In spoofing attacks on UAVs, reference [75] comprehensively considers practical application contexts and evaluates the effects of various environmental conditions on spoofing attacks, including indoor/outdoor scenarios and signal propagation paths (line-of-sight and non-line-of-sight). As shown in Fig. 1.3 , 16 potential spoofing scenarios are designed. Spoofing tests are conducted on a low-cost UAV using multi-software radio-based spoofing techniques. These scenarios are evaluated by monitoring variations in GPS-related parameters, such as horizontal accuracy dilution, vertical accuracy dilution, the number of GPS satellites, and the average signal-to-noise ratio power density.

In spoofing attacks on unmanned surface vehicles (USVs), [46] and [76] point out the influence of strong winds and ocean currents on spoofing effects during USV travel. As weather conditions deteriorate with strong winds and currents, the yaw angle of the USV

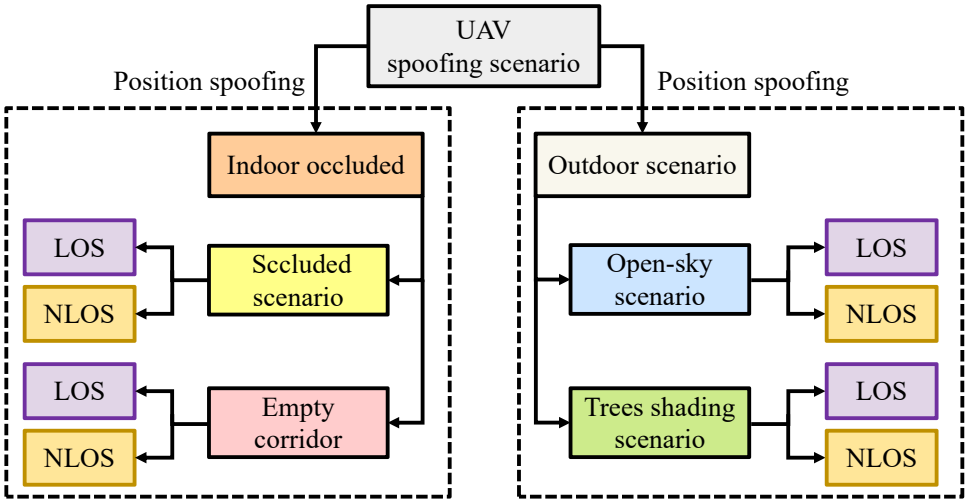

Figure 1.3: Position spoofing scenarios of UAV

changes, affecting the positioning accuracy of the navigation system and causing the USV to oscillate on the desired path. Additionally, the conventional integral line-of-sight guidance law assumes a slowly varying yaw angle, which is not satisfied under harsh weather conditions, further impacting navigation positioning accuracy.

In summary, these evaluation techniques have advanced spoofing technology. However, existing theoretical evaluation methods primarily study the effects of different types and intensities of spoofing signals on GNSS satellite receivers. While there are studies on spoofing effectiveness evaluation for UAVs and unmanned ships, no evaluation method exists for AV MSF systems. Furthermore, the influence of the environment on spoofing attacks is not fully considered, especially in complex geographical and adverse weather scenarios. GNSS and LiDAR uncertainties are not considered in the evaluation of spoofing effectiveness, resulting in incomplete evaluation outcomes. Thus, constructing an effective mathematical model to evaluate the spoofing effectiveness of AV MSF systems in different external environments remains a challenge.

1.3 Critical Technical Challenges in the Investigated Field

Analyzing and summarizing the above domestic and international literature and research status reveals several key technical challenges in the field of MSF system spoofing technology that require urgent resolution in three aspects: error mechanism modeling methods, covert spoofing techniques, and methods for evaluating spoofing effectiveness.

Problem 1: Traditional Kalman Filter-Based Error Transfer Models Are Encumbered by Complex Inversion Operations Due to Varying Sensor Update Frequencies,

Obscuring the Mathematical Relationship Between State Errors and System Parameters. For AVs equipped with GNSS/INS/LiDAR navigation systems, existing research has analyzed some key parameters, such as the initial covariance matrix and LiDAR uncertainty, but neglected the impact of sensor update frequencies when constructing system error transfer models. This results in incomplete system parameter studies. Typically, GNSS update frequencies are much lower than those of INS and LiDAR. The influence of multiple INS and LiDAR signals on state error propagation during a spoofing cycle remains unclear. Additionally, due to differences in measurement sensor update frequencies, traditional Kalman filter-based error transfer models involve complex inversion operations during the measurement update process, complicating derivation and obscuring the analytical model. This prevents a clear representation of the mathematical relationship between state errors and system parameters. Optimizing the error state analytical model under spoofing attacks and identifying key influencing factors of spoofing attacks are the first challenges to address in current research.

Problem 2: Traditional Spoofing Methods Lack Adaptive Spoofing Parameter Adjustment, Leading to Easy Detection by MSF Systems and Low Spoofing Success Rates. The traditional Fusion-ripper algorithm combines constant-value and exponential-value spoofing but does not thoroughly analyze the covertness of the spoofing process. This results in easy detection of spoofing signals by MSF systems during attacks. Furthermore, the Fusion-ripper algorithm employs a traversal method for setting spoofing parameters. While it is suitable for theoretical research to determine the possibility of successful spoofing in specific scenarios, its practical applications are limited. Additionally, the algorithm sets safety thresholds without theoretical justification. When the position error of the target AV exceeds the first threshold, the system transitions from constant-value to exponential-value spoofing. This lack of theoretical foundation complicates the judgment of the take-over effect. Therefore, improving spoofing methods to enable the dynamic adjustment of spoofing parameters based on attack outcomes, preventing detection by MSF systems, and enhancing spoofing success rates are key challenges in current research.

Problem 3: Existing Studies Lack Evaluation of Spoofing Effectiveness for AV MSF Systems in Different Geographical and Weather Scenarios. AVs must operate across diverse environments. However, traditional spoofing technologies indiscriminately broadcast spoofing signals, leading to easy detection and low success rates in challenging scenarios. Existing research has not evaluated the spoofing effectiveness of GNSS/SINS/LiDAR navigation systems for AVs in different scenarios. In complex geographical scenarios, traditional LiDAR Point-cloud Height (LPH)-based methods inaccurately estimate building

heights, which in turn affects the estimation of sky visibility and the number of visible satellites. Traditional regression models also have low accuracy, which reduces the precision of GNSS uncertainty estimation and leads to inaccurate evaluations of spoofing effectiveness. In adverse weather scenarios, existing methods do not assess spoofing effectiveness. Thus, no conclusions are available on how weather impacts spoofing results, the degree of adverse weather required to improve success rates, or which weather types offer the highest spoofing success rates.

1.4 Main Research Content

This thesis investigates spoofing technologies for AV MSF systems based on scientific challenges in the field. Aiming to enhance the efficiency of spoofing attacks on AV MSF systems under complex environments, this study first explores the state error transfer model of MSF systems based on a lightweight information filter. Then, it investigates covert spoofing methods based on fuzzy inference models and evaluates spoofing effectiveness based on sensor uncertainty estimation. Ultimately, the research enhances the concealment and spoofing success rate of AVs in complex environments, providing a theoretical foundation for defense technology research. The overall flowchart of the research content is shown in Fig. 1.4 .

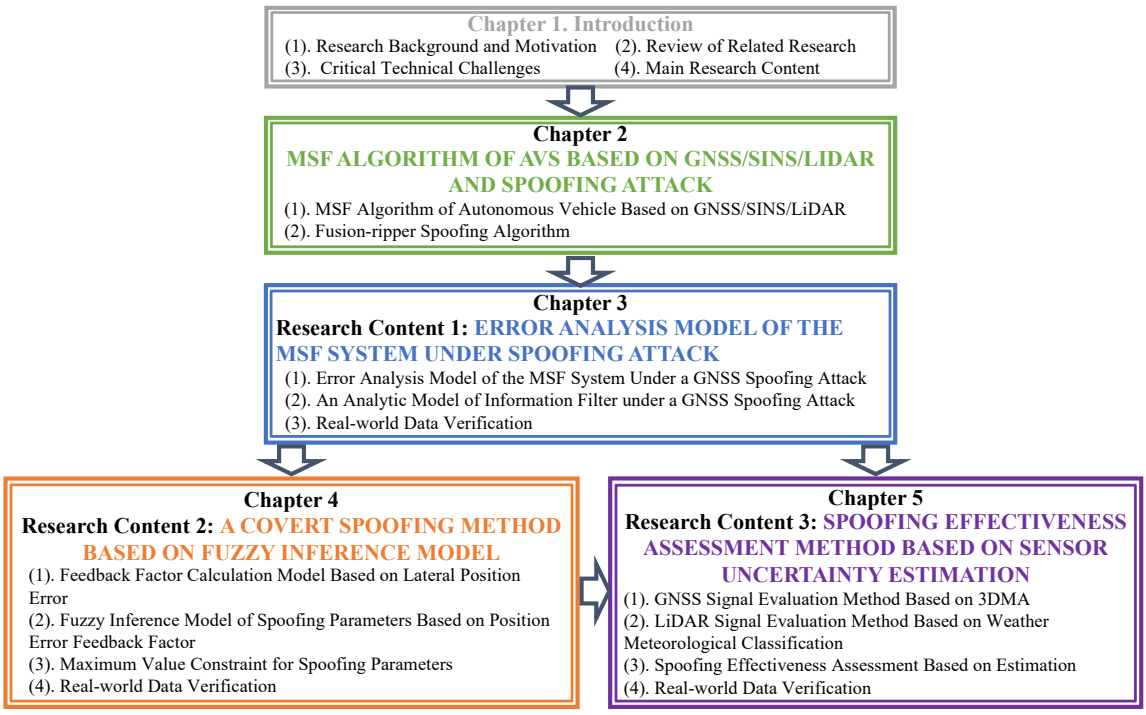

Figure 1.4: Overall flowchart of each chapter of this thesis.

The logical relationships between the main research components are as follows: Research content 1 investigates the state error transfer mechanism of MSF systems under spoofing attacks, identifying key factors that lead to increased state errors. This forms the basis for research content 2 and 3. Research content 2 and 3 focus on the GNSS/SINS/LiDAR-based MSF system of AVs, aiming to improve spoofing success rates in complex environments. Research content 2 delves into covert spoofing methods to address spoofing parameter settings, while research content 3 evaluates spoofing effectiveness under complex environments to address the problem of spoofing timing.

The primary studies are as follows:

Conduct research on the state error propagation mechanism of MSF systems based on a lightweight information filter. Considering the impact of different sensor update frequencies on the state error propagation mechanism of MSF systems under spoofing attacks, establish a Kalman filter analytical model of error states. Derive the update processes of GNSS and LiDAR measurements and the recursive process of INS states under spoofing attacks. To avoid complex inversion operations in LiDAR measurement updates, establish an analytical model of the state error transfer mechanism of MSF systems based on a lightweight information filter. This fully explores the state error transfer mechanism of MSF systems under spoofing attacks and clarifies key factors leading to increased state errors, including the initial state covariance matrix of the MSF system, LiDAR uncertainty, GNSS uncertainty, and the update frequency ratio between LiDAR and GNSS.

Develop covert spoofing methods based on a fuzzy inference model. By real-time monitoring of the target AV, calculate the position error feedback adjustment factor to analyze spoofing difficulty quantitatively. Construct fuzzy knowledge bases and fuzzy rule bases based on the feedback adjustment factor. Establish a fuzzy inference model using the multi-Zadeh method to adjust spoofing parameters and improve spoofing success rates dynamically. By comparing position error feedback adjustment factors, determine whether the spoofing process triggers the take-over effect. Upon triggering the take-over effect, constrain spoofing parameters to maximum values to ensure spoofing success while preventing detection by the MSF system, thereby enhancing spoofing success rates.

Establish a method for evaluating spoofing effectiveness based on sensor uncertainty estimation. For different geographical scenarios, construct a 3D building model of the target area. Estimate the sky visibility mask based on the maximum elevation angle of building edges, calculate the sky visibility at each position in the area, and quantify the

degree to which surrounding buildings block the AV. Combine satellite ephemeris information to estimate the number of visible satellites. Establish a kernel function-based partial least squares nonlinear regression model to estimate GNSS uncertainty. For different weather scenarios, assess the impact of weather on LiDAR performance by establishing LiDAR impulse response functions based on meteorological weather categories, including rain, snow, and fog. Construct LiDAR uncertainty estimation based on B-spline regression models to quantitatively analyze the relationship between different weather types, meteorological categories, and spoofing success rates. Align the estimated uncertainty with the actual uncertainty of the AV's internal sensors and determine whether the AV's environment is 'easy' or 'difficult' for spoofing. This enables the evaluation of spoofing effectiveness in complex geographical and severe weather environments.

Chapter 2

MSF ALGORITHM OF AVS BASED ON GNSS/SINS/LIDAR AND SPOOFING ATTACK

2.1 Introduction

With the advancement of navigation and positioning technologies, autonomous vehicle navigation systems have transitioned from relying on single sensors to adopting MSF systems, significantly enhancing positioning accuracy and robustness. Modern AVs are typically equipped with high-performance navigation sensors and employ MSF frameworks to achieve efficient integration of multiple navigation sensors.

In GNSS receiver spoofing attacks, adversaries transmit counterfeit GNSS signals designed to mimic authentic ones, misleading the target receiver into providing incorrect navigation information. The core of this technique involves disrupting normal GNSS information to deceive the MSF system of AVs. However, MSF systems exhibit strong resistance to spoofing attacks. Compared to direct attacks on GNSS receivers, spoofing MSF systems presents unique technical challenges. Due to the inherent robustness of MSF algorithms, attackers must avoid random spoofing patterns or abrupt signal changes. Instead, they must meticulously plan their attack strategies.

This chapter begins with a detailed introduction to the GNSS/INS/LiDAR-based MSF algorithm for AVs. From the vehicle's perspective, data fusion aims to enhance navigation and positioning performance. Consequently, defense mechanisms such as the Chi-square detection algorithm are typically integrated into MSF systems to provide resistance against faults and disturbances. Subsequently, a mathematical model for a spoofing attack algorithm based on maximizing lateral deviation is established, demonstrating the effectiveness of a two-stage lateral deviation maximization spoofing method. The MSF algorithm and spoofing attack methodologies introduced in this chapter provide a theoretical foundation

for subsequent research on error propagation mechanisms under spoofing attacks and for developing spoofing technologies targeting MSF systems.

2.2 Based on GNSS/SINS/LiDA-Based MSF Algorithm of Autonomous Vehicle

In recent years, with the rapid development of the AV industry, MSF algorithms based on error-state Kalman filter models have been widely applied in AV navigation systems [77]. These algorithms fuse navigation information from various sensors through error recursive models to achieve optimal state parameter estimation. Due to cost constraints and navigation accuracy requirements, AVs are equipped with SINS integrated with GNSS and LiDAR. This configuration leverages the advantages of each sensor to improve navigation system accuracy and provide real-time localization and navigation services for AVs. This chapter investigates the GNSS/SINS/LiDAR-based MSF algorithm for AVs, laying the groundwork for in-depth research on spoofing technologies for MSF systems.

Although numerous MSF frameworks exist, including factor graph optimization, all-source navigation systems, and artificial intelligence-based navigation systems, these fusion algorithms are computationally intensive and complex. The error-state Kalman filter model, characterized by high stability and reduced computational cost, remains widely used in practical AV navigation systems. This thesis focuses on GNSS/SINS/LiDAR-based MSF systems, which fuse heterogeneous positioning measurements through an error-state Kalman filter model. Additionally, a defense algorithm is established to detect and reject abnormal measurements that may occur during system operation, such as outliers and fault signals, preventing contamination of the entire system.

This section establishes the error state Kalman filter model based on GNSS/SINS/LiDAR, encompassing state recursion and measurement equations. It then investigates the Chi-square detection-based defense algorithm, which is extensively applied in practice.

1) State Equation Establishment

In practical applications, most external error parameters between sensors and internal SINS error parameters are compensated through calibration methods [78, 79]. However, residual errors remain, which are regarded as measurement errors in SINS. When the AV MSF system operates normally, for instance, when INS calibration is relatively accurate and the vehicle maneuvers smoothly, only the gyroscope and accelerometer zero-bias errors are considered in error modeling. The state vector includes attitude error, velocity error,

position error, accelerometer zero bias, and gyroscope zero bias, comprising 15 inertial sensor parameters [80]. It can be expressed as:

$$\mathbf{X}(t)_{15\times1} = \begin{bmatrix} \phi_{3\times1}^T & (\delta\mathbf{v}_{3\times1})^T & (\delta\mathbf{p}_{3\times1})^T & \left(\boldsymbol{\varepsilon}_{3\times1}^b\right)^T & \left(\nabla_{3\times1}^b\right)^T \end{bmatrix}^T \tag{2.1}$$

where $\phi_{3\times1}$ denotes attitude error; $\delta\mathbf{v}_{3\times1}$ denotes velocity error; $\delta\mathbf{p}_{3\times1}$ denotes position error; and $\boldsymbol{\varepsilon}_{3\times1}^b$ denotes accelerometer zero bias; $\nabla_{3\times1}^b$ denotes gyro zero bias. The Kalman filter state equation is derived from the velocity, attitude, and position error equations of the SINS, establishing a 15-dimensional continuous Kalman filter state equation:

$$\dot{\mathbf{X}}(t)_{15\times1} = \mathbf{F}(t)_{15\times15}\mathbf{X}(t)_{15\times1} + \mathbf{\Gamma}(t)_{15\times6}\mathbf{W}(t)_{6\times1} \tag{2.2}$$

where $\mathbf{X}(t)_{15\times1}$ denotes a 15-dimensional state variable; $\mathbf{F}(t)_{15\times15}$ denotes the 15-dimensional state transition matrix; $\mathbf{C}_b^n$ denotes the attitude transformation matrix from the body coordinate frame (b-frame) to the navigation coordinate frame (n-frame); $\mathbf{\Gamma}(t)_{15\times6}$ denotes the system noise allocation matrix, which is primarily related to $\mathbf{C}_b^n$ and can be ignored; $\mathbf{W}^b = \begin{bmatrix} \left(\mathbf{w}_g^b\right)^T & \left(\mathbf{w}_a^b\right)^T \end{bmatrix}^T$ denotes the process noise matrix, where $\mathbf{w}_g^b$ and $\mathbf{w}_a^b$ denote the white noise of the gyroscope angular velocity measurements and the accelerometer specific force measurements, respectively. Then, the system state equations of the continuous state space model are discretized, and the results are expressed as:

$$\mathbf{X}_k = \Phi_{k/k-1}\mathbf{X}_{k-1} + \mathbf{W}_{k-1} \tag{2.3}$$

where $\mathbf{X}_k$, $\mathbf{X}_{k-1}$ and $\mathbf{W}_{k-1}$ denote the results of the discretization of the state quantities and the process noise matrices; and $\Phi_{k/k-1} \approx \mathbf{I} + \frac{\mathbf{F}(t_{k-1})}{f_I}$ denotes the discretization result of the state transfer matrix $\mathbf{F}(t)_{15\times15}$, with f_I denoting the SINS update frequency. Assuming the state noise is white noise, its mean and variance are given by:

$$E[\mathbf{W}_k] = 0 \tag{2.4a}$$

$$E[\mathbf{W}_k\mathbf{W}_j^T] = \mathbf{Q}\text{ffi}_{kj} \tag{2.4b}$$

where $\mathbf{Q}$ denotes the noise variance matrix; δ_{kj} denotes the Dirac function.

2) Measurement Equation Establishment

The MSF system studied in this thesis incorporates two measurement sensors: GNSS and LiDAR. The filter measurement update process includes GNSS and LiDAR measurement update equations. When the MSF system receives GNSS signals, it executes the

GNSS measurement update process; when it receives LiDAR signals, it executes the LiDAR measurement update process. Additionally, the uncertainties of GNSS and LiDAR are calculated in real-time and updated based on signal quality in different environments [77].

For the GNSS measurement update equation, GNSS provides 3-dimensional position information of the AV, including longitude, latitude, and altitude. The measurement vector of the Kalman filter model can be expressed as:

$$\mathbf{Z}_1(t)_{3\times1} = \widetilde{\mathbf{p}}_G(t)_{3\times1} - \widetilde{\mathbf{p}}_I(t)_{3\times1} \tag{2.5}$$

where $\widetilde{\mathbf{p}}_I(t)_{3\times1}$ denotes SINS position update information; $\widetilde{\mathbf{p}}_G(t)_{3\times1}$ denotes GNSS measurement information. The final measurement equation can be expressed as:

$$\mathbf{Z}_1(t)_{3\times1} = \mathbf{H}_G(t)_{3\times15}\mathbf{X}(t)_{15\times1} + \mathbf{V}^G(t)_{3\times1} \tag{2.6}$$

where $\mathbf{V}^G(t)_{3\times1}$ denotes the 3-dimensional measurement noise and is assumed to have a mean value of 0, and $\mathbf{H}_G(t)_{3\times15}$ denotes the continuous measurement matrix:

$$\mathbf{H}_G(t)_{3\times15} = \begin{bmatrix}\mathbf{0}_{3\times6} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times6}\end{bmatrix} \tag{2.7}$$

In practical GNSS/SINS/LiDAR-based MSF algorithms, the measurement equations themselves are discrete. Therefore, the measurement equation of the Kalman filter model can be directly expressed as:

$$\mathbf{Z}_k = \mathbf{H}_G\mathbf{X}_k + \mathbf{V}_k^G \tag{2.8}$$

where $\mathbf{H}_G$ denotes the GNSS measurement matrix; $\mathbf{V}_k^G$ denotes the GNSS measurement noise vector. Assuming GNSS measurement noise is white noise and uncorrelated with system state noise, the mean and variance of the measurement noise are:

$$E\left[\mathbf{V}_k^G\right] = 0 \tag{2.9a}$$

$$E\left[\mathbf{V}_k^G\left(\mathbf{V}_j^G\right)^T\right] = \mathbf{R}_G\delta_{kj} \tag{2.9b}$$

$$E\left[\mathbf{W}_k\left(\mathbf{V}_k^G\right)^T\right] = 0 \tag{2.9c}$$

where $\mathbf{R}_G$ denotes the GNSS measurement noise variance matrix, which describes GNSS measurement uncertainty. In AV MSF systems, GNSS uncertainty is typically calculated based on satellite signal observation residuals and weight matrices, as detailed in [77]; δ_{kj}

denotes the Dirac function.

For the LiDAR measurement update equation, LiDAR also provides three-dimensional position information of the AV. , including longitude, latitude, and altitude information. Therefore, the measurement vector in the Kalman filter model can be expressed as:

$$\mathbf{Z}_2(t)_{3\times1} = \tilde{\mathbf{p}}_L(t)_{3\times1} - \tilde{\mathbf{p}}_I(t)_{3\times1} \tag{2.10}$$

where $\tilde{\mathbf{p}}_L(t)_{3\times1}$ denotes the LiDAR measurement. The final measurement equation can be expressed as:

$$\mathbf{Z}_2(t)_{3\times1} = \mathbf{H}_L(t)_{3\times15}\mathbf{X}(t)_{15\times1} + \mathbf{V}^L(t)_{3\times1} \tag{2.11}$$

where $\mathbf{V}_2(t)_{3\times1}$ denotes the 3-d measurement noise and is assumed to have a mean of 0. Similarly, the measurement noise covariance matrix is set to be $\mathbf{R}_L$, which describes the uncertainty of the LiDAR, and the measurement matrix $\mathbf{H}_L(t)_{3\times15}$ can be expressed as follows.

$$\mathbf{H}_L(t)_{3\times15} = \begin{bmatrix}\mathbf{0}_{3\times6} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times6}\end{bmatrix} \tag{2.12}$$

Consistent with the GNSS measurement equation, the LiDAR measurement equation for the Kalman filter model can be directly expressed as:

$$\mathbf{Z}_k = \mathbf{H}_L\mathbf{X}_k + \mathbf{V}_k^L \tag{2.13}$$

where $\mathbf{H}_L$ denotes the LiDAR measurement matrix and $\mathbf{V}_k^L$ denotes the LiDAR measurement noise vector. Similarly, assuming that the LiDAR measurement noise is white noise and the state noise in the system is uncorrelated with the measurement noise, the mean and variance of the measurement noise can be expressed as.

$$E\left[\mathbf{V}_k^L\right] = 0 \tag{2.14a}$$

$$E\left[\mathbf{V}_k^L\left(\mathbf{V}_j^L\right)^T\right] = \mathbf{R}_L\delta_{kj} \tag{2.14b}$$

$$E\left[\mathbf{W}_k\left(\mathbf{V}_k^L\right)^T\right] = 0 \tag{2.14c}$$

where $\mathbf{R}_L$ denotes the LiDAR measurement noise variance matrix, describing LiDAR measurement uncertainty. In MSF systems, LiDAR uncertainty is typically calculated based on LiDAR point cloud alignment time, the number of iterations, and the probability score of the normal distribution transformation (NDT) matching algorithm, as detailed in [21] and [81]; δ_{kj} denotes the Dirac function.

The MSF system estimates and updates state errors in real-time based on the established state and measurement equations, compensates these errors into navigation information, and computes accurate navigation data such as velocity, attitude, and position.

3) Defence Algorithm Based on Chi-square Detection

The Chi-square detection algorithm is commonly employed in AV MSF systems to enhance robustness. It effectively detects and suppresses measurement values with large errors and can identify some GNSS spoofing signals with significant positioning errors, thereby providing a degree of anti-spoofing capability. The anti-spoofing algorithm based on Chi-square detection can detect unexpected outliers or faults. The specific process of this algorithm is as follows.

In the Kalman filter, the innovation is computed using observations and error state predictions:

$$\boldsymbol{\gamma}_k = \mathbf{Z}_k - \mathbf{H}\mathbf{X}_k \tag{2.15}$$

where $\boldsymbol{\gamma}_k$ denotes the new interest of the kth epoch; $\mathbf{Z}_k$ denotes the observation vector; and $\mathbf{X}_k$ denotes the error estimation vector. Under normal operation of the MSF system, $\boldsymbol{\gamma}_k$ obeys a Gaussian distribution with mean 0, and its covariance matrix can be expressed as:

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k/k-1}\mathbf{H}^T + \mathbf{R}_k \tag{2.16}$$

According to the statistical properties of the innovation sequence, the statistic defined by the following equation follows a Chi-square distribution with m degrees of freedom:

$$\boldsymbol{\gamma}_k^T \mathbf{S}_k^{-1} \boldsymbol{\gamma}_k \sim \chi^2(m) \tag{2.17}$$

where m denotes the dimension of the measurement vector. The threshold is calculated according to the following hypothesis detection equation:

$$T_D = \chi^2_{1-P_M}(m) \tag{2.18}$$

where P_M denotes the desired false alarm rate and T_D denotes the statistical significance threshold, obtainable from Chi-square distribution tables. The relevant parameters satisfy:

$$P_M = 1 - \alpha \tag{2.19}$$

$$p\left\{\chi^2(m) > \chi^2_\alpha(m)\right\} = \alpha \tag{2.20}$$

where α denotes the tail probability.

If the Chi-square value exceeds the threshold T_D, the measurement is deemed abnormal. The weight of such measurements is set to 0 to prevent contamination of the entire MSF system, achieving defense against spoofing signals. In summary, spoofing MSF systems poses unique technical challenges, so for some spoofing signals with large positioning errors, the AV can effectively detect and reject the influence of these spoofing signals on the MSF system by the defence algorithm based on Chi-square detection. To successfully attack an AV's MSF system, accurate spoofing signals and well-designed spoofing algorithms are essential.

2.3 Fusion-ripper Spoofing Algorithm

In the MSF algorithm based on the error state Kalman filter model, the navigation system of AVs can detect measurement anomalies or accidental faulty measurement values through the defense algorithm, preventing these signals from contaminating the entire system, and thus providing a certain degree of resistance to spoofing attacks. However, due to the dynamic motion of the vehicle, sensor noise increases, environmental changes, and other factors, the MSF system will have a vulnerable period, during which the system will mainly rely on GNSS or LiDAR as the primary source of position measurement information. When the MSF system relies primarily on GNSS, it is more likely to be susceptible to successful spoofing. In some LiDAR scenarios with low confidence, GNSS becomes the primary measurement update source of the Kalman filter in the MSF system. In contrast, LiDAR measurements will be regarded as anomalies by the anti-alternative algorithm in the MSF system, and thus will not be able to provide effective correction information for the positioning error. Then, the false GNSS signals will become the primary source of position information updates in the MSF system, a phenomenon known as the take-over effect. This phenomenon is called the take-over effect, which is the reason why the spoofing attack can fundamentally destroy the MSF framework of AVs.

The Fusion-ripper spoofing algorithm was first proposed by researchers at the University of California, Irvine, in 2020 [35], and is a spoofing parameter setting algorithm designed primarily for GNSS/SINS/LiDAR-based MSF systems for AVs. Its spoofing process mainly consists of two stages: vulnerability analysis and enhancement attack, which essentially involve constant value spoofing and exponential value spoofing of the MSF system over multiple spoofing signal cycles. The ultimate goal is to find suitable constant value spoofing parameters and exponential spoofing parameters, so that the lateral position error under the b-frame exceeds the safety threshold, deviates from the intended trajectory, and finally realizes the spoofing attack. The primary purpose of this thesis is to study the

spoofing technology to the MSF system of AVs, so to quantify the spoofing model easily, the following assumptions are made before the spoofing algorithm is specifically introduced [35, 77, 82]:

1. The MSF system of the target AV is based on the error state Kalman filter model. The relevant navigation and positioning systems include GNSS, SINS, and LiDAR, and the spoofing source is aware of the specific models and parameters of the sensors inside these systems.

2. The spoofing source is equipped with high-precision navigation sensors and an information monitoring system, which can track and effectively detect the actual position and speed of the target AV in real-time. At the same time, it is equipped with a high-precision environmental perception sensor, which can obtain the three-dimensional building information of the target area in advance and carry out further processing. It can get the visibility of the sky at any location by calculating the target area. It is also able to monitor the weather conditions of the target area in real-time.

3. The AV is assumed to travel along a predetermined trajectory. The vehicle's internal controller continuously corrects off-center errors at a high frequency (100 Hz) [35]. Thus, the lateral deviation of the MSF system's positioning results directly reflects the AV's actual deviation but in the opposite direction.

Based on the above assumptions, the specific process of the Fusion-ripper spoofing algorithm is as follows:

Firstly, a spoofed position error increment $\Delta\tilde{d}^{bx}$ is given laterally along the intended trajectory of the AV, superimposed on the true position information of the AV when it is not spoofed and interfered with. The spoofing source generates spoofed signals using a spoofed signal generation technique based on existing information. Then, the spoofed signals are sent to the AV, which induces its internal receivers to decode the spoofing source and set the false localization result set by the spoofing source. Firstly, the three-dimensional coordinate representation of the spoofing error value under the b-frame is:

$$\Delta\tilde{\mathbf{d}}^{b} = \begin{bmatrix} \Delta\tilde{d}^{bx} & 0 & 0 \end{bmatrix}^{T} \tag{2.21}$$

where $\Delta\tilde{d}^{bx}$ denotes the spoofing position error increment. Since the navigation solution model of AVs is generally established in the n-frame, the position error in the b-frame is converted to the n-frame, so the value of the spoofing error in the n-frame is:

$$\Delta\tilde{\mathbf{d}}^{n} = \mathbf{C}_{b}^{n}\Delta\tilde{\mathbf{d}}^{b} \tag{2.22}$$

where $\mathbf{C}_b^n$ denotes the rotation matrix from the b-frame to the n-frame, and since the AV drives on the ground, the rotation matrix between the n-frame and the b-frame is established by the azimuth angle, which can be expressed as:

$$\mathbf{C}_b^n = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.23}$$

where γ denotes the azimuth of the b-frame for the n-frame, thus:

$$\Delta\tilde{\mathbf{d}}^b = \begin{bmatrix} \Delta\tilde{d}^{bx}\cos\gamma & \Delta\tilde{d}^{bx}\sin\gamma & 0 \end{bmatrix}^T \tag{2.24}$$

In this thesis, we use latitude and longitude to denote the spoofing parameter $\Delta\tilde{\mathbf{p}}_j^{spo}$ in the n-frame, which can be expressed as:

$$\Delta\tilde{\mathbf{p}}_k^{spo} = \begin{bmatrix} \Delta\tilde{L}_k & \Delta\tilde{\lambda}_k & 0 \end{bmatrix}^T \tag{2.25}$$

where k denotes the spoofing sequence; $\Delta\tilde{L}_k$ denotes the longitude spoofing error value, and $\Delta\tilde{\lambda}_k$ denotes the latitude spoofing error value, which can be expressed as respectively:

$$\begin{cases} \Delta\tilde{L}_k = \frac{\Delta\tilde{d}^{bx}\cos\gamma}{R_e\cos\lambda} \\ \Delta\tilde{\lambda}_k = \frac{\Delta\tilde{d}^{bx}\sin\gamma}{R_e} \end{cases} \tag{2.26}$$

where R_e denotes the radius of the earth; λ denotes the latitude information of the location of the AV. Thus, the spoofing parameter $\Delta\tilde{\mathbf{p}}_k^{spo}$ can be expressed as:

$$\Delta\tilde{\mathbf{p}}_k^{spo} = \begin{bmatrix} \frac{\Delta\tilde{d}^{bx}\cos\gamma}{R_e\cos\lambda} & \frac{\Delta\tilde{d}^{bx}\sin\gamma}{R_e} & 0 \end{bmatrix}^T \tag{2.27}$$

Therefore, the spoofing parameter $\Delta\tilde{\mathbf{p}}_k^{spo}$ can be changed by setting the spoofing position error increment $\Delta\tilde{d}^{bx}$. When the spoofing attack begins, the first spoofing sequence is defined as 1. Therefore, the maximum number of spoofed GNSS epochs that the spoofing source can implement is related to the spoofing time and the frequency of the GNSS updates, which can be expressed as follows:

$$k_{\max}^{spo} = T_{\max}^{spo} \cdot f_G \tag{2.28}$$

where $k_{\max}^{spo}$ denotes the maximum number of GNSS epochs that can realize spoofing attack from the beginning to the end of spoofing; $T_{\max}^{spo}$ denotes the maximum time that the spoofing

source can implement spoofing attack, and f_G denotes the update frequency of GNSS. If the number of spoofing elements is too many, that means the spoofing time is too long, which leads to spoofing failure. Therefore, the sequence of multiple spoofing parameters set is:

$$\left\{\Delta\tilde{\mathbf{p}}_1^{spo}, \Delta\tilde{\mathbf{p}}_2^{spo}, \cdots, \Delta\tilde{\mathbf{p}}_k^{spo}\right\}, k \leq k_{\max}^{spo} \tag{2.29}$$

where $\Delta\tilde{\mathbf{p}}_k^{spo}$ denotes the spoofing parameter set by the spoofing source; k denotes the spoofing sequence. During the period of spoofing attack on the MSF system, assuming that the receiver in the target AV receives the spoofing signal and decodes the wrong position information, and that the position information is the same as the position information set by the spoofing source, the false GNSS measurements received by the AV can be expressed as:

$$\tilde{\mathbf{p}}_k^G = \mathbf{p}_k^G + \Delta\tilde{\mathbf{p}}_k^{spo}, k = 1, 2, \cdots and \quad k \leq k_{\max}^{Spo} \tag{2.30}$$

where $\mathbf{p}_k^G$ denotes the true position information of the target AV when it is not interfered by spoofing attacks. Due to the effect of spurious GNSS measurements $\tilde{\mathbf{p}}_k^G$, the MSF system will decode an erroneous localization result, which can be monitored in real-time by the spoofing source, ignoring the upward error, and the spoofing signals cause a positional error in the horizontal plane of the n-frame caused by the MSF system of the target AV $\Delta\widehat{\mathbf{p}}_1^n$ can be represented by the longitude error and the latitude error:

$$\Delta\widehat{\mathbf{p}}_k^n = \begin{bmatrix} \Delta\widehat{L}_k & \Delta\widehat{\lambda}_k & 0 \end{bmatrix}^T \tag{2.31}$$

where $\Delta\widehat{L}_k$ and $\Delta\widehat{\lambda}_k$ denotes the longitude error and latitude error, respectively. The models usually built by b-frame are generally based on the n-frame, so they are converted to the b-frame at first. For ease of representation, the magnitude of the spoofing attack is usually measured in terms of the deviation from the true position of the AV when it is not spoofed.

$$\Delta\widehat{\mathbf{d}}_k^b = \frac{\mathbf{C}_n^b \Delta\widehat{\mathbf{p}}_k^n}{R_e} \tag{2.32}$$

where the positional deviation $\Delta\widehat{\mathbf{d}}_k^b$ in the b-frame can be expressed by the deviation of position in three directions: horizontal, vertical and celestial $\Delta\widehat{\mathbf{d}}_k^b = \begin{bmatrix} \Delta\widehat{d}_k^{bx} & \Delta\widehat{d}_k^{by} & \Delta\widehat{d}_k^{bz} \end{bmatrix}^T$. Longitudinal and vertical deviations of the AV are less likely to cause dangerous outcomes for the AV compared to lateral deviations, so from the spoofing source, longitudinal and vertical deviations are less rewarding, so the spoofing source focuses on the lateral deviation of the target AV for the intended trajectory $\Delta\widehat{d}_k^{bx}$.

The Fusion-ripper spoofing process is divided into two steps: a vulnerability analysis

and an enhancement attack. Vulnerability analysis is to find the vulnerable period of the MSF system through active spoofing testing. Once the spoofing source monitors that the MSF system is in a vulnerable period, it carries out a continuous enhancement attack, so that the lateral deviation of the MSF system for the predetermined trajectory rapidly exceeds the safety threshold and achieves the spoofing purpose of the MSF system of the AV. The spoofing scheme is mainly divided into two stages, as shown in Fig. 2.1 . According

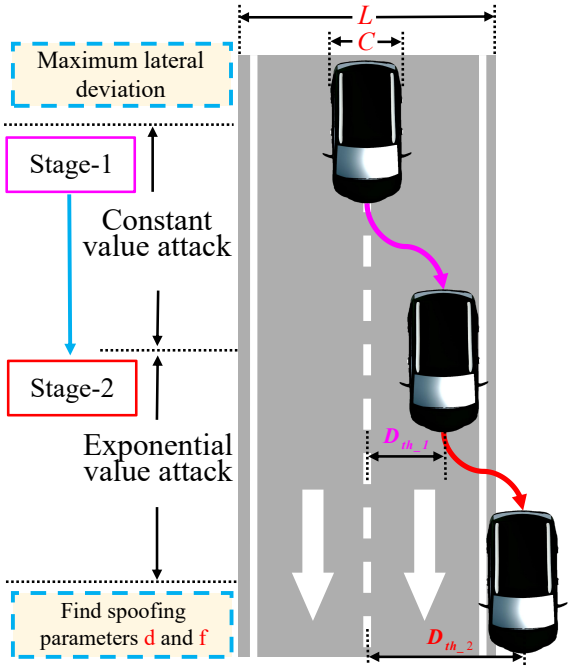

Figure 2.1: Spoofing scheme diagram based on maximizing lateral deviation

to the Fusion-ripper spoofing algorithm, the core of the algorithm is to find constant and exponential parameters that maximize the lateral deviation of the AV MSF system for the intended trajectory.

$$\{d, f\} = \mathrm{M}\left\{\left\|\Delta\widehat{d}_k^{bx}\right\|\right\} \tag{2.33}$$

where d and f denote the constant and exponential parameters to be searched for, respectively; and $\mathrm{M}\{*\}$ denotes the computational process of searching for the parameters d and f by maximizing the lateral deviation. The principle of spoofing attack is to maximize the lateral deviation absolute value $\Delta\widehat{d}_k^{bx}$ of the vehicle for the intended trajectory while satisfying the basic conditions. Additionally, the spoofing process must fulfill the following two basic conditions.

1. Each spoofing attack must generate lateral deviations in the AV that cannot be detected by the Chi-square detection algorithm of the MSF system.

2. Finite spoofing sequence. In practice, the time for tracking and spoofing the MSF system cannot be too long, so the number of spoofing sequences must be smaller than the maximum spoofing number set by the spoofing source.

1) Stage I: Vulnerability analysis

In the vulnerability analysis stage, the spoofing source begins actively broadcasting spoofing signals to the AV's MSF system. By real-time monitoring of the target AV, it obtains feedback position information to determine whether the target AV is in a vulnerable period. Vulnerability analysis essentially involves continuous constant-value spoofing of the target to bypass the Chi-square detection algorithm and reduce the number of spoofing parameters. The purpose of continuous constant-value spoofing is to identify the MSF system's vulnerable periods, i.e., scenarios where LiDAR has low localization confidence, and GNSS is relatively reliable, thereby improving spoofing success rates and achieving the spoofing objective. During this period, the MSF system primarily relies on GNSS-provided navigation information.

In spoofing stage I, the spoofing Position Error Increment is set to a constant value:

$$\Delta\tilde{d}^{bx} = d \tag{2.34}$$

where d denotes the constant value of the Fusion-ripper parameter. According to Eq. (2.27), Eq. (2.29) and Eq. (2.34), the sequence of spoofing parameters for the spoofing source to implement persistent constant value spoofing is:

$$\begin{aligned} &\{\Delta\tilde{\mathbf{p}}_1^{spo}, \Delta\tilde{\mathbf{p}}_2^{spo}, \cdots, \Delta\tilde{\mathbf{p}}_k^{spo}\} \\ &st.1 \quad \Delta\tilde{\mathbf{p}}_k^{spo} = \begin{bmatrix} \frac{d\cos\gamma}{R_e\cos\lambda} & \frac{d\sin\gamma}{R_e} & 0 \end{bmatrix}^T \\ &st.2 \quad \chi^2 < \chi^{Threshold} \\ &st.3 \quad \Delta\hat{\tilde{d}}_k^{bx} < D_{th-1} \end{aligned} \tag{2.35}$$

where $\chi^{Threshold}$ denotes the Chi-square detection threshold; D_{th-1} denotes the safety threshold for stage one, for the setting of this threshold, which is calculated in this thesis by the width of the lane line and the width of AV, the safety threshold can be expressed as:

$$D_{th-1} = \frac{L-C}{2} \tag{2.36}$$

where L denotes lane width; C denotes vehicle width. When lateral deviation exceeds D_{th-1}, as shown in Fig. 2.1 , the vehicle may depart from the current lane under spoofing attacks. Exceeding the lane line indicates that constant-value spoofing has destabilized the

MSF system. When the lateral deviation of the target AV reaches the spoofing threshold, it can be inferred that the target AV's MSF system may be in a vulnerable period, i.e., LiDAR positioning reliability is low, GNSS positioning reliability is high, and the target AV is more susceptible to successful deception.

2) Stage II: Enhanced Attack

Upon identifying a vulnerable period, it signifies that the MSF system places greater trust in GNSS positioning results. Stage II involves an enhanced attack aimed at rapidly increasing the AV's lateral deviation to trigger the take-over effect, where the MSF system fully trusts the position measurement information from the spoofing signal's GNSS error. During the enhanced attack, by designing an exponential deviation sequence and generating spoofing signals based on the AV's actual position, the positioning error of the AV's multi-source navigation system is rapidly increased, allowing the spoofing process to be completed quickly and achieving the spoofing objective.

In stage II, the position error increment is set to increase exponentially:

$$\Delta\tilde{d}^{bx} = d \cdot f^{i} \tag{2.37}$$

where f denotes the exponential parameter; i denotes the exponential value spoofing sequence. According to Eq. (2.27), Eq. (2.29), and Eq. (2.37), the sequence of GNSS spoofing parameters for the spoofing source to implement persistent exponential value spoofing is:

$$\begin{aligned} &\left\{\Delta\tilde{\mathbf{p}}_1^{spo}, \Delta\tilde{\mathbf{p}}_2^{spo}, \cdots, \Delta\tilde{\mathbf{p}}_i^{spo}\right\} \\ &st.1 \quad \Delta\tilde{\mathbf{p}}_i^{spo} = \left[\begin{array}{ccc} \frac{d\cdot f^i \cos\gamma}{R_e \cos\lambda} & \frac{d\cdot f^i \sin\gamma}{R_e} & 0 \end{array}\right]^T \\ &st.2 \quad \chi^2 < \chi^{Threshold} \\ &st.3 \quad \Delta\tilde{d}_i^{bx} < D_{th-2} \end{aligned} \tag{2.38}$$

where D_{th-2} denotes the stage II safety threshold, representing the minimum lateral deviation value for the spoofing source to successfully spoof by adding exponential GNSS sequences during the enhanced attack stage. It is calculated based on lane width and vehicle width:

$$D_{th-2} = \frac{L+C}{2} \tag{2.39}$$

Within a specified period, after the spoofing source has completed the spoofing attack on the target AV, the positioning error of the AV's MSF system is monitored to determine if it exceeds the safety threshold. Once the lateral deviation of the MSF system surpasses the safety threshold D_{th-2}, as shown in Fig. 2.1 , the vehicle's lateral deviation exceeds the

safety distance. There is a high probability of safety accidents, such as the AV departing the entire lane or colliding with roadside structures or oncoming vehicles.

$$\Delta\widehat{d}_i^{bx} \geq D_{th-2} \tag{2.40}$$

In that case, it indicates that the spoofing source can implement successful spoofing attacks on the MSF system. It should be noted that, in the course of practical application, the spoofing source can typically set different security thresholds according to varying strategic requirements.

During the actual process, the spoofing source can dynamically adjust the spoofing parameters according to different spoofing targets. Therefore, it must be acknowledged that the parameter universality of this thesis is generally limited, and it mainly provides some research methods. For commercial AVs, the model of each product is generally public. Attackers can dynamically set the spoofing threshold based on the width of the vehicle. In addition, the spoofing source can also choose different positions for the spoofing attack, because the road width at different positions may vary.

2.4 Summary

This chapter introduces the theoretical foundations related to the research on GNSS/INS/LiDAR-based MSF systems and spoofing attack algorithms. Based on the SINS error model, an error-state Kalman filter-based MSF mathematical model is established to achieve high-precision estimation of positioning information. Furthermore, The widely used Chi-square detection tolerance algorithm in AV MSF systems is also introduced. Meanwhile, a mathematical model for a spoofing attack algorithm based on maximizing lateral deviation is established. This model leverages inherent MSF system vulnerabilities to conduct two-stage spoofing attacks. The usability of this model is demonstrated, providing a theoretical foundation for subsequent research on MSF state error propagation mechanisms under spoofing attacks and on spoofing attack technologies in complex environments.

Chapter 3

ERROR ANALYSIS MODELS OF THE MSF SYSTEM UNDER SPOOFING ATTACK

3.1 Introduction

As a crucial technical measure for countering the navigation system of the target AV in a battlefield environment, spoofing technology holds significant strategic importance in paralyzing the opponent's AV combat system and gaining battlefield advantages. However, most research mainly focuses on spoofing technology for satellite terminal receivers or GNSS/INS combined navigation systems, while research on spoofing technology for MSF systems is limited. In addition, during the analysis of the error mechanism propagation model, existing studies do not comprehensively consider the main parameters affecting spoofing, and the established state error analytical model is unclear. Therefore, constructing a clear analytical model to analyze the influence mechanism of spoofing attacks is a fundamental issue in the field of implementing spoofing attacks. Existing research results analyze some key parameters, such as the initial covariance matrix and LiDAR measurement uncertainty, but do not consider the influence of the sensor update frequency in quantifying the error propagation mechanism, which is different in the actual MSF system, so it is necessary to further consider the influence of this parameter on the MSF state error propagation mechanism under spoofing attack. In addition, the traditional MSF state error propagation model based on the Kalman filter involves more complicated inverse operations in the measurement update process, leading to a cumbersome derivation process and failing to visualize the mathematical relationship between the state error and the system parameters.

Based on the GNSS/INS/LiDAR-based MSF algorithm for AVs and the spoofing algorithm based on maximizing lateral deviation introduced in Chapter 2, this section considers the effects of different measurement sensor update frequencies on the MSF system, establishes the error state Kalman filter analytical model to study in detail the state error transfer

mechanism of the MSF system under the spoofing attack, and further deduces a clear error transfer analytical model based on the lightweight information filter recursive model avoids the complex inverse operation in the LiDAR measurement updating process, and establishes a clear error transfer analytical model, which can visualize the mathematical relationship between the state error and the system parameters. Finally, the main influencing factors that lead to the increase of the state error of the MSF system under the GNSS spoofing are analyzed, including the initial state covariance matrix of the MSF system, the uncertainty of LiDAR, the uncertainty of GNSS, and the ratio of the update frequency between LiDAR and GNSS, which provide the theoretical basis for the research of the spoofing technology of the MSF system under the complex scenario.

3.2 Error Analysis Model of the MSF System Under a GNSS Spoofing Attack

In this thesis, the spoofing parameter is set to superimpose a spoofing position increment on the real GNSS output position signal when the AV is not subject to a spoofing attack. It is assumed that the AV is traveling on a predetermined trajectory, its motion state is smooth, and the system noise characteristics are unchanged. When the GNSS position information is spoofed, then it will result in a spoofed position error superimposed on the real GNSS signal. Assuming that the measured value of the GNSS position information without spoofing attack is $\mathbf{p}_k^G$, and the measured value of the spoofed signal with spoofing attack is $\tilde{\mathbf{p}}_k^G$, then the relationship between them is satisfied:

$$\tilde{\mathbf{p}}_k^G = \mathbf{p}_k^G + \Delta\tilde{\mathbf{p}}_k^{spo} \tag{3.1}$$

where k denotes the spoofing epoch; $\Delta\tilde{\mathbf{p}}_k^{spo}$ denotes the position spoofing parameter. Since we mainly focus on spoofing attacks for AVs and do not consider spoofing attacks for the upward position, the spoofing parameter can be essentially expressed as the spoofing position increment in the horizontal direction added by the spoofing source based on the real GNSS signals, including the longitude spoofing position increment and latitude spoofing position increment, and the final spoofing parameter can be expressed as:

$$\Delta\tilde{\mathbf{p}}_k^{spo} = \begin{bmatrix} \Delta\tilde{L}_k & \Delta\tilde{\lambda}_k & 0 \end{bmatrix}^T \tag{3.2}$$

where $\Delta\tilde{L}_k$ and $\Delta\tilde{\lambda}_k$ denote the longitude spoofing position increment and latitude spoofing position increment, respectively. To fully explore which parameters in the MSF system

directly affect the spoofing results, this section derives a detailed MSF system state error transfer model during the GNSS spoofing signal update cycle. It analyzes the influence of the spoofing signal on the GNSS measurement update process, the SINS state recursion process, and the LiDAR measurement update process in the Kalman filter. The main factors affecting system errors under spoofing attacks are explored in depth, thus laying the foundation for the development of an effective spoofing scheme.

In a GNSS/SINS/LiDAR-based MSF and positioning framework, there is usually a difference between the sensor update frequencies, e.g., the GNSS update frequency f_G is typically 1 Hz or 5 Hz, the LiDAR update frequency f_L is typically 5 Hz or 10 Hz, and the SINS update frequency f_I is typically 100 Hz, 200 Hz, or 400 Hz. Typically, $f_G \leq f_L < f_I$. In a GNSS measurement update cycle, the filter update process usually contains multiple SINS state recursion processes and LiDAR measurement update processes, as shown in Fig. 3.1 .

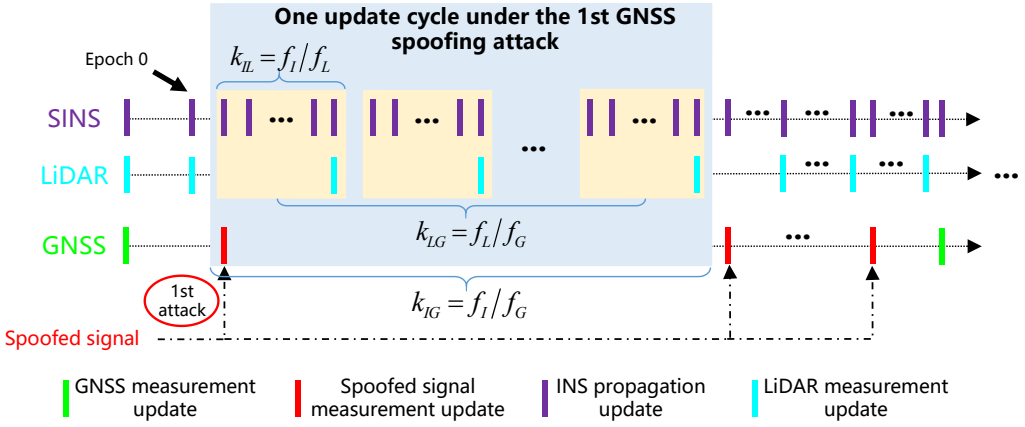

Figure 3.1: A KF update process considering sensors update frequency

Therefore, to improve the accuracy of the error transfer model for MSF systems under spoofing attack, compared to the traditional approach, this chapter considers the difference in update frequency between different sensors.

3.2.1 GNSS Measurement Update Error Transfer Model under Spoofing attack

To facilitate the subsequent derivation of the state error model, the update frequency ratio between each sensor is set to be, respectively:

$$k_{IL} = f_I / f_L \tag{3.3a}$$

$$k_{IG} = f_I / f_G \tag{3.3b}$$

$$k_{LG} = f_L / f_G \tag{3.3c}$$

Assuming that the LiDAR measurement information has just finished updating and the measurement information of the next epoch is the positioning information of GNSS, derive the influence of the satellite signal without spoofing and the spoofing signal on the GNSS measurement updating process, and calculate the state error before and after spoofing. For the convenience of presentation, this moment is used as the starting point of the state update, and the state one-step prediction value is obtained by performing the state recursion according to the error state Kalman filter recursion formula. Thus, the state one-step prediction value $\hat{\mathbf{X}}_{1/0}$ and the one-step prediction mean-square error array $\mathbf{P}_{1/0}$ are denoted as:

$$\hat{\mathbf{X}}_{1/0} = \mathbf{\Phi}_{1/0}\hat{\mathbf{X}}_0 \tag{3.4a}$$

$$\mathbf{P}_{1/0} = \Phi_{1/0}\mathbf{P}_0\Phi_{1/0}^T + \mathbf{Q} \tag{3.4b}$$

where $\Phi_{1/0}$ denotes the initial state transfer matrix; $\hat{\mathbf{X}}_0$ denotes the initial state quantity; $\mathbf{P}_0$ denotes the initial covariance matrix, a parameter that represents the uncertainty in the state of the initial MSF system; $\mathbf{Q}$ denotes the noise covariance matrix. The quantitative update equation after state recursion can be expressed as:

$$\hat{\mathbf{X}}_1 = \hat{\mathbf{X}}_{1/0} + \mathbf{K}_1(\mathbf{Z}_1 - \mathbf{H}_G\hat{\mathbf{X}}_{1/0}) \tag{3.5}$$

where $\mathbf{K}_1$ denotes the Kalman filter gain matrix, which can be expressed as:

$$\mathbf{K}_1 = \mathbf{P}_{1/0}\mathbf{H}_G^T(\mathbf{H}_G\mathbf{P}_{1/0}\mathbf{H}_G^T + \mathbf{R}_G)^{-1} \tag{3.6}$$

where $\mathbf{R}_G$ denotes the measurement noise covariance matrix describing the uncertainty of the GNSS measurements. The state estimation mean square error matrix can be expressed

as:

$$\mathbf{P}_1 = (\mathbf{I} - \mathbf{K}_1\mathbf{H}_G)\mathbf{P}_{1/0} \tag{3.7}$$

Assuming the AV is traveling on a predetermined trajectory at this time, its motion state is smooth, and the noise characteristics of the system remain unchanged. When the GNSS position information is spoofed, it results in a position error superimposed on the actual GNSS signal. Assuming that the measured value of the GNSS position information without spoofing attack is $\widetilde{\mathbf{p}}_1^G$, and the measured value of the spoofed signal with spoofing attack is $\mathbf{p}_1^G$, then the relationship between them is satisfied:

$$\widetilde{\mathbf{p}}_1^G = \mathbf{p}_1^G + \Delta\widetilde{\mathbf{p}}_1^{spo} \tag{3.8}$$

where $\Delta\widetilde{\mathbf{p}}_1^{spo} = \begin{bmatrix}\Delta\widetilde{L}_1 & \Delta\widetilde{\lambda}_1 & 0\end{bmatrix}^T$ denotes the spoofing parameter, which can essentially be expressed as the incremental spoofing position in the horizontal direction that the spoofing source adds to the true GNSS signal when it is not spoofed.

At this point, the actual Kalman filter measurement value $\widetilde{\mathbf{Z}}_1$ and the Kalman filter measurement value $\mathbf{Z}_1$ that has not been spoofed satisfy:

$$\widetilde{\mathbf{Z}}_1 = \mathbf{Z}_1 + \Delta\widetilde{\mathbf{p}}_1^{spo} \tag{3.9}$$

Therefore, the measurement update equation affected by the spoofing signal and the measurement update equation when it is not disturbed by spoofing can be expressed as follows, respectively:

$$\widetilde{\mathbf{X}}_1 = \hat{\mathbf{X}}_{1/0} + \mathbf{K}_1^G(\widetilde{\mathbf{Z}}_1 - \mathbf{H}_G\hat{\mathbf{X}}_{1/0}) \tag{3.10a}$$

$$\hat{\mathbf{X}}_1 = \hat{\mathbf{X}}_{1/0} + \mathbf{K}_1^G(\mathbf{Z}_1 - \mathbf{H}_G\hat{\mathbf{X}}_{1/0}) \tag{3.10b}$$

where $\mathbf{K}_1^G$ denotes the filter gain matrix at the current moment. Then, the state error of the MSF system before and after the spoofing attack can be expressed as:

$$\Delta\widetilde{\mathbf{X}}_1 = \widetilde{\mathbf{X}}_1 - \hat{\mathbf{X}}_1 \tag{3.11}$$

The state errors before and after the spoofing attack are further obtained:

$$\Delta\widetilde{\mathbf{X}}_1 = \mathbf{K}_1^G \cdot \Delta\widetilde{\mathbf{p}}_1^{spo} \tag{3.12}$$

3.2.2 SINS State Recursive Error Transfer Model under Spoofing Attack

After the GNSS and LiDAR have just been updated, since the frequency of SINS updates is generally higher than that of GNSS and LiDAR, only SINS information is received in the MSF system of AVs in the subsequent k_{IL} epochs, and no positional metrics are received from GNSS and LiDAR, so the filter process only performs a state recursive process without metrics updating.

Firstly, the state one-step prediction value and the one-step prediction mean square error array are denoted as:

$$\widetilde{\mathbf{X}}_{2/1} = \mathbf{\Phi}_{2/1}\widetilde{\mathbf{X}}_1 \tag{3.13a}$$

$$\mathbf{P}_{2/1} = \mathbf{\Phi}_{2/1}\mathbf{P}_1\mathbf{\Phi}_{2/1}^T + \mathbf{Q} \tag{3.13b}$$

Since there is no measurement update process, the result of the final state recursive process and the mean square error matrix are unchanged and can be expressed as follows:

$$\tilde{\mathbf{X}}_2 = \tilde{\mathbf{X}}_{2/1} \tag{3.14a}$$

$$\mathbf{P}_2 = \mathbf{P}_{2/1} \tag{3.14b}$$

LiDAR is generally updated more frequently than GNSS, and after k_{IL} epochs, the MSF system will receive the LiDAR position measurement information. Therefore, the final state recursive estimation result before the LiDAR measurement information is updated can be expressed as:

$$\tilde{\mathbf{X}}_{k_{IL}+1} = \prod_{\eta=1}^{k_{IL}} \mathbf{\Phi}_{(\eta+1)/\eta} \cdot \tilde{\mathbf{X}}_1 \tag{3.15a}$$

$$\mathbf{P}_{k_{IL}+1} = \mathbf{\Phi}_{(k_{IL}+1)/k_{IL}}\mathbf{P}_{k_{IL}}\mathbf{\Phi}_{(k_{IL}+1)/k_{IL}}^T + \mathbf{Q} \tag{3.15b}$$

From the results of the above equation, it can be seen that the state variance array of the system gradually increases during the SINS state recursion process. The difference between the results of the state recursion process before and after spoofing can be expressed as:

$$\tilde{\mathbf{X}}_{k_{IL}+1} = \hat{\mathbf{X}}_{k_{IL}+1} + \Delta\tilde{\mathbf{X}}_{k_{IL}+1} \tag{3.16}$$

Therefore, the state error due to spoofing can be expressed as:

$$\Delta\tilde{\mathbf{X}}_{k_{IL}+1} = \prod_{\eta=1}^{k_{IL}} \mathbf{\Phi}_{\eta+1/\eta} \cdot \Delta\tilde{\mathbf{X}}_1 \tag{3.17}$$

Since $\mathbf{\Phi}_{k/k-1} \approx \mathbf{I} + \mathbf{F}(t_{k-1})T_s$ and it has been assumed that the motion of the AV is relatively stable, the state of the system can be regarded as an invariant for a short period, i.e., it can be assumed that the system matrix is invariant, i.e., $\mathbf{\Phi}_{k/k-1} \approx \mathbf{I} + \mathbf{F}_0/f_I$, therefore:

$$\Delta\tilde{\mathbf{X}}_{k_{IL}+1} \approx (\mathbf{\Phi}_{1/0})^{k_{IL}} \cdot \Delta\tilde{\mathbf{X}}_1 \tag{3.18}$$

Therefore:

$$\Delta\tilde{\mathbf{X}}_{k_{IL}+1} \approx [\mathbf{I} + \mathbf{F}_0/f_I]^{k_{IL}} \cdot \Delta\tilde{\mathbf{X}}_1 \tag{3.19}$$

Expand the above equation binomially:

$$\Delta\tilde{\mathbf{X}}_{k_{IL}+1} = \left[\mathbf{I} + C_{k_{IL}}^1 \cdot \mathbf{F}_0/f_I + C_{k_{IL}}^2 \cdot [\mathbf{F}_0/f_I]^2 + \cdots + C_{k_{IL}}^{k_{IL}} \cdot [\mathbf{F}_0/f_I]^{k_{IL}}\right] \cdot \Delta\tilde{\mathbf{X}}_1 \tag{3.20}$$

where $C_{k_{IL}}^i = \frac{k_{IL}!}{i!(k_{IL}-i)!}$ denotes the binomial coefficients, where $i = 1, \cdots, k_{IL}$. Simplify the above equation as follows:

$$\Delta\tilde{\mathbf{X}}_{k_{IL}+1} = [\mathbf{I} + \mathbf{A}_1 + \mathbf{A}_2 + \cdots + \mathbf{A}_{k_{IL}}] \cdot \Delta\tilde{\mathbf{X}}_1 \tag{3.21}$$

where:

$$\mathbf{A}_i = C_{k_{IL}}^i \cdot \left[\mathbf{F}(t_{k-1})/f_I\right]^i, i = 1, \cdots, k_{IL} \tag{3.22}$$

Further, one can derive $\mathbf{A}_i$ the ratio relation between the previous and the next epoch:

$$\frac{\mathbf{A}_i}{\mathbf{A}_{i-1}} = \frac{C_{k_{IL}}^i}{C_{k_{IL}}^{i-1}} \cdot \mathbf{F}(t_{k-1}) \tag{3.23}$$

Then, the maximum value of the ratio is calculated:

$$\max\left(\frac{\mathbf{A}_i}{\mathbf{A}_{i-1}}\right) = \left(\frac{1}{2f_L} - \frac{1}{2f_I}\right) \cdot \mathbf{F}(t_{k-1}) \tag{3.24}$$

The final state error is obtained by simplifying by ignoring higher-order terms:

$$\Delta\tilde{\mathbf{X}}_{k_{IL}+1} \approx \left[\mathbf{I} + C_{k_{IL}}^1 \cdot \mathbf{F}(t_{k-1})/f_I\right] \cdot \Delta\tilde{\mathbf{X}}_1 = \left[\mathbf{I} + \mathbf{F}(t_{k-1})/f_L\right] \cdot \Delta\tilde{\mathbf{X}}_1 \tag{3.25}$$

Therefore, the final state error is expressed as:

$$\Delta\tilde{\mathbf{X}}_{k_{IL}+1} = [\mathbf{I} + \mathbf{F}_0 / f_L] \cdot \mathbf{K}_1^G \cdot \Delta\tilde{\mathbf{p}}_1^{spo} \tag{3.26}$$

The above derivation results show that the amount of position error change due to spoofing attack is minimal during the finite number of state recursions, thus theoretically explaining that the SINS has little effect on the result of position information updates through state recursion between two LiDAR measurement information updates. Therefore, when SINS operates normally, it cannot effectively correct the position error that has been generated.

3.2.3 LiDAR Measurement Update Error Transfer Model under Spoofing Attack

During the state recursion process, the spoofing signal does not lead to a change in the system covariance matrix. Since the state estimation mean-square error matrix is unchanged, the gain matrix is unchanged when the LiDAR's positional information is measured and updated, at which point the measurement update equation can be expressed as follows:

$$\tilde{\mathbf{X}}_{k_{IL}+1} = \tilde{\mathbf{X}}_{(k_{IL}+1)/k_{IL}} + \mathbf{K}_{k_{IL}+1}^L \left(\mathbf{Z}_{k_{IL}} - \mathbf{H}_L \tilde{\mathbf{X}}_{(k_{IL}+1)/k_{IL}} \right) \tag{3.27}$$

where $\mathbf{K}_{k_{IL}+1}^G = \mathbf{P}_{(k_{IL}+1)/(k_{IL})} \mathbf{H}_L^{\mathrm{T}} (\mathbf{H}_L \mathbf{P}_{(k_{IL}+1)/(k_{IL})} \mathbf{H}_L^{\mathrm{T}} + \mathbf{R}_L)^{-1}$. Theoretically, the quantitative update result when there is no spoofing attack is:

$$\hat{\mathbf{X}}_{k_{IL}+1} = \hat{\mathbf{X}}_{(k_{IL}+1)/k_{IL}} + \mathbf{K}_{k_{IL}+1}^L \left(\mathbf{Z}_{k_{IL}} - \mathbf{H}_L \hat{\mathbf{X}}_{(k_{IL}+1)/k_{IL}} \right) \tag{3.28}$$

The difference between the state recursion results before and after the spoofing attack can be expressed as:

$$\tilde{\mathbf{X}}_{(k_{IL}+1)/(k_{IL})} = \hat{\mathbf{X}}_{(k_{IL}+1)/k_{IL}} + \Delta\tilde{\mathbf{X}}_{k_{IL}} \tag{3.29}$$

Further derivation leads to:

$$\Delta\tilde{\mathbf{X}}_{k_{IL}+1} = \left(\mathbf{I} - \mathbf{K}_{k_{IL}+1}^L \mathbf{H}_L \right) \Delta\tilde{\mathbf{X}}_{k_{IL}} \tag{3.30}$$

From the above equation, when the LiDAR position result is still correct, the LiDAR's ability to correct the position error caused by the spoofing attack is mainly related to the gain matrix at this time, which is mainly related to the state covariance matrix and the

measurement noise array at this time. The state error at this time can be expressed as:

$$\Delta\tilde{\mathbf{X}}_{k_{IL}+1} \approx \left(\mathbf{I}-\mathbf{K}_{k_{IL}+1}^{L}\mathbf{H}_{L}\right)\Delta\tilde{\mathbf{X}}_{1} \tag{3.31}$$

Therefore, the error in the state of the MSF system due to the previous spoofed signal before the next GNSS message comes can be expressed as:

$$\Delta\tilde{\mathbf{X}}_{k_{IG}+1} \approx \left(\mathbf{I}-\mathbf{K}_{k_{IG}+1}^{L}\mathbf{H}_{L}\right)\cdots\left(\mathbf{I}-\mathbf{K}_{k_{IL}+1}^{L}\mathbf{H}_{L}\right)\mathbf{K}_{1}^{G}\cdot\Delta\tilde{\mathbf{p}}_{1}^{spo} \tag{3.32}$$

where:

$$\begin{cases} \mathbf{K}_{1}^{G}=\mathbf{P}_{1/0}\mathbf{H}_{G}^{\mathrm{T}}\left(\mathbf{H}_{G}\mathbf{P}_{1/0}\mathbf{H}_{G}^{\mathrm{T}}+\mathbf{R}_{G}\right)^{-1} \\ \mathbf{K}_{k+1}^{L}=\mathbf{P}_{k+1/k}\mathbf{H}_{L}^{\mathrm{T}}\left(\mathbf{H}_{L}\mathbf{P}_{k+1/k}\mathbf{H}_{L}^{\mathrm{T}}+\mathbf{R}_{L}\right)^{-1} \\ \mathbf{P}_{1/0}=\mathbf{\Phi}_{1/0}\mathbf{P}_{0}\mathbf{\Phi}_{1/0}^{\mathrm{T}}+\mathbf{Q} \\ k_{IG}=f_{I}/f_{G} \\ k_{IL}=f_{I}/f_{L} \end{cases} \tag{3.33}$$

In this section, the state recursive error transfer model and the measurement update error transfer model of the Kalman filter under a spoofing attack in a spoofing signal update cycle are derived in detail. The results are shown in Fig. 3.2 .

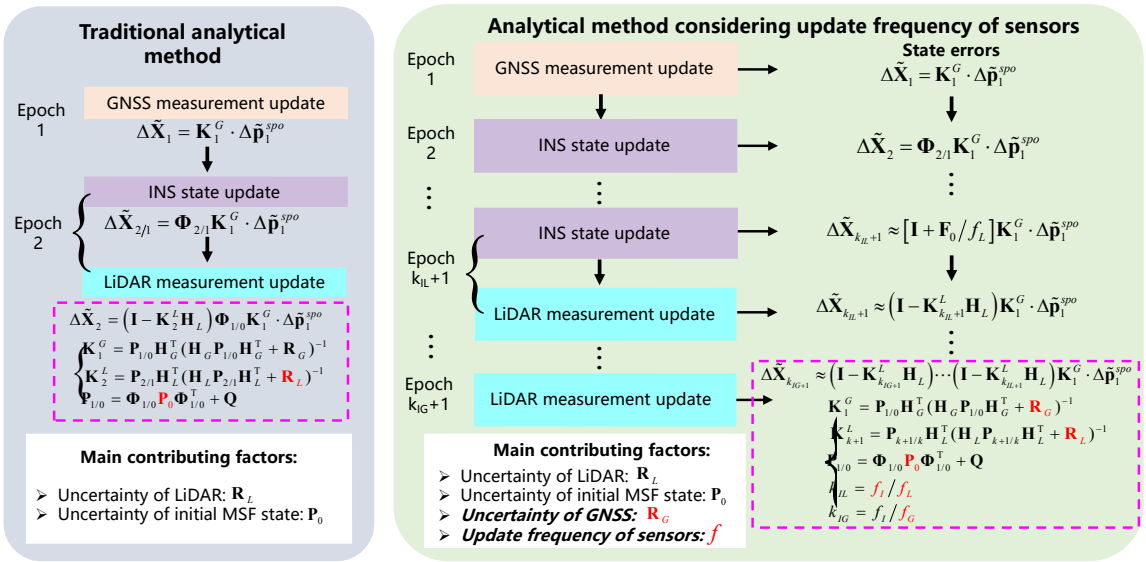

Figure 3.2: Information filter update process and state error of the analytical method considering sensors update frequency

It can be seen from the final derivation results that, on the one hand, the state error caused by spoofing signal $\Delta\tilde{\mathbf{X}}_{k_{IG}+1}$ is not only related to the LiDAR uncertainty $\mathbf{R}_L$ and the initial state covariance matrix $\mathbf{P}_0$ in the spoofing cycle. Uncertainty $\mathbf{R}_L$ and the initial MSF

state covariance matrix $\mathbf{P}_0$ in the spoofing cycle, but also with the uncertainty of GNSS $\mathbf{R}_G$ and the sensor update frequency f. According to the formula of the final state error of spoofing attack in the GNSS update cycle, after further analysis, the relationship between these parameters and the state error is as follows: the state error is positively correlated with $\mathbf{R}_L$ and $\mathbf{P}_0$, i.e., the smaller the $\mathbf{R}_L$ and $\mathbf{P}_0$ are, The smaller the state error of the navigation system is. The state error is negatively correlated with $\mathbf{R}_G$, i.e., the larger $\mathbf{R}_G$ is, the smaller the state error of the MSF system caused by the spoofing attack is. On the other hand, the state error $\Delta\tilde{\mathbf{X}}_{k_{IG}+1}$ caused by the spoofing signal is directly related to the spoofing parameter $\Delta\tilde{\mathbf{p}}_1^{spo}$ set by the spoofing source, and the size of the spoofing parameter will have a direct impact on the final result of the error state. Therefore, setting reasonable spoofing parameters is crucial to the success of a spoofing attack. The error transfer model of the MSF system under a spoofing attack, established in this section, takes into account the difference between sensor update frequencies, which is more accurate than the traditional model.

3.3 An Analytic Model of Information Filter under a GNSS Spoofing Attack

To make the relationship between the positioning error caused by the spoofing attack and the factors of $\mathbf{R}_L$, $\mathbf{R}_G$, $\mathbf{P}_0$, and f more explicit, we firstly ignore the INS state update process in the MSF system. As described above, the INS state update process has a minimal impact on positioning errors caused by GNSS spoofing, which has been explained. Therefore, we ignore this process in the MSF system. Secondly, due to the complicated measurement update process in the standard Kalman filter, we re-establish an analytical model using an information filter, thereby avoiding the complex inverse process inherent in the standard measurement update process.

3.3.1 Error Analysis of GNSS Information Update Process

Similar to Section 3, we also assume the LiDAR measurement information has just been updated at epoch 0, then a GNSS signal exists at epoch 1. According to the information filter process, the information vector prediction value and the one-step information matrix can be expressed as:

$$\mathbf{I}_{1/0}^{-1} = \Phi_{1/0}\mathbf{P}_0\Phi_{1/0}^T + \mathbf{Q} \tag{3.34a}$$

$$\hat{\mathbf{S}}_{1/0} = \mathbf{I}_{1/0}\hat{\mathbf{X}}_{1/0} \tag{3.34b}$$

Then, the information matrix can be expressed as:

$$\mathbf{I}_1 = \mathbf{I}_{1/0} + \mathbf{H}_1^T \mathbf{R}_G^{-1} \mathbf{H}_1 \tag{3.35}$$

The spoofed information vector update equation and the actual information vector update equation are:

$$\widetilde{\mathbf{S}}_1 = \hat{\mathbf{S}}_{1/0} + \mathbf{H}_1^T \mathbf{R}_G^{-1} \widetilde{\mathbf{Z}}_1 \tag{3.36a}$$

$$\hat{\mathbf{S}}_1 = \hat{\mathbf{S}}_{1/0} + \mathbf{H}_1^T \mathbf{R}_G^{-1} \hat{\mathbf{Z}}_1 \tag{3.36b}$$

As we can see, the information matrix $\hat{\mathbf{S}}_{1/0}$ of the information filter is identical. Then, the information vector error caused by the GNSS spoofing attack can be expressed as:

$$\Delta\widetilde{\mathbf{S}}_1 = \mathbf{H}_1^T \mathbf{R}_G^{-1} \Delta\widetilde{\mathbf{Z}}_1 \tag{3.37}$$

where $\Delta\widetilde{\mathbf{S}}_1 = \mathbf{I}_1 \Delta\widetilde{\mathbf{X}}_1$, $\Delta\widetilde{\mathbf{Z}}_1 = \Delta\widetilde{\mathbf{p}}_1^{Spo}$, and then the state error after the GNSS spoofing can be further obtained.

$$\Delta\widetilde{\mathbf{X}}_1 = \mathbf{I}_1^{-1} \mathbf{H}_1^T \mathbf{R}_G^{-1} \Delta\widetilde{\mathbf{p}}_1^{Spo} \tag{3.38}$$

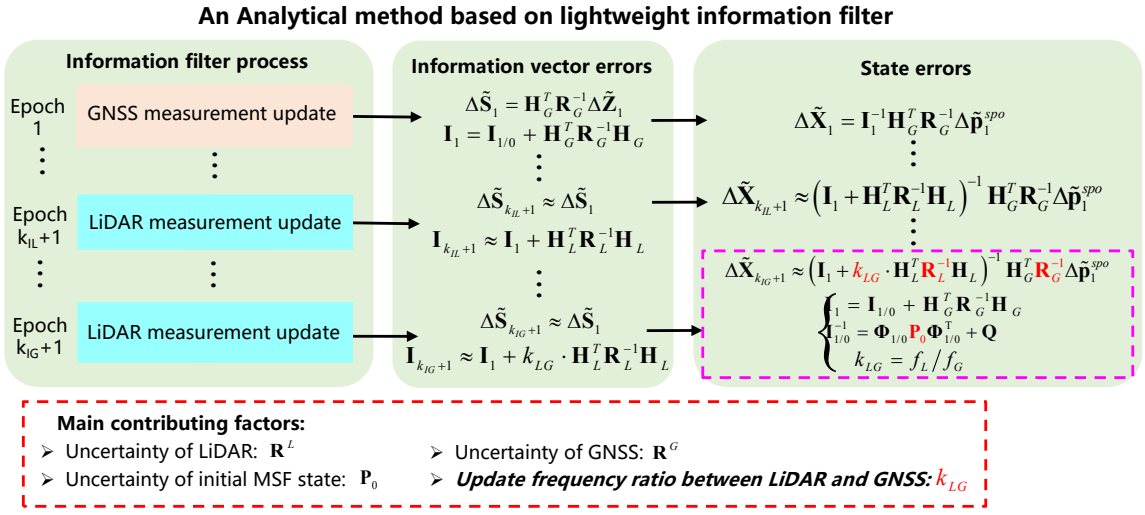

Figure 3.3: Information filter update process and state error of the analytical method considering sensors update frequency

3.3.2 Error Analysis of LiDAR Information Update Process

Since the LiDAR update frequency f_L is greater than the GNSS update frequency f_G in general, the following position measurement information is LiDAR at epoch k_{IL}. Based on the above analysis, we ignore the INS status update process in the MSF system, so there is little change in the information vector error and the information matrix.

$$\Delta\widetilde{\mathbf{S}}_{k_{IL}+1/k_{IL}} \approx \Delta\widetilde{\mathbf{S}}_1 \tag{3.39a}$$

$$\mathbf{I}_{k_{IL}+1/k_{IL}} \approx \mathbf{I}_1 \tag{3.39b}$$

The information update equation can be expressed as:

$$\widetilde{\mathbf{S}}_{k_{IL}+1} = \widetilde{\mathbf{S}}_{k_{IL}+1/k_{IL}} + \mathbf{H}_2^T \mathbf{R}_L^{-1} \mathbf{Z}_{k_{IL}+1} \tag{3.40a}$$

$$\mathbf{I}_{k_{IL}+1} = \mathbf{I}_{k_{IL}+1/k_{IL}} + \mathbf{H}_2^T \mathbf{R}_L^{-1} \mathbf{H}_2 \tag{3.40b}$$

Theoretically, when there is no GNSS spoofing attack, the information update equation is:

$$\hat{\mathbf{S}}_{k_{IL}+1} = \hat{\mathbf{S}}_{k_{IL}+1/k_{IL}} + \mathbf{H}_2^T \mathbf{R}_L^{-1} \mathbf{Z}_{k_{IL}+1} \tag{3.41}$$

The difference between the information vector before and after the GNSS spoofing attack can be expressed:

$$\Delta\widetilde{\mathbf{S}}_{k_{IL}+1} = \Delta\widetilde{\mathbf{S}}_{k_{IL}+1/k_{IL}} \tag{3.42}$$

Because $\Delta\widetilde{\mathbf{S}}_{k_{IL}+1/k_{IL}} \approx \Delta\widetilde{\mathbf{S}}_1$ and $\mathbf{I}_{k_{IL}+1/k_{IL}} \approx \mathbf{I}_1$, the information vector error and information matrix can be simplified.

$$\Delta\widetilde{\mathbf{S}}_{k_{IL}+1} \approx \Delta\widetilde{\mathbf{S}}_1 \tag{3.43a}$$

$$\mathbf{I}_{k_{IL}+1} \approx \mathbf{I}_1 + \mathbf{H}_2^T \mathbf{R}_L^{-1} \mathbf{H}_2 \tag{3.43b}$$

Further derivation can be obtained:

$$\Delta\widetilde{\mathbf{X}}_{k_{IL}+1} \approx \mathbf{I}_{k+1}^{-1} \mathbf{I}_{k_{IL}+1/k_{IL}} \Delta\widetilde{\mathbf{X}}_{k_{IL}+1/k_{IL}} \tag{3.44}$$

Then, the state error can be expressed as:

$$\Delta\widetilde{\mathbf{X}}_{k_{IL}+1} \approx \left(\mathbf{I}_1 + \mathbf{H}_2^T \mathbf{R}_L^{-1} \mathbf{H}_2\right)^{-1} \mathbf{H}_1^T \mathbf{R}_G^{-1} \Delta\widetilde{\mathbf{p}}_1^{Spo} \tag{3.45}$$

From the above formula, when the position results of the LiDAR are still correct, the correction capability of the positioning error caused by GNSS spoofing is mainly related to the filter information matrix at this epoch and the initial information matrix $\mathbf{I}_1$.

Theoretically, after the information update process of LiDAR, the state update process is re-entered until the MSF system receives the next LiDAR epoch, then the measurement update is performed. However, we ignore the state update process during this analysis process, so when the MSF system receives the following LiDAR values at epoch $2k_{IL}+1$. Then, the difference between the information vector error before and after the GNSS spoofing attack can be expressed:

$$\Delta\widetilde{\mathbf{S}}_{2k_{IL}+1} = \Delta\widetilde{\mathbf{S}}_{k_{IL}+1/k_{IL}} \tag{3.46a}$$

$$\mathbf{I}_{2k_{IL}+1} = \mathbf{I}_{k_{IL}+1/k_{IL}} + \mathbf{H}_2^T\mathbf{R}_L^{-1}\mathbf{H}_2 \tag{3.46b}$$

where $\mathbf{I}_{k_{IL}+1/k_{IL}} \approx \mathbf{I}_{k_{IL}+1}, \Delta\widetilde{\mathbf{S}}_{k_{IL}+1/k_{IL}} \approx \Delta\widetilde{\mathbf{S}}_{k_{IL}+1}$. Then, further derivation can be obtained:

$$\Delta\widetilde{\mathbf{S}}_{2k_{IL}+1} \approx \Delta\widetilde{\mathbf{S}}_1 \tag{3.47a}$$

$$\mathbf{I}_{2k_{IL}+1} \approx \mathbf{I}_1 + 2\cdot\mathbf{H}_2^T\mathbf{R}_L^{-1}\mathbf{H}_2 \tag{3.47b}$$

The state error can be expressed as:

$$\Delta\widetilde{\mathbf{X}}_{2k_{IL}+1} \approx \left(\mathbf{I}_1 + 2\cdot\mathbf{H}_2^T\mathbf{R}_L^{-1}\mathbf{H}_2\right)^{-1}\mathbf{H}_1^T\mathbf{R}_G^{-1}\Delta\widetilde{\mathbf{p}}_1^{Spo} \tag{3.48}$$

Due to the low GNSS update frequency, the information update process of LiDAR is repeated before the next GNSS epoch is received, and there are k_{LG} cycles. There are only the measurement update process of LiDAR and the state update process of INS. If the vehicle is relatively stable, the system noise characteristics are unchanged. From epoch 1 to epoch $k_{IG}+1$, the state noise and measurement noise are changed little. Therefore, before the next GNSS epoch comes at epoch $k_{IG}+1$, the information vector error changes little so that it can be approximately simplified as:

$$\Delta\widetilde{\mathbf{S}}_{k_{IG}+1} \approx \Delta\widetilde{\mathbf{S}}_1 \tag{3.49a}$$

$$\mathbf{I}_{k_{IG}+1} \approx \mathbf{I}_1 + k_{LG}\cdot\mathbf{H}_2^T\mathbf{R}_L^{-1}\mathbf{H}_2 \tag{3.49b}$$

Because $\Delta\widetilde{\mathbf{S}}_1 = \mathbf{H}_1^T \mathbf{R}_G^{-1} \Delta\widetilde{\mathbf{p}}_1^{Spo}$ and $\Delta\widetilde{\mathbf{X}}_{k_{IG}+1} = \mathbf{I}_{k_{IG}+1}^{-1} \Delta\widetilde{\mathbf{S}}_1$, Then the state error caused by the GNSS spoofing attack can be approximately simplified as:

$$\Delta\widetilde{\mathbf{X}}_{k_{IG}+1} \approx \left(\mathbf{I}_1 + k_{LG} \cdot \mathbf{H}_2^T \mathbf{R}_L^{-1} \mathbf{H}_2\right)^{-1} \mathbf{H}_1^T \mathbf{R}_G^{-1} \Delta\widetilde{\mathbf{p}}_1^{Spo} \tag{3.50}$$

3.3.3 Analysis

We summarize the actual filter process and the state errors due to the GNSS spoofing attack, as shown in Fig. 3.3 . By comparing Fig. 3.2 and Fig. 3.3 , although the lightweight information filter is essentially the same as the standard KF, it is more intuitive in the expression form of measurement information update.

For the MSF system, through the analytical model based on the information filter, the state error $\Delta\widetilde{\mathbf{X}}_{k_{IG}+1}$ caused by the attacker is not only mainly related to LiDAR uncertainty $\mathbf{R}_L$, initial MSF state uncertainty $\mathbf{P}_0$, and GNSS uncertainty $\mathbf{R}_G$, but also related to the update frequency ratio k_{LG} between LiDAR and GNSS. Different from the results obtained by the standard KF process, the form of the state error is simpler using the simplified analytical model proposed in this chapter. From the final state error formula due to the GNSS spoofing attack in one GNSS update cycle, we can easily analyze the relationship between these essential parameters and the state error.

1. The state error is positively correlated with the $\mathbf{R}_L$ and $\mathbf{P}_0$. That is to say, the smaller the $\mathbf{R}_L$ and $\mathbf{P}_0$, the smaller the state error caused by the GNSS spoofing attack.

2. The state error is negatively correlated with the $\mathbf{R}_G$ and k_{LG}, which means the larger the $\mathbf{R}_G$ and k_{LG}, the smaller the state error caused by the GNSS spoofing attack.

3.4 Real-world Data Verification

Due to the limitation of test conditions, it is not possible to directly conduct real spoofing attack tests on AVs, so the algorithm proposed in this section is mainly verified using real-world data simulation tests. In this chapter, the environment of a spoofing attack is simulated by the existing dataset and simulated spoofing signals, and the spoofing parameter, i.e., spoofing position increment, is directly added to the real GNSS position output when it is not subjected to spoofing attack so that spoofing signals targeting GNSS can be generated by simulation.

Firstly, establish an autonomous development software test platform based on the unmanned open-source platform Autoware [21] and PSINS C++ toolkit [83], on which the

GNSS/SINS/LiDAR multi-source navigation-based on the Kalman filter model is implemented. Under the simulated spoofing attack conditions, the constant Fusion-ripper parameter and the exponential Fusion-ripper parameter that maximize the lateral deviation of the positioning results of the MSF system are found by setting different MSF system parameters for the Fusion-ripper-based spoofing attack algorithm. When the lateral deviation of the vehicle exceeds the safety threshold, it indicates that spoofing can be successful. Conversely, when the lateral deviation of the vehicle cannot exceed the safety threshold, it indicates that spoofing cannot be successful in this scenario. Since the uncertainty of LiDAR and the initial covariance matrix associated with SINS are strongly correlated with MSF system state errors in the existing literature, this chapter focuses on the validation of the spoofing algorithm under conditions of varying GNSS uncertainty and the update frequency ratio between LiDAR and GNSS.

3.4.1 Setup

In this chapter, the KAIST dataset is selected to validate the results of the theoretical analysis, which encompasses various types of urban scenarios [84] and features diverse navigation sensors. The specifications and related parameters of the sensors required in this chapter are described as shown in Tab. 3.1 .

Table 3.1: Main specifications of relevant sensors

Sensors	Version	Update frequency	Related parameters
3D LiDAR	VLP-16 Velodyne	10Hz	16 CH LiDAR (360° FOV), 100m
IMU	Xsens Mti-300	200Hz	Consumer grade AHRS, Zero bias: 10°/h
GNSS	U-Blox EVK-7P	5Hz	Consumer-grade GPS, Accuracy: 2.5m

Two 16-wire 3D LiDARs are installed on the left and right sides of the test vehicle to maximize data acquisition coverage with an update frequency of 10 Hz. The core component of SINS is the Inertial Measurement Unit (IMU), which measures the acceleration and angular velocity information of the vehicle with an update frequency of 200 Hz.GNSS provides absolute position information with an update frequency of 5 Hz. In addition, the dataset utilizes an incremental smoothing and map attitude map SLAM framework to estimate the reference values of the vehicle's accurate trajectory [85].

The navigation data is selected according to the following conditions to ensure successful spoofing.

1. The data collection scenario is a relatively empty suburban scenario with good GNSS signals (high positioning accuracy and low uncertainty).

2. The feature points on both sides of the vehicle body during data collection are relatively limited, so LiDAR has moderate localization accuracy and uncertainty. Suppose there are insufficient feature points on both sides of the road where the vehicle is located. In that case, the NDT matching algorithm may be completely ineffective, resulting in LiDAR failing to provide effective localization results.

According to the above conditions, the KAIST dataset in Urban-07 is selected as the real data for validation, as it contains GNSS signals with high credibility in the initial stage and has limited feature points on both sides of the lane. The real scenario graph of this dataset is shown in Fig. 3.4 (a). Since high-precision point cloud maps are not provided in the KAIST dataset, this thesis first constructs LiDAR point cloud maps using the localization module in Autoware. Then, it obtains the final LiDAR localization results based on the NDT matching method. The final LiDAR point cloud and NDT matching results are shown in Fig. 3.4 (b).

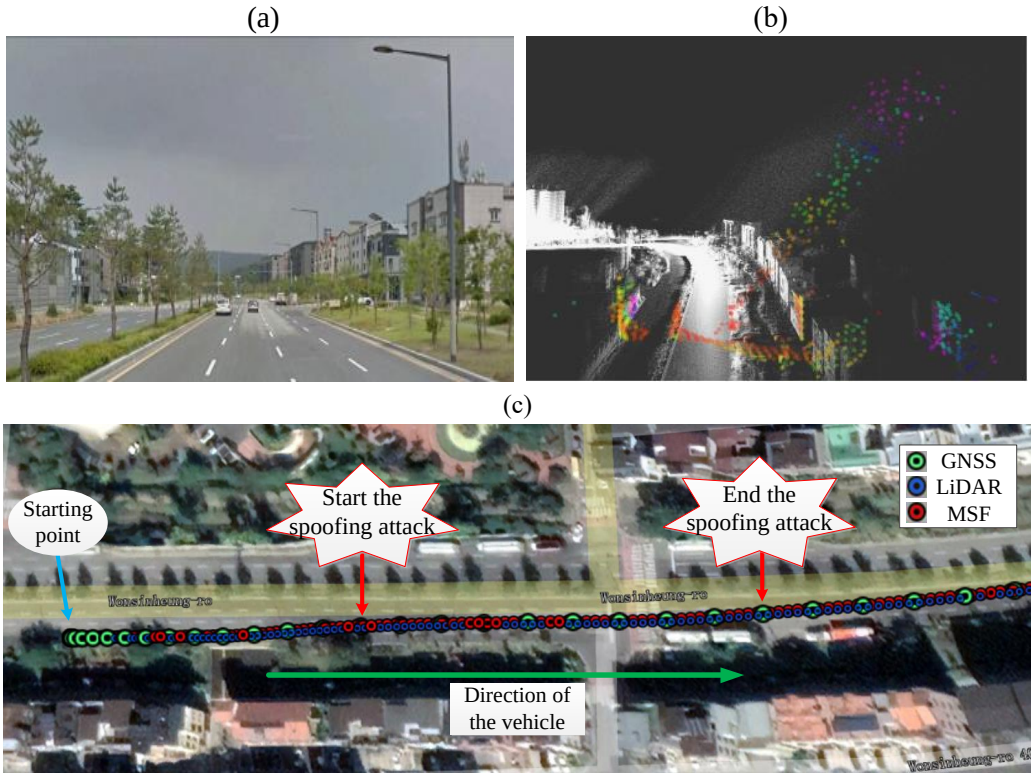

Figure 3.4: Actual scenario (a) and LiDAR point cloud (b) of Urban-07 in the KAIST dataset and the MSF system results (c) without spoofing attack

According to the data from Urban-07, the test vehicle starts from a stationary state and travels from west to east along a predetermined trajectory, beginning at 5 seconds. The update frequencies of GNSS, LiDAR, and SINS are 5 Hz, 10 Hz, and 200 Hz, respectively.

To fully validate the effects of different sensor update frequencies on the spoofing results, in this chapter, the sensor update frequencies are changed by downsampling (5 Hz or 1 Hz for GNSS, and 10 Hz or 5 Hz for LiDAR) and through the different combinations the ratio of sensor update frequencies can be set to 1, 2, 5 and 10. In this section, the update frequencies of GNSS and LiDAR are initially set to 1 Hz and 5 Hz, respectively. Then, the data from GNSS, LiDAR, and SINS are used to implement the MSF system positioning using a Kalman filter. The final positioning results of the MSF system in the n-frame before the simulated spoofing attack are shown in Fig. 3.4 (c), where the location where the spoofing attack started and the location where the spoofing is stopped are also marked in the figure. Then, simulation experiments are conducted under different conditions to analyze the positioning results of the MSF system under a spoofing attack.

3.4.2 Results of MSF System Localization under Spoofing Attack

Under different system parameter settings, the spoofing sources simulate the GNSS positioning results after the spoofing attack by the spoofing algorithm introduced in Chapter 2. Firstly, the parameters d and f need to be found to maximize the lateral deviation of the output positioning results within the spoofing window. If the parameters are too small, the spoofing time will be longer; conversely, if the parameters are too large, the spoofed signals will be easily recognized by the Chi-square detection algorithm within the MSF system. Therefore, the constant value parameter d is set to range from 0.3m to 1.5m with a sampling interval of 0.1m, and the exponential parameter f is set to vary from 1.1 to 1.5, with a sampling interval of 0.1. The parameter combinations [35] are found by enumeration under different conditions to maximize the lateral bias of the positioning results of the MSF system. For the uncertainty of GNSS, it can be expressed by the standard deviation of the position in three directions.

$$\mathbf{R}_G = \begin{bmatrix} R_{Gx}^2 & 0 & 0 \\ 0 & R_{Gy}^2 & 0 \\ 0 & 0 & R_{Gz}^2 \end{bmatrix} \tag{3.51}$$

where R_{Gx}, R_{Gy} and R_{Gz} denote the standard deviation of the GNSS position in the three directions in the b-frame, so changing the standard deviation of the GNSS position represents changing the GNSS uncertainty. The spoofing attack starts at 20 seconds. The spoofing window is set to 10 s, and the MSF system is simulated to have been spoofed by setting up a simulated spoofing signal. To maintain consistent spoofing conditions, the spoofing values are kept constant for 1 second. A successful spoofing attack must ensure that it is not detected by the defence algorithm of the MSF system. We assume that the target AV

internally adopts the Chi-square detection defence algorithm to monitor the measurement information in real-time, to prevent the outliers and fault information from contaminating the whole MSF system of AVs, and the threshold of the Chi-square detection algorithm is calculated by the Eq. (2.18).

In addition, according to the safety threshold calculation method in Chapter 2, this chapter calculates the safety thresholds by vehicle width and lane width, and the final D_{th-1} and D_{th-2} are set to 0.75m and 2.86m [35], respectively. Constant value spoofing is performed at first, and the spoofing scheme shifts to exponential value spoofing when the lateral deviation exceeds the final D_{th-1}. When the lateral deviation of the output of the MSF system exceeds the final D_{th-2}, it indicates that the spoofing can be successful.

Ultimately, the Fusion-ripper parameters d and f that can be found in the spoofing window under different R_{Gx} and k_{LG} are counted to maximize the output lateral deviation within the spoofing window, as shown in Fig. 3.5 . The horizontal axis represents the standard deviation of the GNSS position between 0.1m and 4m, respectively, and the vertical axis represents the ratio of the update frequency of LiDAR and GNSS, which are 1, 2, 5, and 10, respectively. In Fig. 3.5 , the value of 0 indicates that no parameter could be found to make the maximum lateral deviation exceed 2.86m and, therefore, cannot successfully spoof the MSF system in this epoch.

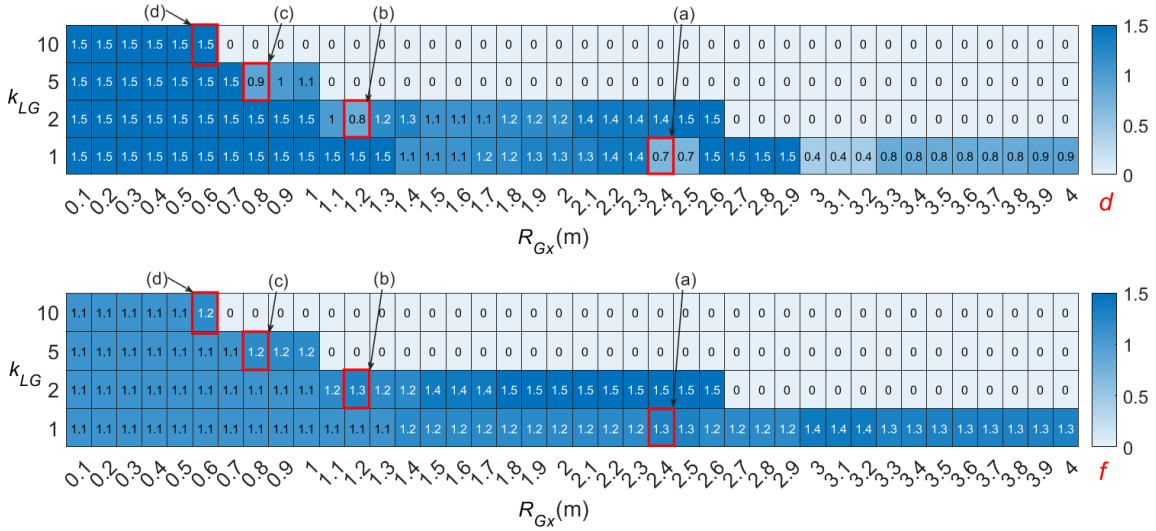

Figure 3.5: Parameters d and f under different parameter settings

In Fig. 3.5 , the blue part represents that the MSF system is in a scenario where it can be effectively spoofed, where the R_{Gx} and k_{LG} parameter values are small. Therefore, effective parameter combinations can be found to perform a successful spoofing attack. The white part represents that the MSF system is in a relatively safe scenario, where the

R_{Gx} and k_{LG} parameter values are large. Thus, no suitable parameter combination can be found that can cause the lateral deviation of the MSF system to exceed the safety threshold in the successive spoofing window.

According to Chapter 2, it can be seen that the Fusion-ripper spoofing algorithm aims to find the parameters d and f in the 2 spoofing stages to maximize the lateral deviation of the MSF system. Excessive parameter combinations may cause the positioning error of the MSF system to exceed the threshold of the Chi-square detection algorithm. At this point, the GNSS measurements will be regarded as outliers by the MSF system. Therefore, four typical successful parameter combinations are selected for analysis, which is marked with red boxes in Fig. 3.5 . This analysis aims to detail how the Chi-square detection defense algorithm limits the spoofing attack and to explain the conditions under which a successful spoofing attack can be performed. In the four cases, the exponential parameter f is gradually increased from 1.1 to 1.5 with a sampling interval of 0.1. The constant Fusion-ripper parameter d is found such that the lateral deviation of the output position of the MSF system is maximized, i.e., the parameter that exceeds the maximum will be recognized by the Chi-square detection algorithm in the system. Eventually, under the spoofing attack of these four sets of Fusion-ripper parameter settings, the position error results of the MSF system are shown in Fig. 3.6 .

The spoofing attack (20s-30s) on the MSF system is simulated by emulating the spoofing signals. Then, the lateral deviations due to Fusion-ripper spoofing parameters and spoofing attacks are counted for different parameter settings, as shown in Tab. 3.2 .

From the results, it is clear that although LiDAR can correct the positioning error between two GNSS systems, the positioning error of the MSF system will gradually increase as the spoofing attack continues. Four typical parameter combinations are selected for detailed analysis:

1. Fig. 3.6 (a) in the case of $f = 1.5$, there is no statistical data. In this case, no matter what value is set for the constant value parameter d, it will lead to the Chi-square value in the Chi-square detection algorithm in the MSF system exceeding the threshold value, so in this case, it is not possible to spoof successfully.

2. Fig. 3.6 (b) and Fig. 3.6 (c) in certain Fusion-ripper parameter combinations will trigger the GNSS take-over effect on the MSF system, i.e., with the gradual increase of the MSF system's positioning error, the correct LiDAR positioning results are regarded as outliers by the system's Chi-square detection algorithm is viewed as outliers, resulting in the MSF system completely trusting the spoofed GNSS measurements, so that the positioning

Table 3.2: Fusion-ripper spoofing parameters under different parameter settings and the corresponding lateral deviation

k_{LG}, R_{Gx}	f	d	Lateral deviation (m)
(a) k_{LG}=1, R_{Gx}=0.6m	1.1	1.5	3.58
	1.2	1.4	6.35
	1.3	**0.7**	**6.37**
	1.4	0.3	5.35
(b) k_{LG}=2, R_{Gx}=0.8m	1.1	1.5	4.60
	1.2	1.1	8.07
	1.3	**0.8**	**8.18**
	1.4	0.7	4.31
	1.5	0.7	8.13
(c) k_{LG}=5, R_{Gx}=1.2m	1.1	1.5	3.92
	1.2	**0.9**	**10.85**
	1.3	0.5	4.22
	1.4	0.3	0.94
	1.5	0.3	0.94
(d) k_{LG}=10, R_{Gx}=2.4m	1.1	1.5	2.40
	1.2	**1.5**	**3.18**
	1.3	1.4	3.08
	1.4	1.3	2.56
	1.5	1.3	2.84

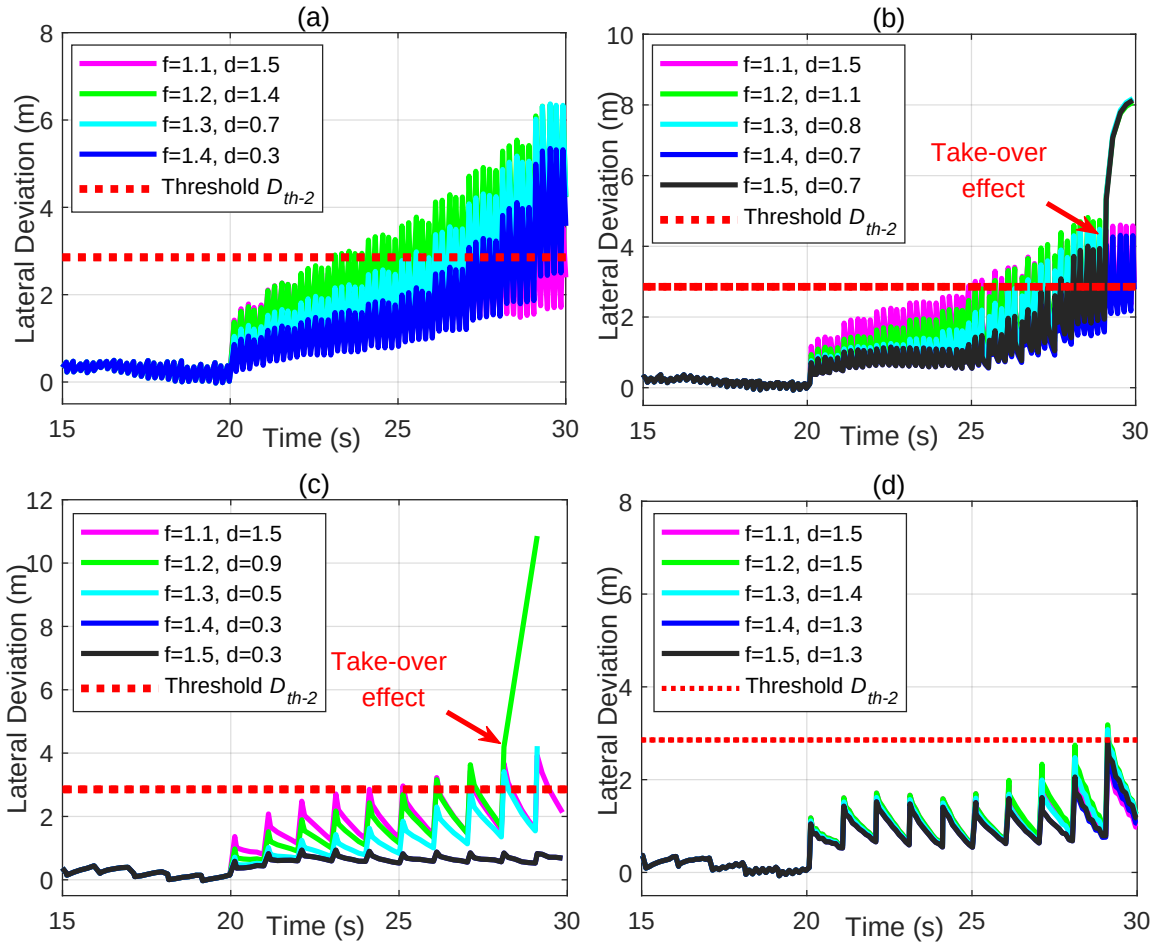

Figure 3.6: Lateral deviation curve in the spoofing window of 10s under different parameter settings

error of the output of the MSF system increases rapidly, thus realizing successful spoofing attack.

3. Fig. 3.6 (d) represents a critical scenario in which very few parameter combinations can cause the lateral deviation of the vehicle to exceed the threshold value. Although a successful spoofing attack can be executed with limited Fusion-ripper parameter configurations, it is difficult to find suitable Fusion-ripper parameters in practical applications due to the limitations of the Chi-square detection algorithms, making it challenging to realize spoofing attack in this scenario successfully.

According to the above results and analysis, on the one hand, the maximum value of the spoofing parameter d that can be found decreases with the increase of the parameter f due to the limitation of the Chi-square detection algorithm. Therefore, the maximum lateral deviation that can be achieved by one spoofing attack will be limited. On the other hand, for some spoofing schemes where the spoofing signals change slowly, it is challenging to

monitor the spoofing signals using the Chi-square detection algorithm due to the gradual increase in positioning error. Even the take-over effect may be triggered, i.e., the correct LiDAR positioning result may be rejected after being regarded as anomalous metric information in the MSF system. At this time, the system completely trusts the spoofed GNSS positioning result. Therefore, due to the vulnerability of the MSF system, under certain MSF parameter settings, a suitable combination of spoofing parameters can be identified, allowing the spoofing source to spoof the MSF system of the AV successfully.

3.5 Summary

In this chapter, the state error model of the MSF system under spoofing attack is established, including the GNSS measurement update error transfer model, the SINS state recursive error transfer model, and the LiDAR measurement update error transfer model. The final state error analytical equation is then obtained. The analytical results show that in one GNSS spoofed signal update cycle, the MSF system parameters and the size of the spoofing parameters directly affect the spoofing results. Eventually, the results of the real-world data simulation test show that under some MSF parameter settings, a suitable combination of spoofing parameters can be found so that the spoofing source can implement a successful spoofing attack on the MSF system. In this chapter, the error transfer model of the MSF system under a spoofing attack is investigated, which lays the theoretical foundation for subsequent in-depth research on the spoofing technology of the MSF system.

Chapter 4

A COVERT SPOOFING METHOD BASED ON FUZZY INFERENCE MODEL

4.1 Introduction

The MSF systems in AVs typically integrate multiple navigation sensors and exhibit strong resistance to spoofing signals. Consequently, spoofing attacks on MSF systems present unique technical challenges compared to direct attacks on GNSS receivers. To avoid detection by the MSF system's defense mechanisms, setting appropriate spoofing parameters is critical.

Traditional spoofing algorithms often suffer from poor concealment and low success rates. For instance, constant-value spoofing parameters may be too small to achieve effective deception within limited time frames, while exponential-value spoofing parameters, though requiring shorter spoofing durations, risk detection by the MSF system's defense algorithms due to excessively large parameters. The Fusion-ripper algorithm, which combines these two approaches, does not fully consider spoofing concealment. In some scenarios, spoofing signals may still be detected. Additionally, the algorithm employs an exhaustive search method for spoofing parameters, which is suitable for theoretical research but impractical for real-world applications. Furthermore, traditional Fusion-ripper algorithms lack clear criteria for judging the take-over effect, potentially leading to spoofing signal detection.

To address these challenges, this chapter proposes a covert spoofing method based on a fuzzy inference model. By dynamically adjusting spoofing parameters based on the target AV's state, this method enhances the success rates of spoofing. Specifically, real-time monitoring of the target AV calculates position error feedback adjustment factors, which are used to construct a fuzzy knowledge base and fuzzy rules. A fuzzy inference model based on the multi-Zadeh method dynamically adjusts spoofing parameters. During spoofing, the

system determines whether the take-over effect has been triggered by comparing position error feedback adjustment factors. If triggered, spoofing parameters are constrained to maximum values to prevent detection by the MSF system.

4.2 Framework for Covert Spoofing Based on Fuzzy Inference Model

The covert spoofing framework developed in this Chapter builds directly upon the foundational work presented in Chapter 2. While the Fusion-ripper algorithm effectively demonstrates the vulnerability of AV MSF systems and establishes a two-stage spoofing strategy (constant value attack followed by exponential value attack) to maximize lateral deviation, its reliance on static parameters limits its covertness and practical adaptability. This chapter proposes an advanced covert spoofing method that retains the core strategic goal but replaces the static parameterization with a dynamic, fuzzy inference-based control system. This evolution enables real-time adaptation of the spoofing attack based on the target AV's state, thereby significantly improving both the success rate and stealth of the spoofing attack.

This chapter establishes a covert spoofing model based on a fuzzy inference model. It utilizes the concept of fuzzy inference to comprehensively evaluate the position error of the target MSF system in the implementation of a spoofing attack, thereby enabling the dynamic adjustment of spoofing parameters. The framework for the proposed method is illustrated in Fig. 4.1 .

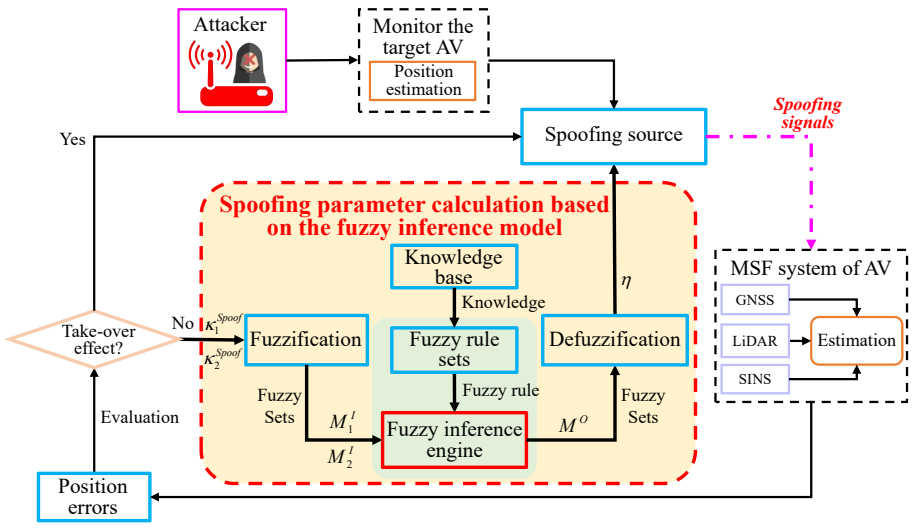

Figure 4.1: The framework of covert spoofing attack based on fuzzy inference model

In this chapter, a covert spoofing algorithm is proposed for AV, and a fuzzy inference model dynamically calculates more reasonable spoofing parameters. It mainly includes the following two aspects:

When the spoofing source starts broadcasting spoofing signals to the target AV, the 2 position error feedback factors in one spoofing cycle are calculated by real-time monitoring the lateral error of the target MSF system during the spoofing process. Then, based on the previous spoofing parameter, the following spoofing parameter is dynamically calculated based on the fuzzy inference model. After that, the next optimized spoofing signal is generated and sent to the target AV.

During the spoofing process, it is determined in real-time whether the spoofing signal has caused the MSF system to trigger a take-over effect, i.e., the MSF system completely believes in the spoofed GNSS signals, and the system mistakenly considers the correct LiDAR to be completely invalid. Suppose it is detected that the MSF system has triggered a take-over effect. In that case, the spoofing parameter is then constrained to a maximum value to prevent excessive spoofing parameters from being detected by the Chi-square detection algorithm in the MSF system. If no take-over effect is triggered, the spoofing parameters continue to be adjusted according to the fuzzy inference model.

4.3 Feedback Factor Calculation Model Based on Lateral Position Error

Spoofing attacks aim to deviate the AV from its intended trajectory. Lateral position accuracy is more critical than longitudinal or vertical positions. Incorrect lateral position information may cause the AV to depart from its trajectory, potentially leading to collisions with surrounding objects. Therefore, spoofing sources tend to be more concerned about the lateral deviation relative to the intended trajectory of the AV.

Based on research from Chapters 2 and 3, assume the spoofing source monitors the AV's real position and generates corresponding spoofing signals. Firstly, given a lateral spoofing position error value $\Delta\tilde{d}^{bx}$ along the predetermined trajectory of the AV, the 3D coordinates of the spoofing error value in the b-frame can be expressed as:

$$\Delta\tilde{\mathbf{d}}^{b} = \begin{bmatrix} \Delta\tilde{d}^{bx} & 0 & 0 \end{bmatrix}^{T} \tag{4.1}$$

where $\Delta\tilde{d}^{bx}$ denotes the value of lateral spoofing position error set by the spoofing source, and since the AV navigation model is generally established in the n-frame, the position error

in the b-frame is converted to the n-frame:

$$\Delta\tilde{\mathbf{d}}^n = \mathbf{C}_b^n \Delta\tilde{\mathbf{d}}^b \tag{4.2}$$

where $\mathbf{C}_b^n$ denotes the rotation matrix from the b-frame to the n-frame, and since the AV drives on the ground, the rotation matrix between the n-frame and the b-frame is established through the azimuth. Therefore:

$$\Delta\tilde{\mathbf{d}}^n = \begin{bmatrix} \Delta\tilde{d}^{bx}\cos\gamma & \Delta\tilde{d}^{bx}\sin\gamma & 0 \end{bmatrix}^T \tag{4.3}$$

where γ denotes the azimuth of the b-frame relative to the n-frame. In this chapter, the spoofing parameter $\Delta\tilde{\mathbf{p}}_j^{spo}$ in latitude and longitude is expressed in terms of the n-frame, which can be expressed as:

$$\Delta\tilde{\mathbf{p}}_j^{spo} = \begin{bmatrix} \Delta\tilde{L}_k & \Delta\tilde{\lambda}_k & 0 \end{bmatrix}^T \tag{4.4}$$

where $\Delta\tilde{L}_k$ denotes the value of longitude spoofing error; $\Delta\tilde{\lambda}_k$ denotes the value of latitude spoofing error, which can be expressed as respectively:

$$\begin{cases} \Delta\tilde{L}_k = \frac{\Delta\tilde{d}^{bx}\cos\gamma}{R_e\cos\lambda} \\ \Delta\tilde{\lambda}_k = \frac{\Delta\tilde{d}^{bx}\sin\gamma}{R_e} \end{cases} \tag{4.5}$$

where R_e denotes the radius of the earth; λ denotes the latitude information of the AV.

Thus, the spoofing parameter $\Delta\tilde{\mathbf{p}}_j^{spo}$ can be expressed as:

$$\Delta\tilde{\mathbf{p}}_j^{spo} = \begin{bmatrix} \frac{\Delta\tilde{d}^{bx}\cos\gamma}{R_e\cos\lambda} & \frac{\Delta\tilde{d}^{bx}\sin\gamma}{R_e} & 0 \end{bmatrix}^T \tag{4.6}$$

Therefore, the spoofing parameter $\Delta\tilde{\mathbf{p}}_j^{spo}$ can be changed by setting the spoofing position error increment $\Delta\tilde{d}^{bx}$, and then generating the spoofing signal corresponding to the target spoofing position. Typically, the spoofing source must complete the entire spoofing process within a limited time. As the number of spoofing instances increases, the spoofing parameters change. The spoofing source performs multiple spoofing actions with varying parameters:

$$\left\{\Delta\tilde{\mathbf{p}}_1^{spo}, \Delta\tilde{\mathbf{p}}_2^{spo}, \cdots, \Delta\tilde{\mathbf{p}}_j^{spo}\right\}, j \leq k_{\max}^{spo} \tag{4.7}$$

where $k_{\max}^{spo}$ denotes the maximum number of spoofing attacks.

Assuming the target AV receives the spoofing signal, its internal receiver's position solution aligns with the spoofing source's intended target position. During one spoofing cycle, the GNSS measurement value of the target AV (the spoofing source's position information) can be expressed as:

$$\tilde{\mathbf{p}}_1^G = \mathbf{p}_1^G + \Delta\tilde{\mathbf{p}}_j^{spo}, j = 1,2,\cdots and \quad j \leq k_{\max}^{spo} \tag{4.8}$$

where $\mathbf{p}_1^G$ denotes the true position information of the spoofed AV when it is not spoofed.

According to Chapter 3, the impact of measurement sensors on the error transfer model must be considered in practical MSF systems. Typically, GNSS updates less frequently than LiDAR and SINS. LiDAR can partially correct spoofed position errors. The final position error of the MSF system after one GNSS update cycle can be expressed as [86]:

$$\Delta\tilde{\mathbf{X}}_{k_{IG}+1} \approx \left(\mathbf{I}_1 + k_{LG}\cdot\mathbf{H}_2^T\mathbf{R}_L^{-1}\mathbf{H}_2\right)^{-1}\mathbf{H}_1^T\mathbf{R}_G^{-1}\Delta\tilde{\mathbf{p}}_1^{spo} \tag{4.9}$$

Similarly, according to the calculation process of $\Delta\widehat{d}_1^{bx}$. The projection in the lateral direction $\Delta\widehat{\mathbf{d}}_{k_{IG}+1}^{b}$ of the AV is represented as:

$$\Delta\widehat{d}_{k_{IG}+1}^{bx} = \Delta\widehat{L}_{k_{IG}+1}\cdot R_e\cos\lambda_{k_{IG}+1}\cos\gamma_{k_{IG}+1} + \Delta\widehat{\lambda}_{k_{IG}+1}\cdot R_e\sin\gamma_{k_{IG}+1} \tag{4.10}$$

where k_{IG} denotes the update frequency ratio of LiDAR to GNSS, $\Delta\widehat{L}_{k_{IG}+1}and\Delta\widehat{\lambda}_{k_{IG}+1}$ denotes the longitude error and latitude error at the current moment, respectively. $\lambda_{k_{IG}+1}$ denotes the latitude of the AV at this moment, $\gamma_{k_{IG}+1}$ denotes the azimuth of the AV relative to the n-frame.

4.4 Fuzzy Inference Model of Spoofing Parameters Based on Position Error Feedback Factor

A successful spoofing process is mainly related to the following factors: 1) The ability of the MSF system to suppress the spoofing signal; 2) The ability of the navigation information provided by LiDAR in the MSF system to suppress the position error that has been generated; 3) Whether the defence algorithm of the MSF system will detect the spoofing parameter. However, traditional spoofing schemes, including constant value spoofing, exponential value spoofing, and Fusion-ripper, do not fully consider and analyze these factors

and are unable to adjust the spoofing parameters according to the actual situation dynamically.

A fuzzy inference system can simulate the way the human brain thinks, especially when faced with problems with unclear models or high uncertainty, and it does so by creating fuzzy sets and defining fuzzy rules. These fuzzy sets are divided based on the affiliation function, while fuzzy rules are built based on qualitative knowledge and experience [87]. Then, the system imitates the human brain to perform logical reasoning and ultimately transforms the fuzzy quantities obtained from reasoning into precise results through the defuzzification process [87].

The spoofing source does not fully calculate the specific navigation parameters within the MSF system of the target AV. However, it can be used to calculate its positioning error and evaluate the effect of the spoofing attack based on the position information fed back by the AV. Since the spoofing difficulty is a fuzzy concept, it is difficult to establish a clear analytical model for calculating the optimal spoofing parameters. Therefore, this chapter proposes a covert spoofing method based on a fuzzy inference model, which evaluates the spoofing difficulty in real-time and dynamically adjusts the spoofing parameters.

Firstly, a parameter η is defined, which indicates the rate of change of the lateral spoofing position error of the next epoch relative to the lateral spoofing position error of the previous epoch. By constructing a fuzzy inference model, the mapping relationship between the two position feedback adjustment factors κ_1^{spo}, κ_2^{spo} and the change rate of the spoofing error parameter η is established.

$$\eta = FuzzyIM\left(\kappa_1^{spo}, \kappa_2^{spo}\right) \tag{4.11}$$

where $FuzzyIM(*)$ denotes the fuzzy inference model developed in this chapter.

4.4.1 Establishment of knowledge base based on position error feedback factor

1) Structure of the Fuzzy Inference Model

The overall anti-spoofing attack ability of the GNSS/SINS/LiDAR-based MSF system of AVs is mainly embodied in two aspects during the spoofing process, so the fuzzy inference model is set up to have two inputs: fuzzy input 1: the position error feedback factor κ_1^{spo} embodying the inhibition ability of the MSF system to the spoofing attack, and setting its thesis domain U_1^I is [0, 1]; fuzzy input 2: position error feedback factor κ_2^{spo} that reflects LiDAR's ability to correct the generated position error, and sets its domain U_2^I to

[0,1]. These two position error feedback factors can be computed in real-time by monitoring the position of the target AV. The fuzzy inference model has one output, i.e., the rate of change of the next spoofing parameter relative to the previous spoofing parameter η, to prevent the spoofing parameter from changing too drastically and set its thesis domain U^O to [0.6, 1.4] based on the experience, so the established fuzzy inference model adopts the structure of dual-input and single-output.

2) Fuzzy Sets and Membership Functions for Inputs and Outputs

Since it is impossible to establish a clear mathematical model to evaluate the suppression ability of the MSF system against the spoofing attack, it is empirically set that when κ_1^{spo} is larger than 0.6, it indicates that the MSF system has a weak suppression ability against the spoofing attack; when κ_1^{spo} is larger than 0.2 and smaller than 0.6, it indicates that the MSF system has a medium suppression ability; when κ_1^{spo} is less than 0.2, it indicates that the suppression ability of MSF system on spoofing attack is strong. Similarly, when κ_2^{spo} is larger than 0.6 and smaller than 1, it indicates that LiDAR has a weak suppression ability of position error caused by the spoofing attack; when κ_2^{spo} is larger than 0.2 and smaller than 0.6, it indicates that LiDAR has medium suppression ability of position error caused by spoofing attack; when κ_2^{spo} is less than 0.2, it indicates that LiDAR has a strong ability to suppress the position error caused by spoofing attack. Therefore, we set fuzzy input 1 with three fuzzy subsets: Strong (S), Medium (M), and Weak (W). Similarly, fuzzy input 2 is also set with three fuzzy subsets: Strong (S), Medium (M), and Weak (W).

Each element in a fuzzy set corresponds to an affiliation degree, which is between [0,1], and the closer it is to 1, the more it belongs to the set. The affiliation degree is used to objectively state which class of sets it belongs to, and the affiliation function of the two input variables is set to be Gaussian by the assignment method [88]. The specific form can be expressed as:

$$M^I(u,\sigma,c) = \exp\left(-\frac{(u-c)^2}{2\sigma^2}\right) \tag{4.12}$$

where, u denotes the input variable. The parameter σ determines the width of the curve, and the parameter c determines the center of the curve. According to the Zadeh representation [88], the fuzzy sets can be represented by the affiliation function, and when the domain U is infinite, the fuzzy set M on U is denoted as:

$$M = \int_{u\in U} \frac{M(u)}{u} \tag{4.13}$$

The fuzzy distribution can characterize the degree of convergence of the indicator

values, thereby determining the difficulty in the actual spoofing process. Usually, the fuzzy distribution of affiliation includes skewed large distribution, skewed small distribution, and intermediate distribution, and the fuzzy set M_1^I of the fuzzy input κ_1^{spo} can be expressed as:

$$M_1^I = \int_{u\in[0,0.4]} \frac{1-M(u,0.1,0.4)}{u} + \int_{u\in[0,1]} \frac{M(u,0.1,0.4)}{u} + \int_{u\in[0.4,1]} \frac{1-M(u,0.1,0.4)}{u} \tag{4.14}$$

In Eqs.(4-20) and Eqs.(4-21), $\int$ does not denote integration and $+$ does not denote summation. They have only symbolic meanings and denote a generalization of the relationship between all the elements that make up a fuzzy set and its corresponding degree of affiliation over the domain of the argument. $\frac{M(u)}{u}$ also does not denote a division relation; it means that the degree of affiliation of u to the fuzzy set M is $M(u)$. Setting the fuzzy set M_2^I of the fuzzy input κ_2^{spo} is the same as the fuzzy set M_1^I of the fuzzy input κ_1^{spo}.

Finally, the affiliation function curves for the two fuzzy inputs are shown in Fig. 4.2 .

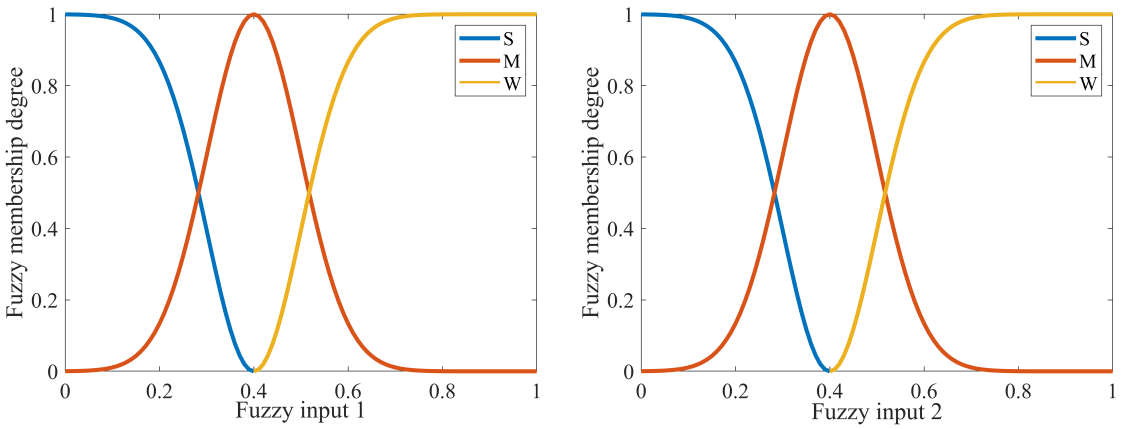

Figure 4.2: Membership curves of fuzzy input $\kappa_1^{spo}$and κ_2^{spo}

The fuzzy output is to set the rate of change η of the parameters of the next spoofing epoch relative to the previous spoofing parameter. To prevent the spoofing parameter from changing drastically, the fuzzy output is set with five fuzzy subsets, Increase (I), Slight Increase (SI), Unchangeable (U), Slight Decrease (SD), Decrease (D), and selecting the triangular affiliation function, which can be expressed as:

$$M^O(v,a,b,c) = \begin{cases} 0, & v \le a \\ \frac{v-a}{b-a}, & a < v \le b \\ \frac{c-v}{c-b}, & b < v \le c \\ 0, & v > c \end{cases} \tag{4.15}$$

where v is the output variable. a, b and c are the parameters of the triangular affiliation function, where $a < b < c$. The fuzzy output η of the fuzzy set M^O can be expressed as:

$$M^O = \int_{v\in[0.6,0.8]} \frac{M^O(v,0.4,0.6,0.8)}{v} + \int_{v\in[0.6,1]} \frac{M^O(v,0.6,0.8,1)}{v} + \int_{v\in[0.8,1.2]} \frac{M^O(v,0.8,1,1.2)}{v} + \int_{v\in[1,1.4]} \frac{M^O(v,1,1.2,1.4)}{v} + \int_{v\in[1.2,1.4]} \frac{M^O(v,1.2,1.4,1.6)}{v} \tag{4.16}$$

Eventually, the affiliation functions of the output variables are shown in Fig. 4.3 .

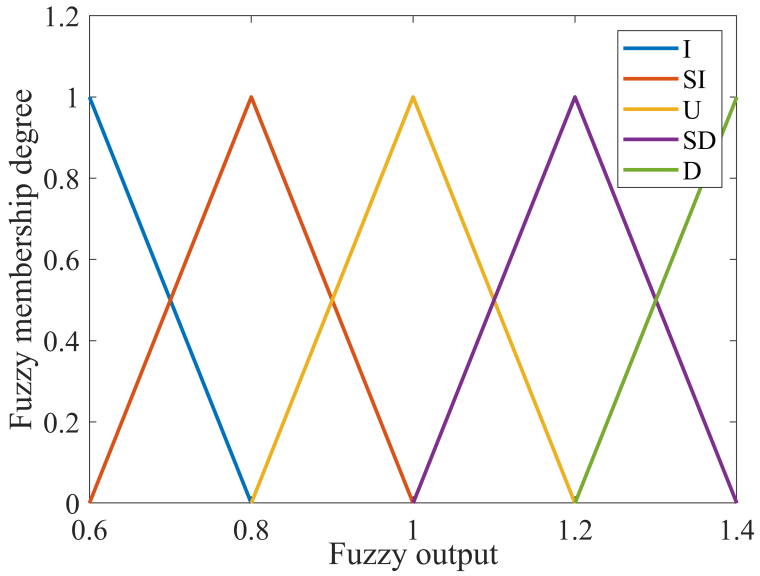

Figure 4.3: Membership curve of fuzzy output η

3) Establishment of linguistic variables based on fuzzy five-tuples

The affiliation functions of the fuzzy input and output variables are determined to build a fuzzy knowledge base based on the linguistic variables. Linguistic variables were proposed initially by Zadeh as a basic concept of fuzzy modeling [89], which refers to variables that are assigned linguistic values rather than numerical values. Referring to Zadeh's defined concept, a linguistic variable L is defined as a five-tuple:

$$L = (X, T(x), U, G, M) \tag{4.17}$$

where X denotes the name of a linguistic variable, $T(x)$ denotes the set of names of linguistic values of a linguistic variable, U denotes the argument domain, G denotes the syntactic rule, and M denotes the semantic rule.

Using fuzzy five-tuples to represent fuzzy inputs κ_1^{spo} and κ_2^{spo} can be expressed as:

$$\begin{cases} L_1^I = \left(X_1^I, T_1^I\left(x_1^I\right), U_1^I, G_1^I, M_1^I\right) \\ L_2^I = \left(X_2^I, T_2^I\left(x_2^I\right), U_2^I, G_2^I, M_2^I\right) \end{cases} \tag{4.18}$$

The linguistic variables of the fuzzy output η can be expressed as:

$$L^O = \left(X^O, T^O\left(x^O\right), U^O, G^O, M^O\right) \tag{4.19}$$

The final fuzzy knowledge base is represented by a five-tuple of three linguistic variables as shown in Fig. 4.4 .

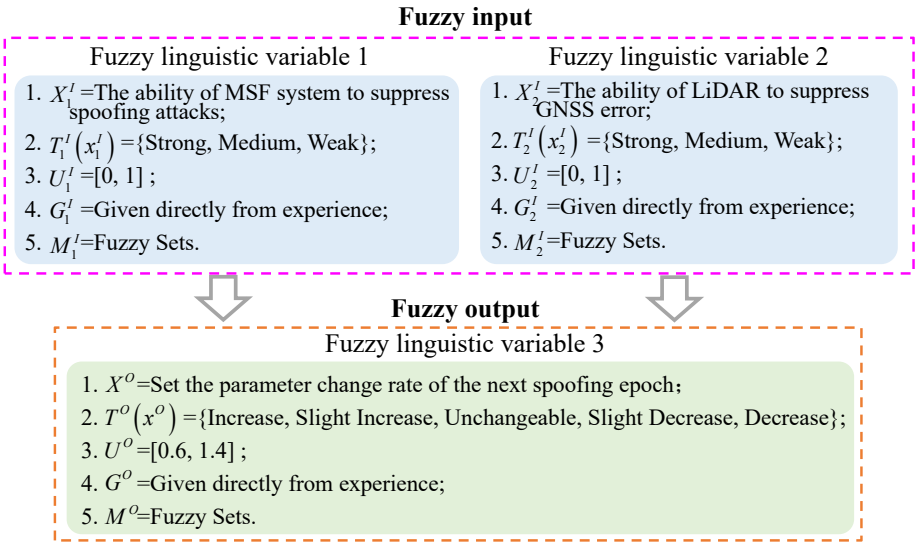

Figure 4.4: The five-tuple of three language variables

4.4.2 Fuzzy Rules Based on Position Error Feedback Factor

The fuzzy inference model does not rely on the precise mathematical model but on the fuzzy rules transformed from the operational experience and expression knowledge. According to the relationship between system parameters and state error in Chapter 3, the fuzzy relationship between position error after spoofing and spoofing parameters is established. If the MSF system is weak in suppressing the GNSS error, then the spoofing parameter can be increased appropriately so that the spoofing purpose can be realized quickly. If the MSF system has a medium suppression ability for GNSS error, set the spoofing parameter to change slightly. If the MSF system's ability to suppress GNSS errors is strong, it means that it is difficult for the spoofing source to deceive the MSF system at this time.

The spoofing parameter can be appropriately reduced to prevent larger spoofing parameters from being detected.

Similarly, suppose the LiDAR's ability to suppress the position error due to spoofing attack is weak. In that case, the spoofing parameter can be further increased appropriately to enable a quick realization of the spoofing attack. If the LiDAR's ability to suppress the position error is moderate, set the spoofing parameter to change little. Suppose the LiDAR's ability to suppress the position error due to a spoofing attack is strong. In that case, the spoofing parameter can be appropriately reduced to prevent excessive spoofing parameters from being detected.

Based on the above fuzzy relationship, fuzzy rules are established. Any inference rule contains two parts, the premise (also known as the antecedent) and the conclusion (also known as the consequent), and has the form of 'If ....., then .....', or the IF-THEN rule. Fuzzy rules can be expressed by conditional clauses of the following form:

If <fuzzy proposition 1>, Then <fuzzy proposition 2>

Further, nine IF-THEN fuzzy rules are established based on fuzzy relationships:

Rule 1 ($\Re_1$): If (Input 1 is W) and (Input 2 is W), then (Output is I);

Rule 2 ($\Re_2$): If (Input 1 is M) and (Input 2 is W), then (Output is SI);

Rule 3 ($\Re_3$): If (Input 1 is S) and (Input 2 is W), then (Output is U);

Rule 4 ($\Re_4$): If (Input 1 is W) and (Input 2 is M), then (Output is SI);

Rule 5 ($\Re_5$): If (Input 1 is M) and (Input 2 is M), then (Output is U);

Rule 6 ($\Re_6$): If (Input 1 is S) and (Input 2 is M), then (Output is SD);

Rule 7 ($\Re_7$): If (Input 1 is W) and (Input 2 is S), then (Output is U);

Rule 8 ($\Re_8$): If (Input 1 is M) and (Input 2 is S), then (Output is SD);

Rule 9 ($\Re_9$): If (Input 1 is S) and (Input 2 is S), then (Output is D).

To facilitate the description, a fuzzy rule table is established based on the above fuzzy rules, as shown in Tab. 4.1 .

Table 4.1: Fuzzy rule table ((1)-(9) indicates the serial number of the nine rules)

Fuzzy rule		GNSS		
		W	M	S
LiDAR	W	$I^{(1)}$	$SI^{(2)}$	$U^{(3)}$
	M	$SI^{(4)}$	$U^{(5)}$	$SD^{(6)}$
	S	$U^{(7)}$	$SD^{(8)}$	$D^{(9)}$

4.4.3 Spoofing Parameter Fuzzy Inference Based on Multiple Zadeh Method

Multiple multidimensional fuzzy inference modeling based on fuzzy rules, also known as chained fuzzy inference modeling, is widely used in fuzzy control. In multiple multidimensional fuzzy inference, the major premise has multiple cases, and each major premise has multiple sets of antecedents, which are generally represented as:

$$\begin{array}{l} A_{11}, A_{12}, \cdots, A_{1n} \rightarrow B_1 \\ A_{21}, A_{22}, \cdots, A_{2n} \rightarrow B_2 \\ \qquad \vdots \\ A_{t1}, A_{t2}, \cdots, A_{tn} \rightarrow B_t \\ A_1^*, A_2^*, \cdots, A_n^* \\ \hline \qquad\qquad\qquad\qquad B^* \end{array} \tag{4.20}$$

where A_{tn} and B_t are the fuzzy sets on the known major premise domains U^I and U^O, respectively; t denotes the number of fuzzy rules; n denotes the number of antecedent variables; '$\rightarrow$' denotes implication, i.e., connecting two propositions that have a dependence relationship to form a composite proposition; A_n^* denotes the given fuzzy set; B^* denotes the fuzzy set of solution.

The spoofing parameter fuzzy inference model based on the position error feedback factor has two antecedent variables for each fuzzy rule, and there are nine fuzzy rules in total, so the specific form of the multiple multidimensional fuzzy inference model is:

$$\begin{array}{ll} M_{1,W}^I & M_{2,W}^I \rightarrow M_{I^{(1)}}^O \\ M_{1,W}^I & M_{2,M}^I \rightarrow M_{SI^{(2)}}^O \\ & \vdots \\ M_{1,S}^I & M_{2,S}^I \rightarrow M_{D^{(9)}}^O \\ M_{1,*}^I & M_{2,*}^I \\ \hline & \qquad\qquad\qquad M_*^O \end{array} \tag{4.21}$$

Fuzzy inference is performed according to the multiple Zadeh method, which starts with a Cartesian product of each rule and the two antecedents of the given premise, respectively, which can be expressed as:

$$\begin{aligned} &M^I_{1,W} \times M^I_{2,W} = M^I_{I^{(1)}}, \\ &M^I_{1,W} \times M^I_{1,W} = M^I_{SI^{(2)}}, \\ &\qquad \cdots \\ &M^I_{1,S} \times M^I_{2,S} = M^I_{D^{(9)}}, \\ &M^I_{1,*} \times M^I_{2,*} = M^I_* \end{aligned} \tag{4.22}$$

Then, the multiple multidimensional fuzzy inference model (MMFIM) is simplified into a multiple inference model, and the inference result M^O_* is calculated according to the First Infer Then Aggregate (FITA) method. The specific procedure of the FITA method is to process the antecedent M^I_* with each rule separately according to the simple fuzzy rules and then get the intermediate variables $M^O_{I^{(1)}*}, M^O_{SI^{(2)}*}, \cdots, M^O_{D^{(9)}*}$, which can be expressed as:

$$\frac{M^I_{I^{(1)}} \to M^O_{I^{(1)}} \quad M^I_*}{M^O_{I^{(1)}*}}, \qquad \frac{M^I_{SI^{(2)}} \to M^O_{SI^{(2)}} \quad M^I_*}{M^O_{SI^{(2)}*}}, \qquad \frac{M^I_{D^{(9)}} \to M^O_{D^{(9)}} \quad M^I_*}{M^O_{D^{(9)}*}} \tag{4.23}$$

After that, aggregating all the intermediate variables together to get the final inference result M^O_*, which can be expressed as:

$$M^O_{I^{(1)}*} \oplus M^O_{SI^{(2)}*} \oplus \cdots \oplus M^O_{D^{(9)}*} = M^O_{D*} \tag{4.24}$$

where $\oplus$ denotes the aggregation operation.

Then, the area center method is used to denazify the fuzzy set, to determine the center of the area surrounded by the image of the fuzzy set's affiliation function and the horizontal coordinate axes, and then take the horizontal coordinate of this center as the whole fuzzy set output. This method takes into account the affiliation of all the elements in the fuzzy set and their corresponding values. Therefore, it can reflect the characteristics of the fuzzy set. Assuming that the horizontal coordinate of the center of the area enclosed by the affiliation function and the horizontal coordinate axis is v_*, it can be determined according to the following formula:

$$u_* = \frac{\int_{U^O} M^O_*(u)\, u du}{\int_{U^O} M^O_*(u)\, du} \tag{4.25}$$

where U^O denotes the domain; M^O_* denotes the affiliation function of the fuzzy set; and u

denotes the element in the domain. Defuzzification using the area-centered method yields the final output u_*, i.e., the change rate η of the next spoofing parameter relative to the previous spoofed parameter. Based on the fuzzy input and fuzzy output specific affiliation functions, as well as the fuzzy inference model established above, the final mapping surface of fuzzy input and fuzzy output fuzzy relationships based on the fuzzy inference model is shown in Fig. 4.5 :

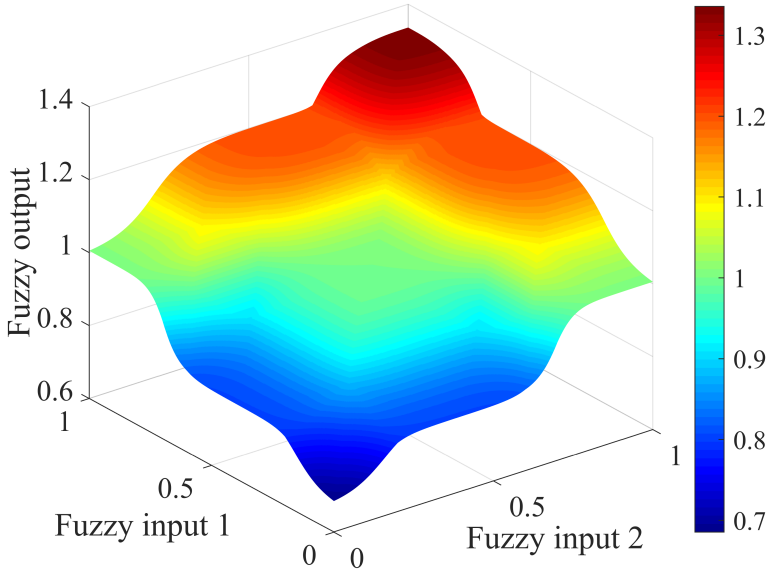

Figure 4.5: Mapping surface of fuzzy relation between fuzzy input κ_1^{spo}, $\kappa_2^{spo}$and fuzzy output η

4.5 Maximum Value Constraint for Spoofing Parameters

According to the results in Section 4.3.1, two error feedback factors κ_1^{spo} and κ_2^{spo} are defined by the position error monitoring results to evaluate the effect of spoofing attack comprehensively. When the spoofing attack starts, the position error of the MSF system will increase, and in the middle of the two spoofing epochs, the presence of the LiDAR signal will lead to a decrease in the position error of the MSF system, i.e., $\Delta\tilde{d}^{bx}_{k_{IG}+1} < \Delta\tilde{d}^{bx}_{1}$. Especially in regions that are easily spoofed, the position error cannot be completely corrected. Therefore, we dynamically adjust the change rate of the spoofing parameter using the fuzzy inference model. As the spoofing parameter gradually increases, the lateral error of the MSF system also increases. At this time, The deviation between the deceived GNSS signal and the LiDAR signal gradually increases.

During a spoofing process, the schematic diagram of the lateral error variation curve of the MSF system is shown as Fig. 4.6 (due to space limitations, some LiDAR signals and the MSF signals are not marked).

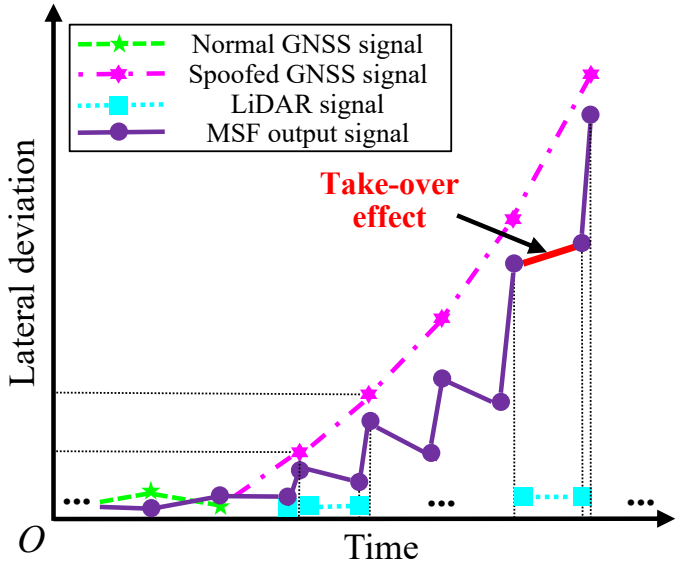

Figure 4.6: Lateral error curve diagram of MSF system during spoofing

As the spoofing attack leads to a gradual increase in the lateral position error of the MSF system, it is monitored $\Delta\widehat{d}^{bx}_{k_{IG}+1} > \Delta\widehat{d}^{bx}_{1}$, which indicates that LiDAR has no suppression ability for the position error caused by spoofing attack, i.e., in a GNSS update cycle, the LiDAR with higher update frequency completely loses its correction ability to the MSF system, and at this time, LiDAR is completely invalid in the MSF system, which indicates that the matching result of LiDAR is completely invalid or has triggered the take-over effect, i.e., the MSF system sees the correct positioning result of LiDAR as a faulty signal, which leads to the MSF system completely trusting the positioning result of GNSS. Therefore, the spoofed GNSS signal completely takes over the MSF system.

In some easier spoofing scenarios, the spoofing success rate is higher theoretically. However, some traditional spoofing methods, such as exponential value spoofing and Fusion-ripper, actually reduce the spoofing success rate. Analyzing the reason, the probability is that the spoofing parameter caused by exponential value spoofing is too large, making it easy to detect. This enables the spoofing source to execute a successful spoofing attack. Still, the defense algorithm in the system detects the spoofing signal, so there is no need to set overly large spoofing parameters in the actual spoofing process. Thus, the method proposed in this chapter establishes a mathematical relationship between the maximum upper

limit of the spoofing parameter and the take-over effect. When the take-over effect is triggered, there is no need to increase the spoofing parameters, which prevents the oversized spoofing parameter from being detected by the MSF system during the subsequent spoofing process, thereby improving the covertness of the spoofing attack. Eventually, the rate of change of the parameters of the next spoofing epoch relative to the previous spoofing parameter η_i can be expressed as:

$$\eta_i = \begin{cases} FuzzyIM\left(\kappa_1^{spo}, \kappa_2^{spo}\right), \widehat{\Delta d}_{k_{IG}+1}^{bx} < \widehat{\Delta d}_1^{bx} \\ 1, \widehat{\Delta d}_{k_{IG}+1}^{bx} > \widehat{\Delta d}_1^{bx} \end{cases} \tag{4.26}$$

where i denotes the spoofing epoch. Finally, the spoofing parameter $\Delta\hat{d}_i$ of the previous epoch and the rate of change of the spoofing error parameter η_i are used to quantify the spoofing parameter of the next epoch.

$$\Delta\hat{d}_{i+1}^{bx} = \eta_i \cdot \Delta\hat{d}_i^{bx} \tag{4.27}$$

4.6 Real-world Data Verification

Due to the limitations of the test conditions, this chapter mainly uses real-world data tests for algorithm verification. In this chapter, the simulation-generated spoofing signals are also used to spoof the MSF system. Then, the spoofing position increment is superimposed on the real GNSS position output when it is not subject to a spoofing attack, simulating the generation of GNSS spoofing signals. Firstly, the test equipment and condition settings required in this chapter are explained. Then, different test conditions are selected to conduct spoofing tests on MSF systems, further verifying the effectiveness of the proposed spoofing method.

4.6.1 Setup

Based on the navigation data platform from the Intelligent Positioning and Navigation Laboratory at the Hong Kong Polytechnic University, positioning and navigation data are collected in urban environments. The data acquisition platform and the navigation sensors associated with this chapter are shown in Fig. 4.7 , which is equipped with LiDAR, GNSS, SINS, and other navigation sensors.

In Fig. 4.7 , the green box denotes the LiDAR installed on the data acquisition vehicle in the experiments, the dark blue box denotes the IMU, and the light blue box represents the

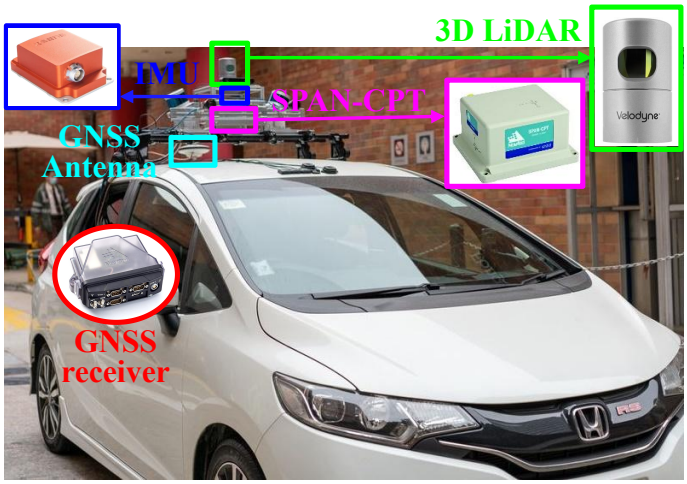

Figure 4.7: Data acquisition platform and associated navigation sensors

GNSS antenna. The red rounded box denotes the GNSS receiver, which is placed inside the vehicle along with the other hardware devices. These sensors comprise the onboard MSF system, which simulates an advanced vehicle navigation system. The pink box denotes the SPAN-CPT system, which is used to provide high-precision navigation references. The specifications of the sensors equipped on the data acquisition platform are shown in Tab. 4.2 .

Table 4.2: Main parameters of relevant sensors and GNSS/SINS system parameters

Sensors	Version	Update frequency	Others
3D LiDAR	HDL32E Velodyne	10Hz	360HFOV, +10∼-30VFOV, 80m
IMU	Xsens Mti10	400Hz	AHRS
GNSS	NovAtel FlexPak6	1Hz	GPS L1/L2, Galileo
GNSS/SINS	NovAtel SPAN-CPT	100Hz	RMSE: 5cm

SPAN-CPT is a high-precision post-processing GNSS/SINS navigation system, which can provide high-precision navigation results and be used to verify the absolute positioning accuracy of the MSF system and the Root Mean Square Error (RMSE) of the SPAN-CPT used in the test is 5 cm. High-precision GNSS positioning results are provided by Real Time Kinematic (RTK). In the experimental data processing, the open-source data processing software RTKLIB is used to obtain the final GNSS positioning results [90]. In RTKLIB, the positioning model is set to 'Kinematic'. The integer ambiguity resolution is set to 'Fix and Hold'. In addition, the distance between the mobile station and the reference station is about 7 kilometers. Similarly, the GNSS/INS/LiDAR-based MSF algorithm is implemented using a software platform developed based on the autonomous platform Autoware

and PSINS C++ toolbox. To fully validate the effectiveness of the fuzzy inference model-based spoofing method, test validation is conducted in various scenarios.

In the constant-value spoofing method, the constant-value spoofing parameter d is set from 0.6 m to 1.5 m, and the sampling interval is 0.1 m. To ensure the consistency of the test conditions, the deception is repeated 5 times for each position so that the number of deceptions carried out at the same position is 50 times.

In the parameter setting of the exponential value spoofing method, the constant value spoofing parameter d is from 0.6m to 1.5m, and the sampling interval is 0.1m. The exponential parameter f is from 1.1 to 1.5, the sampling interval is 0.1, and the number of times spoofing is performed at the same location is 50 times.

Fusion-ripper is divided into two stages of spoofing attack: constant value spoofing, when the lateral error exceeds the threshold, then exponential value spoofing, consistent with the experimental setup in the previous chapter, set the constant value spoofing parameter d to 0.6m to 1.5m, and the interval of the sampling is 0.1m. set the exponential parameter f to 1.1 to 1.5, and the interval of the sampling is 0.1. Therefore, the Fusion-ripper performs deception at the same location 50 times. Then, count the number of times it succeeds in deception.

For the spoofing method proposed in this chapter, the theoretical number of times of deception at each position is 1 time, but to ensure the consistency of the test conditions, the initial value of the spoofing parameter is likewise set to be between 0.6m and 1.5m, with a sampling interval of 0.1m, and the deception is repeated for 5 times at each position. After that, the spoofing parameters are dynamically adjusted according to the fuzzy inference model, so the number of deceptions at the same location is also 50 times. Then, the average success rate of the whole interval is calculated. In the process of practical application, the initial value of the spoofing parameter can be directly set as an empirical value. Afterward, the spoofing parameters are dynamically adjusted according to the fuzzy inference model established in this chapter.

Additionally, during the spoofing process as a whole, since the time for spoofing is limited in the actual process, the time for each deception is set to a fixed 10s. If the spoofing signal is not detected and the lateral deviation of the AV is caused to exceed the safety threshold, it means that the spoofing is successful.

4.6.2 Spoofing Results of Different Spoofing Methods

The first scenario is a typical urban canyon near Tsim Sha Tsui, Hong Kong, where the surrounding environment during data collection features high-rise buildings and numerous dynamic vehicles. In this scenario, the vehicles are mostly obscured by the high-rise buildings on both sides, resulting in poor-quality navigation information output by GNSS, while LiDAR provides high-accuracy navigation results. The sensors configured on the test vehicle are used to collect actual navigation data, and the total duration of this set of data is 380s. The vehicle trajectory and the localization processing results of the relevant sensors are shown in Fig. 4.8. Due to the occlusion of the surrounding buildings or trees, only 170 epochs of RTK GNSS positioning information are solved. Additionally, the scenario features some relatively empty areas (the vehicle travels through these areas from 317s to 333s). There are no high-rise buildings on both sides, at which time the quality of the navigation signal output from GNSS is higher, and the positioning error reaches the decimetre level, as shown in Fig. 4.9. The positioning reference is provided by the high-precision GNSS/INS navigation system SPAN-CPT. It is easier to succeed in this scenario; therefore, this chapter primarily simulates the spoofing process using real-world data within this interval to verify the effectiveness of the proposed method.

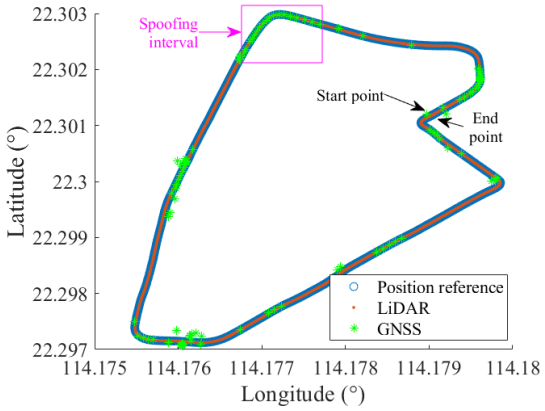

Figure 4.8: Location results of vehicle trajectory and related sensors

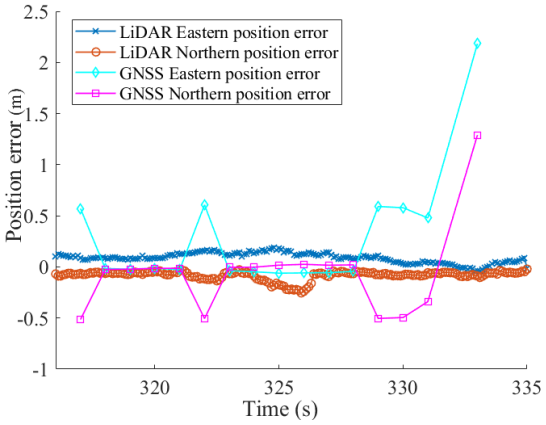

Figure 4.9: Positioning errors of LiDAR and GNSS

To facilitate the experimental verification, the simulation spoofing interval is set from 317s to 326s (in this interval, the vehicle passes through the upper left the empty area in Fig. 4.8. Theoretically, the spoofing success rate is higher in these scenarios), and the spoofing attack is performed every 1s, so a spoofing attack is performed on 10 positions in this scenario. Two conditions are satisfied by successful spoofing: 1) when the maximum lateral deviation of the AV exceeds 2.86m (refer to Chapter 2 for the calculation rules of the threshold); 2) the spoofing signals are not detected by the defence algorithm inside the

MSF system. The traditional constant value spoofing method, exponential value spoofing method, and Fusion-ripper are compared with the covert spoofing method proposed in this chapter. The final metric for comparison is the success rate of implementing spoofing attacks on the MSF system of AVs, which can be expressed as:

$$r_s = \frac{n_s}{N} \times 100\% \tag{4.28}$$

where n_s denotes the number of successful spoofs; N denotes the total number of spoofs performed by the spoofing source.

The statistics of successful spoofing counts for the traditional method and the fuzzy inference model-based covert spoofing method proposed in this chapter are shown in Fig. 4.10 .

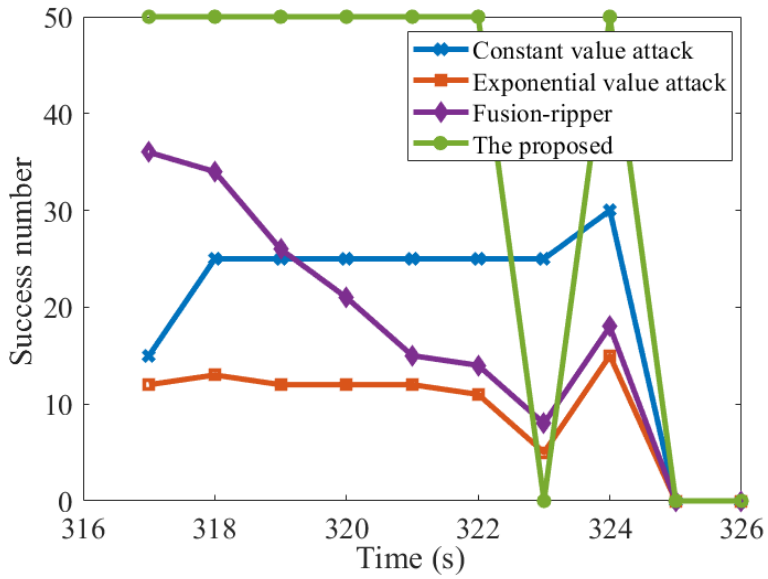

Figure 4.10: The success number of different spoofing methods

The total number of spoofing times for each method is 500, and all successful spoofing times within the spoofing interval are counted. The corresponding success rate is calculated according to Eq. (4.28), and the final result is shown in Tab. 4.3 .

Table 4.3: Average success rate of different spoofing methods (%)

Statistical results	Constant value spoofing	Exponential value spoofing	Fusion-ripper	**The proposed**
n_s	195	92	172	350
r_s	39.0	18.4	34.4	70.0

By analyzing the experimental results, it is evident that the traditional exponential value spoofing method has a low success rate, primarily due to the fact that the spoofing parameters change significantly during the exponential value spoofing process, resulting in spoofing signals that are easily detected. The traditional constant value spoofing method also has a lower success rate of 39%, mainly because the spoofing parameters are constant in size, and the lateral deviation of the autonomous vehicle cannot be made to exceed the threshold value within a limited period. The Fusion-ripper spoofing method encompasses both the exponential value spoofing process and the constant value spoofing process, resulting in a success rate that falls between the other two methods. The covert spoofing method proposed in this chapter achieves a success rate of 70%, which is superior to that of other traditional methods.

Figure 4.11: Data acquisition platform in Google Earth test scenario and test route

Since the spoofing parameters of the proposed spoofing attack algorithm are set according to specific fuzzy rules, they are relatively regular. This inherent regularity might lead to poorer adaptability to sudden environmental changes, resulting in scenarios where the success rate is either very high or very low at a given moment. This outcome is primarily related to the nature of the proposed algorithm itself and the quality of the sensors. In contrast, the other compared algorithms exhibit weaker regularity and higher randomness in their parameter settings. Therefore, at some specific moments (such as 323s), these traditional methods might coincidentally succeed under certain parameter configurations. This explains why in Fig. 4-9, for the spoofing attack starting at 323s, the success number of the proposed method is 0, while the traditional methods have success counts larger than 0.

The second scenario is chosen to collect real-world data under sunny weather conditions in the suburbs of Kowloon Tong, Hong Kong, with a total duration of 200s. The test scenario and route trajectory are shown in Fig. 4.11 . During the test, the vehicle starts from a standstill at the starting point and travels clockwise in a predetermined route to reach the endpoint, where it stops. The black circular trajectory in the figure is the chosen test route, with the start and end points in close proximity. The buildings on both sides of the road are not high but relatively dense. Therefore, the signal quality of GNSS and LiDAR is highly credible in clear weather. The positioning results and position errors of the relevant sensors are shown in Fig. 4.12 and Fig. 4.13 , and the positioning reference values are provided by the high-precision GNSS/SINS navigation system SPAN-CPT.

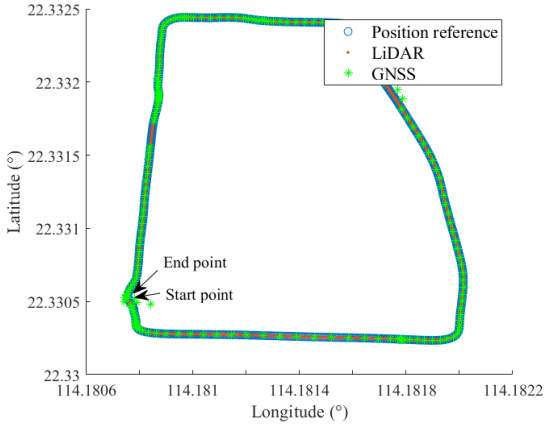

Figure 4.12: Location results of vehicle trajectory and related sensors

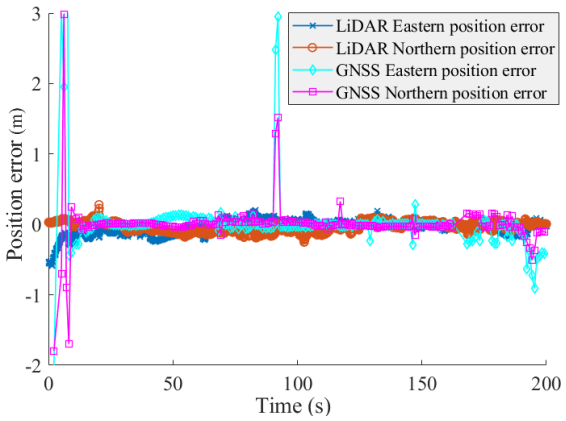

Figure 4.13: Positioning errors of LiDAR and GNSS

According to the theoretical analysis above, the spoofing success rate on the MSF system in this scenario is not high. To improve the success rate of spoofing attacks, we generate LiDAR data [91] with different meteorological weather types and meteorological weather classes based on real-world LiDAR data simulation through a weather simulator, which reduces the quality of the LiDAR signal. To effectively verify the effectiveness of the proposed method in this chapter, navigation data under different weather scenarios are selected for the experiment. Firstly, the spoofing attack is carried out at the position of 5s. Another spoofing attack is carried out every 5s, and the last spoofing is carried out at the position of 190s, so there are a total of 38 positions spoofed and jammed for this set of data, and thus the total number of spoofing for each method is 1900 times. There are differences in the number of spoofing attacks performed for each location depending on the spoofing method. In addition, the time for each spoofing is 10s. Based on the data collected under clear weather, the LiDAR data under severe weather are simulated according to different meteorological weather classes. The different parameter sizes of rainfall rate, visibility,

and snowfall rate are used to indicate the degree of severity of the three types of weather, namely, rain, snow, and fog, respectively. Finally, a spoofing attack is carried out under the same conditions to illustrate the effects of different spoofing methods quantitatively.

For rainy weather scenarios with varying rainfall rates, the success rates of spoofing attacks are compared. Taking the rainfall rate of 60m m/h on a rainy day as an example, the statistics of the number of successful spoofing times of different spoofing methods are shown in Fig. 4.14 . Spoofing attack is performed at different rainfall rates, and the average

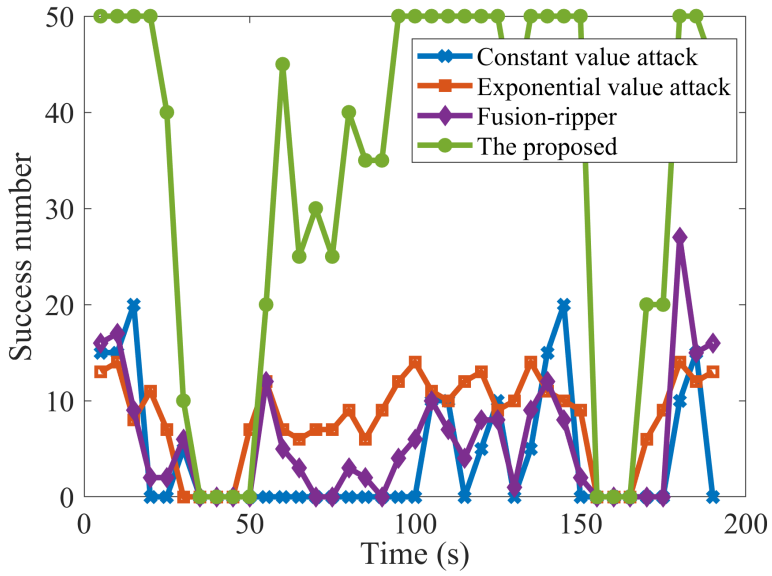

Figure 4.14: Success number of different spoofing methods in rainy weather scenario (%)

success rate of all spoofing methods is calculated according to Eq. (4.28). The final statistics are shown in Tab. 4.4 .

Table 4.4: Average success rate of each spoofing method under different rainfall rates (%)

Meteorological parameter	Constant value spoofing	Exponential value spoofing	Fusion-ripper	**The proposed**
100mm/h	12.37	17.32	13.47	71.32
80mm/h	10.26	17.11	13.11	69.47
60mm/h	8.16	16.42	11.26	67.37
40mm/h	3.42	14.00	7.16	58.95

From the results, it can be seen that the spoofing success rates of all methods show a trend of gradual increase with the worsening of the weather and meteorological conditions, indicating that the more severe the weather environment, the greater the influence on LiDAR is. The easier it is to realize a successful spoofing attack. However, for different spoofing methods, there are obvious differences in spoofing success rates. The success rate

of the constant value spoofing method is lower, and the exponential spoofing method, due to its excessive spoofing parameters, generates spoofing signals that are easily detected by the system, directly affecting the spoofing success rate and resulting in a success rate of no more than 20%. The success rate of traditional Fusion-ripper is in between the two. The covert spoofing method proposed in this chapter is more successful.

For foggy weather conditions, different fog visibility scenarios are selected for spoofing attacks on MSF systems, and the success rates of all spoofing methods are compared. Taking the visibility of 37.5m on a foggy day as an example, the statistics of successful spoofing counts of the constant value spoofing method, the exponential value spoofing method, the Fusion-ripper as well as the hidden spoofing method based on the fuzzy inference model proposed in this chapter are shown in Fig. 4.15 .

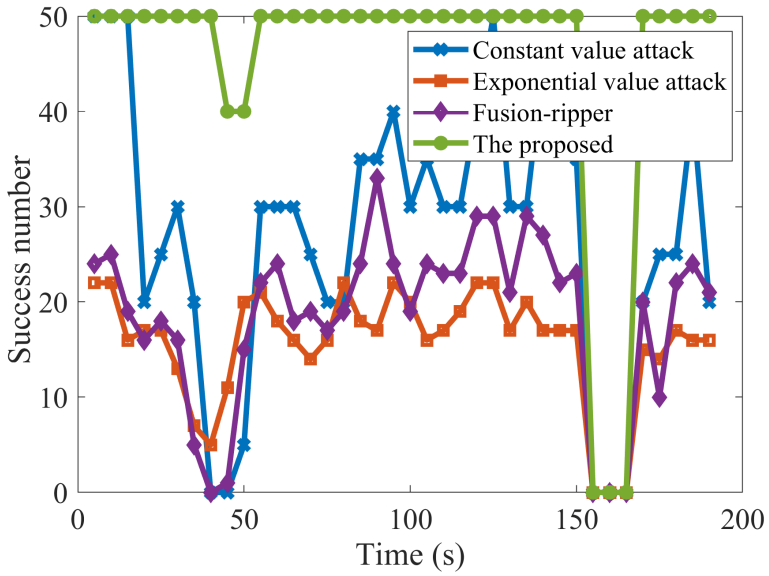

Figure 4.15: Success number of different spoofing methods in foggy scenarios (%)

The spoofing attack is performed under different fog visibility conditions, and similarly, the average success rate of all spoofing methods is calculated according to Eq. (4.28). The final statistics are shown in Tab. 4.5 .

From the results, it is evident that the spoofing success rate increases gradually as the fog visibility decreases. For different spoofing methods, the spoofing results have large differences. But comprehensively, the spoofing method proposed in this chapter still has the highest spoofing accuracy.

For snowy weather scenarios with varying snowfall rates, spoofing attacks are selected, and the success rates of all spoofing methods are compared. Taking the snowfall

Table 4.5: Average success rates of spoofing methods under different fog visibility (%)

Meteorological parameter	Constant value spoofing	Exponential value spoofing	Fusion-ripper	**The proposed**
15m	86.32	46.53	60.26	92.11
20m	80.00	42.26	54.53	92.11
37.5m	55.00	31.37	37.11	91.05
60m	10.00	18.68	14.53	73.95
75m	3.16	15.47	9.58	67.89

rate of 15 mm/h on snowy days as an example, the statistics of successful spoofing counts of the constant value spoofing method, the exponential spoofing method, the Fusion-ripper, and the fuzzy inference model-based covert spoofing method proposed in this chapter are shown in Fig. 4.16 .

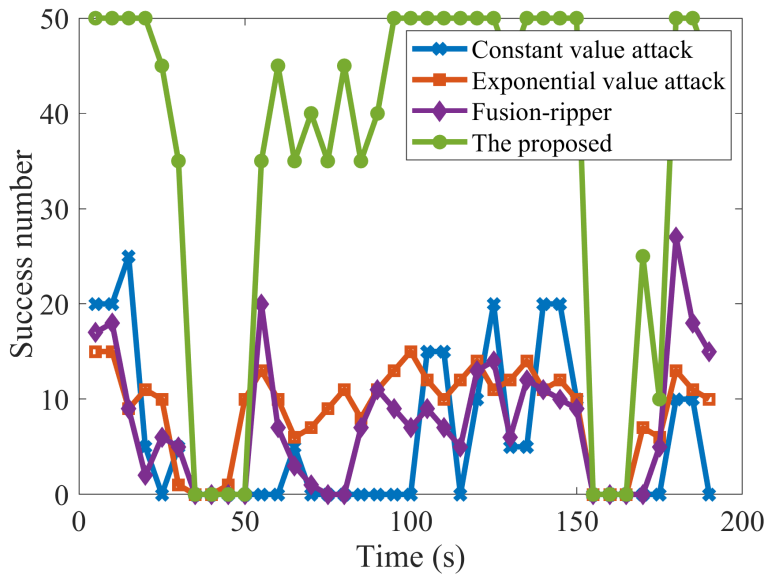

Figure 4.16: Success number of different spoofing methods in snowy scenarios (%)

A spoofing attack is performed at different snowfall rates, and similarly, the average success rate of all spoofing methods is calculated according to Eq. (4.28). The final statistics are shown in Tab. 4.6 .

From the final results, the covert spoofing algorithm proposed in this chapter achieves a high success rate, and the spoofing results are consistent with those obtained on rainy and foggy days. Therefore, the validity of the proposed method is verified, and it can be effectively used for the spoofing process in practical applications.

Overall, for the three different types and intensities of weather scenarios, the fuzzy inference model-based spoofing method proposed in this chapter significantly outperforms

Table 4.6: Average success rate of each spoofing method under different snowfall rates (%)

Meteorological parameter	Constant value spoofing	Exponential value spoofing	Fusion-ripper	**The proposed**
25mm/h	22.11	21.05	19.89	77.37
20mm/h	17.63	19.21	17.37	75.00
15mm/h	11.58	17.89	14.89	71.84
10mm/h	6.05	14.79	9.68	62.89
5mm/h	3.42	13.68	8.26	62.37

the other three traditional spoofing methods in terms of success rate. The spoofing success rate remains above 58% even in some weather scenarios with low weather levels. To summarize, under all typical test scenarios, the proposed method has the least increase in spoofing success rate compared to the traditional method when the fog visibility is 15m (86.32% for the constant value spoofing method to 92.11% for the proposed method), so the proposed method has an increase in spoofing success rate of more than 5% compared to the traditional method.

In conclusion, this chapter has presented a covert spoofing method that advances the state-of-the-art beyond the traditional Fusion-ripper algorithm of Chapter 2. By integrating a fuzzy inference model for dynamic parameter adjustment, the proposed method overcomes the limitations of static spoofing parameters, leading to a more robust and adaptive spoofing attack framework that provides a more powerful theoretical basis for the research on security defense of multi-sensor fusion systems in AVs.

4.7 Summary

This chapter proposes a covert spoofing method based on a fuzzy inference model to realize the dynamic adjustment of spoofing parameters. Firstly, a fuzzy knowledge base and a rule base are constructed based on the position error feedback factor, a multiple multidimensional fuzzy inference model is established based on the fuzzy rule base, and fuzzy implication and aggregation are performed through the multiple Zadeh method inference model to evaluate the spoofing effect in real-time and dynamically set the spoofing parameter of the next spoofing epoch, to realize the adaptive adjustment of spoofing parameters. Meanwhile, the maximum value of the spoofing parameter is limited when the MSF system triggers the take-over effect. Finally, data simulation tests are conducted to verify the effectiveness of the covert spoofing method based on the fuzzy inference model proposed in this chapter. To facilitate validation, this chapter primarily selects test data collected under the

scenario of easy deception for validation. The simulation test results show that, compared with several traditional spoofing methods, the spoofing signals sent by the proposed method are not easily detected by the MSF system in the target AV, and high spoofing success rates can be achieved in the typical scenarios.

Chapter 5

SPOOFING EFFECTIVENESS ASSESSMENT METHOD BASED ON SENSOR UNCERTAINTY ESTIMATION

5.1 Introduction

Autonomous vehicles (AVs) inevitably operate in diverse and dynamic environments, including challenging geographic terrains and varying weather conditions. The susceptibility to spoofing attacks varies across these scenarios. Blindly broadcasting spoofing signals can lead to low success rates and threaten the safety of the spoofing source. To address this, this chapter develops a method for assessing spoofing effectiveness based on sensor uncertainty estimation, focusing on the MSF system of AVs. The sensors involved include GNSS, SINS, and LiDAR. Since SINS operates autonomously with navigation accuracy independent of external conditions, its uncertainty cannot be effectively estimated by the spoofer. Thus, this chapter excludes SINS uncertainty from the spoofing effectiveness evaluation. However, the signal quality of GNSS and LiDAR in MSF systems varies dynamically across different scenarios, significantly impacting the success rates of spoofing. Therefore, accurately estimating sensor uncertainties is crucial for evaluating the effectiveness of spoofing in various environments.

Traditional spoofing techniques often overlook the influence of varying scenarios on spoofing success rates. Constant spoofing in challenging environments not only prolongs attack duration and reduces overall success rates but also diminishes the covert nature of the attack, increasing the risk of signal detection and threatening spoofer security. Given that GNSS and LiDAR uncertainties directly impact spoofing success rates, accurately estimating these uncertainties is crucial for assessing the spoofing effectiveness. However,

spoofers typically lack access to internal vehicle data and must rely on external environmental information for estimation. This creates technical challenges in ensuring the estimated uncertainties align with actual sensor uncertainties.

To resolve these issues, this chapter proposes a spoofing effectiveness assessment method based on sensor uncertainty estimation. First, a sky visibility estimation algorithm leveraging 3D Map Assistance (3DMA) is developed. Using building model vertex coordinates, a spatial geometric model is constructed to calculate the maximum occlusion elevation angles in all directions of the target area. This improves sky visibility estimation accuracy and enables estimation of the number of visible satellites in the target region by combining ephemeris information. Subsequently, a Kernel Partial Least Squares Regression (KPLSR) model is established to align estimation results with actual GNSS uncertainties. For LiDAR, an impulse response function under severe weather conditions is constructed, and a LiDAR uncertainty estimation method based on B-spline regression is developed. Finally, based on GNSS and LiDAR uncertainty estimation results, scenarios are categorized as 'easy' or 'difficult' for spoofing, providing a theoretical foundation for efficient MSF system spoofing attacks.

5.2 GNSS Signal Evaluation Method Based on 3DMA

In urban environments, satellite signals may not reach GNSS receivers in AVs directly due to building obstructions but instead arrive via reflection or scattering, forming multipath signals. These multipath signals can alter the phase and amplitude of received signals, affecting GNSS positioning accuracy. To evaluate GNSS performance across various geographic scenarios, this chapter utilizes 3D building models to calculate sky visibility, thereby quantitatively assessing building occlusion and evaluating GNSS signal quality.

Firstly, the spoofing source selects a target area and calibrates all 3D buildings in that area. The vertex coordinates of the building models are used to construct a spatial geometric model, and the maximum occlusion elevation angle of the buildings in each direction is calculated, which in turn estimates the sky visibility at that location. Combined with the ephemeris information, it determines whether the observation satellites are obscured or not and calculates the number of visible satellites. With this information, the degree to which the surrounding buildings occlude the GNSS signals received by the autonomous vehicle in the area is quantitatively assessed. The geometric relationship between the 3D building model and the sky visibility is derived in detail below.

5.2.1 Sky Visibility Mask Estimation Algorithm based on 3D Building Models

A mathematical model of all 3D buildings in the region is established using vertex coordinates. Each 3D building is modeled as a closed geometry with vertical ribs perpendicular to the base, varying in length, with all lower base vertices lying in the same plane. The location and shape of the building can be represented by the set of upper base vertices. Each vertex is defined by 3D coordinates (latitude, longitude, and altitude). The first vertex of the 3D building model in Figure 5.1 can be expressed as:

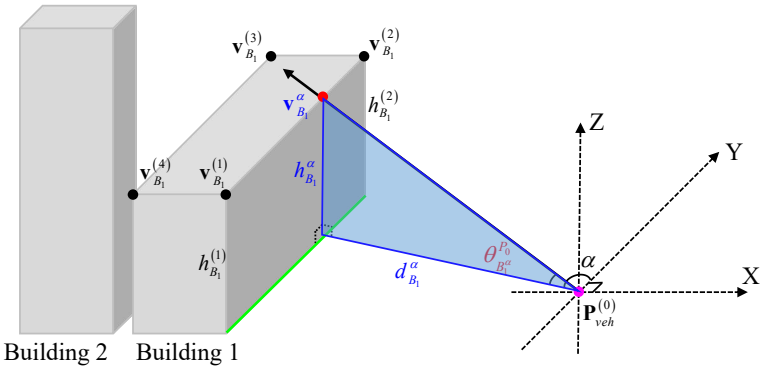

Figure 5.1: 3D building model in the region based on vertex coordinates

Each vertex is represented by a 3D coordinate (latitude, longitude, and altitude), and the first vertex $\mathbf{v}_{B_1}^{(1)}$ of the 3D building model 1 in Fig. 5.1 , for example, can be represented by its 3D coordinates:

$$\mathbf{v}_{B_1}^{(1)} = \begin{pmatrix} Lat_{B_1}^{(1)} & Lon_{B_1}^{(1)} & h_{B_1}^{(1)} \end{pmatrix} \tag{5.1}$$

where $Lat_{B_1}^{(1)}$, $Lon_{B_1}^{(1)}$, and $h_{B_1}^{(1)}$ denote the vertex's latitude, longitude, and altitude, respectively; parentheses in the superscripts are only meant to differentiate them from the indices. Each 3D building model thus consists of at least three vertices, and the ordinal order in this chapter is indicated counterclockwise. Taking Building 1 in Fig. 5.1 as an example, this 3D building model can be uniquely represented by the 3D coordinates of the four vertices $Building^{(1)} = \begin{Bmatrix} v_{B_1}^{(1)} & v_{B_1}^{(2)} & v_{B_1}^{(3)} & v_{B_1}^{(4)} \end{Bmatrix}$. Thus, any 3D building model can be represented by all n vertices of its calibration:

$$Building^{(m)} = \begin{Bmatrix} v_{B_m}^{(1)} & v_{B_m}^{(2)} & \cdots & v_{B_m}^{(n)} \end{Bmatrix}, n \geq 3 \tag{5.2}$$

where m denotes the number of all calibrated 3D building models in the region; n denotes the number of vertices of the 3D building model, whose vertex coordinates can be expressed as

follows: $\mathbf{v}_{B_n}^{(1)} = \begin{pmatrix} Lat_{B_n}^{(1)} & Lon_{B_n}^{(1)} & h_{B_n}^{(1)} \end{pmatrix}^T$. For the other 3D building models around the target vehicle, the buildings around it are numbered starting from the building closest to the location of the target autonomous vehicle, as shown in Fig. 5.1 . All the buildings in the region are calibrated to form a comprehensive collection of 3D building models in the region. Eventually, the 3D building model of the region can be represented as:

$$Building_{Zone}^{3D} = \begin{Bmatrix} Building^{(1)} & Building^{(2)} & \cdots & Building^{(m)} \end{Bmatrix} \tag{5.3}$$

Then, the maximum occlusion elevation angles for all azimuthal building obstructions in the target area are calculated, and the sky visibility mask for any position of the AV (excluding positions inside 3D buildings) is estimated to assess the degree of building occlusion quantitatively.

A sky visibility mask is generated based on the 3D building model. Using the vertex coordinates of the 3D building model, the spatial geometric model of the 3D building and the location of the AV are constructed. Given the limited distance between the target AV and the building, it is assumed that the vehicle's height is 0, placing it in the same plane as the lower base of the 3D building model. The coordinates of the target AV are represented as:

$$\mathbf{P}_{veh}^{(0)} = \begin{pmatrix} Lat_P^{(0)} & Lon_P^{(0)} & 0 \end{pmatrix}^T \tag{5.4}$$

where $Lat_P^{(0)}$ and $Lon_P^{(0)}$ denote the latitude and longitude of the AV's location, respectively. To facilitate the quantitative calculation of the extent of building occlusion at the target vehicle's position, a three-dimensional Cartesian coordinate frame is established with the vehicle's location as the origin, where the eastward direction is defined as the x-axis and the northward direction is defined as the y-axis. The upward direction is the z-axis, as shown in Figure 5.1. The latitude and longitude coordinates in degrees are converted to distance coordinates in meters.

$$\mathbf{v}_{B_n}^{(1)} = \mathbf{C}_e^n \left(\mathbf{v}_{B_n}^{(1)} - \mathbf{P}_{veh}^{(0)} \right) \tag{5.5}$$

where $\mathbf{C}_e^n$ denotes the attitude transformation matrix from the Earth coordinate frame to the geographic coordinate frame.

To calculate the maximum elevation angle due to building obstructions in azimuth, the 3D coordinates of the intersection points are determined. Taking the two vertices on the azimuthal angle of building model 1 as an example, the 3D coordinates of the intersection point $\mathbf{v}_{B_1}^{\alpha}$ need to be calculated, so the 3D coordinates of the vertices $\mathbf{v}_{B_1}^{(1)}$ in the spatial

coordinate frame $\mathbf{P}_{veh}^{(0)} - XYZ$ under the 3D coordinates are denoted as:

$$\begin{cases} x_{B_1}^{(1)} = \left(Lon_{B_1}^{(1)} - Lon_{B_1}^{(0)}\right) R_e \cos\left(Lat_{B_1}^{(1)}\right) \\ y_{B_1}^{(1)} = \left(Lat_{B_1}^{(1)} - Lat_{B_1}^{(0)}\right) R_e \\ z_{B_1}^{(1)} = h_{B_1}^{(1)} \end{cases} \tag{5.6}$$

where R_e denotes the radius of the earth. Similarly, the 3D coordinates of the vertex $\mathbf{v}_{B_2}^{(1)}$ can be expressed as respectively:

$$\begin{cases} x_{B_1}^{(2)} = \left(Lon_{B_1}^{(2)} - Lon_{B_1}^{(0)}\right) R_e \cos\left(Lat_{B_1}^{(2)}\right) \\ y_{B_1}^{(2)} = \left(Lat_{B_1}^{(2)} - Lat_{B_1}^{(0)}\right) R_e \\ z_{B_1}^{(2)} = h_{B_1}^{(2)} \end{cases} \tag{5.7}$$

The coordinates through the vertices $\mathbf{v}_{B_1}^{(1)}$ and $\mathbf{v}_{B_1}^{(1)}$ denote the bottom edge of the combination of the two intersections of the lateral prongs corresponding to these two vertices with the lower bottom face of the geometry. In Fig. 5.1 , the green line represents this bottom edge. Let the equations of the bottom edge of the building corresponding to the vertices $\mathbf{v}_{B_1}^{(1)}$ and $\mathbf{v}_{B_1}^{(1)}$ be:

$$Ax + By + C = 0 \tag{5.8}$$

The parameters for this bottom edge equation are calculated from the vertex coordinates:

$$\begin{cases} A = y_{B_1}^{(2)} - y_{B_1}^{(1)} \\ B = x_{B_1}^{(1)} - x_{B_1}^{(2)} \\ C = y_{B_1}^{(1)} x_{B_1}^{(2)} - y_{B_1}^{(2)} x_{B_1}^{(1)} \end{cases} \tag{5.9}$$

Assuming the azimuth of the target vehicle is α, the maximum elevation angle obscured by the building is calculated. The ray in that direction can be expressed as:

$$\cos\alpha \cdot x + \sin\alpha \cdot y = 0 \tag{5.10}$$

Therefore, the x axis coordinates and y axis coordinates of the intersection of Eq. (5.8) and Eq. (5.10) are:

$$\begin{cases} x = \frac{-C\sin\alpha}{A\sin\alpha + B\cos\alpha} \\ y = \frac{-C\cos\alpha}{A\sin\alpha + B\cos\alpha} \end{cases} \tag{5.11}$$

By further simplification, the horizontal and vertical coordinates of the intersection of the ray on this final azimuth with the 3D building model can be expressed as, respectively:

$$\begin{cases} x_{B_1}^{\alpha} = \dfrac{\sin\alpha\left(y_{B_1}^{(1)}x_{B_1}^{(2)} - y_{B_1}^{(2)}x_{B_1}^{(1)}\right)}{\left(x_{B_1}^{(2)} - x_{B_1}^{(1)}\right)\cos\alpha - \left(y_{B_1}^{(2)} - y_{B_1}^{(1)}\right)\sin\alpha} \\ y_{B_1}^{\alpha} = \dfrac{\cos\alpha\left(y_{B_1}^{(1)}x_{B_1}^{(2)} - y_{B_1}^{(2)}x_{B_1}^{(1)}\right)}{\left(x_{B_1}^{(2)} - x_{B_1}^{(1)}\right)\cos\alpha - \left(y_{B_1}^{(2)} - y_{B_1}^{(1)}\right)\sin\alpha} \end{cases} \tag{5.12}$$

Then, the height coordinates $h_{B_1}^{\alpha}$ of the intersection point are calculated. On the lower bottom surface of the building model, the distance from the coordinate origin to the building is:

$$d_{B_1}^{\alpha} = \sqrt{\left(x_{B_1}^{\alpha}\right)^2 + \left(y_{B_1}^{\alpha}\right)^2} \tag{5.13}$$

Substitute Eq. (5.12) into Eq. (5.13) to get the final target origin to intersection distance:

$$d_{B_1}^{\alpha} = \frac{y_{B_1}^{(1)}x_{B_1}^{(2)} - y_{B_1}^{(2)}x_{B_1}^{(1)}}{\left(x_{B_1}^{(2)} - x_{B_1}^{(1)}\right)\cos\alpha - \left(y_{B_1}^{(2)} - y_{B_1}^{(1)}\right)\sin\alpha} \tag{5.14}$$

After that, the z-axis coordinates of the intersection point, i.e., the height of the intersection point for the ground $h_{B_1}^{\alpha}$, are calculated and can be expressed as:

$$h_{B_1}^{\alpha} = k_{B_1}^{(12)}\left(h_{B_1}^{(2)} - h_{B_1}^{(1)}\right) + h_{B_1}^{(1)} \tag{5.15}$$

where the slope of the line $k_{B_1}^{(12)}$ can be expressed as:

$$k_{B_1}^{(12)} = \frac{\left(x_{B_1}^{\alpha} - x_{B_1}^{(1)}\right)}{\left(x_{B_1}^{(2)} - x_{B_1}^{(1)}\right)} \tag{5.16}$$

Therefore, the height of the intersection $\mathbf{v}_{B_1}^{\alpha}$ corresponding to the maximum elevation angle shaded by the building can be expressed as:

$$h_{\alpha}^{P_0} = \frac{\left(h_{B_1}^{(2)} - h_{B_1}^{(1)}\right)}{\left(x_{B_1}^{(2)} - x_{B_1}^{(1)}\right)}\left[\frac{\sin\alpha\left(y_{B_1}^{(1)}x_{B_1}^{(2)} - y_{B_1}^{(2)}x_{B_1}^{(1)}\right)}{\left(x_{B_1}^{(2)} - x_{B_1}^{(1)}\right)\cos\alpha - \left(y_{B_1}^{(2)} - y_{B_1}^{(1)}\right)\sin\alpha} - x_{B_1}^{(1)}\right] + h_{B_1}^{(1)} \tag{5.17}$$

After a series of simplifications, the height of the final intersection $\mathbf{v}_{B_1}^{\alpha}$ can be expressed as:

$$h_{B_1}^{\alpha} = \frac{h_{B_1}^{(2)}\left(y_{B_1}^{(1)}\sin\alpha - x_{B_1}^{(1)}\cos\alpha\right) - h_{B_1}^{(1)}\left(y_{B_1}^{(2)}\sin\alpha - x_{B_1}^{(2)}\cos\alpha\right)}{\left(x_{B_1}^{(2)} - x_{B_1}^{(1)}\right)\cos\alpha - \left(y_{B_1}^{(2)} - y_{B_1}^{(1)}\right)\sin\alpha} \tag{5.18}$$

Thus, the elevation angle of the intersection point $\mathbf{v}_{B_1}^{\alpha}$ with respect to the origin can be expressed as:

$$\theta_{\alpha}^{P_0} = \tan^{-1}\left(\frac{h_{B_1}^{\alpha}}{d_{B_1}^{\alpha}}\right) \tag{5.19}$$

Substituting $h_{B_1}^{\alpha}$ and $d_{B_1}^{\alpha}$ into Eq. (5.19), and after simplification, the elevation angle of the final intersection $\mathbf{v}_{B_1}^{\alpha}$ can be expressed as:

$$\theta_{\alpha}^{P_0} = \tan^{-1}\left[\frac{h_{B_1}^{(2)}\left(y_{B_1}^{(1)}\sin\alpha - x_{B_1}^{(1)}\cos\alpha\right) - h_{B_1}^{(1)}\left(y_{B_1}^{(2)}\sin\alpha - x_{B_1}^{(2)}\cos\alpha\right)}{y_{B_1}^{(1)}x_{B_1}^{(2)} - y_{B_1}^{(2)}x_{B_1}^{(1)}}\right] \tag{5.20}$$

After obtaining the elevation angle on this edge, the elevation angles of all edges of the building combined on the same azimuth are calculated, and the largest elevation angle is selected as the elevation angle of building 1 obscuring that azimuth.

$$\theta_{\alpha,B^1}^{P_0} = \max\left\{\begin{array}{cccc}\theta_{\alpha^1}^{P_0} & \theta_{\alpha^2}^{P_0} & \cdots & \theta_{\alpha^l}^{P_0}\end{array}\right\} \tag{5.21}$$

where l denotes the total number of eligible edges for the building.

After completing building model 1, similarly, calculate the elevation angle of the 3D building model numbered 2 at the same azimuth as shown in Fig. 5.2 .

Calculate the maximum value of the elevation angle for all building edges as the building occlusion elevation angle at position $\mathbf{P}_{veh}^{(0)}$ and azimuth α:

$$\theta_{\alpha,B}^{P_0} = \max\left\{\begin{array}{cccc}\theta_{\alpha,B^1}^{P_0} & \theta_{\alpha,B^2}^{P_0} & \cdots & \theta_{\alpha,B^n}^{P_0}\end{array}\right\} \tag{5.22}$$

where n denotes the number of eligible buildings. Meanwhile, to reduce the amount of calculation, set $\theta_{\alpha}^{\min} = 1^{\circ}$. When $\theta_{\alpha}^{P_0} < \theta_{\alpha}^{\min}$, the effect of the building is not considered.

Based on the aforementioned elevation calculation model, the elevation angles for all directions at the same location are calculated to quantitatively determine the extent of building occlusion around the target vehicle. To facilitate computation, the azimuth sampling

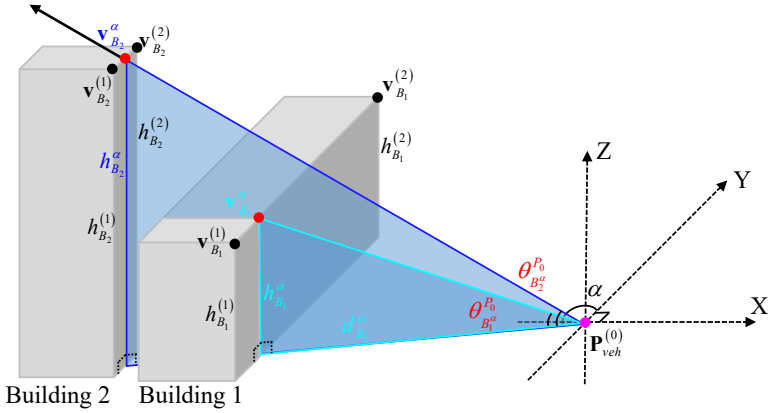

Figure 5.2: Elevation calculation model of different buildings at the same position and azimuth

interval is set to 1° in this chapter. Thus, the azimuth angles at the same target vehicle location are expressed as:

$$\alpha = \left\{ \begin{array}{cccc} 1^\circ & 2^\circ & \cdots & 360^\circ \end{array} \right\} \tag{5.23}$$

The north direction is set as 0°, with positive angles measured counterclockwise. Starting from 1°, there are 360 azimuths at each target position. Based on the modeling, 360 maximum elevation angles corresponding to each azimuth are calculated due to building occlusion. Ultimately, the calculation of all maximum elevation angles for that position can be expressed as:

$$\theta^{P_0}_{\alpha,B} = \left\{ \begin{array}{cccc} \theta^{P_0}_{\alpha,1^\circ} & \theta^{P_0}_{\alpha,2^\circ} & \cdots & \theta^{P_0}_{\alpha,360^\circ} \end{array} \right\} \tag{5.24}$$

Since spoofers cannot control the target AV's location, they need to monitor the entire target area. First, all location information in the target area, except for buildings, is determined, with a resolution of 2 m. Then, for each azimuth of the AV's location, the maximum elevation angle due to building occlusion is calculated using the 3D building model. Finally, based on the 3D building model, a sky visibility mask library is generated for all locations in the target area except buildings, containing the latitude and longitude coordinates of the location and the maximum elevation angle corresponding to building occlusion for 360 azimuths at that location. In this chapter, building models are calibrated on Google Earth, and sky visibility masks are estimated based on the 3D building model, as shown in Fig. 5.3 .

In practical applications, the sky visibility mask corresponding to the position with

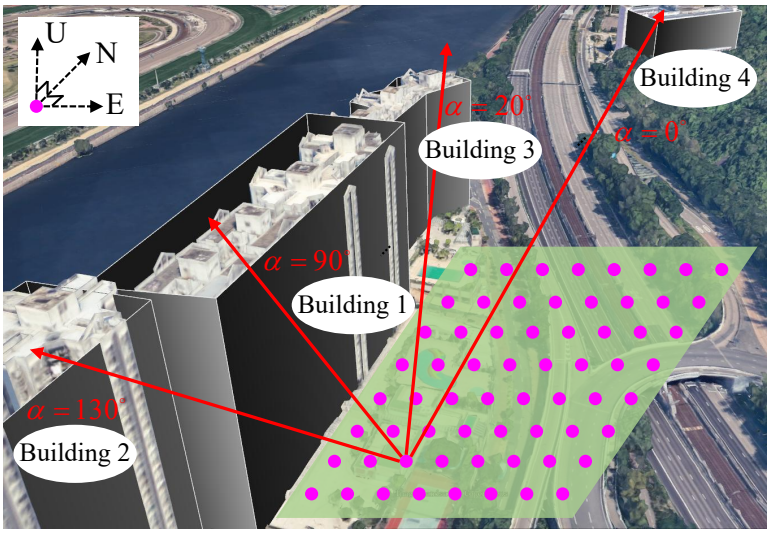

Figure 5.3: Schematic diagram for generating a sky visibility mask in an area based on the 3DMA model

the closest straight-line distance to the actual value of the vehicle is searched for in a pre-established sky visibility library.

5.2.2 Algorithm for Estimating Sky Visibility and Number of Visible Satellites Based on Sky Visibility Masks

Sky visibility assesses the degree to which buildings on both sides of a roadblock satellite signals and is an effective tool for processing GNSS data. Currently, sky visibility masks can be generated using 3D building models, LiDAR, cameras, etc. [92]. Inspired by these prior studies, although spoofers cannot access the target AV's GNSS and LiDAR data, they can generate sky visibility masks from the 3D models of buildings around the vehicle to assess the extent of building occlusion. Fig. 5.4 illustrates the sky visibility masks of two scenarios in Mongkok, Hong Kong: one is a relatively open area (Scenario 1) with few buildings on both sides of the lane; the other is an urban canyon scenario (Scenario 2) with denser buildings on both sides, including some tall structures, commonly found in modern cities.

In the sky visibility mask, the gray background area is the area occluded by buildings, and the white background area is the area not occluded by buildings. In this chapter, the ratio of the occluded area of buildings is used to assess the degree of occlusion, i.e., the ratio of the area of the gray area to the whole area, and this parameter is defined as the sky

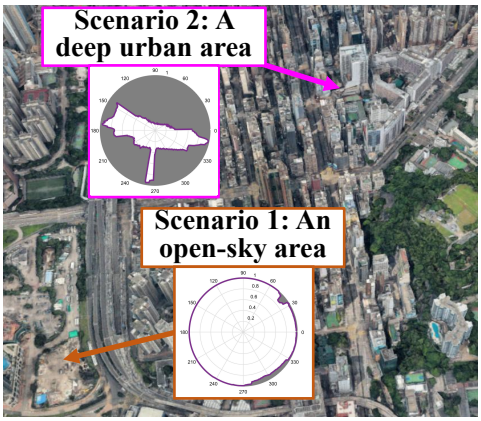

Figure 5.4: Sky visibility for two real scenarios

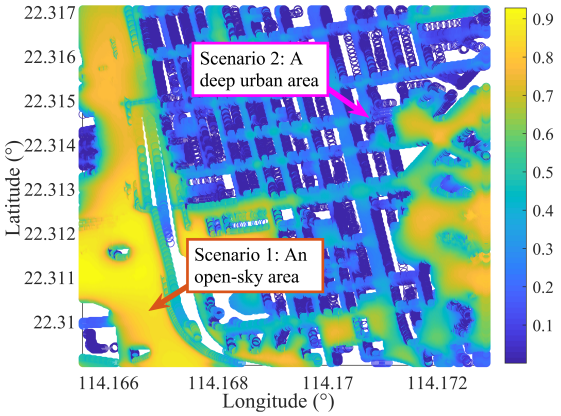

Figure 5.5: Sky visibility map of the entire region

visibility.

$$S_k = S\left(\mathbf{P}_{veh}^{(k)}\right) = \frac{\text{Area}\left(\mathbf{P}_{veh}^{(k)}\right)}{\pi} \tag{5.25}$$

where $S(*)$ denotes the area ratio between the masked area and the whole area, between 0 and 1, which indicates the sky visibility at the vehicle location $\mathbf{P}_{veh}^{(k)}$. $\text{Area}(*)$ denotes the area of the gray shaded area of the sky visibility mask at that location (the radius of the mask is set to 1 for ease of computation) is computed, as shown in Fig. 5.5 . Sky visibility for two real scenarios. Scenario 1 is a relatively empty area, and Scenario 2 is an urban area with denser buildings.

When the ratio is smaller, it indicates a higher degree of building occlusion around the AV, likely surrounded by taller and denser structures. In such environments, GNSS signals may be affected by NLOS and multipath effects, reducing the reliability of positioning information. Conversely, a larger ratio indicates lower building occlusion, with a value of 1 implying no surrounding buildings, i.e., a completely open scenario. Sky visibility is calculated for the entire scenario, yielding the overall sky visibility, as shown in Fig. 5.5 . The yellow areas represent relatively open scenarios, while the blue areas indicate regions heavily shaded by surrounding buildings.

The number of visible satellites for GNSS positioning is influenced by sky visibility. The more satellite signals received by the satellite receiver, the higher the positioning accuracy and the lower the uncertainty. Using the sky visibility mask generated with the assistance of a 3D building model and combining it with satellite ephemeris information, the elevation and azimuth angles of each satellite can be calculated to determine if the satellite signal is blocked by buildings, thereby estimating the number of visible satellites the target AV can receive.

First, the satellite elevation angle is calculated. The satellite elevation angle has a significant impact on satellite signals and is widely used in GNSS positioning to determine the weight of satellite measurement information. A larger elevation angle reduces the likelihood of satellite obstruction by obstacles and minimizes multipath effects. When the elevation angle is small, satellite signals are more prone to being blocked or reflected by buildings and experience greater tropospheric refraction bias due to longer atmospheric propagation paths, potentially leading to more severe atmospheric delays. Satellite elevation angle can be estimated from GNSS measurements:

$$\theta_{El}^{(j)} = \arcsin\left(\frac{u_{sv}^{(j)}}{r^{(j)}}\right) \tag{5.26}$$

where $u_{sv}^{(j)}$ denotes the altitude of the celestial component of the position of the j satellite in the Eastward-Northward-Upward coordinate frame (ENU-frame) for the location of the target autonomous vehicle receiver; and $r^{(j)}$ represents the distance between the satellite and the target AV receiver. Although the exact positions of the satellite and receiver are unknown, their positioning errors are negligible compared to the satellite-receiver distance. Thus, the satellite position estimated from satellite ephemeris and the receiver position estimated from measurements can still provide an acceptable estimation of the elevation angle.

Then, the satellite azimuth angle is calculated. Unlike satellite elevation angle, satellite azimuth angle has an indirect effect on observation quality. Similar to the elevation angle, the satellite azimuth angle can be estimated from the positions of the satellite and target AV receivers. For a specific satellite numbered j:

$$\theta_{Az}^{(j)} = \arctan\left(\frac{e_{sv}^{(j)}}{n_{sv}^{(j)}}\right) \tag{5.27}$$

where $e_{sv}^{(j)}$ and $n_{sv}^{(j)}$ denote the distances of the eastward and northward components of the satellite's position in the ENU-frame for the location of the target autonomous vehicle receiver, respectively.

Its azimuth and elevation angles are $\theta_{Az}^{(j)}$ and $\theta_{El}^{(j)}$, respectively, and find the maximum elevation angle $\theta_{\alpha}^{P_0}$ of the 3D building model on that azimuth $\theta_{Az}^{(j)}$, and then compare its magnitude with the elevation angle $\theta_{El}^{(j)}$. When the elevation angle of the satellite is greater than the maximum elevation angle of the 3D building model, it means that the building has not obstructed the satellite. Conversely, it means that the building has obstructed the

satellite. Taking the sky visibility mask at a certain location as an example, the degree of occlusion of the GPS satellite is shown in Fig. 5.6 .

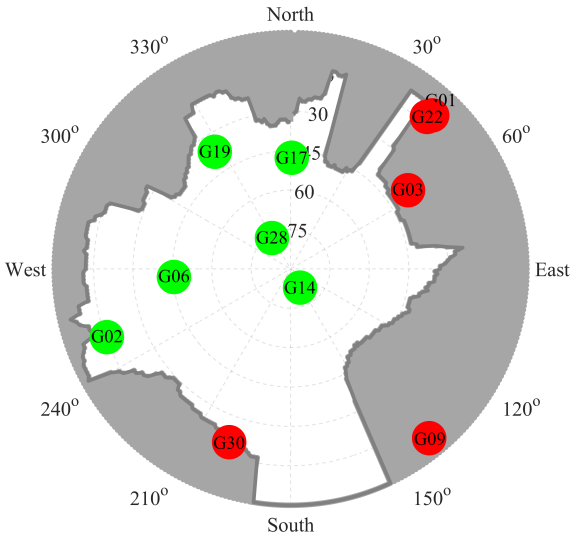

Figure 5.6: Sky visibility mask and occlusion of GPS satellites

The green satellites indicate unobstructed signals, while the red satellites indicate blocked signals. The number of visible satellites the target AV can receive at that location is then calculated. The calculated sky visibility and number of visible satellites are used to quantitatively assess the extent of GNSS signal obstruction by surrounding buildings at any location within the target area.

5.3 LiDAR Signal Evaluation Method Based on Weather Meteorological Classification

Relevant studies indicate that the total loss of GNSS signals during atmospheric propagation is only 2 dB, including all signal loss factors such as rain, clouds, snow, and fog. Compared to other error variables, this loss is minimal, and weather factors like clouds, rain, and snow typically do not significantly affect the positioning accuracy of satellite signal receivers. In contrast, severe weather conditions significantly impact LiDAR signals, substantially reducing the credibility of their positioning information and diminishing their role in MSF systems. Since severe weather has minimal effects on GNSS and SINS signals, this section focuses on evaluating the mechanisms by which severe weather impacts LiDAR signals and the performance of the NDT matching algorithm.

5.3.1 LiDAR Impulse Response Function for Clear Weather Scenarios

Typically, a LiDAR broadcasts a laser signal outward, and when the signal encounters a target, it is reflected. The receiver in the LiDAR then receives the reflected laser signal. After that, the distance between the LiDAR and the target around the autonomous vehicle is calculated based on the speed of light and the time difference between the transmitter and the receiver. Ultimately, a one-frame point cloud is formed for that moment in time based on all the reflected signals received.

Firstly, the LiDAR impulse response function is investigated in clear weather. The impulse response function can convert the distance-dependent received power into the time-dependent transmitted signal power. Therefore, in the presence of scattering particles, it can represent the optical signal strength of the pulse propagation. Typically, the impulse response function under clear weather can be obtained by multiplying the product of the optical channel and the target channel [93].

$$H^C(l) = H_O^C(l) \cdot H_T^C(l) \tag{5.28}$$

where l denotes the detection distance of LiDAR; $H_O^C(l)$ denotes the impulse response function of the optical channel under clear weather; $H_T^C(l)$ denotes the impulse response function of the target channel under clear weather, where $H_O^C(l)$ can be expressed as:

$$H_O^C(l) = \frac{\kappa(l)}{l^2} \tag{5.29}$$

where $\kappa(l)$ denotes the area ratio of the area detected by the LiDAR laser transmitter to the area received by the laser receiver, which can ultimately be expressed as:

$$\kappa(l) = \begin{cases} 0, l \leq l_1 \\ \frac{l-l_1}{l_2-l_1}, l_1 < l < l_1 \\ 1, l \geq l_2 \end{cases} \tag{5.30}$$

where l_1 and l_2 denote parameters related to the optical configuration of the LiDAR transmitter and receiver. The impulse response function H_T^C of the target channel in clear weather can be expressed as:

$$H_T^C(l) = \lambda_0 \delta(l - l_0) \tag{5.31}$$

where l_0 denotes the distance of a stationary target over an optical channel; $\delta(l - l_0)$ denotes the Dirac impact function; and λ_0 denotes the differential reflectance of the target, usually $\lambda_0 \in \left(0, \frac{1}{\pi}\right)$. Therefore, the LiDAR impulse response function in clear weather can

be finally expressed as:

$$H^C(l) = \frac{\kappa(l)}{l^2} \cdot \lambda_0 \delta(l - l_0) \tag{5.32}$$

5.3.2 LiDAR Impulse Response Function for Adverse Weather Scenarios

Compared to clear weather, adverse weather primarily affects LiDAR through attenuation and backscattering effects. Attenuation affects the propagation of laser signals in the air, mainly influencing the optical channel impulse response function. Rain, snow, and fog, composed essentially of water molecules, reflect laser signals. Unlike stationary targets (hard targets) under normal conditions, LiDAR point clouds produce noise in the air (soft targets) under severe weather. The effect of water molecules on LiDAR laser signal transmission under severe weather conditions is illustrated in Fig. 5.7. These water molecules can impact the navigation and localization results of LiDAR in MSF systems for AVs.

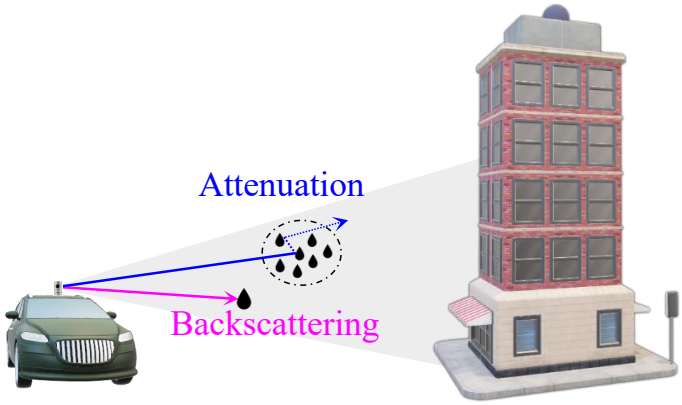

Figure 5.7: Diagram of the effect of water droplets on LiDAR

Assuming that the attenuation coefficient and backscattering coefficient remain constant for the same weather meteorological class during laser signal propagation through the air, the impulse response function under severe weather conditions can be expressed as:

$$H^A(l) = H_O^A(l) \cdot H_T^A(l) \tag{5.33}$$

where $H_O^A(l)$ and $H_T^A(l)$ denote the optical channel impulse response function and the target channel impulse response function under severe weather, respectively. The $H_O^A(l)$ can be expressed as:

$$H_O^A(l) = \frac{\kappa(l)}{l^2} \cdot e^{-\alpha l} \tag{5.34}$$

where α denotes the attenuation coefficient, which is related to the meteorological type and meteorological class of the weather. In addition, $H_T^A(l)$ can be expressed as:

$$H_T^A(l) = H_T^O(l) + \beta \cdot \varepsilon(l_0 - l) \tag{5.35}$$

where $\varepsilon(*)$ denotes the unit step function; l_0 denotes the distance of the hard target on the optical channel; and β denotes the backward scattering coefficient. Therefore, the impulse response function of LiDAR in bad weather can be expressed as:

$$H^A(l) = e^{-\alpha l} \cdot H^C(l) + \frac{\kappa(l)}{l^2} \cdot e^{-\alpha l} \beta \cdot \varepsilon(l_0 - l) \tag{5.36}$$

To visualize the effects of attenuation and backscattering on point clouds, the same frame of point cloud data is used to compare the point clouds before and after fog addition through a simulator, as shown in Fig. 5.8 .

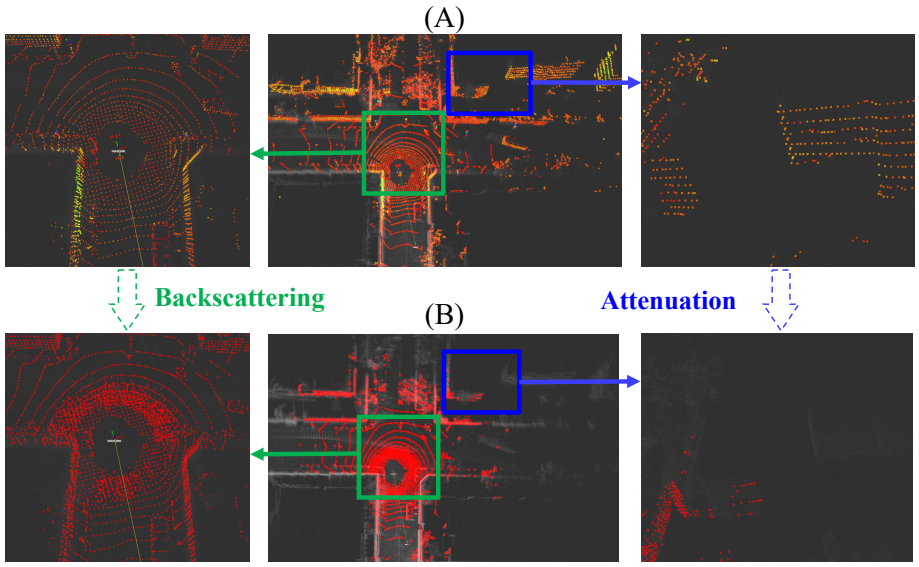

Figure 5.8: Comparison of LiDAR point clouds in clear weather (a) and adverse weather (b)

The changes in LiDAR point clouds indicate that the number of distant point clouds is significantly reduced under fog conditions due to attenuation effects. Fog minimizes the number of effective point clouds. Additionally, increased water molecules in the air lead to backscattering impacts, creating noise areas near the LiDAR that do not exist on the prior map, thereby increasing LiDAR signal noise. Both factors reduce the accuracy and credibility of LiDAR's NDT matching algorithm in adverse weather conditions, thereby affecting the reliability of LiDAR localization results.

5.3.3 LiDAR Signal Evaluation Algorithm Based on Weather Meteorological Classification

Compared with the impulse response function $H^C(l)$ under clear weather, the impulse response function $H^A(l)$ under severe weather conditions is affected by the attenuation coefficient α and the backscattering coefficient β, which are primarily related to the type and intensity of severe weather. This chapter establishes mathematical models for different severe weather conditions (focusing on three common types: rain, snow, and fog) based on weather meteorological classification. The mathematical relationship between weather meteorological classification and the two key parameters in the impulse response function (attenuation coefficient α and backscattering coefficient β) is derived. In the following, three common types of weather (rain, snow, and fog) are modeled and analyzed separately.

1) Rainy day scenarios. Rain is prevalent year-round in some regions, particularly in coastal cities, where heavy rain and storms are common in summer. Rain weather can be categorized into slight, moderate, heavy, and heavy rain based on different rainfall rates. The probability distribution function of rain and corresponding intensity levels can be expressed as [94, 95]:

$$\begin{aligned} &N_R\left(D_R\right) = 8000 \cdot \exp\left(-4.1 \cdot r_{\text{Rain}}^{-0.48} \cdot D_R\right) \\ &\text{st.} \begin{cases} 0 < r_{\text{Rain}} \leq 2, \text{Slight} \\ 2 < r_{\text{Rain}} \leq 10, \text{Moderate} \\ 10 < r_{\text{Rain}} \leq 50, \text{Heavy} \\ r_{\text{Rain}} > 50, \text{Violent} \end{cases} \end{aligned} \tag{5.37}$$

where N_R denotes the rainfall probability distribution; r_{Rain} denotes the rainfall rate; and D_R denotes the raindrop diameter. The attenuation and backscattering coefficients can be calculated based on the rain probability distribution, allowing for the quantitative simulation of LiDAR point clouds under various rain intensities [93, 96].

$$\begin{cases} \alpha = f_\alpha\left(D_R\right) = \frac{\pi}{8} \int_0^\infty D_R^2 Q_A\left(D_R\right) N_R\left(D_R\right) dD_R \\ \beta = f_\beta\left(D_R\right) = \frac{\pi}{8} \int_0^\infty D_R^2 Q_B\left(D_R\right) N\left(D_R\right) dD_R \end{cases} \tag{5.38}$$

where $Q_A\left(D_R\right)$ and $Q_B\left(D_R\right)$ denote the attenuation and backscattering efficiencies of water droplets with a diameter of D_R, respectively. By adding different intensities of rain to LiDAR point clouds collected under clear weather, point clouds under various rainy conditions can be simulated.

2) Foggy day scenarios. Fog commonly occurs in high-latitude regions during autumn

and winter, including major cities such as Beijing, Berlin, and Washington. Fog is typically assessed based on visibility, which refers to the degree of clarity of objects at a certain distance. It can be categorized into light, moderate, heavy, and dense fog [97]. Corresponding to different visibility ranges, the LiDAR attenuation coefficient and backscattering coefficient can be calculated based on visibility [93].

$$\begin{cases} \alpha = \frac{300}{v_{Fog}} \\ \beta = \frac{0.046}{v_{Fog}} \end{cases}, \text{st.} \begin{cases} 200 < v_{Fog} \leq 400, \text{Slight} \\ 100 < v_{Fog} \leq 200, \text{Moderate} \\ 50 < v_{\text{Fog}} \leq 100, \text{Heavy} \\ v_{\text{Fog}} \leq 50, \text{Violent} \end{cases} \tag{5.39}$$

where α denotes the attenuation coefficient; β denotes the backscattering coefficient, and v_{Fog} denotes the visibility of fog. Similarly, point clouds under foggy conditions can be simulated by adding different intensities of fog to the initial point clouds collected under clear weather.

3) Snowy day scenarios. Snow is common in some countries, particularly in high-latitude cities such as Harbin, Washington, and Helsinki. In mid-winter, these cities experience extreme snowy weather annually, often disrupting transportation. Compared with rain and fog, snow has large and irregularly shaped water molecule particles. Still, its density is relatively small, so its effect on LiDAR localization is usually weaker than that of fog. However, its effect on LiDAR is more significant than that of rain at the same meteorological level scale. Snowfall rate is a metric for evaluating the meteorological level of snowfall size, which can usually be categorized as slight, medium, and heavy [98]. Similar to the probability distribution function for rain, the snowfall rate is used to calculate the probability distribution function, which can eventually be expressed as [99]:

$$\begin{gathered} N_S(D_S) = 7600 \cdot r_{\text{Snow}}^{-0.87} \cdot \exp\left(-2.55 \cdot r_{\text{Snow}}^{-0.48} \cdot D_S\right) \\ \text{st.} \begin{cases} 0 < r_{\text{Snow}} \leq 1, \text{Slight} \\ 1 < r_{\text{Snow}} \leq 5, \text{Moderate} \\ r_{\text{Snow}} > 5, \text{Heavy} \end{cases} \end{gathered} \tag{5.40}$$

where N_S denotes the probability distribution of snow; r_{Snow} denotes the snowfall rate of snow; and D_S denotes the diameter of snowflake. Then Eq. (5.40) can be used to calculate the attenuation coefficient and backscattering coefficient under different snowfall rates. Therefore, based on the initial point clouds collected under clear weather, LiDAR point cloud data under different severe weather conditions, including rain, snow, and fog, can be simulated according to weather meteorological classification. The attenuation and

backscattering coefficients for different weather types are used to evaluate LiDAR signals based on weather classification.

5.4 Spoofing Effectiveness Assessment Method Based on GNSS and LiDAR Uncertainty Estimation

Assessing the effectiveness of spoofing on target AVs requires estimating the sensor uncertainty in the MSF system based on observed information about the target vehicle, ensuring that the estimation results closely match the real sensor uncertainty. Spoofing scenarios are then categorized based on the magnitude of sensor uncertainty.

5.4.1 GNSS uncertainty estimation algorithm based on KPLSR

For different geographical scenarios, the degree of occlusion varies, leading to varying signal quality in GNSS. Since the spoofing source cannot access the internal information of the sensors, it can only assess the quality of the navigation information output from the sensors based on limited external information. In addition, the uncertainty of GNSS in the same set of experimental data with the same sky visibility may be different, which is caused by many factors, such as the change of the motion state of the target AV, the change of the surrounding environment, the change of the parameters inside the AV, the random noise, etc., because it is difficult to establish a complete analytical model to derive a clear mathematical relationship between these parameters. This chapter establishes a suitable regression model based on the observation data, thereby establishing the mapping relationship between the observation information and sensor uncertainty and subsequently determining the spoofing difficulty in the area.

Traditional linear and nonlinear regression models have limitations, including poor estimation accuracy, insensitivity to outliers, and incomplete consideration of factors. Based on GNSS uncertainty and monitoring information (sky visibility, number of visible satellites, and speed of the target AV) in complex environments, a GNSS uncertainty estimation method using partial least squares regression with a kernel function is proposed to enhance the accuracy of GNSS uncertainty estimation further.

The basic idea of KPLSR is to first map each point in the original space to the feature space by a nonlinear kernel function and then build a linear PLSR model [100] in the mapped feature space. According to Cover's theorem, the nonlinear data in the original space is most likely to be linear after the high-dimensional nonlinear mapping. KPLSR can efficiently compute the latent variables in the eigenspaces by using integral operators and

nonlinear kernel functions. The main advantage of KPLSR over other nonlinear methods is that it does not involve complex nonlinear optimization. Additionally, since KPLSR can utilize various kernel functions, it can effectively handle a wide range of nonlinear regression problems.

The data are first mapped from the original space to a high-dimensional feature space, and then linear partial least squares regression (PLSR) is performed in the feature space. The data table of independent variables set up in this chapter consists of the sky visibility S_k, and the number of visible satellites N_k estimated by the spoofing source, as well as the monitored speed of the autonomous vehicle V_k, and the dependent variable is the GNSS uncertainty at this point. Thus, the independent variable data table $\mathbf{X}$ and the dependent variable data table $\mathbf{Y}$ are constituted:

$$\begin{cases} \mathbf{X} = \begin{bmatrix} S_k & N_k & V_k \end{bmatrix} \\ \mathbf{Y} = R_k^G \end{cases}, k = 1 \cdots l \tag{5.41}$$

where l denotes the total number of training samples to build the PLSR estimation model. The PLSR measure estimator can be trained to obtain the PLSR measure by utilising the independent variable data table and the dependent variable data table. The purpose of training is to build a nonlinear regression model between the independent variable data table $\mathbf{X}$ and the dependent variable data table $\mathbf{Y}$ by extracting the components. Borrowing from the PLSR theory in multivariate statistical methods, KPLSR transforms the nonlinearly correlated independent variable sample data into a high-dimensional feature space: $\mathbf{X} \to \mathbf{\Phi}(\mathbf{X})$. The independent variable data table $\mathbf{X}$ will get a column vector in the higher dimensional mapping space after nonlinear mapping, if there exists a function $K_{i,j} = K[x(i), x(j)]$ in the primal space that satisfies $K[x(i), x(j)] = \varphi^T(i)\varphi(j)$, $\varphi^T(i)\varphi(j)$ is the inner product of the higher dimensional space, then $K[x(i), x(j)]$ is said to be the kernel function. If every element $\mathbf{K}_0^{\mathbf{\Phi}} = \mathbf{\Phi}_0 \mathbf{\Phi}_0^{\mathrm{T}}$ of the matrix $\varphi^T(i)\varphi(j)$ in the high-dimensional space is represented by a kernel function, then it is said to be a kernel function. Then the matrix $\mathbf{K}^{\mathbf{\Phi}}$ is said to be a kernel function matrix, or kernel matrix for short.

$$\mathbf{K}^{\mathbf{\Phi}} = \begin{bmatrix} \boldsymbol{\varphi}(1)^{\mathrm{T}} \\ \boldsymbol{\varphi}(2)^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\varphi}(n)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi}(1) & \boldsymbol{\varphi}(2) & \cdots & \boldsymbol{\varphi}(n) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}^{\mathrm{T}}(1)\boldsymbol{\varphi}(1) & \cdots & \boldsymbol{\varphi}^{\mathrm{T}}(1)\boldsymbol{\varphi}(n) \\ \vdots & \ddots & \vdots \\ \boldsymbol{\varphi}^{\mathrm{T}}(n)\boldsymbol{\varphi}(1) & \cdots & \boldsymbol{\varphi}^{\mathrm{T}}(n)\boldsymbol{\varphi}(n) \end{bmatrix} \tag{5.42}$$

After simplification, $\mathbf{K}^{\mathbf{\Phi}}$ can be expressed as:

$$\mathbf{K}^{\mathbf{\Phi}} = \begin{bmatrix} K_{1,1} & \cdots & K_{1,n} \\ \vdots & \ddots & \vdots \\ K_{n,1} & \cdots & K_{n,n} \end{bmatrix} \tag{5.43}$$

The use of kernel functions to map nonlinear relations in low-dimensional space to linear relations in high-dimensional space makes PLSR applicable to nonlinear structures. Among many kernel functions, the Gaussian kernel function is characterised by strong localisation and a small number of parameters, which makes it easy to control in practical applications and exhibits excellent performance and strong learning ability. Because the feature space corresponding to the Gaussian kernel function is infinite-dimensional, the Gaussian kernel function is widely used. Therefore, in this chapter, the Gaussian kernel function is selected as the kernel function introduced into the established KPLSR model:

$$K_{i,j} = e^{-\frac{\|\mathbf{X}(i)-\mathbf{X}(j)\|^2}{2\sigma^2}} \tag{5.44}$$

where σ denotes the kernel parameter. Based on the introduction of the kernel function, PLSR is made between the samples mapped by the independent variable and the samples of the dependent variable in a high-dimensional space. If the explicit expression of the mapping function $\mathbf{\Phi}(\cdot)$ is known, PLSR is made between the samples of the X-mapped samples φ and the samples of the dependent variable Y in a high-dimensional linear mapping space. The specific calculation process of the regression model can be referred to in [101] and [102].

The uncertainty of GNSS is estimated using the above method, and then the scenarios of the target AV are evaluated based on the magnitude of the GNSS uncertainty. From the analytical model of error states under spoofing attack established in Chapter 2, it can be seen that the smaller the uncertainty of GNSS, the higher the spoofing success rate. Therefore, if the lateral deviation is higher than the safety threshold, the spoofing difficulty in this scenario is judged to be 'easy', at which the spoofing will reduce the risk of spoofing signals being detected. The spoofing success rate is higher. Otherwise, the spoofing difficulty in this scenario will be judged to be 'difficult', and no spoofing will be performed. Otherwise, the difficulty of spoofing in this scenario will be determined as 'difficult' and no spoofing will be performed.

5.4.2 LiDAR Uncertainty Estimation Method Based on B-Spline Regression

Spoofers can determine the degree of building occlusion around AVs using vehicle location and 3D building models, thereby estimating GNSS uncertainty to assess spoofing effectiveness. LiDAR uncertainty can also be calculated by real-time monitoring of weather conditions at the target vehicle's location. This section establishes a regression model to align estimation results with true sensor uncertainty, thereby assessing spoofing difficulty based on the magnitude of LiDAR uncertainty. Through research in Section 5.3, it is found that different meteorological weather types and classes have varying impacts on LiDAR positioning performance. This chapter further evaluates spoofing effectiveness by quantitatively analyzing the relationship between different weather types, meteorological classes, and LiDAR uncertainty. Experimental data and weather simulators are used to synthesize navigation data under severe weather conditions, simulate spoofing attacks on the MSF system, and compare and analyze spoofing results [91].

Multiple factors influence LiDAR positioning accuracy, including external environmental factors such as weather conditions and atmospheric refraction errors, as well as internal systematic factors, including scanning angle errors and data processing methods. Since LiDAR uncertainty is related to various internal parameters and external environmental factors, real-time estimation of LiDAR uncertainty from limited external observations is unrealistic. However, spoofing effectiveness can be assessed by estimating the average LiDAR uncertainty over an interval.

This chapter establishes a LiDAR uncertainty estimation method based on B-spline regression under severe weather conditions. B-spline regression offers advantages over other nonlinear regression methods, such as polynomial regression and neural network regression. It is suitable for regression models with limited data, requiring no extensive parameter tuning or complex training processes. It generates smoother curves, avoids polynomial regression's overshooting issues, reduces regression curve fluctuations and oscillations, and improves result reliability and stability [103, 104]. Additionally, B-spline regression exhibits excellent local control, meaning local curve changes only affect nearby regions without impacting the entire curve globally, enhancing robustness against local outliers or noise. Therefore, this chapter establishes a spline regression curve with the horizontal coordinate as the meteorological weather class and the vertical coordinate as the LiDAR uncertainty. A spline is a flexible band that generates a smooth curve through a set of specified point

sets. A B-spline curve is a curve whose shape is controlled locally through control points.

$$P(t) = \sum_{i=0}^{n-1} B_{i,d}(t) p_i, \quad t_{\min} \le t \le t_{\max}, 2 \le d \le n \tag{5.45}$$

where $P(t)$ denotes the vector of point coordinates on the regression curve, and the horizontal coordinate of this vector denotes the meteorological weather class, which is rainfall rate r_{Rain}, fog visibility v_{Fog}, and snowfall rate r_{Snow} for different weather types, and the vertical coordinate denotes the need to estimate the LiDAR uncertainty mean $\overline{R}_L$ at that meteorological class; n denotes the number of control points p_i, i.e., the known coordinates involved in the regression model; p_i is the control point coordinates (i starting from 0); $B_{i,d}(t)$ denotes a polynomial coefficient of the weights influenced by the control points' coordinates. The parameter d affects the number of B-spline curves, $d-1$ denotes the number of curves, and t represents the value taken when plotting the curves. The polynomials are computed using the Cox-deBoor recursive formula:

$$B_{k,1}(u) = \begin{cases} 1, & u \in [u_k, u_{k+1}] \\ 0, & u \notin [u_k, u_{k+1}] \end{cases} \tag{5.46}$$

$$B_{k,d}(u) = \frac{u - u_k}{u_{k+d-1} - u_k} B_{k,d-1}(u) + \frac{u_{k+d} - u}{u_{k+d} - u_{k+1}} B_{k+1,d-1}(u) \tag{5.47}$$

Weather scenarios are categorized based on the magnitude of the estimated LiDAR uncertainty. The larger the LiDAR uncertainty, the higher the spoofing success rate through the error state transfer model under the spoofing attack established in Chapter 2. The estimated LiDAR uncertainty is used to evaluate the weather scenario where the autonomous vehicle is located, the spoofing source selectively broadcasts spoofing signals, and the lateral deviation is higher than the safety threshold. The spoofing difficulty in this scenario is judged to be 'easy', and deception at this time will reduce the risk of the spoofing signals being detected, and the spoofing success rate will be higher. Otherwise, the spoofing difficulty in this scenario will be determined as 'easy', and the spoofing success rate will be higher. Otherwise, the spoofing difficulty in this scenario will be judged as 'difficult', and no spoofing will be performed.

5.4.3 GNSS and LiDAR Uncertainty Assessment of Spoofing Effectiveness

This chapter investigates the spoofing technology of the MSF system for AVs. SINS, an autonomous navigation sensor, offers navigation accuracy that is independent of external

environments. Spoofers cannot effectively estimate SINS uncertainty during the spoofing process, so this chapter excludes SINS uncertainty from the spoofing effectiveness assessment. The signal quality of GNSS and LiDAR in MSF systems varies across scenarios, with their uncertainties dynamically changing. The degree of building occlusion impacts GNSS positioning performance, while different meteorological weather levels affect LiDAR positioning performance. During spoofing, the signal quality of GNSS and LiDAR is evaluated through uncertainty estimation results to determine spoofing difficulty. Since spoofers cannot access internal vehicle data, they must estimate sensor uncertainty using environmental information around the vehicle. Regression models are established based on external information such as 3D building models, weather, and meteorological levels to effectively estimate sensor uncertainty and align estimation results with true sensor uncertainty.

GNSS uncertainty can be accurately estimated using 3DMA and KPLSR algorithms, and scenarios can be assessed based on the magnitude of the estimated GNSS uncertainty. According to the state error analysis model under spoofing attack established in Chapter 2, lower GNSS uncertainty indicates better signal quality, higher GNSS positioning accuracy, lower spoofing difficulty, and higher spoofing success rates. Therefore, when the estimated GNSS uncertainty is below the threshold, the scenario is judged as 'easy' for spoofing. At this time, spoofing reduces the risk of signal detection and increases the success rate. Otherwise, it is deemed 'difficult', and no spoofing signals are sent. Similarly, the LiDAR uncertainty estimation algorithm based on B-spline regression can accurately estimate LiDAR uncertainty and evaluate different weather scenarios. According to the state error transfer model under spoofing attacks, when the estimated LiDAR uncertainty exceeds the threshold, it indicates poorer LiDAR signal quality, lower spoofing difficulty, and higher spoofing success rates. Thus, scenarios with LiDAR uncertainty above the threshold are judged as 'easy' for spoofing, while others are deemed 'difficult'. Ultimately, the spoofing effectiveness based on GNSS and LiDAR uncertainty can be expressed as:

$$Ass_k^{spo} = \begin{cases} 1, & \tilde{R}_k^G < R_{Threshod}^G \quad \text{or} \quad \tilde{R}_k^L > R_{Threshod}^L \\ 0, & \text{otherwise} \end{cases} \tag{5.48}$$

where $\tilde{R}_k^G$ and $\tilde{R}_k^L$ denote the estimated GNSS and LiDAR uncertainties, respectively, and $R_{Threshod}^G$ and $R_{Threshod}^L$ denote the uncertainty thresholds set for GNSS and LiDAR, respectively. $Ass_k^{spo} = 1$ indicates that the spoofing scenario is evaluated as 'easy', and $Ass_k^{spo} = 0$ indicates that the spoofing scenario is evaluated as 'difficult'.

In practical applications, spoofers only conduct spoofing attacks when the scenario is

assessed as ‘easy’, selectively broadcasting spoofing signals to the target AV. This prevents blind spoofing attacks, improves spoofing success rates, and reduces the risk of spoofing signals being detected by the target AV.

5.5 Real-world Data Verification

This chapter utilizes a software platform developed based on Autoware and the PSINS C++ toolbox to implement the MSF algorithm. Spoofing signals are simulated by directly superimposing position spoofing parameters onto the GNSS position output. The test condition settings are first explained, followed by real-world data simulation spoofing tests for the MSF system under different scenarios.

5.5.1 Setup

The data platform of the Intelligent Positioning and Navigation Laboratory (IPNL) at the Hong Kong Polytechnic University is used to collect positioning and navigation data in urban environments, which is then used to validate the proposed method in this chapter. The relevant navigation sensors include GNSS, LiDAR, and SINS, with specific sensor parameter descriptions and test environment configurations provided in Chapter 4.

5.5.2 Spoofing Effectiveness Validation under Different Geographical Scenarios

1) Scenario Selection

The first test scenario is selected as an urban canyon near Tsim Sha Tsui, Hong Kong (the same as the first scenario in Chapter 4). During data acquisition, the vehicle’s trajectory and speed information are obtained using the configured sensors, as shown in Fig. 5.9. Different colors indicate the vehicle’s speed in m/s. The total duration of this dataset is 380 seconds. Due to occlusion by surrounding buildings or trees, RTK positioning information is solved for only 170 epochs. The sky visibility information of the area is generated using the 3D building model-assisted sky visibility estimation algorithm proposed in this chapter, as shown in Fig. 5.10. From the sky visibility information, it can be seen that the vehicle spends most of its time in heavily obscured areas, with only part of the trajectory in the upper left passing through a more open area.

The second scenario is around the coast in Whampoa, Hong Kong. In this test scenario, the vehicle starts in an open environment near the sea and then enters a narrow street

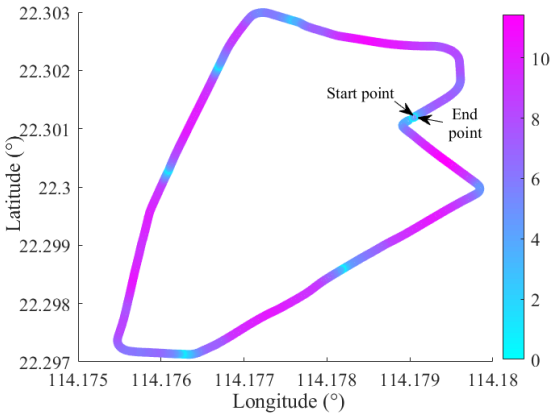

Figure 5.9: Vehicle trajectory and speed in scenario 1

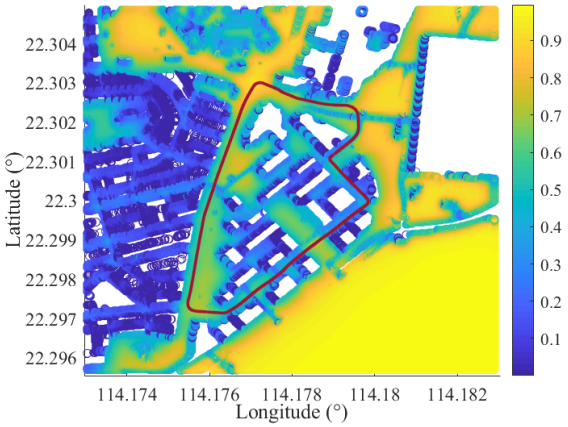

Figure 5.10: Sky visibility map for scenario 1

along a wide road. At this point, the buildings on both sides of the lane are denser, leading to higher GNSS uncertainty due to increased satellite signal blockage. The vehicle's trajectory and speed information are shown in Fig. 5.11 , with different colors indicating the vehicle's speed in m/s. The total duration of this dataset is 1538 seconds, with RTK positioning information solved for 1230 epochs. The sky visibility information of the area is also generated using the 3D building model-assisted sky visibility estimation algorithm proposed in this chapter, as shown in Fig. 5.12 .

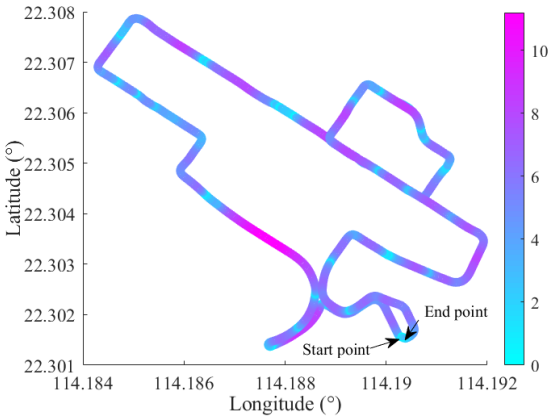

Figure 5.11: Vehicle trajectory and speed in scenario 2

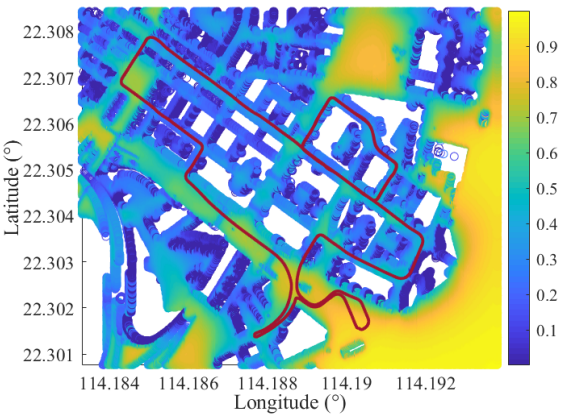

Figure 5.12: Sky visibility map for scenario 2

The sky visibility information indicates that the vehicle is in a more open area during the initial and final phases, while in the middle phase, it is more heavily obscured by buildings on both sides of the lane.

2) Validation of 3DMA Model via Correlation Analysis

This section compares the proposed 3DMA model with the traditional LiDAR Point-cloud Height (LPH)-based model for calculating sky occlusion masks in each azimuthal region through correlation analysis [105].

First, sky visibility masks of the target area are generated using both methods. Then, sky visibility and the number of visible satellites are estimated from the masks for the two scenarios. In Scenario 2, the last 380 epochs are selected for comparison. In the experimental data processing of this chapter, the open-source software RTKLIB is used to obtain the final positioning results of GNSS. The standard deviation of the position estimate from RTKLIB is used as the reference value for GNSS uncertainty [90]. In RTKLIB, the GNSS uncertainty reference value is calculated based on the residuals and weight matrix of satellite signal observations. The sky visibility and GNSS uncertainty variation curves generated by the two methods for the two scenarios are shown in Fig. 5.13 and Fig. 5.14 .

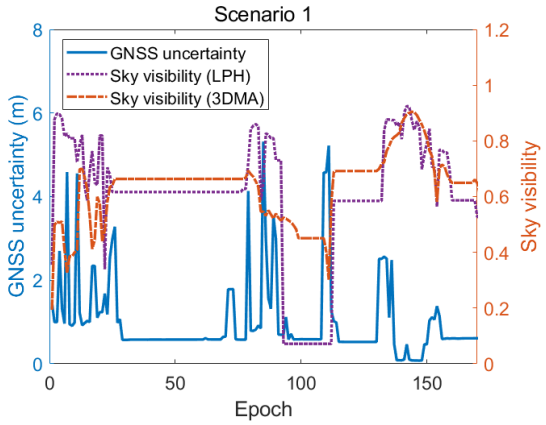

Figure 5.13: Sky visibility and GNSS uncertainty in scenario 1

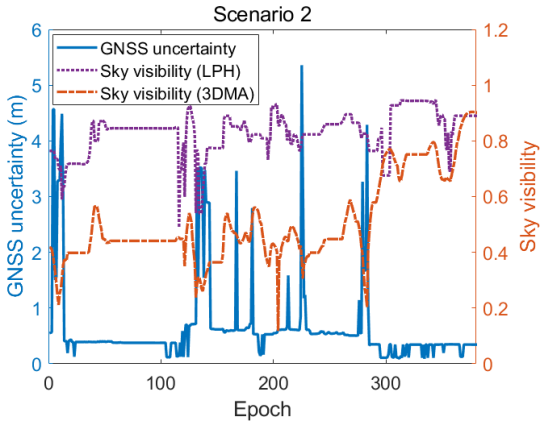

Figure 5.14: Sky visibility and GNSS uncertainty in scenario 2

To verify the accuracy of the estimation results, the correlation between the parameters is calculated. The correlation coefficient is a statistical measure indicating the degree of correlation between two sets of variables, ranging from -1 to 1. Positive and negative values indicate positive and negative correlations, respectively. The larger the absolute value, the stronger the correlation. A correlation coefficient close to 1 or -1 indicates a strong correlation; a value close to 0 indicates a weak correlation. When the absolute value of the correlation coefficient exceeds 0.5, the two variables are considered strongly correlated; between 0.3 and 0.5, they are moderately correlated; and between 0.1 and 0.3, they are weakly correlated [106, 107]. The correlation coefficients between the sky visibility estimated by the traditional LPH model and the 3DMA model proposed in this chapter and the GNSS uncertainty reference values are calculated to validate the proposed method.

The Kendall correlation coefficient is suitable for non-linear, non-normally distributed data. Since the relationship between sky visibility and GNSS uncertainty is not necessarily linear, the Kendall method is used in this chapter to calculate the correlation coefficient between the two variables. There are two formulas for calculating the Kendall correlation coefficient. The Tau-a method is used in this chapter as an example. First, all possible pairs of elements are generated. For n samples, each corresponding to the values of variables X and Y, all samples are combined pairwise to generate $C(n,2) = \frac{n(n-1)}{2}$ pairs of elements. Then the sample is $(X_1,Y_1),(X_2,Y_2),\cdots,(X_n,Y_n)$, then all possible pairs of elements are: (i,j), where $1 \leq i < j \leq n$. Each pair of elements is compared to determine consistency and inconsistency:

1. Consistency: if $X_i > X_j$ and $Y_i > Y_j$, or $X_i < X_j$ and $Y_i < Y_j$, then the pair of elements is consistent and counts as c.

2. Inconsistency: if $X_i > X_j$ but $Y_i < Y_j$, or $X_i < X_j$ but $Y_i > Y_j$, then the pair of elements is inconsistent and counts as d.

3. Juxtaposition: if $X_i = X_j$ or $Y_i = Y_j$, then the pair of elements is neither categorized as a consistent pair nor an inconsistent pair.

For n samples, iterate over all pairs of $i < j$ elements and count the final consistent pairs c and inconsistent pairs d :

$$\begin{cases} c = \sum\limits_{i<j} \mathbf{1}\left[(X_i - X_j)(X_i - X_j) > 0\right] \\ d = \sum\limits_{i<j} \mathbf{1}\left[(X_i - X_j)(X_i - X_j) < 0\right] \end{cases} \tag{5.49}$$

where $\mathbf{1}[*]$ denotes the indicator function, which is 1 when the condition is satisfied and 0 otherwise. The final Kendall correlation coefficient is calculated:

$$Tau - a = \frac{c-d}{\frac{1}{2}n(n-1)} \tag{5.50}$$

where, c denotes the number of pairs of elements in the two variables that are concordant (i.e., the relative order of the two elements in the two variables is the same), and d denotes the number of discordant pairs, representing the number of pairwise combinations of all samples.

The correlation coefficients between the sky visibility calculated by the two methods and the GNSS uncertainty reference value are calculated for the two scenarios using the above correlation coefficient formula, and the statistical results are shown in Tab. 5.1 .

Table 5.1: Correlation parameter comparison of the estimated sky visibility and GNSS uncertainty

Scenario	LPH	**3DMA**
Scenario 1	-0.13	-0.41
Scenario 2	-0.27	-0.49

From the statistical results, all correlation coefficients are negative, indicating a negative correlation between sky visibility and GNSS uncertainty. The traditional LPH model's sky visibility estimation shows a weak correlation with the GNSS uncertainty reference value in both scenarios. In contrast, the sky visibility estimation by the model proposed in this chapter exhibits a moderate correlation with the GNSS uncertainty reference value in both scenarios. The absolute value of the correlation coefficient for the 3DMA model proposed in this chapter is significantly higher than that of the conventional LPH model for both scenarios. Therefore, the 3DMA model proposed in this chapter outperforms the traditional LPH model.

Subsequently, sky visibility masks are generated using the traditional LPH model and the 3DMA model proposed in this chapter. The number of visible satellites observable by the AV is estimated by combining the satellite ephemeris information at that time, with GPS and Galileo selected as the satellite ephemeris. The final results are shown in Fig. 5.15 and Fig. 5.16 .

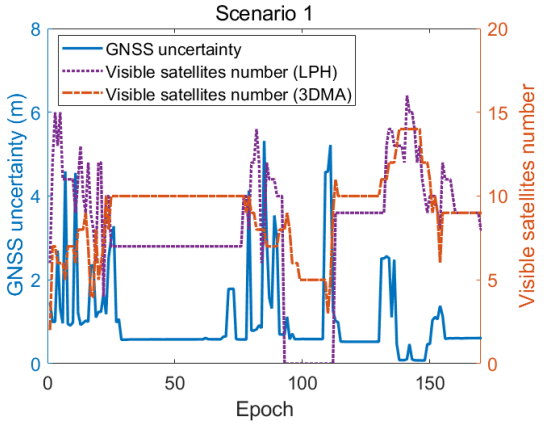

Figure 5.15: The number of visible satellites and GNSS uncertainty in scenario 1

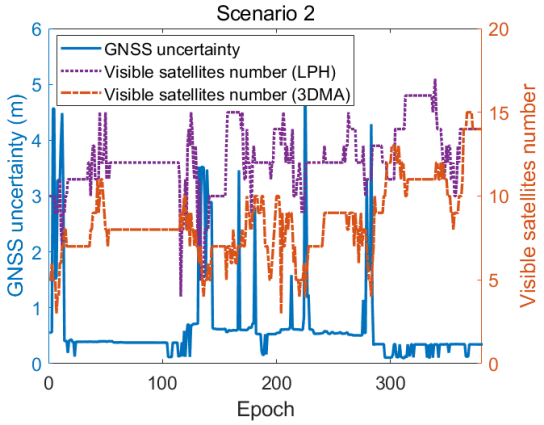

Figure 5.16: The number of visible satellites and GNSS uncertainty in scenario 2

Similarly, the correlation coefficients of the number of visible satellites estimated by the two methods with the GNSS uncertainty reference value are calculated using the Kendall correlation coefficient formula, and the statistical results are shown in Tab. 5.2 .

Table 5.2: Correlation comparison of the estimated number of visible satellites and GNSS uncertainty

Scenario	LPH	**3DMA**
Scenario 1	-0.09	-0.51
Scenario 2	-0.32	-0.52

From the statistical results, all correlation coefficients are negative, indicating a negative correlation between the number of visible satellites and GNSS uncertainty. The results of the correlation coefficients in the two different scenarios show that the correlation between the number of visible satellites estimated by the 3DMA proposed in this chapter and the GNSS uncertainty reference value is significantly higher than that of the traditional LPH method. The number of visible satellites estimated by the traditional LPH model in both scenarios shows no correlation or moderate correlation with the GNSS uncertainty reference value. In contrast, the number of visible satellites estimated by the model proposed in this chapter in both scenarios exhibits a strong correlation with the GNSS uncertainty reference value. Therefore, the 3DMA model proposed in this chapter is superior to the conventional LPH model.

3) GNSS Uncertainty Estimation Results and Assessment of Spoofing Effectiveness

During the experiment, based on the sky visibility and the number of visible satellites estimated by the 3DMA model, the regression model is trained by selecting the first 140 epochs in Scenario I and the first 1130 epochs in Scenario II. The regression model is validated by choosing the last 30 epochs in Scenario I and the last 100 epochs in Scenario II. Then, the GNSS uncertainty is estimated by kernel function partial least squares regression. In the regression analysis, the F test is used to test the overall significance of the regression model, i.e., to test whether at least one of the independent variables in the model has a significant effect on the dependent variable. Specifically, it is used to test the following null hypothesis:

1. H_0 null hypothesis: the regression coefficients of all independent variables in the model are equal to zero, i.e., the model has no predictive power.

2. H_1 alternative hypothesis: at least one of the independent variables in the model has a regression coefficient not equal to zero, i.e., the model has predictive power.

It is common to use the training set to fit the model and compute the F statistic. This

is because the purpose of the F test is to assess the overall explanatory power of the independent variables in the model for the dependent variable. This assessment is based on the statistics generated during the model fitting process, such as Regression Sum of Squares (SSR), Residual Sum of Squares (Error Sum of Squares, SSE), and so on. Firstly, SSR and SSE are calculated separately:

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \tag{5.51a}$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{5.51b}$$

where $\hat{y}_i$ denotes the value of the dependent variable for the ith observation; $\bar{y}$ denotes the mean value of the dependent variable; y_i denotes the predicted value for the ith observation; and n denotes the number of observations.

To verify the effectiveness of the proposed GNSS uncertainty estimation method based on kernel function partial least squares regression, the proposed model is compared with the traditional linear and nonlinear regression models. Common traditional methods include the Partial Least Squares Regression (PLSR) model and the Second Order Nonlinear Regression (SONR) model. The SONR model and the Least Squares Support Vector Regression (LS-SVR) model. The final GNSS uncertainty estimation results of different regression models are shown in Fig. 5.17 and Fig. 5.18 .

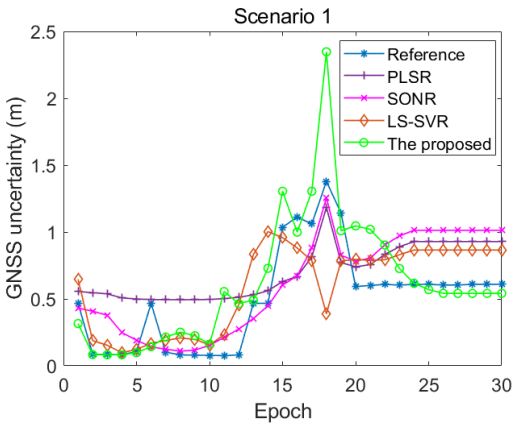

Figure 5.17: GNSS uncertainty estimation results in scenario 1

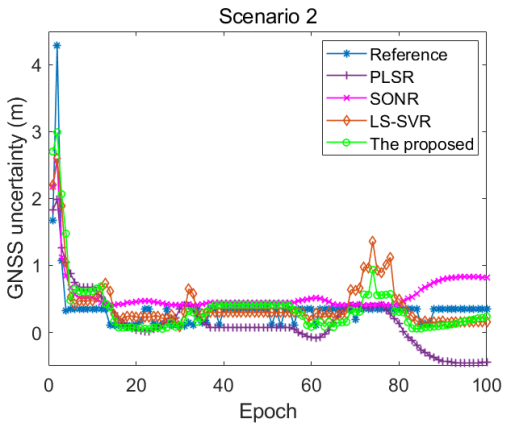

Figure 5.18: GNSS uncertainty estimation results in scenario 2

Calculate the RMSE between the estimated GNSS uncertainties of the two scenarios and the reference values, and the final results are statistically presented in Tab. 5.3 .

By comparing with the traditional regression models, the RMSE of the proposed method between the estimated GNSS uncertainty and the reference value is minimum in both urban

Table 5.3: The RMSE between the estimated GNSS uncertainty and the reference value (m)

Scenario	PLSR	SONR	LS-SVR	**The proposed**
Scenario 1	0.33	0.28	0.29	0.27
Scenario 2	0.45	0.33	0.32	0.29

test scenarios. Therefore, the KPLSR model based on 3DMA established in this thesis has the highest GNSS uncertainty estimation accuracy, thus verifying the effectiveness of the proposed method.

Then, based on the estimation results of GNSS uncertainty, the spoofing effectiveness is evaluated, which further validates the effectiveness of the proposed method in this chapter. For different geographical scenarios, it is defined that when the uncertainty of GNSS is less than 0.6m, the quality of the satellite signal is better, and the accuracy of GNSS positioning is higher, so the spoofing difficulty in this scenario is judged as 'easy'. On the contrary, when the uncertainty of GNSS is greater than 0.6m, the quality of the satellite signal is poorer, and the accuracy of GNSS positioning is lower, which is judged as 'difficult'. On the other hand, when the uncertainty of GNSS is greater than 0.6m, the quality of the satellite signal is poor, and the positioning accuracy of GNSS is low, which is judged as 'difficult'. Since the uncertainty estimate of the method proposed in this chapter is the most accurate. Theoretically, the success rate is higher in the area that is judged as easy by the method proposed in this chapter.

In the spoofing test, the spoofing window is still set to 10s. The thresholds for successful spoofing are also calculated according to the settings in Chapter 2, and the two thresholds are set to 0.75 m and 2.86 m. A successful spoofing is considered to have occurred when the lateral deviation of the positioning information output from the MSF system exceeds 2.86 m. The spoofing attack is then implemented in two different urban scenarios to verify the effectiveness of the method proposed in this chapter.

The starting point for the spoofing attack is randomly selected in the geographic scenarios judged to be 'easy', and the total spoofing number is set to 50. Different regression models are used to estimate uncertainty, and then the scenarios where the AVs are located are categorized. The spoofing attack is then carried out in the easy-to-deceive areas. In the end, the number of successful attempts is 28 for conventional PLSR, 31 for SONR, and 32 for LS-SVR. According to the effectiveness evaluation model proposed in this chapter, the number of successful spoofing attacks for broadcasting spoofing signals in 'easy' spoofing

scenarios is 36. All of these regression models are based on the sky visibility mask generated by 3DMA. The number of successful spoofs in the 'easy' spoofed area evaluated with the sky visibility mask generated with the aid of LPH is 23. Without categorization, the number of successful spoofs in random areas across the interval is only 8. Finally, the spoofing success rate in the 'easy' spoofed regions estimated by the different methods is counted, and the results are shown in Fig. 5.19 .

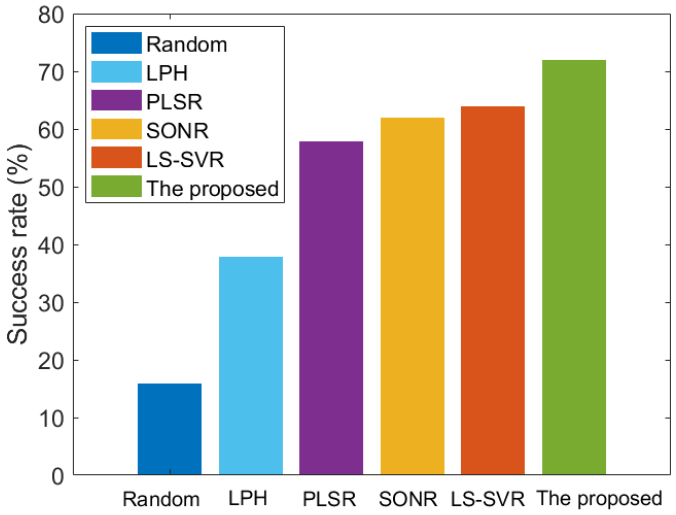

Figure 5.19: The spoofing success rate in the 'easy' spoofing scenario

Therefore, the proposed method estimates the highest success rate for spoofing in the 'easy' spoofing scenarios compared to traditional effectiveness evaluation methods. It is worth noting that the two selected urban scenarios contain some open areas, i.e., areas with low GNSS and LiDAR uncertainties. Thus, successful spoofing attacks of the MSF system can be achieved. However, suppose the AV is always traveling in an urban canyon area with denser buildings on both sides of the lane. In that case, it is possible that the geo-environmental classification model proposed in this chapter will not be able to find easy to spoof scenarios. Therefore, no spoofing is performed in these challenging scenarios, thereby preventing ineffective spoofing signals from being generated by spoofing sources.

In summary, the effectiveness of the proposed method is finally verified in various geographic scenarios through simulated spoofing tests using actual data and different scenarios.

5.5.3 Spoofing Effectiveness Validation Under Different Weather Scenarios

1) Scenario Selection

For the validation of spoofing effectiveness under different weather scenarios, the navigation data of the sensors under clear weather conditions is first collected through the test platform. A suburban scenario is selected for the test (the same as the second scenario in Chapter 4, with the test scenario, route trajectory, as well as the start and end points shown in Fig. 4.11). The weather simulator [91] is used to simulate and generate LiDAR signals under different meteorological weather levels, including rain, fog, and snow, based on the original LiDAR point cloud of the whole interval. Then, the localization results under different weather conditions for the MSF system, with real-world data simulation and spoofing attacks, are analyzed.

The localization performance of LiDAR in clear weather is analyzed using the NDT matching algorithm to process laser point cloud data collected under clear weather conditions. Firstly, the a priori map of the route is built by Autoware. Then, the LiDAR localization results are obtained by the NDT matching algorithm. The total duration of the whole interval is 200s, and the matching result at the 10th second is shown in Fig. 5.20 (b). From the results, it can be seen that there are sufficient feature points under sunny weather conditions, resulting in better matching results.

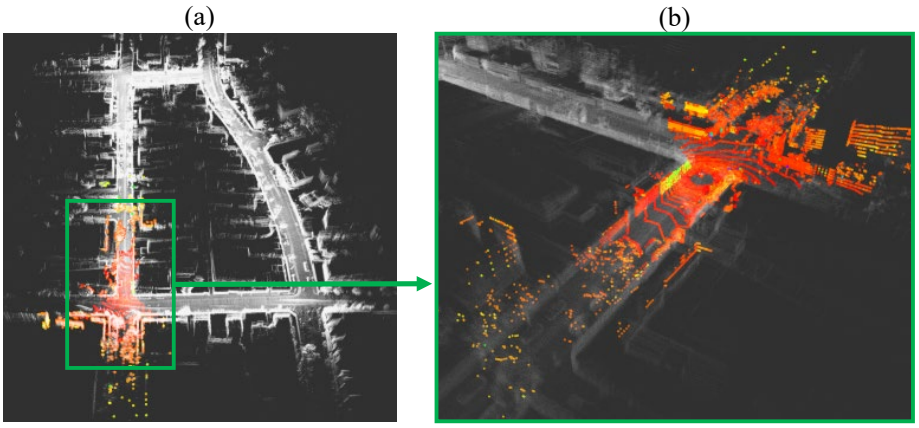

Figure 5.20: Under clear weather conditions, (a) prior point cloud map and (b) NDT matching results of the 10s

2) Variation of LiDAR Point Cloud Response Ratios Under Different Weather Conditions

The LiDAR point cloud under different weather conditions is simulated by adding various weather signals with different intensities to the LiDAR point cloud collected under clear weather conditions. Then, the NDT matching algorithm is performed to obtain the localization results, including localization error and uncertainty, followed by data fusion and spoofing attacks. Firstly, the NDT matching results of three rainfall rates are compared as an example, corresponding to moderate rain, heavy rain and heavy rain. At 10s, the matching results are shown in Fig. 5.21 .

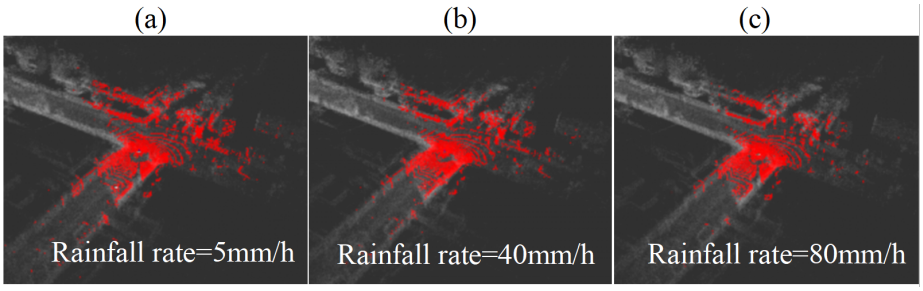

Figure 5.21: NDT matching results for different rainfall rates at 10s

Similarly, the point cloud is simulated by adding fog of varying intensities to the initial point cloud. The NDT matching results for three visibility levels are compared as an example, corresponding to slight fog, heavy fog, and dense fog, respectively. At 10s, the matching results are shown in Fig. 5.22 .

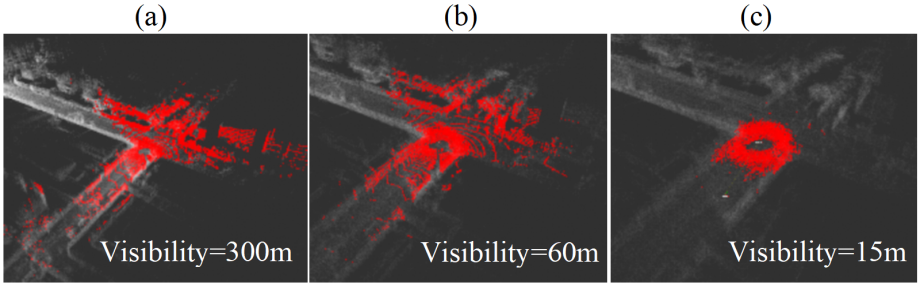

Figure 5.22: NDT matching results for different fog visibility at 10s

Then, the same method is used to obtain the point clouds under different snowfall intensities. As an example, the NDT matching results for three snowfall rates are compared, corresponding to slight, medium, and heavy snow. The results of the NDT matching algorithm at 10s are shown in Fig. 5.23 .

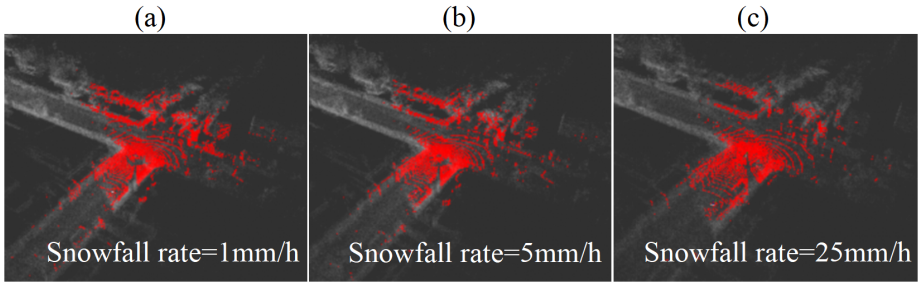

Figure 5.23: NDT matching results for different snowfall rates at 10s

In addition, a response ratio parameter is defined to evaluate the changes in the LiDAR point cloud before and after the inclusion of severe weather effects.

$$\gamma = \frac{n_\alpha + n_\beta}{n_0} \times 100\% \tag{5.52}$$

where γ denotes the response ratio parameter; n_0 denotes the number of initial point clouds of LiDAR in clear weather; n_α and n_β denote the number of attenuation of LiDAR point cloud signals and the number of backscattering in bad weather, respectively, which are affected by water molecules in terms of the magnitude of the two parameters. If the attenuation number of the point cloud is larger, it means that water molecules obscure more feature points. Therefore, some stationary targets cannot be effectively detected due to the long distance. If the number of backscattering of the point cloud is larger, it means that the laser signal is more likely to be reflected by water molecules, which forms more noise in the air and thus increases the error of the NDT matching algorithm. Therefore, the larger the response ratio value, the greater the impact of weather on the LiDAR point cloud and the output localization results.

Finally, the response ratios for different weather types and meteorological levels are calculated at 10s to evaluate the effect of weather conditions on the LiDAR point cloud. The final statistical results are shown in Tab. 5.4. As weather conditions intensify from slight rain, fog, or snow to heavy conditions, the response ratio gradually increases. Among the three weather types, fog has the most significant impact on LiDAR point clouds. At 15m of visibility, dense noise surrounds LiDAR, affecting 85.61% of the point cloud. This severely limits LiDAR's detection range and the accuracy of the NDT matching algorithm, ultimately reducing LiDAR's positioning accuracy and its weight in the MSF system of AVs.

3) Uncertainty Estimation of LiDAR NDT Matching Under Different Meteorological Parameters

Table 5.4: Response ratio under different weather types and weather levels

Weather type	Meteorological level	Meteorological parameter	Response ratio
Rain	Moderate rain	Rainfall rate=5mm/h	13.13%
	Heavy rain	Rainfall rate=40mm/h	15.96%
	Violent rain	Rainfall rate=80mm/h	18.73%
Fog	Slight fog	Visibility=300m	4.83%
	Heavy fog	Visibility=60m	48.03%
	Violent fog	Visibility=15m	85.61%
Snow	Slight snow	Snowfall rate=1mm/h	13.60%
	Moderate snow	Snowfall rate=5mm/h	16.12%
	Heavy snow	Snowfall rate=25mm/h	20.70%

Theoretically, higher meteorological classes lead to worse LiDAR localization accuracy. To quantify the impact of severe weather on localization accuracy, we analyze the 2D plane uncertainty, which directly affects LiDAR's weight in the MSF system of AVs. Using the standard deviation of uncertainty to evaluate NDT matching reliability, we reference the LiDAR uncertainty output from Autoware software [21, 81]. We analyze the effect of rainfall on localization uncertainty, as shown in Fig. 5.24. The mean uncertainty value across the entire trajectory interval under the same rainfall rate is calculated using a B-spline regression model, as depicted in Fig. 5.25. For different weather scenarios, LiDAR uncertainty exceeding 2 meters is classified as 'easy' to spoof, while under 2 meters is classified as 'difficult'. When rainfall rates surpass 35.28 mm/s, scenarios are deemed 'easy' to spoof.

Results indicate that the uncertainty of the NDT matching algorithm gradually increases with rainfall. When rainfall rates exceed 20 mm/h, the average uncertainty of localization results significantly increases compared to clear weather, substantially reducing LiDAR's weight in the MSF system.

For foggy days, the final statistics of the uncertainty of the NDT matching algorithm for the whole interval are shown in Fig. 5.26. When the fog visibility is less than 85.52m, it is defined as an easy spoofing scenario, as shown in Fig. 5.27.

Results show that NDT matching uncertainty tends to increase as visibility decreases. Slight or moderate fog has minimal effects on LiDAR uncertainty, but heavy and dense fog significantly increasess uncertainty. When fog visibility reaches 15 meters, the average

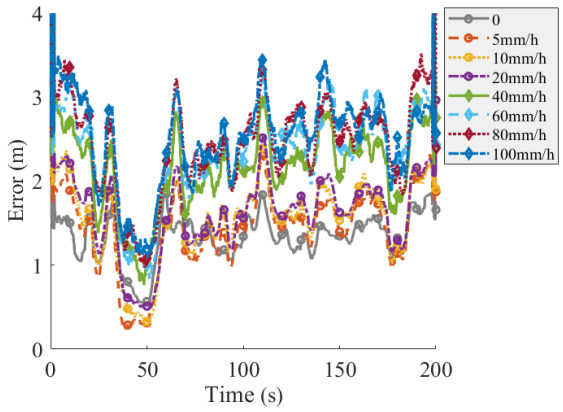

Figure 5.24: Changes of LiDAR uncertainty under different rainfall rates

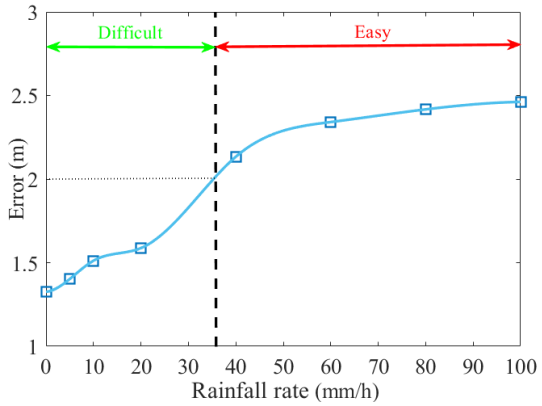

Figure 5.25: Classification results of spoofing scenarios under different rainfall rates

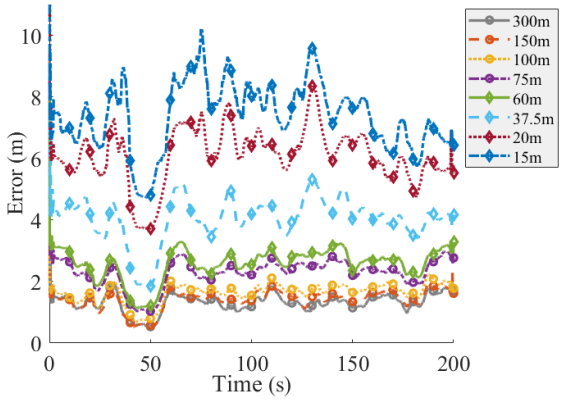

Figure 5.26: Changes of LiDAR uncertainty under different fog visibility conditions

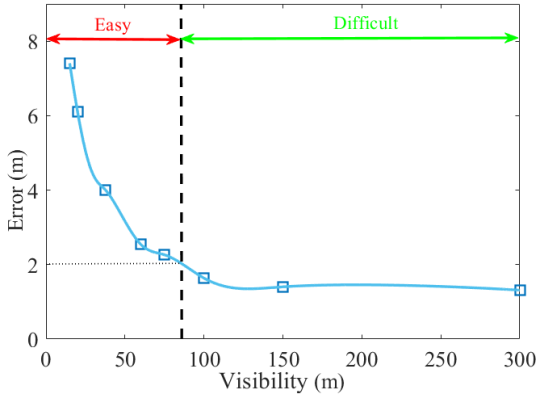

Figure 5.27: Classification results of spoofing scenarios under different fog visibility conditions

uncertainty across the vehicle's entire trajectory interval exceeds 7 meters.

For snowy days, the final statistics of LiDAR uncertainty for the whole interval are shown in Fig. 5.28 . When the snowfall rate is larger than 3.68 mm/h, it is defined as an easily spoofed scenario, as shown in Fig. 5.29 .

Results indicate that NDT matching uncertainty gradually increases with snowfall rates. Snowfall also affects LiDAR localization results, reducing LiDAR's weight in the MSF system under heavy snow conditions.

As an example, the final statistics of the results under partial weather levels are shown in Tab. 5.5 .

In summary, the uncertainty results of the NDT matching algorithm align with the

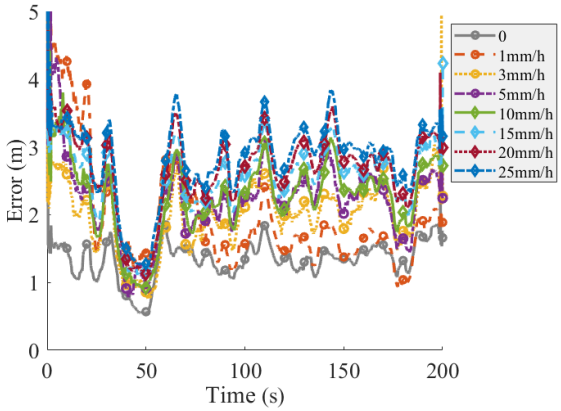

Figure 5.28: Changes of LiDAR uncertainty under different snowfall rates

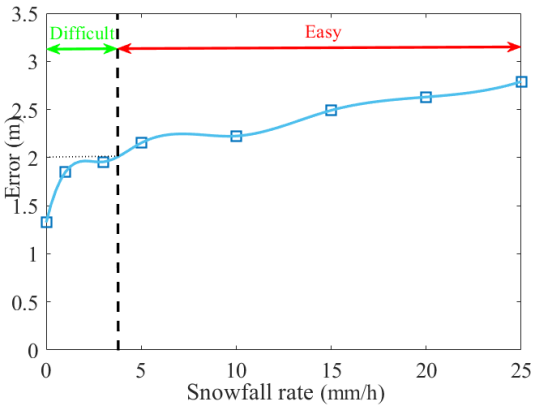

Figure 5.29: Classification results of spoofing scenarios under different snowfall rates

Table 5.5: The mean value of uncertainty under different weather conditions

Weather type	Meteorological level	Meteorological parameter	Mean value of uncertainty
Rain	Moderate rain	Rainfall rate=5mm/h	1.40m
	Heavy rain	Rainfall rate=40mm/h	2.13m
	Violent rain	Rainfall rate=80mm/h	2.42m
Fog	Slight fog	Visibility=300m	1.33m
	Heavy fog	Visibility=60m	2.54m
	Violent fog	Visibility=15m	7.41m
Snow	Slight snow	Snowfall rate=1mm/h	1.85m
	Moderate snow	Snowfall rate=5mm/h	2.15m
	Heavy snow	Snowfall rate=25mm/h	2.79m

trends in localization error across various weather meteorological classes. At lower weather classes, conditions have a minimal impact on localization results, as sufficient effective point clouds maintain NDT matching accuracy. However, as weather classes intensify, LiDAR's effective scanning distance decreases, and airborne noise from severe weather increases. Among the three severe weather types analyzed, fog exhibits the most significant impact on LiDAR.

4) Spoofing Effectiveness Evaluation Under Different Meteorological Parameters

This chapter further evaluates the effectiveness of spoofing on MSF systems under various meteorological parameters, including the impact of rain, fog, and snow weather scenarios on the spoofing results. It also investigates the relationship between weather and meteorological parameters and the spoofing success rate.

Since there is no RTK Fix solution between 162s-168s, the intervals of spoofing attack are set to be 5s-160s as well as 170s-200s in this set of data. The interval between each implementation of spoofing is 5s, and the duration of one spoofing is 10s, which makes a total of 35 spoofing epochs implemented. Similarly, the MSF system is spoofed using the covert spoofing method proposed in Chapter 4. In addition, the initial spoofing lateral deviation is set to 1m, and other parameters are consistent with the experimental settings in the above section. When the positioning error of the MSF system is greater than 2.86m, it indicates that this spoofing is successful, i.e., the lateral deviation of the MSF system exceeds the threshold value. Then, the number of successful spoofing times in the whole trajectory interval is counted, and the spoofing success rate is calculated under different weather and meteorological classes, respectively.

First, we conduct spoofing attacks on the AV's MSF system under clear weather. Among 35 implemented attacks, only one results in a lateral deviation exceeding 2.86m. Thus, the spoofing success rate under clear weather conditions is very low at 2.86%, consistent with theoretical analysis. Subsequently, we execute spoofing attacks on the MSF system of AVs under various meteorological conditions. The spoofing success rates under different rainfall rates for MSF systems are illustrated in Fig. 5.30 .

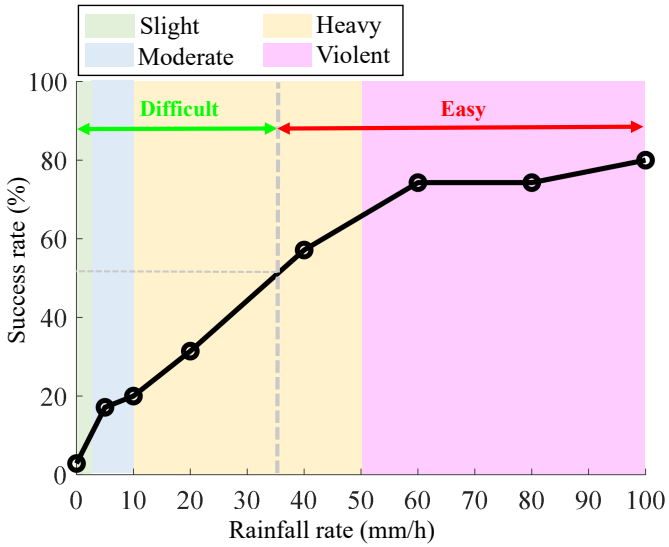

Figure 5.30: Relationship between spoofing success and rainfall rates for MSF systems

The relationship between spoofing success rate and rainfall rate for the pair of MSF systems of AVs under spoofing attack is statistically analyzed, and the results are shown in Tab. 5.6 .

Results indicate that as rainfall rates increase, spoofing success rates also rise. When rainfall rates exceed 40mm/h, the GNSS spoofing success rate on MSF systems surpasses

Table 5.6: Relationship between spoofing success and rainfall rates for MSF system

Weather type	Meteorological parameters	**Spoofing difficulty level**	**Spoofing success rate (%)**
Rain	Rainfall rate=5mm/h	Difficult	17.14
	Rainfall rate=10mm/h		20.00
	Rainfall rate=20mm/h		31.43
	Rainfall rate=40mm/h	Easy	57.14
	Rainfall rate=60mm/h		74.29
	Rainfall rate=80mm/h		74.29
	Rainfall rate=100mm/h		80.00

57%. At 100mm/h rainfall rates, the GNSS spoofing success rate on MSF systems reaches 80%. Thus, spoofing success rates are higher in rainy weather compared to clear conditions.

For the spoofing results in different foggy weather conditions, the statistics of spoofing success rates for various visibility levels in MSF systems are shown in Fig. 5.31 .

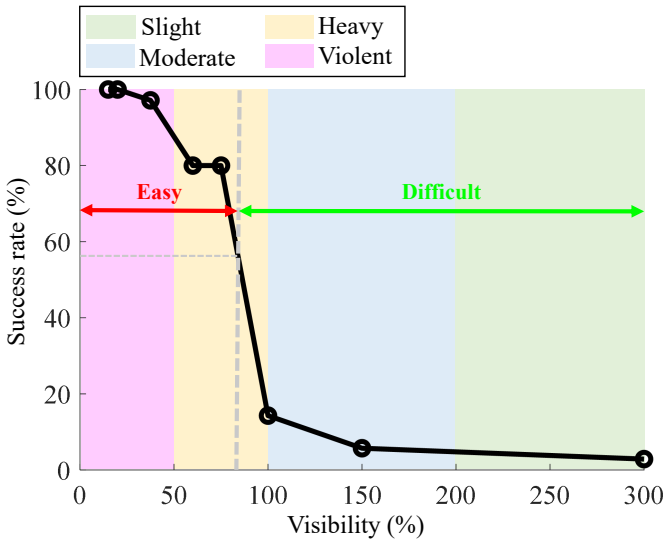

Figure 5.31: Relationship between spoofing success against the MSF system and fog visibility

The relationship between the spoofing success rate and fog visibility of the MSF system for autonomous vehicles under spoofing attack is statistically determined, and the results are shown in Tab. 5.7 .

Results show that as visibility decreases in foggy conditions, the spoofing success rate gradually increases. When visibility drops below 75 meters, spoofing success rates are at

Table 5.7: Relationship between spoofing success against MSF system and fog visibility

Weather type	Meteorological parameters	**Spoofing difficulty level**	**Spoofing success rate (%)**
Fog	Visibility=300m	Difficult	2.86
	Visibility=150m		5.71
	Visibility=100m		14.29
	Visibility=75m	Easy	80.00
	Visibility=60m		80.00
	Visibility=37.5m		97.14
	Visibility=20m		100.00
	Visibility=15m		100.00

least 80%. Below 20 meters visibility, all 35 spoofing attempts on the MSF system succeed, indicating a 100% success rate for vehicle-mounted MSF systems under such weather conditions.

For different snowfall conditions, spoofing attacks are implemented on the MSF system at varying snowfall rates, with success rates shown in Fig. 5.32 .

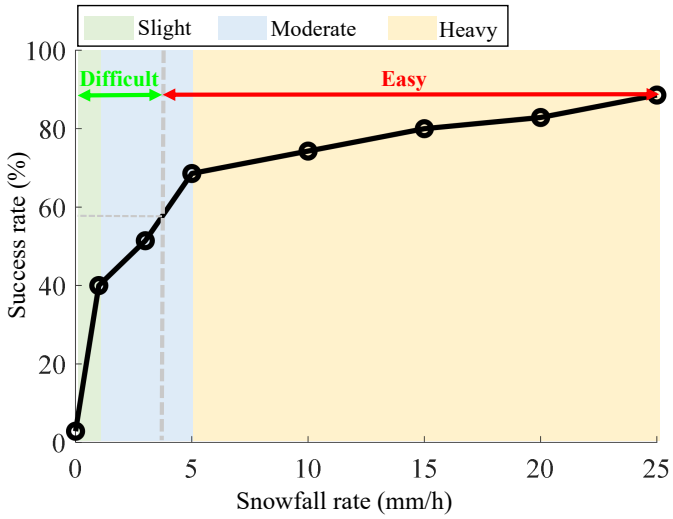

Figure 5.32: Relationship between spoofing success and snowfall rates for MSF system

The relationship between spoofing success rates and snowfall rates for MSF systems under spoofing attacks is detailed in Tab. 5.8 .

Results indicate that spoofing success rates gradually increase with snowfall rates, suggesting that spoofing the MSF system of AVs becomes less challenging as snowfall intensifies. The success rate of spoofing can exceed 68% in scenarios that are easy to spoof.

Table 5.8: Relationship between spoofing success and snowfall rates for MSF system

Weather type	Meteorological parameters	**Spoofing difficulty level**	**Spoofing success rate (%)**
Snow	Snowfall rate=1mm/h	Difficult	40.00
	Snowfall rate=3mm/h		51.43
	Snowfall rate=5mm/h	Easy	68.57
	Snowfall rate=10mm/h		74.29
	Snowfall rate=15mm/h		80.00
	Snowfall rate=20mm/h		82.86
	Snowfall rate=25mm/h		88.57

In summary, among the three weather types (rain, snow, and fog), fog exerts the most significant impact on spoofing outcomes under equivalent weather class conditions. Snow's effect on spoofing is less than fog's but greater than rain's. This aligns with the impact of severe weather on LiDAR localization errors and uncertainties. In contrast, the spoofing success rate for MSF systems in clear weather is only 2.86%.

The results demonstrate that spoofing success rates significantly improve when executed in scenarios with higher weather and meteorological classes. Thus, the assessment method proposed in this chapter proves effective in enhancing spoofing success rates for MSF systems of AVs under adverse weather conditions. By quantitatively analyzing the relationship between weather classes and spoofing success rates using meteorological information, the proposed method allows spoofing sources to evaluate spoofing effectiveness. When weather classes are severe, spoofing attacks on targets are likely to disable the navigation systems of AVs with high probability.

This study provides a theoretical foundation for designing defense algorithms of GNSS /SINS/LiDAR-based AVs in the event of potential malicious spoofing attacks, thereby facilitating further research into GNSS anti-spoofing algorithms in adverse weather conditions. Thereby, it enables vehicles to implement proactive measures, such as activating emergency plans, slowing down, or even stopping in the event of safety risks. Immediate restoration of the MSF system performance ensures vehicle safety and reduces the risk of catastrophic traffic accidents. Ultimately, this study ensures the position safety of the MSF system in various weather scenarios.

5.6 Summary

This chapter proposes a spoofing effectiveness assessment method based on sensor uncertainty estimation. By constructing a nonlinear regression model to estimate GNSS and LiDAR uncertainties, the proposed method ensures that the estimation results closely align with the actual sensor uncertainties. For diverse geographical scenarios, a 3DMA-based sky visibility estimation algorithm is established. This algorithm utilizes the vertex coordinates of 3D building models to construct a spatial geometric model, calculates the maximum occlusion elevation angle of the target area, and enhances the accuracy of sky visibility masks. Combined with ephemeris information, it estimates sky visibility and the number of visible satellites. A KPLSR model is then developed for precise GNSS uncertainty estimation, enabling the classification of target scenarios as 'easy' or 'difficult' for spoofing. Focusing on lane-side three-dimensional building models while excluding special scenarios such as viaducts and tunnels, simulation results demonstrate that the proposed method's estimation accuracy surpasses that of traditional regression models. For varying weather conditions, the chapter establishes a LiDAR impulse response function under severe weather and constructs a LiDAR uncertainty estimation method based on B-spline regression. By simulating LiDAR point clouds under various severe weather conditions (rain, snow, fog) and estimating LiDAR uncertainty over intervals, the relationship between weather types, meteorological classes, and spoofing success rates is quantitatively analyzed to evaluate the effectiveness of spoofing. Unlike traditional continuous spoofing methods, this chapter evaluates spoofing effectiveness across various scenarios using the established model, addressing the issue of indiscriminate spoofing signal broadcasting and thereby effectively enhancing spoofing success rates.

Chapter 6

CONCLUSIONS

This dissertation focuses on the research demands for spoofing technology of AVs in complex environments. It delves into the error transfer mechanisms of spoofing attacks, establishes an analytical model based on a lightweight information filter, and identifies the primary factors contributing to state errors in MSF systems under spoofing attacks. A covert spoofing method based on a fuzzy inference model is proposed to adaptively adjust spoofing parameters in real-time, thereby enhancing the covertness of spoofing. Furthermore, a spoofing effectiveness assessment method based on sensor uncertainty estimation is introduced. By constructing a nonlinear regression model tailored to different geographical and weather scenarios, the method accurately estimates sensor uncertainties, improves the estimation accuracy of measurement sensor uncertainties across various scenarios, and categorizes target AV environments as ‘easy’ or ‘difficult’ for spoofing. This prevents indiscriminate spoofing signal broadcasting, which often results in low success rates. The findings provide a robust theoretical foundation for researching MSF system spoofing technology for AVs in complex environments.

The dissertation accomplishes the following research tasks:

(1)Investigated the state error transfer mechanism of MSF systems under spoofing attacks. For the first time, the influence of different sensor update frequencies was considered. The state error transfer mechanism of MSF systems under spoofing attacks was thoroughly examined, and an error state Kalman filter analytical model was established. Subsequently, an error transfer model based on a lightweight information filter was developed. This model simplifies the state error analytical model and clarifies its relationship with system parameters. The main factors affecting state errors under spoofing attacks were identified, including GNSS uncertainty and the update frequency ratio of different sensors, as well as the initial state uncertainty of MSF and LiDAR uncertainty. Theoretical analysis was validated with actual data. When the update frequency ratios of GNSS and LiDAR were 1, 2, 5, and 10, respectively, the MSF system could be successfully spoofed within a 10-second attack window if the GNSS uncertainty was less than 4m, 2.7m, 1.1m, and 0.7m, respectively. This confirmed that the update frequency ratio of GNSS and LiDAR,

along with GNSS uncertainty, are key factors influencing spoofing success rates. The error transfer model established in this section is more comprehensive compared to traditional models.

(2) Researched a covert spoofing method based on a fuzzy inference model to ensure spoofing success rates while enhancing covertness. By real-time monitoring of the target AV, a position error feedback adjustment factor was calculated to quantify the difficulty of spoofing. A fuzzy knowledge base and rule base were constructed based on the feedback adjustment factor, and a fuzzy inference model was built using the multiple Zadeh method to adjust spoofing parameters and improve success rates dynamically. The magnitude of the feedback adjustment factor was compared to determine whether the spoofing process triggered the take-over effect. When activation was detected, the maximum value of spoofing parameters was constrained to prevent detection by the MSF system, thereby enhancing covertness and reducing detection risks. Real-world data simulation tests verified the effectiveness of the proposed covert spoofing method. The results indicated that the proposed method improves spoofing success rates by over 5% compared to traditional approaches.

(3) Developed a spoofing effectiveness assessment method based on sensor uncertainty estimation. GNSS and LiDAR uncertainties were estimated using a nonlinear regression model to evaluate spoofing effectiveness and determine spoofing difficulty across different scenarios. For various geographical scenarios, a 3DMA-based sky visibility estimation algorithm was established. This algorithm constructs a spatial geometric model using the vertex coordinates of 3D building models and calculates the maximum occlusion elevation angle of the target area. Sky visibility and visible satellites in the target region were estimated using current ephemeris information. Correlation analysis results demonstrated the superiority of the 3DMA-based sky visibility estimation algorithm over the traditional LPH method. Subsequently, a KPLSR model was established for accurate estimation of GNSS uncertainty. Vehicle data simulation test results indicated that spoofing success rates in ‘easy’ scenarios exceeded 70%, outperforming other traditional methods. To address the challenge of LiDAR uncertainty estimation under varying meteorological conditions, a LiDAR impulse response function was established for different weather scenarios, and a LiDAR uncertainty estimation method based on B-spline regression was developed. Real-world data simulation tests were used to simulate LiDAR point clouds under various weather conditions (rain, snow, fog). By estimating the average LiDAR uncertainty over intervals, the relationship between weather types, meteorological classes, and spoofing success rates was quantitatively analyzed to assess spoofing effectiveness. The results showed that under different weather scenarios, when scenarios were classified as ‘easy’

based on uncertainty estimation methods, the spoofing success rate exceeded 57%, significantly higher than in ‘difficult’ scenarios.

The main innovative work is as follows:

(1) Established an MSF system state error transfer model based on a lightweight information filter. The error transfer characteristics of the MSF system under spoofing attacks were deeply analyzed, and an analytical model based on a lightweight information filter was developed. This model decouples the state errors caused by spoofing attacks, avoids complex inverse operations during information vector updates, optimizes the recursive update process of INS, and elucidates the mathematical relationship between state errors and system parameters. It also reveals the influence mechanism of the update frequency ratio of measurement sensors and uncertainty on spoofing attacks.

(2) Proposed a covert spoofing method based on a fuzzy inference model. A position error feedback factor computational model under spoofing attacks was established, and a multidimensional fuzzy inference model for spoofing parameters was constructed using a fuzzy knowledge base and rule base. The mapping relationship between the position feedback factor and the rate of change of spoofing parameters was revealed. The amplitude of spoofing parameters was constrained using maximum value techniques to enable adaptive adjustment of spoofing parameters. Under all typical test scenarios, the proposed method improved spoofing success rates by over 5% compared to traditional methods.

(3) Constructed a spoofing effectiveness assessment method based on sensor uncertainty estimation. A 3DMA-based estimation method for sky visibility and visible satellites was established, and a KPLSR-based GNSS uncertainty estimation algorithm was developed. The relationship between meteorological parameters and NDT matching accuracy was analyzed, and a LiDAR uncertainty estimation algorithm based on B-spline regression was constructed. This method evaluates spoofing effectiveness across different scenarios and addresses the issue of traditional methods where indiscriminate spoofing signal broadcasting leads to low success rates.

As the study progresses, the following issues remain to be further studied and explored:

(1) In terms of spoofing targets and error mechanism modeling, investigate spoofing technology for LiDAR and SINS. Additionally, consider scenarios where the MSF system of AVs incorporates other navigation sensors, such as visual sensors, odometers, magnetometers, etc. Conduct in-depth research on the impact of various spoofing attack methods on sensor data and establish precise error mechanism models.

(2) Regarding the optimization and improvement of spoofing technology, conduct in-depth research on the latest generation mechanisms and detection methods of spoofing technology and understand their working principles and limitations. Combining the latest spoofing technology generation mechanisms and detection methods, develop more efficient and covert spoofing technology based on an understanding of spoofing and defense mechanisms. For new spoofing detection and suppression algorithms designed for AVs, explore how to design more aggressive spoofing algorithms to counter these advanced defense measures.

(3) Quantitatively analyzing the impact of both non-building static occluders (e.g., trees, billboards) and dynamic occluders (e.g., buses, large trucks) on spoofing attack performance. This will involve developing enhanced environmental models and correction mechanisms to further refine the assessment of spoofing effectiveness in complex urban scenarios.

(4) Due to testing limitations, this dissertation primarily employs real-world data simulation tests for algorithm verification. Simulation-generated spoofing signals are used to attack the MSF system, with spoofing position increments superimposed on real GNSS position outputs to simulate GNSS spoofing signal generation. Subsequent work will gradually incorporate actual spoofing tests to validate the effectiveness of the proposed scheme further.

Reference

[1] Rigas Themistoklis Ioannides, Thomas Pany, and Glen Gibbons. Known vulnerabilities of global navigation satellite systems, status, and potential mitigation techniques. *Proceedings of the IEEE*, 104(6):1174–1194, 2016.

[2] Beomju Shin, Minhuck Park, Sanghoon Jeon, Hyoungmin So, Gapjin Kim, and Changdon Kee. Spoofing attack results determination in code domain using a spoofing process equation. *Sensors*, 19(2):293, 2019.

[3] Junqi Zhang, Shaoyin Cheng, Linqing Hu, Jie Zhang, Chengyu Shi, Xingshuo Han, Tianwei Zhang, Yueqiang Cheng, and Weiming Zhang. The ghost navigator: Revisiting the hidden vulnerability of localization in autonomous driving.

[4] Shengyu Li, Shiwen Wang, Yuxuan Zhou, Zhiheng Shen, and Xingxing Li. Tightly coupled integration of gnss, ins, and lidar for vehicle navigation in urban environments. *IEEE Internet of Things Journal*, 9(24):24721–24735, 2022.

[5] Le Chang, Xiaoji Niu, Tianyi Liu, Jian Tang, and Chuang Qian. Gnss/ins/lidar-slam integrated navigation system based on graph optimization. *Remote Sensing*, 11(9):1009, 2019.

[6] Alexander Carballo, Abraham Monrroy, David Wong, Patiphon Narksri, Jacob Lambert, Yuki Kitsukawa, Eijiro Takeuchi, Shinpei Kato, and Kazuya Takeda. Characterization of multiple 3d lidars for localization and mapping performance using the ndt algorithm. In *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pages 327–334. IEEE, 2021.

[7] Feng Huang, Weisong Wen, Guohao Zhang, Dongzhe Su, and Li-Ta Hsu. Adaptive multi-sensor integrated navigation system aided by continuous error map from rsu for autonomous vehicles in urban areas. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 5895–5902. IEEE, 2023.

[8] Jorrit J Olthuis, Savio Sciancalepore, and Nicola Zannone. Cyberattacks and defenses for autonomous navigation systems: A systematic literature review. *Computer Networks*, page 111331, 2025.

[9] Linqing Hu, Junqi Zhang, Jie Zhang, Shaoyin Cheng, Yuyi Wang, Weiming Zhang, and Nenghai Yu. Security analysis and adaptive false data injection against multi-sensor fusion localization for autonomous driving. *Information Fusion*, 117:102822, 2025.

[10] Henglai Wei, Baichuan Lou, Zezhong Zhang, Bohang Liang, Fei-Yue Wang, and Chen Lv. Autonomous navigation for evtol: Review and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 9(2):4145–4171, 2024.

[11] Weisong Wen, Xiwei Bai, Yin Chiu Kan, and Li-Ta Hsu. Tightly coupled GNSS/INS integration via factor graph and aided by fish-eye camera. *IEEE Transactions on Vehicular Technology*, 68(11):10651–10662, 2019.

[12] Le Chang, Xiaoji Niu, Tianyi Liu, Jian Tang, and Chuang Qian. GNSS/INS/LiDAR-SLAM integrated navigation system based on graph optimization. *Remote Sensing*, 11(9):1009, 2019.

[13] Weisong Wen, Tim Pfeifer, Xiwei Bai, and Li-Ta Hsu. Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter. *NAVIGATION, Journal of the Institute of Navigation*, 68(2):315–331, 2021.

[14] Jonathon S Gipson and Robert C Leishman. Resilience for Multi-filter All-source Navigation Framework with Integrity. 2021.

[15] Juan D Jurado and John F Raquet. Autonomous and resilient management of all-source sensors. In *Proceedings of the ION 2019 Pacific PNT Meeting*, pages 142–159, 2019.

[16] Qian Meng and Li-Ta Hsu. Integrity Monitoring for All-Source Navigation Enhanced by Kalman Filter-Based Solution Separation. *IEEE Sensors Journal*, 21(14):15469–15484, 2020.

[17] Guohao Zhang and Li-Ta Hsu. Intelligent GNSS/INS integrated navigation system for a commercial UAV flight control system. *Aerospace science and technology*, 80:368–380, 2018.

[18] Manuel Mar, Vishnu Chellapandi, Liangqi Yuan, Ziran Wang, and Eric Dietz. A review of full-sized autonomous racing vehicle sensor architecture. *arXiv preprint arXiv:2402.02603*, 2024.

[19] Hee-Yang Jung, Dong-Hee Paek, and Seung-Hyun Kong. Open-source autonomous driving software platforms: Comparison of autoware and apollo. *arXiv preprint arXiv:2501.18942*, 2025.

[20] Shinpei Kato, Eijiro Takeuchi, Yoshio Ishiguro, Yoshiki Ninomiya, Kazuya Takeda, and Tsuyoshi Hamada. An open approach to autonomous vehicles. *IEEE Micro*, 35(6):60–68, 2015.

[21] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296. IEEE, 2018.

[22] Hiroyuki Chishiro, Kazutoshi Suito, Tsutomu Ito, Seiya Maeda, Takuya Azumi, Kenji Funaoka, and Shinpei Kato. Towards heterogeneous computing platforms for autonomous driving. In *2019 IEEE International Conference on Embedded Software and Systems (ICESS)*, pages 1–8. IEEE, 2019.

[23] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *arXiv preprint arXiv:1807.08048*, 2018.

[24] Jiaxuan Xu, Qi Luo, Kecheng Xu, Xiangquan Xiao, Siyang Yu, Jiangtao Hu, Jinghao Miao, and Jingao Wang. An automated learning-based procedure for large-scale vehicle dynamics modeling on baidu apollo platform. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5049–5056. IEEE, 2019.

[25] Zixuan Zou, Guoshuai Wang, Zhenshuo Li, Rui Zhai, and Yonghua Li. Mfo-fusion: A multi-frame residual-based factor graph optimization for gnss/ins/lidar fusion in challenging gnss environments. *Remote Sensing*, 16(17):3114, 2024.

[26] Chongcong Xu, Fangfang Zheng, and Ge Guo. Stealthy false data injection attacks in multi-channel vehicular communication: White-box and gray-box strategies. *IEEE Transactions on Vehicular Technology*, 2025.

[27] Ziheng Zhou, Hong Li, Yimin Deng, and Mingquan Lu. Clock drift monitoring based gnss spoofing detection method for autonomous vehicles. In *Proceedings of the 37th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2024)*, pages 333–345, 2024.

[28] Junbing Cheng, Yunfei Gao, Hongru Wang, Wen Ma, and Jie Wu. Vision-assisted gnss/ins high precision positioning method based on adaptive maximum correntropy criterion in urban traffic environment. *Measurement*, 245:116667, 2025.

[29] Chengzhong Zhang, Dingjie Wang, and Jie Wu. Two-dimensional directions determination for gnss spoofing source based on mems-based dual-gnss/ins integration. *Remote Sensing*, 16(23):4568, 2024.

[30] Ziheng Zhou, Hong Li, and Mingquan Lu. Doppler-based raim for gnss spoofing detection in vehicular applications. *IEEE Transactions on Vehicular Technology*, 2025.

[31] Mahsa Foruhandeh, Hanchao Yang, Xiang Cheng, Angelos Stavrou, Haining Wang, and Yaling Yang. All in one: Improving gps accuracy and security via crowdsourcing. *Computer Networks*, 254:110775, 2024.

[32] Siqi Wang, Jiang Liu, Bai-Gen Cai, Jian Wang, and De-Biao Lu. Gnss spoofing detection and elimination for resilient train positioning using spiking neural network and compressed sensing. In *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2172–2178. IEEE, 2024.

[33] Na Xia, Jiacheng Li, Huazheng Du, and Shuhao Jing. A new framework for gnss spoofing detection based on signal phase and structure. In *2024 4th International Conference on Electronic Information Engineering and Computer Communication (EIECC)*, pages 1395–1399. IEEE, 2024.

[34] Manuel Mar, Vishnu Pandi Chellapandi, Liangqi Yuan, Ziran Wang, and Eric Dietz. Advanced sensor configurations for high-speed autonomous racing vehicles. *IEEE Journal of Selected Areas in Sensors*, 2025.

[35] 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.

[36] Junjie Shen, Jun Yeon Won, Zeyuan Chen, and Qi Alfred Chen. Attacking multi-sensor fusion based localization in high-level autonomous driving. In *2021 IEEE Security and Privacy Workshops (SPW)*, pages 242–242. IEEE, 2021.

[37] Kenneth Mills, Fauzia Ahmad, Moeness G Amin, and Braham Himed. Fast iterative interpolated beamforming for interference doa estimation in gnss receivers using fully augmentable arrays. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–5. IEEE, 2019.

[38] QD Chen, HH Tao, R Liu, and WM Zhen. Time difference estimation method for tdoa location of gnss weak interference. *J. China Acad. Electron. Inf. Technol*, 15(2):135–140, 2020.

[39] R Liu et al. Centroid localization method of jamming source based on gnss receiver c/n0 weighting and jamming signal propagation error correction. *Syst. Eng. Electron.*, 43(8):2083–2089, 2021.

[40] Polona Pavlovčič-Prešeren, Franc Dimc, and Matej Bažec. Exploiting the sensitivity of dual-frequency smartphones and gnss geodetic receivers for jammer localization. *Remote Sensing*, 15(4):1157, 2023.

[41] Minghan Zhong, Xiaoming Zhang, Weiyu Gao, Mingquan Lu, and Hong Li. Prototype development of a flexible covert spoofer using measurement information from lidar and ahrs. In *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*, pages 3580–3591, 2022.

[42] Yan Guo, Meiping Wu, Kanghua Tang, Junbo Tie, and Xian Li. Covert spoofing algorithm of uav based on gps/ins-integrated navigation. *IEEE Transactions on Vehicular Technology*, 68(7):6557–6564, 2019.

[43] Xingshou Geng, Kanghua Tang, Yan Guo, Lu Zhang, Wenqi Wu, and Tiefeng Ma. A distributed uav swarm countermeasure method based on gnss spoofing. *IEEE Internet of Things Journal*, 2025.

[44] Xingshou Geng, Yan Guo, Kanghua Tang, Wenqi Wu, and Yanchao Ren. Research on covert directional spoofing method for ins/gnss loosely integrated navigation. *IEEE Transactions on Vehicular Technology*, 72(5):5654–5663, 2022.

[45] Xiaomeng Ma, Meiguo Gao, Yangguang Zhao, and Mohan Yu. A novel navigation spoofing algorithm for uav based on gps/ins-integrated navigation. *IEEE Transactions on Vehicular Technology*, 73(10):15424–15439, 2024.

[46] Petter Solnør, Øystein Volden, Kristoffer Gryte, Slobodan Petrovic, and Thor I Fossen. Hijacking of unmanned surface vehicles: A demonstration of attacks and countermeasures in the field. *Journal of Field Robotics*, 39(5):631–649, 2022.

[47] Xiaoqin Jin, Xiaoyu Zhang, Shoupeng Li, and Shuaiyong Zheng. Detection of slowly varying spoofing using weighted kalman gain in gnss/ins tightly coupled systems. *GPS Solutions*, 28(1):54, 2024.

[48] Sashank Narain, Aanjhan Ranganathan, and Guevara Noubir. Security of gps/ins based on-road location tracking systems. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 587–601. IEEE, 2019.

[49] Çağatay Tanıl, Samer Khanafseh, Mathieu Joerger, and Boris Pervan. An ins monitor to detect gnss spoofers capable of tracking vehicle position. *IEEE Transactions on Aerospace and Electronic Systems*, 54(1):131–143, 2017.

[50] Xingshou Geng, Yan Guo, Kanghua Tang, and Wenqi Wu. A spoofing algorithm for ground unmanned platform equipped with gnss/ins integrated navigation system. *IEEE Transactions on Instrumentation and Measurement*, 2024.

[51] Yoshiyasu Takefuji. Connected vehicle security vulnerabilities [commentary]. *IEEE Technology and Society Magazine*, 37(1):15–18, 2018.

[52] J Rossouw Van Der Merwe, Xabier Zubizarreta, Ivana Lukčin, Alexander Rügamer, and Wolfgang Felber. Classification of spoofing attack types. In *2018 European Navigation Conference (ENC)*, pages 91–99. IEEE, 2018.

[53] Ali Jafarnia-Jahromi, Ali Broumandan, John Nielsen, and Gérard Lachapelle. Gps vulnerability to spoofing threats and a review of antispoofing techniques. *International Journal of Navigation and Observation*, 2012(1):127072, 2012.

[54] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[55] Kecheng Xu, Xiangquan Xiao, Jinghao Miao, and Qi Luo. Data driven prediction architecture for autonomous driving and its application on apollo platform. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 175–181. IEEE, 2020.

[56] Luciano Baresi and Damian A Tamburri. Architecting artificial intelligence for autonomous cars: The openpilot framework. In *European Conference on Software Architecture*, pages 189–204. Springer, 2023.

[57] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[58] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, et al. Lgsvl

simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*, pages 1–6. IEEE, 2020.

[59] Ruoyu Song, Muslum Ozgur Ozmen, Hyungsub Kim, Raymond Muller, Z Berkay Celik, and Antonio Bianchi. Discovering adversarial driving maneuvers against autonomous vehicles. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2957–2974, 2023.

[60] Abdul Malik Khan, Naveed Iqbal, and Muhammad Faisal Khan. Synthetic gnss spoofing data generation using field recorded signals. *MethodsX*, 5:1272–1280, 2018.

[61] Andreas Finkenzeller, Anshu Mathur, Jan Lauinger, Mohammad Hamad, and Sebastian Steinhorst. Simutack-an attack simulation framework for connected and autonomous vehicles. In *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pages 1–7. IEEE, 2023.

[62] Andrew J Kerns, Daniel P Shepard, Jahshan A Bhatti, and Todd E Humphreys. Unmanned aircraft capture and control via gps spoofing. *Journal of field robotics*, 31(4):617–636, 2014.

[63] Tobias Bamberg, Manuel M Appel, and Michael Meurer. Which gnss tracking loop configuration is most robust against spoofing? In *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, pages 3587–3595, 2018.

[64] Chenxi Peng, Hong Li, and Mingquan Lu. Research on the responses of gnss tracking loop to intermediate spoofing. In *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, pages 943–952, 2019.

[65] Todd E Humphreys, Jahshan A Bhatti, Daniel Shepard, and Kyle Wesson. The texas spoofing test battery: Toward a standard for evaluating gps signal authentication techniques. 2012.

[66] G Aissou, HO Slimane, S Benouadah, and N Kaabouch. A dataset for gps spoofing detection on autonomous vehicles. *IEEE DataPort*, 2022.

[67] Todd Humphreys. Texbat data sets 7 and 8. *The University of Texas*, 2016.

[68] Austin Albright, Sarah Powers, Jason Bonior, and Frank Combs. A tool for furthering gnss security research: The oak ridge spoofing and interference test battery

(oakbat). In *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pages 3697–3712, 2020.

[69] David Gómez-Casco, Paolo Crosta, and Mariana Spangenberg. Validation of evil waveforms in a gnss simulator for gps and galileo signals. In *2022 10th Workshop on Satellite Navigation Technology (NAVITEC)*, pages 1–7. IEEE, 2022.

[70] Muwahida Liaquat, Mohammad Zahidul H Bhuiyan, Saiful Islam, Into Pääkkönen, and Sanna Kaasalainen. An enhanced fgi-gsrx software-defined receiver for the execution of long datasets. *Sensors*, 24(12):4015, 2024.

[71] Meng Zhou, Hong Li, and Mingquan Lu. The modeling and analysis for the assessment of gnss spoofing technology. In *China Satellite Navigation Conference (CSNC) 2013 Proceedings: BeiDou/GNSS Navigation Applications Test & Assessment Technology User Terminal Technology*, pages 627–639. Springer, 2013.

[72] Yue Wang, Fuping Sun, Jinming Hao, Lundong Zhang, and Xian Wang. Evaluation of global navigation satellite system spoofing efficacy. *Journal of Systems Engineering and Electronics*, 33(6):1238–1257, 2022.

[73] Elena Basan, Oleg Makarevich, Maria Lapina, and Massimo Mecella. Analysis of the impact of a gps spoofing attack on a uav. In *CEUR Workshop Proceedings*, volume 3094, pages 6–16, 2022.

[74] Mark Hunter, Francesca Fillipi, and Guy Buesnel. An assessment of gnss receiver behaviour in laboratory conditions when subject to gps meaconing or spoofing scenarios. In *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pages 1533–1555, 2020.

[75] SVSLN Surya Suhas Vaddhiparthy, Garapati Sreya, Prudhvi Raj Turlapati, Deepak Gangadharan, and Harikumar Kandath. A comprehensive evaluation on the impact of various spoofing scenarios on gps sensors in a low-cost uav. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pages 1–6. IEEE, 2023.

[76] Petter Solnør. Applications of cryptographic methods in feedback control. 2023.

[77] 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.

[78] Gongmin Yan, Chunlian Zhao, Feng Wu, and Yong-yuan Qin. An improvement for the calibration of laser gyro strapdown IMU. In *Proceedings of the 32nd Chinese Control Conference*, pages 4861–4865. IEEE, 2013.

[79] Gongmin Yan, Xi Sun, Jun Weng, Qi Zhou, and Yongyuan Qin. Time-asynchrony identification between inertial sensors in SIMU. *Journal of Systems Engineering and Electronics*, 26(2):346–352, 2015.

[80] Gongmin Yan, Xiaokang Yang, Xingjun Su, Jun Weng, and Yongyuan Qin. Error Distribution Method and Analysis of Observability Degree Based on the Covariances in Kalman Filter. In *2018 37th Chinese Control Conference (CCC)*, pages 4900–4905. IEEE, 2018.

[81] Weisong Wen, Li-Ta Hsu, and Guohao Zhang. Performance analysis of ndt-based graph slam for autonomous vehicle in diverse typical driving scenarios of hong kong. *Sensors*, 18(11):3928, 2018.

[82] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016.

[83] Gongmin Yan, Jinling Wang, and Xinyi Zhou. High-precision simulator for strapdown inertial navigation systems based on real dynamics from GNSS and IMU integration. In *China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III*, pages 789–799. Springer, 2015.

[84] Jinyong Jeong, Younggun Cho, Young-Sik Shin, Hyunchul Roh, and Ayoung Kim. Complex urban dataset with multi-level sensors from highly diverse urban environments. *The International Journal of Robotics Research*, 38(6):642–657, 2019.

[85] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. isam: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[86] 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.

[87] Zhiwei Zhao, Guiqiang Ni, Yuanyuan Shen, and Nasruddin Hassan. Multiple multidimensional fuzzy reasoning algorithm based on neural network. *Journal of Intelligent & Fuzzy Systems*, 35(4):4121–4129, 2018.

[88] Kai Meng Tay and Chee Peng Lim. Optimization of gaussian fuzzy membership functions and evaluation of the monotonicity property of fuzzy inference systems. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 1219–1224. IEEE, 2011.

[89] Lotfi Asker Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information sciences*, 8(3):199–249, 1975.

[90] Tomoji Takasu et al. Rtklib. *Available: http:\\www. rtklib. com*, 2013.

[91] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. Lidar snowfall simulation for robust 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16364–16374, 2022.

[92] Max Jwo Lem Lee, Shang Lee, Hoi-Fung Ng, and Li-Ta Hsu. Skymask matching aided positioning using sky-pointing fisheye camera and 3d city models in urban canyons. *Sensors*, 20(17):4728, 2020.

[93] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real lidar point clouds for 3d object detection in adverse weather. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15283–15292, 2021.

[94] Ralph H Rasshofer, Martin Spies, and Hans Spies. Influences of weather phenomena on automotive laser radar systems. *Advances in radio science*, 9:49–60, 2011.

[95] S Jebson. Fact sheet number 3: Water in the atmosphere. 2007.

[96] Craig F Bohren and Donald R Huffman. *Absorption and scattering of light by small particles*. John Wiley & Sons, 2008.

[97] Matti Kutila, Pasi Pyykönen, Hanno Holzhüter, Michele Colomb, and Pierre Duthon. Automotive lidar performance verification in fog and rain. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1695–1701. IEEE, 2018.

[98] T Fahey. Snowfall rate thresholds for light, moderate and heavy. In *Information paper in the Seventh bi-annual meeting of the Aerodrome Meteorological Observation and Forecast Study Group (AMOFSG), Montreal, Canada*, 2008.

[99] KLS Gunn and JS Marshall. The distribution with size of aggregate snowflakes. *Journal of Atmospheric Sciences*, 15(5):452–461, 1958.

[100] Roman Rosipal and Leonard J Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of machine learning research*, 2(Dec):97–123, 2001.

[101] Bai Yifeng, Xiao Jian, and Yu Long. Kernel partial least-squares regression. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1231–1238. IEEE, 2006.

[102] Dhritiman Saha, T Senthilkumar, Chandra B Singh, and Annamalai Manickavasagan. Quantitative detection of metanil yellow adulteration in chickpea flour using line-scan near-infrared hyperspectral imaging with partial least square regression and one-dimensional convolutional neural network. *Journal of Food Composition and Analysis*, 120:105290, 2023.

[103] Sky McKinley and Megan Levine. Cubic spline interpolation. *College of the Redwoods*, 45(1):1049–1060, 1998.

[104] Thibaud Briand and Pascal Monasse. Theory and practice of image b-spline interpolation. *Image Processing On Line*, 8:99–141, 2018.

[105] Jiachen Zhang, Weisong Wen, Feng Huang, Yongliang Wang, Xiaodong Chen, and Li-Ta Hsu. Gnss-rtk adaptively integrated with lidar/imu odometry for continuously global positioning in urban canyons. *Applied Sciences*, 12(10):5193, 2022.

[106] Agustin Garcia Asuero, Ana Sayago, and AG González. The correlation coefficient: An overview. *Critical reviews in analytical chemistry*, 36(1):41–59, 2006.

[107] Rubao Ma, Weichao Xu, Qinruo Wang, and Wei Chen. Robustness analysis of three classical correlation coefficients under contaminated gaussian model. *Signal Processing*, 104:51–58, 2014.