Random Spoofing Attack against LiDAR-Based Scan Matching SLAM

Masashi Fukunaga
Mitsubishi Electric Corporation
Fukunaga.Masashi@ce.MitsubishiElectric.co.jp

Takeshi Sugawara
The University of Electro-Communications
sugawara@uec.ac.jp

Abstract*—Integrity of sensor measurement is crucial for safe and reliable autonomous driving, and researchers are actively studying physical-world injection attacks against light detection and ranging (LiDAR). Conventional work focused on object/obstacle detectors, and its impact on LiDAR-based simultaneous localization and mapping (SLAM) has been an open research problem. Addressing the issue, we evaluate the robustness of a scan-matching SLAM algorithm in the simulation environment based on the attacker capability characterized by indoor and outdoor physical experiments. Our attack is based on Sato et al.'s asynchronous random spoofing attack that penetrates randomization countermeasures in modern LiDARs. The attack is effective with fake points injected behind the victim vehicle and potentially evades detection-based countermeasures working within the range of object detectors. We discover that mapping is susceptible toward the z-axis, the direction perpendicular to the ground, because feature points are scarce either in the sky or on the road. The attack results in significant changes in the map, such as a downhill converted into an uphill. The false map induces errors to the self-position estimation on the x-y plane in each frame, which accumulates over time. In our experiment, after making laser injection for 5 meters (i.e. 1 second), the victim SLAM's self-position begins and continues to diverge from the reality, resulting in the 5m shift to the right after running 125 meters. The false map and self-position significantly affect the motion planning algorithm, too; the planned trajectory changes by 3° with which the victim vehicle will enter the opposite lane after running 35 meters. Finally, we discuss possible mitigations against the proposed attack.

I. Introduction

Since autonomous vehicles (AVs) recognize the world using multiple sensors and make critical decisions, the integrity of sensor measurement is essential for safe AV driving. Consequently, signal injection attacks that compromise the integrity of sensor measurement in the physical domain have become an active research area in the last few years.

Localization and mapping are crucial components of AV motion planning. Dynamically updating the map is important to keep track of the constantly changing world, and light detection and ranging (LiDAR) that measures 3D point cloud is commonly used for the purpose. The global positioning system (GPS) and the inertial measurement unit (IMU) are common for localization (i.e., odometry), but they are not always available; IMUs suffer from error accumulation and GPS are unreliable when tall buildings block a line of sight to the satellite in an urban area. To address the problem, researchers are studying the method for achieving localization using a LiDAR only, which is called LiDAR-based simultaneous localization and mapping (SLAM) [1]–[4]. Scan matching is a common LiDAR-based SLAM and achieves localization using a relative displacement vector obtained by comparing successive LiDAR frames [4]. The vector is then matched with a 3D map using feature points, such as a contour or surface, to update the map.

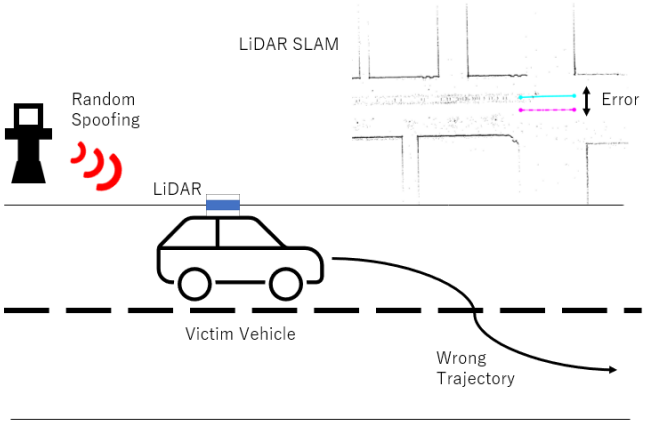

Fig. 1. Overview of the proposed random spoofing attack on LiDAR-Based SLAM. An attacker on the roadside induces random fake points in the victim LiDAR with asynchronous laser illumination. The victim SLAM build a wrong 3D map with the fake points, which results in a wrong trajectory plan. The victim car plans a trajectory that goes across a lane over 35 meters with our 1-second laser-injection experiment.

Meanwhile, researchers are studying signal injection attacks that manipulate sensor measurement in the physical domain [5], and there are several attacks on LiDARs [6]–[8]. LiDARs typically use time-of-flight (ToF) for distance measurement; they send a laser pulse toward the target, measure the time delay until receiving an echo, and translate it into a distance. Exploiting this principle, the attacker sends fake laser pulses that a victim LiDAR recognizes as genuine echoes, thereby adding fake 3D points in the scene. In particular, some previous attacks exploited the predictable pulse timing

Symposium on Vehicles Security and Privacy (VehicleSec) 2024
26 February 2024, San Diego, CA, USA
ISBN 979-8-9894372-7-6
https://dx.doi.org/10.14722/vehiclesec.2024.23014
www.ndss-symposium.org

to inject a fake object with an arbitrary shape [9] or even remove the existing points [10]. More recent LiDARs have countermeasures, such as timing randomization, that make pulse timing unpredictable, efficiently thwarting such a precise injection [11]. Sato et al. took another approach of injecting many random points, i.e., *random spoofing*, and showed that it is sufficient to attack machine learning models [11]. Meanwhile, those conventional LiDAR attacks focused on object detection, and its impact on SLAM has been an open research problem.

A. Research Question

We study the impact of random spoofing attack on scan-matching SLAM algorithms. Failure in either localization or mapping can have serious consequences, such as making a dangerous maneuver or inappropriate acceleration/deceleration. Meanwhile, scan-matching SLAM algorithms may have inherent robustness against attacks because they can reject random fake points during the match. Therefore, we tackle the following research questions in this paper: **(i)** *Can an attacker manipulate the scan-matching SLAM algorithms with random spoofing?* **(ii)** *If yes, what the implications of such attacks on AV motion planning?*

B. Contributions

We approach the research questions through a series of real-world experiments and simulations. This paper provides the following key contributions.

1) Characterization of attacker capability (Section IV): We begin by characterizing the attacker capability on random spoofing with a series of real-world experiments. In the indoor moving-target experiment, we can inject 100 points/frame while tracking a target vehicle moving at 1 m/s from 5 meters away. We also conduct an outdoor long-range experiment to characterize the number of fake points over distance, showing that random spoofing succeeds from 40 meters away, but the number of fake points decreases exponentially with distance.

2) Impact on mapping (Section V): The feasibility of the attack is verified in the MATLAB/Simulink platform that simulates an AV driving in the urban area with an attacker with the capability characterized by the indoor and outdoor physical experiments. We discover that the mapping is susceptible to attack toward the z-axis, the vertical direction perpendicular to the ground. Unlike the remaining x- and y-axes, the z-axis has few features that the scan matching algorithm can rely on because there are few points in the sky and the points on the road are removed with preprocessing. Laser injection successfully changes the map by 5° along the z-axis, converting a downhill into an uphill.

3) Impact on motion planning (Section VI): We further evaluate the impact of laser injection on localization and motion planning. The false map causes a small error in the estimation of x-y plane self-position in each frame, which accumulates over time. As a result, the victim SLAM's self-position shifts by 5 meters to the right over 125 meters after getting laser injection for 5 meters. A motion planning algorithm (the `plannerPRT` motion planner in the MATLAB Navigation toolbox) is also affected by the false map and self-position, and the planned trajectory changes by 3° with laser injection; the victim vehicle will enter the opposite lane after running 35 meters with this angle.

4) Defense (Section VII): Finally, we discuss possible mitigations while highlighting the limitations of the proposed attack. We suggest that sensor fusion, extended anomaly detection, and hardware defenses can be effective countermeasures.

II. Preliminaries

A. Sensors in Autonomous Vehicle

Automated driving refers to the use of a control system to perform safety driving evaluations previously performed by human drivers. Autonomous driving is classified into five levels by human intervention. Levels 1 and 2 are foot-free and hands-free driving and are categorized as driver assistance. Levels 3 and Level 4 are eye-free and driver-free driving and cover automated driving under specific conditions. Level 5 is fully automated driving [12]. AVs recognize surrounding conditions using multiple sensors, including LiDAR, camera, radar, GPS, and IMU. Sensor-based tasks include object detection, trajectory planning, localization, and mapping. The integrity of sensor data is essential for autonomous driving safety.

B. LiDAR-based perception

LiDAR measures objects in the 3D environment as a collection of dots called point cloud. LiDARs typically use ToF, which emits a laser pulse and measures the time until receiving reflected light, i.e. echoes. Velodyne VLP-16 [13] is a popular ToF LiDAR used in previous works. VLP-16 has a mechanically spinning head that scans the scene for 360° and 100 meters. The head covers the vertical angle of $\pm 15^\circ$ using 16 stack of lasers. VLP-16 emits a series of pulses periodically as it rotates, and this predictability of pulse timing has been exploited in the previous works [6], [8], [14], [15]. More recent LiDARs have countermeasures, such as timing randomization, that make the pulse timing unpredictable for attackers [16].

C. Previous Attacks on LiDARs

Petit et al. demonstrated the first signal injection attack on LiDAR that injects a fake replica of a genuine object with a relay attack in 2015 [6]. Then Shin et al. improved the attack to inject fake points instead of a replica in 2017 [7]. Then, many researchers followed this direction and proposed several attacks and evaluated their impacts on later-stage applications [8], [17], [18].

One research direction is to control fake points more precisely. In particular, PLA-LiDAR [9] injected arbitrary-shaped point cloud by carefully scheduling the injection timing. Meanwhile, Physical Removal Attack (PRA) removes part of the genuine point cloud by injecting fake points within the minimum operational threshold of a target LiDAR [10].

These precise attacks assume predictable pulse timing and are no longer possible with LiDARs with unpredictable pulse

timing [16]. The attacker can still inject fake points, but the coordinates of the points becomes uncontrollable without synchronization. In this paper, we call this type of injection *random spoofing*. Sato et al. discovered that such random spoofing efficiently breaks object classifiers and proposed High-Frequency Removal (HFR) that injects many random points [11].

Another line of research is an adversarial example [19] against a 3D point cloud classification model [20], [21]. An adversary can manipulate the point cloud to add, delete, or move points in the whitebox setting because an adversary can compute adversarial point clouds against the autonomous vehicle's 3D point cloud classifier [8]. Additionally, the adversary can create a 3D object that is recognized as an adversarial point cloud [14], [22].

D. SLAM: Simultaneous Localization and Mapping

SLAM updates a map in real time and projects its own position on the map. Real-time mapping and localization are crucial for keeping track of the constantly changing world and for motion planning. GPS and IMU can be used for localization but are not always available. For example, GPS needs to see GPS satellites in the sky, which can be blocked by tall buildings or a ceiling [23]. We can keep track of the position by accumulating the information from an IMU, but it suffers from error accumulation and becomes increasingly unreliable over time [24], and a GPS is necessary to compensate for the accumulation of errors [25], [26]. For example, even a highly accurate IMU has an error of about 2 meters in 30 seconds [26]. Such a multisensor fusion has non-trivial requirements on synchronization and calibration and cannot satisfy certain real-time requirements [27]. Consequently, researchers are seeking localization exclusively using LiDAR, i.e, with LiDAR-based SLAM [27].

Scan matching is commonly used for LiDAR-based SLAM that achieves localization using a relative displacement vector obtained by comparing successive LiDAR frames [28]. In the following, we explain the SLAM algorithm [28] used in this paper. It is an algorithm that performs registration of point clouds and map generation using those point clouds, and is used in the MATLAB environment targeted in this paper.

This SLAM uses point cloud registration and map generation to reconstruct 3D scenes and create road maps for location estimation. Point-cloud registration is the process of aligning two or more 3D point clouds of the same scene into a common coordinate system.

The workflow for map generation and location estimation is executed according to the following steps.

1) Point cloud preprocessing: To prepare the point cloud for registration, it is down-sampled to remove unnecessary features and noise.

2) Point cloud registration: Each point cloud is registered against the previous point cloud. This is the process of accumulating registration estimates across consecutive frames. These registrations are used in odometry. Using odometry alone can cause a drift between measured and ground-truth attitudes.

3) Loop detection: Loop closure detection is used to minimize drift. Loop closure detection is the process of identifying the sensor's return to a previously accessed position, forming a loop in the sensor's trajectory.

4) Drift correction: The detected loops are used to minimize drift by optimizing the attitude graph. Attitude graph optimization is the incremental construction of the attitude graph by adding nodes and edges, and optimizing the attitude graph once sufficient loops are found. Optimization of the posture graph yields an optimized set of absolute postures.

5) Map assembly: The point cloud map is assembled by aligning the registered point clouds using the optimized absolute posture. Such a pre-built point cloud map can be used for position estimation, the process of locating vehicles in the map.

6) Position estimation: Based on the assembled map, the vehicle's attitude is determined.

III. Threat Model

A. Attack Scenario

The attacker is motivated to induce false information by injecting a laser into the SLAM that works in the target vehicle. The attacker targets SLAM to avoid detection-based countermeasures working within the range of object detection; as we will show later, SLAM is affected by fake points on the behind of the car, which are irrelevant for common object detectors. The attacker's ultimate goal is to let the victim make wrong decisions, such as dangerous maneuvers or inappropriate acceleration/deceleration, through false localization and/or mapping.

B. Target

The victim vehicle is fully autonomous using sensor data based on international standards [29], [30]. The target vehicle uses LiDAR-based scan-matching SLAM to achieve mapping and localization, which is then used for autonomous driving decisions, including motion planning. We further assume that the target vehicle relies on LiDAR SLAM for localization possibly because the vehicle is in the region where neither IMU nor GPS is available, as discussed in Section II-D.

C. Attacker Capability

The attacker is at a distance from the victim, for example, on the side of the road, but has a line of sight to the victim and can continuously illuminate the target with a laser. This can be achieved with a turret that has the ability to track the victim's LiDAR. The attacker has information about the target vehicle and knows the specification of the LiDAR, including its position and wavelength. This can be easily accomplished by studying publicly available data sheets or purchasing the same model. These conditions are similar to those of previous works [17].

We restrict the attacker's capability to random spoofing only. This represents the case where the victim LiDAR has

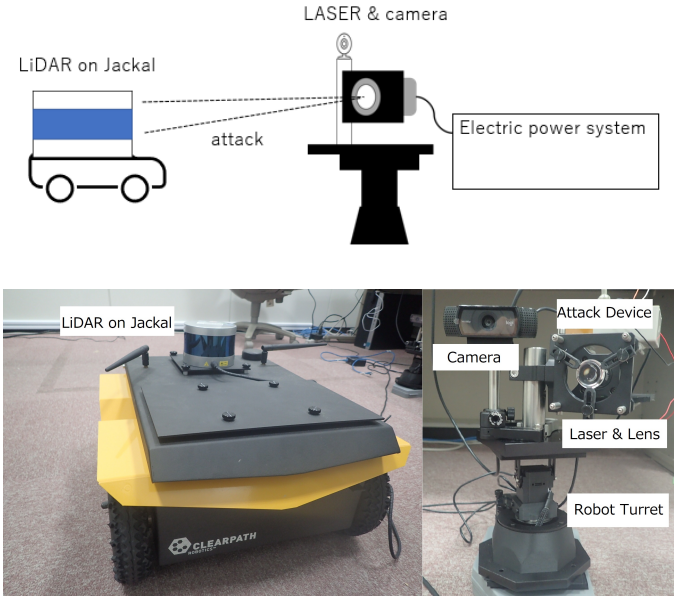

Fig. 2. The setup for indoor evaluation. (Top) A diagram of the entire setup. (Left bottom) The victim LiDAR mounted on an UGV. (Right bottom) The laser, optics, and camera mounted on a pan-tilt turret for aiming.

a countermeasure [11], e.g., the timing of the laser pulse is unpredictable because of timing randomization, as discussed in Section II. We will verify the attacker's capability regarding the number of fake points with experiments in Section IV. Furthermore, we assume that the target LiDAR has a detection-based countermeasure with respect to object detection [31], and random spoofing is limited to the region outside object detection; otherwise, the attack is detected.

IV. Characterization of Random Spoofing

We begin by characterizing the attacker capability on attacking moving targets from long distances with two real-world experiments to fill the gaps in the previous work [11]. The first indoor experiment verifies the feasibility of random spoofing, similar to HFR, on the moving target with motion tracking. The second outdoor experiment characterizes the number of injected points over long distances, i.e., up to 40 meters in contrast to the previous work limited to 15 meters.

A. Indoor Experiments with Tracking

The first experiment verifies the asynchronous random spoofing attack while tracking a moving target.

1) Setup: Fig. 2 shows the diagram and pictures of our indoor setup. The victim LiDAR (VLP-16 from Velodyne [13]) is mounted on an unmanned ground vehicle (UGV; Jackal UGV from Clearpath Robotics [32]) that moves at ∼0.1 m/s during the experiments[1]. The spoofer comprises the optical and tracking systems. The spoofer uses a 903 nm infrared laser diode (SPL PL90 from OSRAM [33]) and a 15mm lens

[1]UGV's speed is limited to meet our safety standard.

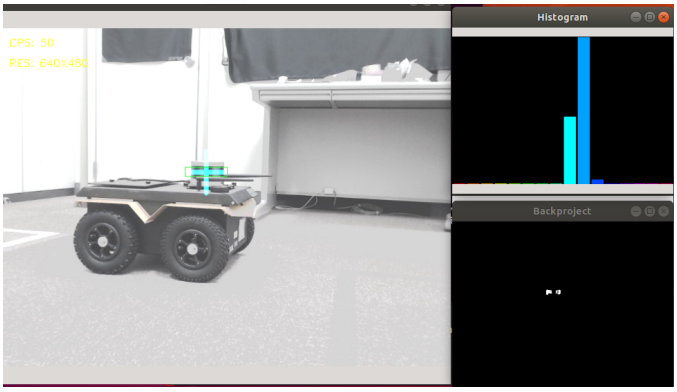

Fig. 3. View from the motion tracker during the moving-target tracking experiment. (Left) Image from the camera capturing the LiDAR on UGV. (Right top) Color histogram used for target detection. (Right bottom) Back projection showing the detected object.

(LA1540 - N-BK7 from Thorlabs [34]) to make a collimated laser beam, assembled on an optical breadboard. We use a laser driver (PCO-7114-50-4 from Directed Energy Inc. [35]) to generate laser pulses from the laser diode. Its operating voltage is 60 volts to satisfy our facility's safety criteria. The pulse repetition frequency is 1 MHz, following the previous study [14][2].

We use a turret (PhantomX XL430 Robot Turret IL-PXT-X from Trossen Robotics [36]) to aim and track the laser beam at the target LiDAR. A computer running Ubuntu 16.04 [37] and ROS Kinetic [38] controls the turret using images from the camera (C920 from Logitech [39]). It detects LiDAR in camera images with the Camshift method [40] that uses color. Then, it moves the turret to keep the target LiDAR in the center of the camera image. Fig. 3 shows the images from the ROS program that reliably detect and follow the LiDAR because its color is significant from the background colors.

2) Result: The setup successfully injects fake points while the target is moving, as shown in Fig. 4. The attack angle is narrower than the conventional work [6] because we use a collimated (cf. focusing) beam for a successful attack over a long distance, which results in a weak light intensity at the target. The results show that the attacker can continuously inject fake points even when the laser injection angle changes as the target moves, confirming the previous LiDAR attack with tracking [10].

B. Outdoor Experiment for Characterizing Distances.

Our attack assumes consistent laser illumination over a certain period of time. The attacker-victim distance can change during the attack and affect the number of fake points. We characterize this relationship between distance and the number of fake points with an outdoor experiment.

1) Model: We assume that the laser power attenuates exponentially with distance r. In the range of a few tens of meters, the light can be approximated as an ideal collimated

[2]The largest number of points are injected at 1 MHz in a preliminary experiment, confirming the result in the previous works [11].

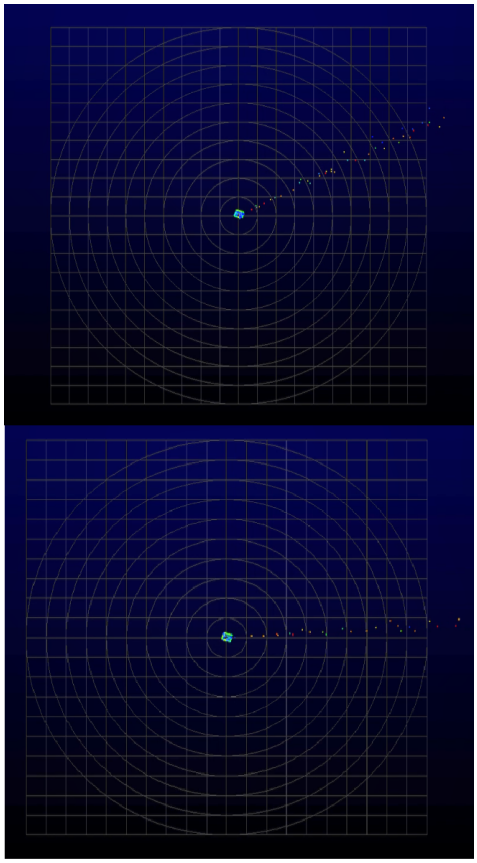

Fig. 4. LiDAR view during the indoor moving-target experiment in two different moments ((top) and (bottom)). The square object on the center is our laboratory room. The fake points distribute on a narrow cone toward the direction of the injected laser beam, as a result of random spoofing.

light, wherein scattering by atmospheric particles dominates the attenuation. Consequently, the light intensity decreases exponentially with distance, following the Lambert-Beer law with atmospheric transmittance as a coefficient. We further assume that the number of fake points is proportional to the laser power. Then, the number of injected points is modeled as

$$P(r) = P_0 \cdot e^{-ar} \ , \tag{1}$$

wherein P_0 and a are the initial the attenuation factors, respectively. We are going to verify the model and determine the constants P_0 and a with the following experiment.

2) Setup: We evaluate the number of injected points over a long distance using the setup in Fig. 5 where the LiDAR and the attack device are placed on the tables. We count the number of fake points while changing the distance between the spoofer and the LiDAR for 5, 10, 20, and 40 meters. Fig. 6 shows a 3D point cloud under laser injection, and the fake points are indicated with a white ellipse. Knowing the positions of the spoofer and LiDAR, we count the number of points distributed between the spoofer and the LiDAR. We repeat the counting 10 times for each distance to get the mean and standard deviation.

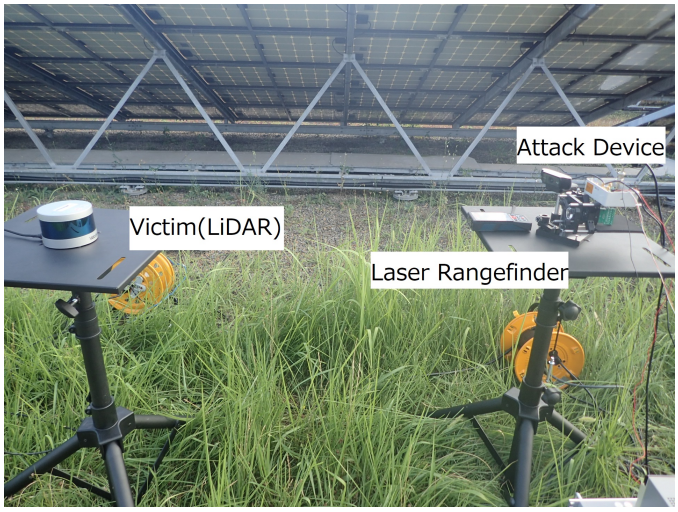

Fig. 5. Setup for outdoor evaluation. The LiDAR and the laser turret are placed on optics are placed on the tables. used. The distance between the LiDAR and the attack device is separated from each other while the distance is measured with a laser rangefinder. In this experiment, the change in the number of point cloud injections during the experiment was measured.

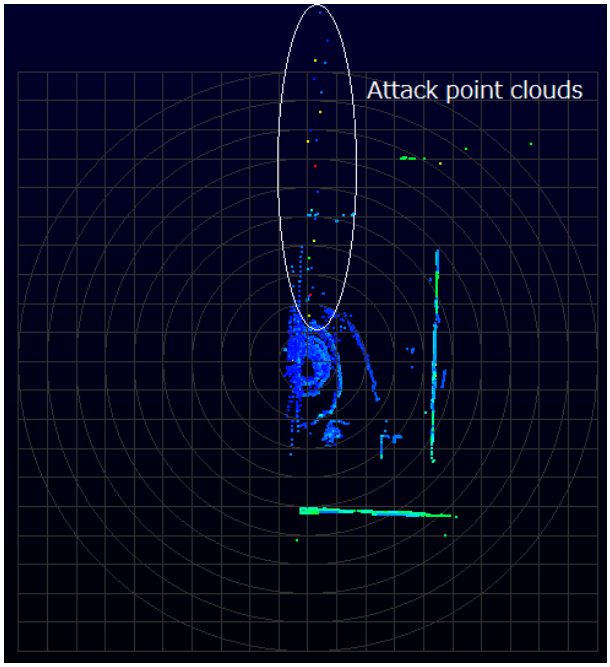

Fig. 6. LiDAR view during the outdoor long-range experiment with laser injection from 10 meters. The injected point cloud is highlighted with the ellipse.

3) Result: The graph in Fig. 7 summarizes the experimental results, showing the relationship between the number of fake points (vertical axis) and the distance (horizontal axis). The model in Eq. (1) greatly described the results, as shown with the fitted curve in Fig. 7, confirming that the number of injected points decreases exponentially with distance. The initial and attenuation factors are $P_0 = 66.7$ and $a = -0.063$ in the fitted curve, which are used in our simulation.

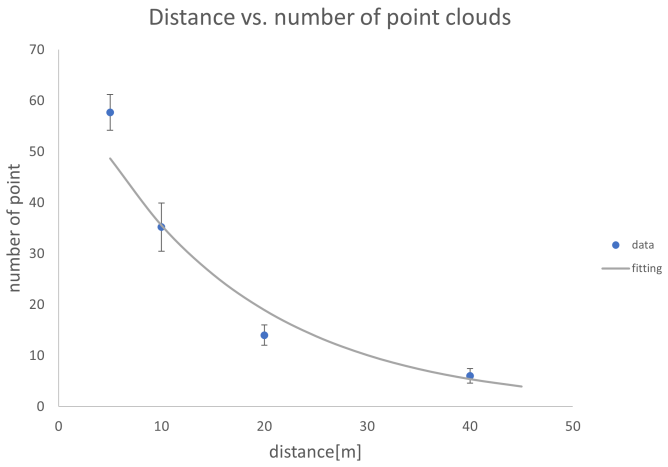

Fig. 7. Relationship between the distance and the number of fake points, based on measurements from outdoor experiments. The gray line is the fitted model with Eq. 1 ($P_0 = 66.7$ and $a = 0.063$). The number of injected points decreases exponentially with distance.

TABLE I
PARAMETERS FOR THE MATLAB/SIMULINK LIDAR MODEL CONFIGURED FOR VELODYNE VLP-16.

Parameter	Values used simulation
Vertical field of view	30°
Vertical resolution	0.2°
Horizontal field of view	360°
Horizontal resolution	0.3°
Detection Range	100 m
Range resolution	0.01 m

V. IMPACT ON MAPPING

We evaluate the impact of random spoofing on localization and mapping with a series of simulations.

A. Simulation Environment

Our evaluation platform is MATLAB/Simulink with the Navigation toolbox [41] that provides the target SLAM algorithm, motion planning, and sensor models. The LiDAR model in the Navigation toolbox has six configurable parameters, and we determine the values based on VLP-16's specification [13], as shown in Table I. The LiDAR model rotates 15 times per second and captures 1,200 points per rotation. The vertical resolution is 16. GPS is not available and the victim vehicle relies on the LiDAR for localization, as discussed in Section III-B.

The target is Point Cloud SLAM [28] in the Navigation toolbox based on a scan matching algorithm. Its processing pipeline comprises six steps, as shown in Fig. 8. Each step performs the following operations:

- Downsample the point cloud to remove unwanted features and noise.
- Register the point cloud.
- Perform loop closure detection to minimize drift.
- Use the detected loops to minimize drift through pose graph optimization.

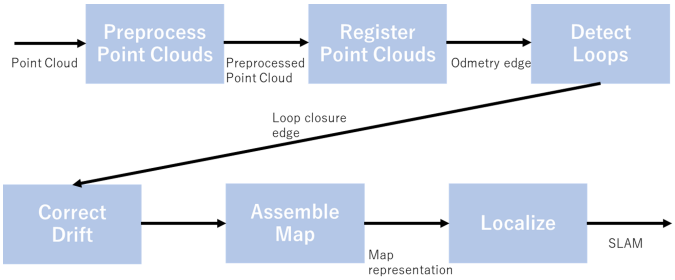

Fig. 8. Processing pipeline of the target LiDAR SLAM [28]. It uses point cloud registration and map generation to reconstruct 3D scenes and create road maps for location estimation. Point cloud registration is the process of aligning two or more 3D point clouds of the same scene into a common coordinate system.

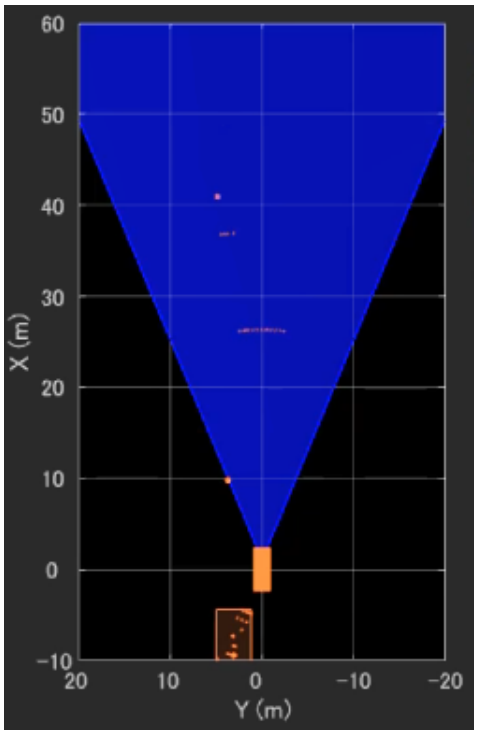

Fig. 9. The range of the LiDAR object detector. Within this range, if objects are detected that interferes with driving, the automated vehicle will safely stop.

- Assemble a point cloud map by aligning the registered point clouds using their optimized absolute poses.
- Find the pose of the vehicle based on the assembled map.

We assume that the MATLAB LiDAR object detector in the LiDAR Toolbox [42] covers $\pm 22.5°$, as shown in Fig. 9. We inject fake points outside of this object detection region to evade a detection-based countermeasure, as discussed in Section III-C.

The victim car drives at 11.1 m/s along the road toward the x direction in the urban area in Fig. 10 built with Unreal Engine 4 [43] [23]. An attacker injects fake points from a sidewalk at $(X, Y) = (0, 5)$. To model random spoofing attack, we generate random values on the line between the victim and

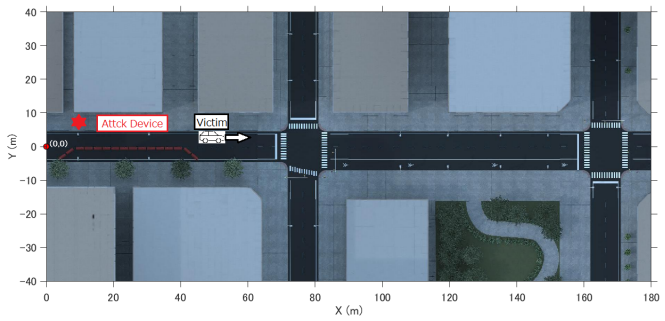

Fig. 10. The urban area scene used during simulation. The victim car travels along the road toward X direction. The attacker is on the sidewalk and inject a laser beam to the victim vehicle from behind. The red dotted line immediately after the start of the run is an obstacle.

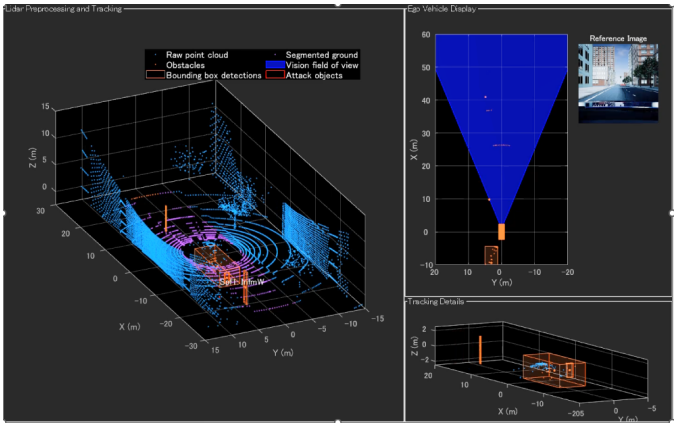

Fig. 11. View from the object detector during random spoofing attack. The object detector does not respond to the the fake points injected on the backside because they are out of the range. The fake point cloud on this figure is the points in the space enclosed by the Bounding box at the rear of the car.Automatic control by object detection is performed in the blue area (vision field of view) in the upper right graph. On the other hand, SLAM uses 360-degree point cloud data, which is the measurement area of LiDAR.

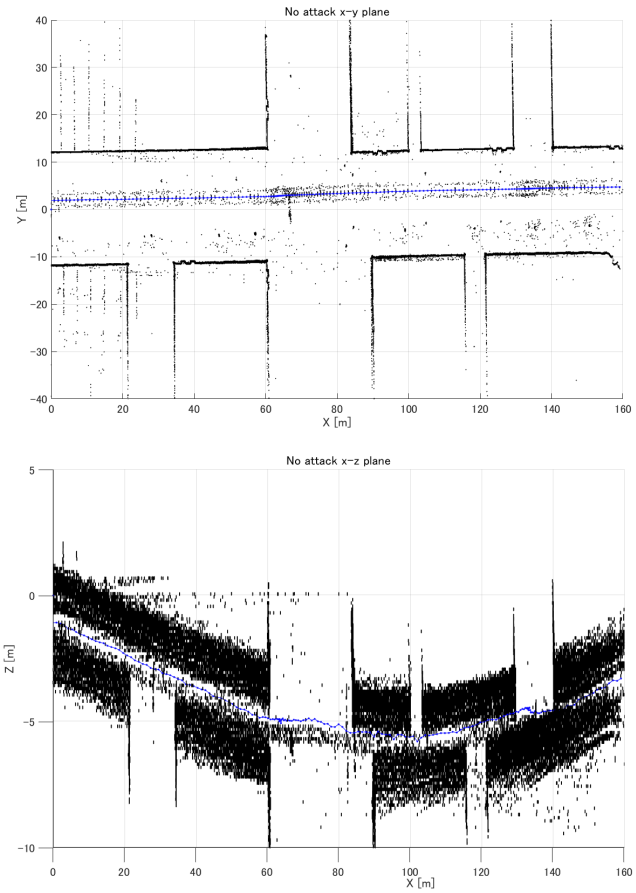

Fig. 12. SLAM map in the benign case without laser injection projected to the x-y and x-z planes. (Top) the x-y plane corresponding to the bird's-eye view showing the boundaries between the buildings and the roads. (Bottom) the x-z plane is a side view where the z-axis is perpendicular to the ground. There is a downhill from $X = 0$ to 100 and an uphill from $X = 100$ to 160. The blue lines represent the transition of self-position. The SLAM map contains building outlines, gradients and self-positions.

the attacker and put them in the `pointCloud` data structure. Fig. 11 shows a 3D scene with injected fake points.

The attacker injects fake points behind the victim to avoid the detection-based countermeasure. The attacker starts fake point injection when the victim car is at $X = 10$ and continues to inject fake points for the next Δ meters. Here, we examine different duration of attack, i.e., $\Delta \in \{5, 10, 20, 40\}$. Finally, we evaluate the map generated by the SLAM algorithm when the car reaches $X = 160$ meters.

B. Baseline Measurement without Laser Injection

Fig. 12 shows a SLAM map in the benign case without laser injection projected to the x-y and x-z planes. The blue lines represent the self-position recognized by the SLAM algorithm. The x-y plane is the bird's eye view showing the boundary between the buildings and the roads, while the x-z plane is its side view. There is a downhill for $0 < X < 100$ with the slope angle of 5° (5m elevation over the 100 meters). The road then changes to uphill for $100 < X < 160$ with a slope angle of 5° (3m elevation over 60 meters).

C. Laser Injection without Attenuation

We first consider an ideal attack with no laser attenuation. Figs. 13 show the same projected SLAM maps with attack durations $\Delta = 5$, 10, 20, and 40 meters. Although the x-y plane is mostly unaffected, the attack causes significant changes in the x-z plane. In all the cases, the the x-z plane shows uphill for $0 < X < 160$, unlike the benign case in Fig. 12. The impact on the map becomes more significant as attack duration increases, and the map with $\Delta = 40$ has the strongest distortion from the benign case.

The results can be explained as follows. The x-y plane is robust against fake points because there are many landmarks and features that the scan matching algorithm can rely on. It is not the case with the x-z plane, because there are few points in the sky and the ground components are down sampled. As a result, the scan matching algorithm fails to correct the errors induced by laser injection, which accumulate over time.

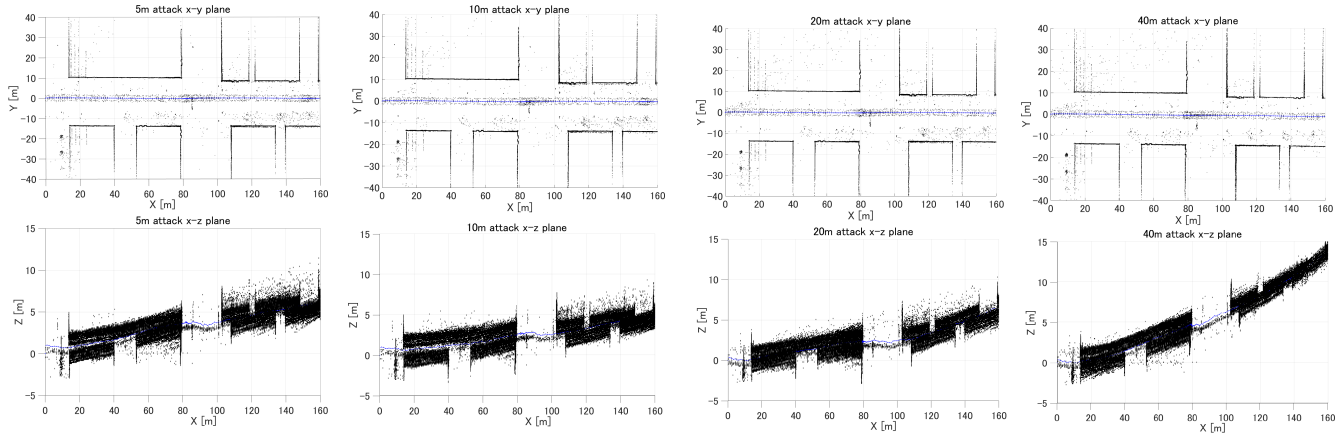

Fig. 13. These figures are cross sections of the attacked SLAM map in the x-y and x-z planes.From left to right, they correspond to attacks at $\Delta = 5$, 10, 20, and 40 meters.

D. Laser Injection with Attenuation

We repeat the previous experiment considering the laser attenuation modeled in Eq. (1) and the concrete parameters ($P_0 = 66.7$ and $a = 0.063$) characterized in Section IV-B. Specifically, attenuation was added to the transformation of random spoofing points in the simulation. The greater the distance between the attacker and the LiDAR, the greater the attenuation.

Fig. 14 shows the the x-z projection of the SLAM maps for the attack durations $\Delta = 5$, 10, 20, and 40 meters, and the attack still succeeds by changing the slope. The changes toward the z-axis is relatively small compared with the previous results in Fig. 13, but this is reasonable considering the reduced number of fake points as a result of laser attenuation.

Laser injection has a significant effect on the z-axis. In the benign case, the road is 5° downhill for $0 < X < 100$ and changes to 5° uphill for $100 < X < 160$, as discussed in Section V-B. A long duration of attack has more impacts on the SLAM map. In the result with $\Delta = 5$, for example, the road is mostly flat. In the result with $\Delta = 40$, on the other hand, the road becomes uphill with a slope angle of 5° for $0 < X < 160$. Changes in the slope angle can affect AV driving decisions, as we will evaluate in the next section.

The results also show that the impact of the attack lasts for a while. In our evaluation, we inject fake points for $10 < X < 10+\Delta$, and the SLAM maps in Fig. 14 are obtained when the car is at $X = 160$. This means that the benign measurements in $10 + \Delta < X < 160$ are insufficient to correct the changed maps.

VI. Impacts on Localization and Motion Planning

The results in the previous section show that the attacker can effectively change the SLAM map on the z-axis by laser injection, and we evaluate its impact on localization and motion planning.

We extend the simulation setup in Section V-A with motion planning. The target motion planner is `plannerPRT` [44] in the MATLAB Navigation toolbox [41] that makes geometric planning based on the rapidly-exploring random tree (RRT) [45], which generates search trees in steps using random samples from a particular state space. `plannerPRT` receives a map and generates a trajectory as a motion plan.

We compare the trajectories generated from the SLAM maps with and without laser injection. We evaluate the most rigorous case with the minimum attack duration $\Delta = 5$ meters; we place a spoofer at $X = 0$ and inject a laser while the victim car is in $25 < X < 30$. The motion planner generates trajectories for $X > 160$ using the SLAM maps obtained when the target vehicle is at $X = 160$.

Fig. 15 shows the localization and motion plan with and without attack in the x-y plane. Solid lines are benign cases; the blue solid line ($X < 160$) is the self-position deduced from the SLAM algorithm, and the cyan solid line ($X > 160$) is the trajectory generated by the motion planner. Dashed lines represent the attack cases; the red dashed line ($X < 160$) is the self-position and the magenta dashed line ($X > 160$) is the trajectory. Fig. 15 shows that the attack significantly affects the self-position in the x-y plane, although the impact on mapping is limited to the z-axis (see Fig. 14).

The attack effect on localization is not immediate, and the self-positions (the blue and red lines) begin to split after the injection has finished at $X = 35$. This result suggests that the false map causes a small error in the self-position estimation in each frame, which accumulates over time. As a result, the false self-position that target SLAM recognizes exceeds the lane around $X = 100$, and the self-position is shifted by 5 meters in the Ydirection at $X = 160$.

The false map and self-position force the motion planner to make a wrong trajectory for $X > 160$, as shown with the cyan and magenta lines in Fig. 15. The vehicle direction changes by 3° towards the opposite lane as a result of the laser injection; the victim vehicle will enter the opposite lane after running 35 meters with this motion plan, which can cause a serious traffic accident. Note that the target motion planner does not recognize a white line as an object in making a

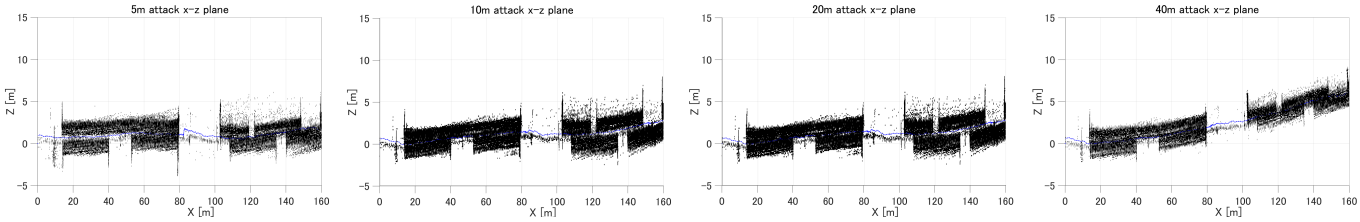

Fig. 14. This is the result of an attack simulation incorporating real-world constraints. The x-z planes of the SLAM maps generated when the attack sections are 5 m, 10 m, 20 m, and 40 m, respectively. The impact of the attack is smaller than before incorporating the constraints, but the attack is successful to the extent that a downhill is mistaken for an uphill.

trajectory, which is one reason behind the above bad decision. Meanwhile, considering a white line can cause another serious problem when the victim tries to keep a lane using a false self-position that continues to deviate from the reality.

VII. Defenses

We discuss possible defenses and mitigations, which also highlights the limitations of the proposed attack.

A. Using secondary sensors

Using secondary sensors that are unaffected by lasers is a promising countermeasure approach. GPS and IMU are popular and already deployed in the field, but are not always available, as discussed in Section II-D. Another candidate is a tilt sensor that measures the slope angle, which is also common in vehicles. Since the proposed attack mainly targets the z-axis, the 1-dimensional tilt sensor is sufficient to detect anomalies or compensate for errors. Visual SLAM algorithms using cameras can be an alternative solution. Since cameras have their own weaknesses [46], combining a LiDAR with cameras can improve the robustness as a system.

B. 360° anomaly detection.

Anomaly detection is a common countermeasure approach considered in previous attacks on LiDAR-based object detection [22]. Detecting random spoofing is relatively easy because the injected points are uncontrollable and distinguishing random points from benign objects is straightforward. Our attack potentially evades anomaly detection regarding object detectors by injecting fake points outside the range of an object detector, e.g., behind the victim vehicle. In other words, the proposed attack will be detectable by conventional anomaly detectors by extending its coverage to 360°.

C. Hardware defenses

Making the LiDAR hardware robust against laser injection attacks will also solve the SLAM problems. Methods to change the internal structure of the sensor include filtering [46], and pulse randomization. In particular, some hardware countermeasures are effective against random spoofing attacks [11].

VIII. Conclusion and Future Work

This paper studies the impact of the laser injection attack on LiDAR-based SLAM. By considering a target LiDAR-based with defenses, we make a random spoofing attack that penetrates the timing randomization countermeasure outside the range of LiDAR-based object detectors in which a detection-based countermeasure is likely in place. Based on the properties of random spoofing attack characterized by indoor and outdoor physical experiments, we evaluate the robustness of LiDAR-based SLAM in the simulation environment. The z-axis in mapping, the vertical direction perpendicular to the ground, is susceptible to random spoofing because of poor existing features; there are few points in the sky and the points on the road are removed with preprocessing. As a result, the map is significantly affected towards the z-axis, e.g., converting a downhill into an uphill.

The false map causes a small error in the estimation of self-position in the x-y plane in each frame, which accumulates over time, which is sufficient to shift the recognized self-position by 5 meters to the right over 160 meters. The false map and self-position significantly affect the motion planning algorithm, and the planned trajectory changes by 3° at the 160m position, and the victim vehicle will enter the opposite lane after running 35 meters with this angle. These impacts on localization and motion planning can cause a serious traffic accident. Finally, we discuss possible mitigations, including sensor fusion, extended anomaly detection, and hardware defenses.

Several important questions are open for future research. Understanding the causality of the vulnerabilities discovered in the SLAM algorithm is necessary to build more robust SLAM algorithms. Our evaluation is limited to a particular scene, and verifying the attack in the other settings, including a flat area with no slope, is important to evaluate the robustness of the attack. Finally, a real-world (cf. simulation-based) end-to-end evaluation is necessary to figure out the practical impact of the attack.

References

[1] M. Aldibaja, R. Yanase, T. H. Kim, A. Kuramoto, K. Yoneda, and N. Suganuma, "Accurate elevation maps based graph-slam framework for autonomous driving," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1254–1261.

[2] X. Yang, X. Lin, W. Yao, H. Ma, J. Zheng, and B. Ma, "A robust LiDAR SLAM method for underground coal mine robot with degenerated scene compensation," *Remote Sensing*, vol. 15, no. 1, 2023.

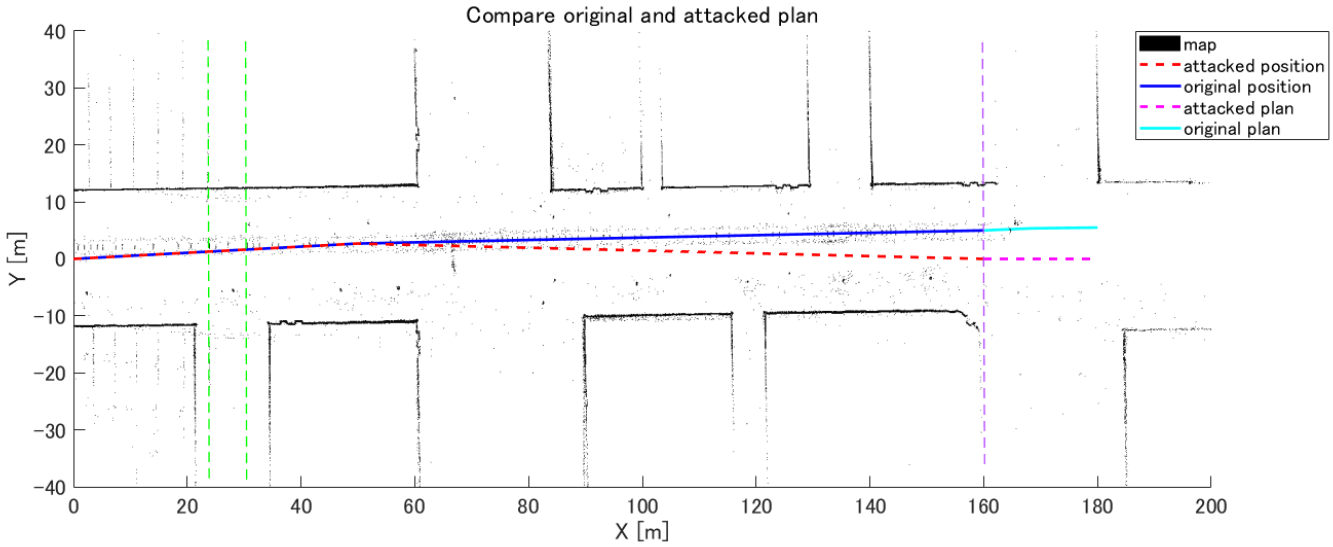

Fig. 15. This is an x-y plane view of the result of motion planning with MATLAB RRT executed on a SLAM map. Fake points are injected in $25 < X < 30$ highlighted with the green dashed lines. The trajectory plan is executed at $X = 160$. The solid blue line is the transition of the self-position before the attack. The red leaf selection is the transition of self-position after the attack. The solid cyan line is the trajectory planning executed on the SLAM map before the attack, and the magenta dashed line is the trajectory planning executed on the SLAM map after the attack. The post-attack SLAM map was used with an attack range of 5 meters. The results show that the automated system can make a trajectory plan that deviates from the lane due to the SLAM attack. The purple dashed line was drawn at x=160m. The section under attack was also enclosed by a green dashed line.

[3] X. Xu, L. Zhang, J. Yang, C. Cao, W. Wang, Y. Ran, Z. Tan, and M. Luo, "A review of multi-sensor fusion SLAM systems based on 3D LIDAR," *Remote Sensing*, vol. 14, no. 12, 2022.

[4] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems X*, 2014.

[5] K. Fu and W. Xu, "Risks of trusting the physics of sensors," *Commun. ACM*, vol. 61, no. 2, pp. 20–23, 2018.

[6] J. Petit, B. Stottelaar, and M. Feiri, "Remote attacks on automated vehicles sensors: Experiments on camera and LiDAR," in *Black Hat Europe 2015*, 2015.

[7] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against Lidars for automotive applications," in *Cryptographic Hardware and Embedded Systems 2017*, 2017, pp. 445–467.

[8] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial sensor attack on LiDAR-based perception in autonomous driving," in *Proceedings of CCS '19*, 2019, p. 2267–2281.

[9] Z. Jin, X. Ji, Y. Cheng, B. Yang, C. Yan, and W. Xu, "PLA-LiDAR: Physical laser attacks against LiDAR-based 3D object detection in autonomous vehicle," in *44th IEEE Symposium on Security and Privacy, SP 2023*, 2023, pp. 1822–1839.

[10] Y. Cao, S. H. Bhupathiraju, P. Naghavi, T. Sugawara, Z. M. Mao, and S. Rampazzi, "You can't see me: Physical removal attacks on LiDAR-based autonomous vehicles driving frameworks," in *USENIX Security 2023*. USENIX Association, 2023, pp. 2993–3010.

[11] T. Sato, Y. Hayakawa, R. Suzuki, Y. Shiiki, K. Yoshioka, and Q. A. Chen, "Revisiting LiDAR spoofing attack capabilities against object detection: Improvements, measurement, and new attack," *CoRR*, vol. abs/2303.10555, 2023.

[12] Society of Automotive Engineers, "SAE levels of driving automation," https://www.sae.org/blog/sae-j3016-update.

[13] Velodyne LiDAR Inc., "VLP-16 product," https://velodynelidar.com/products/puck/, 2019.

[14] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, "Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures," in *USENIX Security 2020*, 2020, pp. 877–894.

[15] R. S. Hallyburton, Y. Liu, Y. Cao, Z. M. Mao, and M. Pajic, "Security analysis of Camera-LiDAR fusion against Black-Box attacks on autonomous vehicles," in *USENIX Security 22*, 2022, pp. 1903–1920.

[16] K. Yoshioka, "A tutorial and review of automobile direct ToF LiDAR SoCs: Evolution of next-generation LiDARs," *IEICE Transactions on Electronics*, p. 2021CTI0002, 2022.

[17] S. H. V. Bhupathiraju, J. Sheldon, L. A. Bauer, V. Bindschaedler, T. Sugawara, and S. Rampazzi, "EMI-LiDAR: Uncovering vulnerabilities of LiDAR sensors in autonomous driving setting using electromagnetic interference," in *Proceedings of WiSec '23*, 2023, p. 329–340.

[18] K. Yoshida, M. Hojo, and T. Fujino, "Adversarial scan attack against scan matching algorithm for pose estimation in LiDAR-based SLAM," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E105.A, 10 2021.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014.

[20] D. Liu, R. Yu, and H. Su, "Extending adversarial attacks and defenses to deep 3D point cloud classifiers," 2019.

[21] K. Lee, Z. Chen, X. Yan, R. Urtasun, and E. Yumer, "ShapeAdv: Generating shape-aware adversarial 3D point clouds," 2020.

[22] Y. Cao, C. Xiao, D. Yang, J. Fang, R. Yang, M. Liu, and B. Li, "Adversarial objects against LiDAR-based autonomous driving systems," *CoRR*, vol. abs/1907.05418, 2019.

[23] C. Chen, J. Ibanez-Guzman, and O. Le-Marchand, "Low-cost loosely-coupled gps/odometer fusion: A pattern recognition aided approach," 08 2008, pp. 1 – 6.

[24] W. S. F. Iv, J. H. Wall, and D. M. Bevly, "Characterization of various IMU error sources and the effect on navigation performance," 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:16816374

[25] C. Raveena, R. Sravya, R. Kumar, and A. Chavan, "Sensor fusion module using imu and gps sensors for autonomous car," in *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 2020, pp. 1–6.

[26] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects," *Information Fusion*, vol. 7, no. 2, pp. 221–230, 2006.

[27] H. Wang, C. Wang, and L. Xie, "Lightweight 3-d localization and mapping for solid-state LiDAR," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1801–1807, 2021.

[28] MathWorks, "Perform SLAM using 3-D Lidar

point clouds," https://www.mathworks.com/help/nav/ug/perform-lidar-slam-using-3d-lidar-point-clouds.html, 2023.
[29] ISO, "ISO 34502," https://www.iso.org/standard/78951.html, 2022.
[30] ——, "ISO 3691-4," https://www.iso.org/standard/83545.html, 2023.
[31] R. Sahba, A. Sahba, M. Jamshidi, and P. Rad, "3D object detection based on LiDAR data," 10 2019, pp. 0511–0514.
[32] CLEARPATH, "Jackal," https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/, 2019.
[33] OSRAM, "SPL PL90," https://ams-osram.com/products/lasers/ir-lasers-eel/osram-radial-t1-34-spl-pl90-3, 2023.
[34] Thorlabs, "LA1540 - N-BK7," https://www.thorlabs.co.jp/thorProduct.cfm?partNumber=LA1540, 2000.
[35] Directed Energy Inc., "PCO-7114-50-4," https://directedenergy.com/product/pco-7114-50-4/, 2018.
[36] Trossen Robotics, "PhantomX XL430 pan and tilt," https://www.trossenrobotics.com/phantomx-x-series-robot-turret.aspx, 2018.
[37] Ubuntu, "Ubuntu 16.04.7 lts (xenial xerus)," https://releases.ubuntu.com/16.04/, 2018.
[38] Open Robotics, "ROS kinetic," http://wiki.ros.org/kinetic, 2016.
[39] logitech, "C920 HD PRO WEBCAM," https://www.logitech.com/en-gb/products/webcams/c920-pro-hd-webcam.960-001055.html, 2020.
[40] OpenCV, "Meanshift and Camshift," https://docs.opencv.org/4.x/d7/d00/tutorial_meanshift.html, 2023.
[41] MathWorks, "Navigation toolbox," https://www.mathworks.com/help/nav/index.html, 2023.
[42] ——, "Lidar toolbox," https://www.mathworks.com/help/lidar/index.html?s_tid=CRUX_lftnav, 2023.
[43] Epic Games, "Unreal Engine," https://www.unrealengine.com/en-US/eula/unreal, 2018.
[44] MathWorks, "plannerRRT," https://www.mathworks.com/help/nav/ref/plannerrrt.html, 2019.
[45] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, 1999, pp. 473–479 vol.1.
[46] B. G. Stottelaar, "Practical cyber-attacks on autonomous vehicles," February 2015. [Online]. Available: http://essay.utwente.nl/66766/