CARMEN: Center for Automated Vehicle Research with Multimodal Assurred Navigation

A USDOT Tier-1 University Transportation Center

THE OHIO STATE UNIVERSITY
COLLEGE OF ENGINEERING

UCI University of California, Irvine

TEXAS
The University of Texas at Austin

University of CINCINNATI

Resilience and Validation of GNSS PNT Solutions

Todd Humphreys (https://orcid.org/0000-0003-0749-6988)

Qi Alfred Chen (https://orcid.org/0000-0003-0316-9285)
Umit Ozguner (https://orcid.org/0000-0003-2241-7547)
Charles Toth (https://orcid.org/0000-0001-9461-4887)

FINAL RESEARCH REPORT - November 20, 2023

Contract # 69A3552047138

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. This report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. The U.S. Government assumes no liability for the contents or use thereof.

Final CARMEN Report for Project 5: Resilience and Validation of GNSS PNT Solutions

Table of Contents

1 Introduction

Highly automated transportation systems rely on a steady stream of signals and information from external sources for localization, route planning, perception, and general situational awareness. This includes reliance on positioning, navigation, and timing (PNT) information: Location is essential autonomous navigation and planning; and accurate timing is a precondition for on-board sensor fusion, cooperative control, and management based on information from other vehicles or the infrastructure. It is crucial to identify schemes for GNSS signal authentication and resilience that are well-suited for highly autonomous vehicles (HAVs). HAVs require PVT sensing techniques that are resilient to unusual natural or accidental events and secure against deliberate attack.

GNSS will no doubt play a significant role in PNT for HAVs, as GNSS is the only positioning system that offers absolutely-referenced meter-level accuracy with global coverage and all-weather operation. Furthermore, carrier-phase differential GNSS (CDGNSS), whose real-time variant for mobile platforms is commonly known as real-time kinematic (RTK) GNSS, is a centimeter-accurate positioning technique that differences a receiver's GNSS observables with those from a nearby fixed reference station to eliminate most sources of measurement error. The trouble is that GNSS is fragile: the harsh multipath and signal blockage conditions of the urban ground vehicle environment often results in degraded position estimation. Furthermore, GNSS is susceptible to deliberate attack, as its service is easily denied jammers, or deceived by spoofers.

There are two core defense strategies against GNSS interference: (1) GNSS hardening and (2) GNSS augmentation. The idea of GNSS hardening is toughen the GNSS receiver against interference specific to GNSS. At the core of GNSS hardened systems is inertial navigation, which is a ubiquitous pairing alongside GNSS receivers. Inertial measurement units (IMU) are impervious to RF interference and signal blockage. Tightly coupling GNSS receivers with IMUs enable precise navigation in challenging multipath environments and provide a powerful defense against spoofing.

The next strategy for PNT resiliency are augmentations to GNSS. Traditional GNSS have been brilliantly successful, yet for some applications they remain inadequate with regard to accuracy, constellation survivability, or robustness to interference—for both civil and military users. To address these limitations, several alternative augmentation systems have been investigated such as: (1) Vision-based; (2) Radar-based; (3) TRNS; (4) Communication systems; (5) LEO PNT; and (6) Signals of Opportunity. These alternate sensors can be either coupled with GNSS, or operate as stand-alone PNT solutions in GNSS-denied environments, providing yet another layer of security.

Under the CARMEN UTC, there have been significant advancements in PNT resiliency and security. The four key areas that have seen tremendous progress are: (1) Low-Cost Ground Vehicle Anti-Spoofing, (2) Physics-Based Anomaly Detection (PBAD), (3) Machine Learning PNT Model Security, and (4) Receiver Data Validation by PNT Solutions Obtained from Other Sources.

We developed, implemented, and validated a powerful single-antenna carrier-phase-based test to detect GNSS spoofing attacks on ground vehicles equipped with a low-cost IMU. This technique exploits road roughness, MEMS IMUs, and carrier phase GNSS to achieve low-cost anti-GNSS-spoofing appropriate for mass-market ground vehicles. The effectiveness of the developed spoofing detection method was evaluated with data captured by a vehicle-mounted sensor suite in Austin, Texas. Artificial worst-case spoofing attacks injected into the dataset were detected within two seconds.

We evaluated existing cyber-layer defense PBAD strategies and developed new ones for PNT threats to HAVs by (1) identifying promising data sources in HAVs (e.g., sensing data and related actuators) that can potentially correlate with the PNT inputs, (2) understanding their correlations and constructing physical invariants, (3) designing online anomaly detection algorithms corresponding to the identified physical invariants, and (4) evaluating the performance in real-world HAVs.

Some PNT sensors used in HAVs (e.g., LiDARs and cameras) predominately rely on deep neural networks (DNNs). We performed risk analysis and identify mitigation methods for security issues at the DNN model level. We explored domain-specific adaptions of existing defense strategies in other application domains to the DNN models used in PNT algorithms and leveraged the insights to design effective solutions specific to PNT threat scenarios. We quantified safety and mobility risks from real-world or simulation-based experiments under DNN-based PNT model attacks, and design and implementation of mitigation methods specific to DNN models such as model architecture changes and adding model input sanitization.

Sensors, such as radar, LiDAR, vision camera, etc., on an HAV provide data in real-time. In contrast, an HD map represents sensor data acquired earlier and structured for some purpose. Connected HAVs may share both their PNT solutions and sensor data. The signals used to observe the environment and then create a PNT solution may come from infrastructure dedicated to positioning, such as GPS or from other sources of natural and man-made ones. Examples of man-made SoPs include cellular and LEO satellites. Natural signals include Earth gravity, magnetic field, and surface-based signals used in terrain-based navigation. We combined both the theoretical and experimental results into a performance assessment/validation document.

2 Low-Cost Ground Vehicle Anti-Spoofing

2a Carrier-phase and IMU based GNSS Spoofing Detection for Ground Vehicles

ABSTRACT

This paper develops, implements, and validates a powerful single-antenna carrier-phase-based test to detect Global Navigation Satellite Systems (GNSS) spoofing attacks on ground vehicles equipped with a low-cost inertial measurement unit (IMU). Increasingly-automated ground vehicles require precise positioning that is resilient to unusual natural or accidental events and secure against deliberate attack. This paper's spoofing detection technique capitalizes on the carrier-phase fixed-ambiguity residual cost produced by a well-calibrated carrier-phase-differential GNSS (CDGNSS) estimator that is tightly coupled with a low-cost IMU. The carrier-phase fixed-ambiguity residual cost is sensitive at the sub-centimeter-level to discrepancies between measured carrier phase values and the values predicted by prior measurements and by the dynamics model, which is based on IMU measurements and on vehicle constraints. Such discrepancies will arise in a spoofing attack due to the attacker's practical inability to predict the centimeter-amplitude vehicle movement caused by roadway irregularities. The effectiveness of the developed spoofing detection method is evaluated with data captured by a vehicle-mounted sensor suite in Austin, Texas. The dataset includes both consumer- and industrial-grade IMU data and a diverse set of multipath environments (open sky, shallow urban, and deep urban). Artificial worst-case spoofing attacks injected into the dataset are detected within two seconds.

INTRODUCTION

The combination of easily-accessible low-cost Global Navigation Satellite System (GNSS) spoofers and the emergence of increasingly-automated GNSS-reliant ground vehicles prompts a need for fast and reliable GNSS spoofing detection [1], [2]. To underscore this point, Regulus Cyber recently spoofed a Telsa Model 3 on autopilot mode, causing the vehicle to suddenly slow and unexpectedly veer off the main road [3].

Among GNSS signal authentication techniques, signal-quality-monitoring (SQM) and multi-antenna could be considered for implementation on ground vehicles [4]. However, SQM tends to perform poorly on dynamic platforms in urban areas where strong multipath and in-band noise are common [4]–[7], and multi-antenna spoofing detection techniques, while effective [8], [9], are disfavored by automotive manufacturers seeking to reduce vehicle cost and aerodynamic drag. Thus, there is a need for a single-antenna GNSS spoofing detection technique that performs well on ground vehicles despite the adverse signal-propagation conditions in an urban environment.

In a concurrent trend, increasingly-automated ground vehicles demand ever-stricter lateral positioning to ensure safety of operation. An influential recent study calls for lateral positioning better than 20 cm on freeways and better than 10 cm on local streets (both at 95%) [10]. Such stringent requirements can be met by referencing lidar and camera measurements to a local high-definition map [11], [12], but poor weather (heavy rain, dense fog, or snowy whiteout) can render this technique unavailable [13]. On the other hand, recent progress in precise (dm-level) GNSS-based ground vehicle positioning, which is impervious to poor weather, has demonstrated surprisingly high (above 97%) solution availability in urban areas [14]. This technique is based on carrier-phase differential GNSS (CDGNSS) positioning, which exploits GNSS carrier phase measurements having mm-level precision but integer-wavelength ambiguities [15].

Key to the promising results in [14] is the tight coupling of CDGNSS and IMU measurements, without which high-accuracy CDGNSS solution availability is significantly reduced due to pervasive signal blockage and multipath in urban areas (compare the improved performance of [14] relative to [16]). Tight coupling brings mm-precise GNSS carrier phase measurements into correspondence with high-sensitivity and high-frequency inertial sensing. The particular estimation architecture of [14] incorporates inertial sensing via model replacement, in which the estimator's propagation step relies on bias-compensated acceleration and angular rate measurements from the IMU instead of a vehicle dynamics model. As a consequence, at each measurement update, an *a priori* antenna position is available whose delta from the previous measurement update accounts for all vehicle motion sensed by the IMU, including small-amplitude high-frequency motion caused by road irregularities. Remarkably, when tracking authentic GNSS signals in a clean (open sky) environment, the GNSS carrier phase predicted by the *a priori* antenna position and the actual measured carrier phase agree to within millimeters.

This paper pursues a novel GNSS spoofing detection technique based on a simple but consequential observation: it is practically impossible for a spoofer to create a false ensemble of GNSS signals whose carrier phase variations, when received through the antenna of a target ground vehicle, track the phase values predicted by inertial sensing. In other words, antenna motion caused by road irregularities, or rapid braking, steering, etc., is sensed with high fidelity by an onboard IMU but is unpredictable at the sub-cm-level by a would-be spoofer. Therefore, the differences between IMU-predicted and measured carrier phase values offer the basis for an exquisitely sensitive GNSS spoofing detection statistic. What is more, such carrier phase fixed-ambiguity residual cost is generated as a by-product of tightly-coupled inertial-CDGNSS vehicle position estimation such as performed in [14].

Two difficulties complicate the use of fixed-ambiguity residual cost for spoofing detection. First is the integer-ambiguous nature of the carrier phase measurement [15], which causes the post-integer-fix residual cost to equal not the difference between the measured and predicted carrier phase, as would be the case for a typical residual, but rather this difference modulo an integer number of carrier wavelengths. Such integer folding complicates development of a probability distribution for a detection test statistic based on carrier phase fixed-ambiguity residual cost.

Second, the severe signal multipath conditions in urban areas create thick tails in any detection statistic based on carrier phase measurements. Setting a detection threshold high enough to avoid false spoofing alarms caused by mere multipath could render the detection test insensitive to dangerous forms of spoofing. Reducing false alarms by accurately modeling the effect of a particular urban multipath environment on the detection statistic would be a Sisyphean undertaking, requiring exceptionally accurate up-to-date 3D models of the urban landscape, including materials properties.

This paper takes an empirical approach to these difficulties. It does not attempt to develop a theoretical model to delineate the effects of integer folding or multipath on its proposed carrier-phase fixed-ambiguity residual cost based detection statistic. Rather, it develops null-hypothesis empirical distributions for the statistic in both shallow and deep urban areas, and uses these distributions to demonstrate that high-sensitivity spoofing detection is possible despite integer folding and urban multipath.

Related Work

The idea of using coupled GNSS and inertial sensing to detect GNSS spoofing was first explored for aviation [17]–[22]. Wind gusts and turbulence cause rapid movement of aircraft that are instantaneously reflected in calibrated inertial measurements. As with road irregularities for ground vehicles, a GNSS spoofer will find it challenging to track and replicate such movements in real-time. However, this prior work either did not exploit carrier-phase measurements or relied on a tactical-grade IMU, rendering solutions either too slow (long time-to-detect) or too expensive.

Accumulating innovation faults within a specified time window in a loosely coupled INS/GNSS Kalman filter was investigated in [23]. For a given time window, there are two ways to accumulate the slowly-drifting faults. One averages the normalized sum-squared innovations of each epoch (innovation averaging); the other averages the measurements within a time window and subsequently performs a snapshot test (measurement averaging). Innovation averaging has little effect on the Kalman filter prediction and filtering process, so it can be easier to deploy and can be designed as an add-on function. Measurement averaging requires small modifications of the Kalman filter measurement update process. The position-domain spoofing detection strategy in [23] required 15 seconds to detect a fairly obvious spoofing attack with position drift of 5 m/s. Such a time to detect is unacceptably long for an automated ground vehicle.

Prior work in spoofing detection specifically for ground vehicles demonstrated that low-cost IMUs could be used to detect GNSS spoofing by constructing a coherency test between the GNSS and inertial measurements [24]. But the test statistic in [24] was constructed from position-domain measurements, and so is much less sensitive than the carrier-phase-based test proposed in the current paper, resulting in an unacceptably-long (3 minute) time to detection.

Contributions

This paper's primary contributions are (i) the development and verification of a highly sensitive all-environment single-antenna GNSS spoofing detection technique based on carrier-phase fixed-ambiguity residual cost produced by a well-calibrated CDGNSS solution that is tightly coupled with a low-cost IMU, (ii) the introduction of an artificial worst-case spoofing methodology, and (iii) a comparison between industrial- and consumer-grade IMUs for spoofing detection within the proposed framework.

MEASUREMENT MODEL

The full formulation of the measurement model for the tightly-coupled GNSS-IMU estimator on which this paper's spoofing detection technique is based may be found in [14]. Key developments are presented here for the reader's convenience.

The estimator ingests N_k pairs of double-difference (DD) GNSS observables at each GNSS measurement epoch, with each pair composed of a pseudorange and a carrier phase measurement. The measurement vector at epoch k is

$$\boldsymbol{z}_k \triangleq \left[\boldsymbol{\rho}_k^{\mathsf{T}}, \boldsymbol{\phi}_k^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{2N_k}$$

where $\boldsymbol{\rho}_k$ and $\boldsymbol{\phi}_k$ are vectors of double-difference pseudorange and carrier phase measurements, both in meters. At epoch k, after linearizing about the *a priori* state estimate, a measurement model can be expressed as

$$\boldsymbol{\nu}_{\mathrm{k}} = \boldsymbol{H}_{\mathrm{r}k}\delta\boldsymbol{x}_k + \boldsymbol{H}_{\mathrm{n}k}\boldsymbol{n}_k + \boldsymbol{w}_{\nu k}, \quad \boldsymbol{w}_\nu \sim \mathcal{N}(\boldsymbol{0}, \Sigma_k) \tag{1}$$

where $\boldsymbol{\nu}_{\mathrm{k}}$ is the difference between the measurement $\boldsymbol{z}_k$ and its modeled value based on the *a priori* state estimate, $\boldsymbol{H}_{\mathrm{r}k}$ and $\boldsymbol{H}_{\mathrm{n}k}$ are Jacobians, $\delta\boldsymbol{x}_k$ is the state estimate error vector, $\boldsymbol{n}_k \in \mathbb{Z}^{N_k}$ is the the integer ambiguity vector, and $\boldsymbol{w}_{\nu k}$ is noise. A short-baseline regime is assumed for the DD measurements, which implies that ionospheric, tropospheric, ephemeris, and clock errors are cancelled in the double differencing, leaving $\boldsymbol{w}_{\nu k}$ to account only for multipath and receiver thermal noise. The prior on the real-valued error state can be expressed in terms of the following data equation:

$$\boldsymbol{0} = \delta\boldsymbol{x}_k + \boldsymbol{w}_{xk}, \quad \boldsymbol{w}_x \sim \mathcal{N}(\boldsymbol{0}, \bar{\boldsymbol{P}}_k) \tag{2}$$

The CDGNSS measurement update of the tightly-coupled GNSS-IMU estimator can be cast in square-root form for greater numerical robustness and algorithmic clarity [25]. Given $\boldsymbol{\nu}_k$, $\boldsymbol{H}_{\mathrm{r}k}$, $\boldsymbol{H}_{\mathrm{n}k}$, and the data equation above, the measurement update can be defined as the process of finding $\delta\boldsymbol{x}_k$ and $\boldsymbol{n}_k$ to minimize the cost function

$$J_k(\delta\boldsymbol{x}_k, \boldsymbol{n}_k) = \|\boldsymbol{\nu}_k - \boldsymbol{H}_{\mathrm{r}k}\delta\boldsymbol{x}_k - \boldsymbol{H}_{\mathrm{n}k}\boldsymbol{n}_k\|^2_{\boldsymbol{\Sigma}_k^{-1}} + \|\delta\boldsymbol{x}_k\|^2_{\bar{\boldsymbol{P}}_k^{-1}}$$

The vector cost components can be normalized by left multiplying with square-root information matrices based on Cholesky factorization $\boldsymbol{R}_{\mathrm{k}} = \texttt{chol}\left(\boldsymbol{\Sigma}_k{}^{-1}\right)$, $\bar{\boldsymbol{R}}_{xxk} = \texttt{chol}\left(\bar{\boldsymbol{P}}_k^{-1}\right)$:

$$\begin{aligned} J_k(\delta\boldsymbol{x}_k, \boldsymbol{n}_k) &= \left\| \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{R}_{\mathrm{k}}\boldsymbol{\nu}_k \end{bmatrix} - \begin{bmatrix} \bar{\boldsymbol{R}}_{xxk} \\ \boldsymbol{R}_{\mathrm{k}}\boldsymbol{H}_{\mathrm{r}k} \end{bmatrix} \delta\boldsymbol{x}_k - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{R}_k\boldsymbol{H}_{\mathrm{r}k} \end{bmatrix} \boldsymbol{n}_k \right\|^2 \\ &= \left\| \boldsymbol{\nu}'_k - \begin{bmatrix} \boldsymbol{H}'_{\mathrm{r}k} \\ \boldsymbol{H}'_{\mathrm{n}k} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}_k \\ \boldsymbol{n}_k \end{bmatrix} \right\|^2 \end{aligned}$$

The cost J_k can be decomposed via QR factorization

$$\left[\tilde{\boldsymbol{Q}}_k, \tilde{\boldsymbol{R}}_k\right] = \texttt{qr}\left(\begin{bmatrix} \boldsymbol{H}'_{\mathrm{r}k} \\ \boldsymbol{H}'_{\mathrm{n}k} \end{bmatrix}\right)$$

where matrix $\tilde{\boldsymbol{Q}}_k$ is orthogonal and $\tilde{\boldsymbol{R}}_k$ is upper triangular. Because $\tilde{\boldsymbol{Q}}_k$ is orthogonal, the components of J_k inside the norm can be left-multiplied by $\tilde{\boldsymbol{Q}}_k^{\mathsf{T}}$ without changing the cost, and J_k can be decomposed into 3 terms:

$$\begin{aligned} J_k(\delta\boldsymbol{x}_k, \boldsymbol{n}_k) &= \left\| \tilde{\boldsymbol{Q}}_k^{\mathsf{T}}\boldsymbol{\nu}'_k - \tilde{\boldsymbol{R}}_k \begin{bmatrix} \delta\boldsymbol{x}_k \\ \boldsymbol{n}_k \end{bmatrix} \right\|^2 \\ &= \left\| \begin{bmatrix} \boldsymbol{\nu}''_{1k} \\ \boldsymbol{\nu}''_{2k} \\ \boldsymbol{\nu}''_{3k} \end{bmatrix} - \begin{bmatrix} \boldsymbol{R}_{xxk} & \boldsymbol{R}_{xnk} \\ \boldsymbol{0} & \boldsymbol{R}_{nnk} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}_k \\ \boldsymbol{n}_k \end{bmatrix} \right\|^2 \\ &= \underbrace{\|\boldsymbol{\nu}''_{1k} - \boldsymbol{R}_{xxk}\delta\boldsymbol{x}_k - \boldsymbol{R}_{xnk}\boldsymbol{n}_k\|^2}_{J_{1k}(\delta\boldsymbol{x}_k, \boldsymbol{n}_k)} + \underbrace{\|\boldsymbol{\nu}''_{2k} - \boldsymbol{R}_{nnk}\boldsymbol{n}_k\|^2}_{J_{2k}(\boldsymbol{n}_k)} + \underbrace{\|\boldsymbol{\nu}''_{3k}\|^2}_{J_{3k}} \end{aligned} \tag{3}$$

If both the measurement model and $\bar{\boldsymbol{R}}_{xxk}$ are not ill-conditioned, then $\boldsymbol{R}_{xxk}$ and $\boldsymbol{R}_{nnk}$ are invertible. J_{1k} can be zeroed for any value of $\boldsymbol{n}_k$ due to the invertibility of $\boldsymbol{R}_{xxk}$. J_{3k} is the irreducible cost, and, under a single-epoch ambiguity resolution scheme, can be shown to be equal to the normalized innovations squared (NIS) associated with the double-difference pseudorange measurements.

J_{2k} is the extra cost incurred by enforcing the integer constraint on $\boldsymbol{n}_k$. If $\boldsymbol{n}_k$ is allowed to take any real value (the *float solution*), J_{2k} can be zeroed due to the invertibility of $\boldsymbol{R}_{nnk}$. The float solution $\{\delta\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{n}}_k\}$ is formed by choosing $\delta\tilde{\boldsymbol{x}}_k$ and

$\tilde{\boldsymbol{n}}_k$ to zero J_{1k} and J_{2k}. Because $\tilde{\boldsymbol{R}}_k$ is upper triangular, these values can be found by efficient backsubstitution. The *fixed solution* $\{\delta\check{\boldsymbol{x}}_k, \check{\boldsymbol{n}}_k\}$ is found via an integer least squares (ILS) solver, yielding

$$\begin{aligned} \check{\boldsymbol{n}}_k &= \arg \min_{\boldsymbol{n}_k \in \mathbb{Z}^{N_k}} J_{2k}(\boldsymbol{n}_k) \\ \delta\check{\boldsymbol{x}}_k &= \boldsymbol{R}_{xxk}^{-1}\left(\boldsymbol{\nu}_{1k}'' - \boldsymbol{R}_{xnk}\check{\boldsymbol{n}}_k\right) \end{aligned} \tag{4}$$

Note that $\boldsymbol{R}_{xxk}$ is the *a posteriori* state vector square-root information matrix conditioned on $\boldsymbol{n}_k = \check{\boldsymbol{n}}_k$.

TEST STATISTIC

Key to this paper's spoofing detection statistic is the integer-fixed carrier-phase residual cost

$$\epsilon_{\phi k} = J_{2k}(\check{\boldsymbol{n}}_k)$$

which can also be thought of as the ILS solution cost [26]. This is small whenever the carrier phase measurements are consistent with the prior state estimate, the pseudorange measurements, and with the assumption of integer-valued carrier-phase ambiguities. It is one of several acceptance test statistics used to decide whether the fixed solution $\check{\boldsymbol{n}}_k$ is correct with high probability [15]. In [27], $\epsilon_{\phi k}$ was incorporated in a statistic used to detect carrier cycle slips. It can similarly be used to detect false integer fixes, just as with other integer aperture acceptance test statistics, or the lingering effects of conditioning the real-valued part of the state $\delta\boldsymbol{x}_k$ on a previous false fix [14].

Furthermore, $\epsilon_{\phi k}$ provides is a highly sensitive statistic for spoofing detection. When no spoofing is present, there is tight agreement between the IMU-propagated *a priori* state estimate and GNSS data resulting in a small $\epsilon_{\phi k}$. If the vehicle hits a bump in the road, the GNSS antenna phase center will rise by a few centimeters, and the inertial sensor will detect a corresponding acceleration, which will get propagated through to the *a priori* state. On the other hand, when spoofing is present, a discrepancy between inertial and GNSS data will arise at the carrier-phase level, leading to $\epsilon_{\phi k}$ being larger than usual.

A windowed sum of $\epsilon_{\phi k}$ offers even greater sensitivity to false fix events at the expense of a longer time to detect. The test statistic used to detect spoofing in this paper is the *windowed fixed-ambiguity residual cost* (WFARC), Ψ_k. This is calculated over a moving window of fixed length l of past GNSS measurement epochs. It has N_{Ψ_k} degrees of freedom and is calculated by

$$\Psi_k \triangleq \sum_{n=k-l+1}^{k} \epsilon_{\phi n}, \quad N_{\Psi_k} \triangleq \sum_{n=k-l+1}^{k} N_n$$

where N_k is the number of DD carrier phase measurements at epoch k. In this paper, a window length of l = 10 past GNSS measurement epochs (amounting to a window of 2 seconds) is used.

If the filter is consistent and the integer ambiguities are correctly resolved, then Ψ_k should be approximately χ^2-distributed with N_{Ψ_k} degrees of freedom. This distribution is approximate due to the "integer-folding" effect: large phase residuals are not possible because of integer-cycle phase wrapping. A statistical consistency test can be performed by choosing a desired false-alarm rate $\bar{P}_{f,\Psi}$ and declaring a false fix if $\Psi_k > \gamma_{\Psi k}$, where the threshold $\gamma_{\Psi k}$ is calculated by evaluating the inverse cumulative distribution function of $\chi^2(N_{\Psi_k})$ at $\bar{P}_{f,\Psi}$.

Compared to a single residual $\epsilon_{\phi k}$, Ψ_k has greater statistical power for consistency testing and helps avoid premature declaration of spoofing due to sporadic measurement outliers. However, increasing the window length l also increases the latency to detect a spoofing event. This statistical test is conducted at each GNSS measurement epoch. The null hypothesis, H_0 denotes no spoofing detected, and the alternate hypothesis, H_1 indicates the detection of spoofing. These are declared according to the rule

$$\delta(\Psi_k) = \begin{cases} H_0 & \text{if} \quad \Psi_k < \gamma_{\Psi k} \\ H_1 & \text{if} \quad \Psi_k \geq \gamma_{\Psi k} \end{cases}$$

DATA COLLECTION

Data was gathered on the UT Radionavigation Laboratory (RNL) *Sensorium*, an integrated platform for automated and connected vehicle perception research. It is equipped with multiple radars, IMUs, GNSS receivers, and a lidar, as shown in

Fig. 1. With the *Sensorium*, the RNL produced a public benchmark dataset collected in the dense urban center of the city of Austin, TX called TEX-CUP [28] for evaluating multi-sensor GNSS-based urban positioning algorithms [16]. The data captured includes a diverse set of multipath environments (open-sky, shallow urban, and deep urban) as shown in Fig. 2. The TEX-CUP dataset provides raw wideband IF GNSS data with tightly synchronized raw measurements from multiple IMUs and a stereoscopic camera unit, as well as truth positioning data. This allows researchers to develop algorithms using any subset of the sensor measurements and compare their results with the true position.

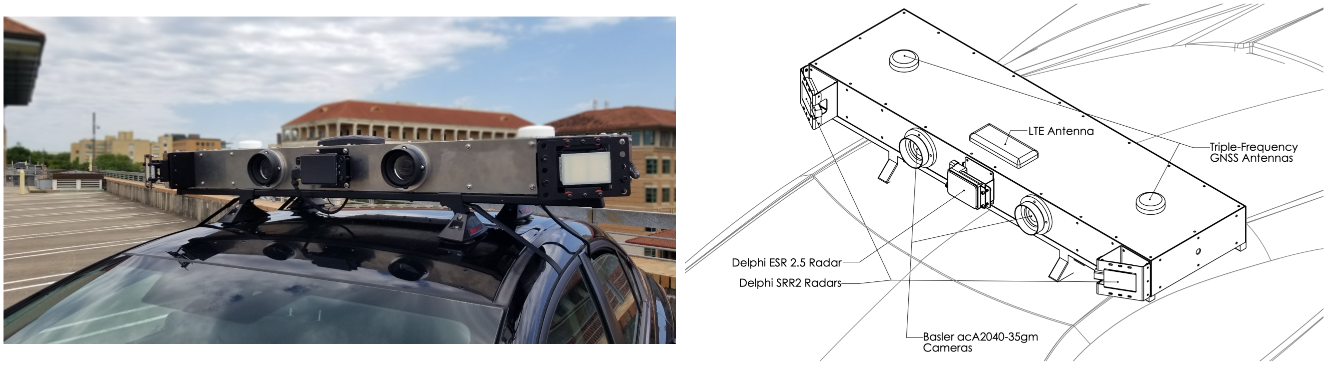


Fig. 1: The UT RNL has developed a multi-modal ground-vehicle-mounted integrated perception platform call the *Sensorium*. It houses three different types of IMU, two triple-frequency GNSS antennas, three radar sensors, and two cameras. An extensive localization pipeline based on these sensors has been developed.

For this paper's analysis, only the raw GNSS intermediate-frequency (IF) samples from the primary antenna and inertial data from TEX-CUP were considered, along with inertial data. Two-bit-quantized IF samples were captured at the Sensorium and at the reference station through the *RadioLynx*, a low-cost L1+L2 GNSS front end with a 5 MHz bandwidth at each frequency, and were processed with the RNL's GRID SDR [29]–[33]. The tightly-coupled CDGNSS estimator described earlier was implemented in C++ as a new version of the GRID's sensor fusion engine. The system's performance was separately evaluated using inertial data from each of the Sensorium's two MEMS inertial sensors. The first, a LORD MicroStrain 3DM-GX5-25, is an industrial-grade sensor. The second, a Bosch BMX055, is a surface-mount consumer-grade sensor.

TEX-CUP provides ground truth data for the vehicle position and orientation. The truth dataset was generated by a combination of sensor fusion and a tactical-grade IMU. The *Sensorium* is equipped with an iXblue ATLANS-C: a high-performance RTK-GNSS coupled fiber-optic gyroscope inertial navigation system. The post-processed fused RTK-INS position solution obtained from the ATLANS-C is taken to be the ground truth trajectory. Post-processing software provided by iXblue generates a forward-backward smoothed position and orientation solution with fusion of AsteRx4 RTK solutions and inertial measurements. The post-processed solution is accurate to better than 10 centimeters throughout the dataset. The effectiveness of the developed spoofing detection method is evaluated with these datasets.

SPOOFING METHODOLOGY

The total signal at the victim receiver antenna is

$$y_{\text{tot}}(t) = y_a(t) + y_s(t) + \nu(t)$$

where $y_a(t)$ is the authentic signal, $y_s(t)$ is the spoofed signal, and $\nu(t)$ is the received noise. Under a challenging spoofing attack, $y_s(t)$ contains a perfect null of the authentic signal and $\nu(t)$ is entirely naturally generated, i.e., not introduced by the spoofer.

Physical-Layer Spoofing

To artificially simulate a spoofing attack, over-the-air, cable injection, and digital signal injection spoofing were considered. Over-the-air attacks are possible [34]–[36], but are not authorized in urban areas. A cable injection attack would be permissible

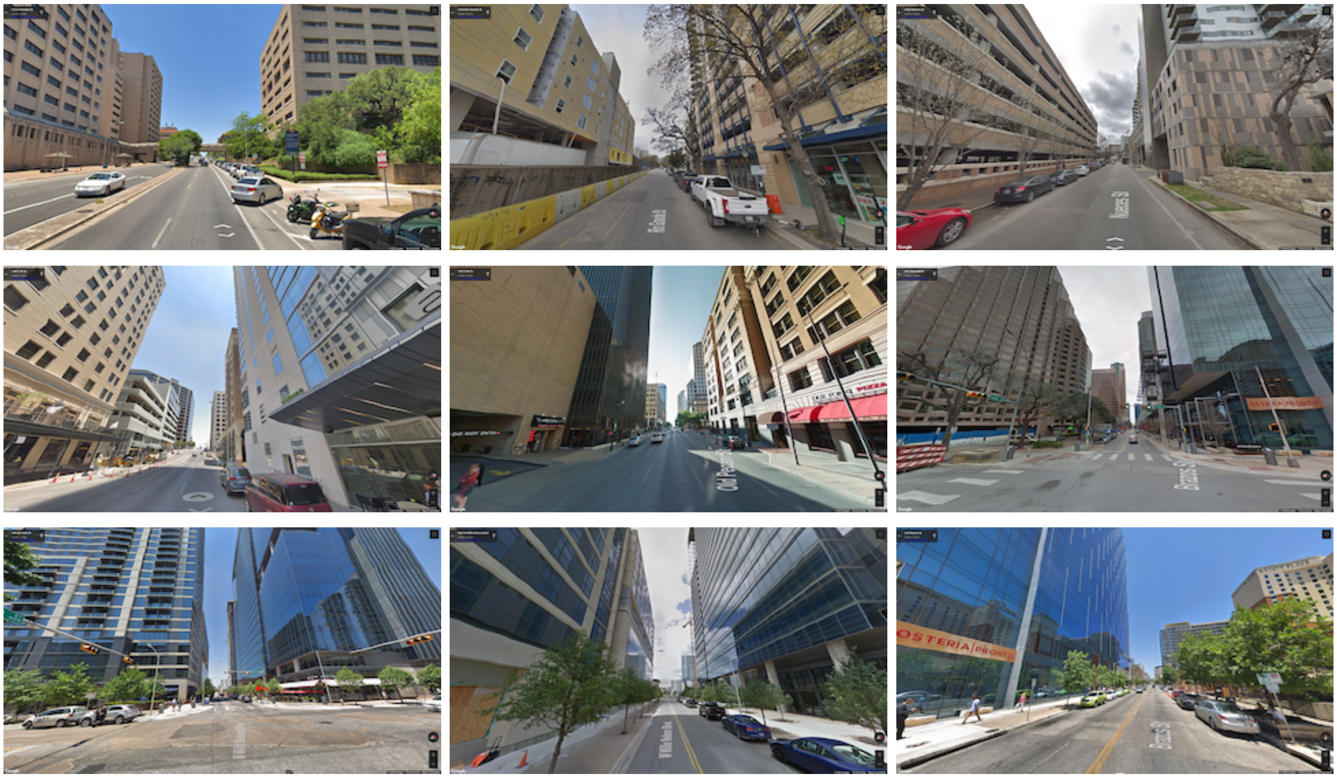


Fig. 2: Google Street View imagery of a few challenging scenarios encountered in the TEXCUP dataset [28].

for a live experiment in an urban area, and digital signal combining, as in [37], is a powerful after-the-fact spoofing technique. But in both cases it is challenging to explore a *worst case* spoofing attack in which the authentic signals y_{a} are entirely nulled by an antipodal spoofing signal, as described in [37]. Experience with `ds7` and `ds8` from the Texas Spofing Test Battery (TEXBAT, [38], [39]) revealed that such antipodal spoofing is difficult to maintain under even static laboratory conditions. The remnant authentic signal from an unsophisticated and imperfect spoofing attack sullies the test statistic, making detection too easy and leading to an overly optimistic performance assessment.

Of course, nulling y_{a} can also be achieved by generating a spoofing signal so powerful that it buries the authentic signal below the receiver's noise floor. But this should not be considered a *worst case* attack because the overwhelmingly high received signal power from the spoofer is easily detected [5].

In short, physical-layer spoofing is challenging to conduct in such a way as to present a convincing worst-case spoofing attack to this paper's detector.

Observation-Domain Spoofing

It is important to evaluate spoofing detection techniques on a worst-case spoofing attack, with the idea being that if the proposed detection strategy is effective on the worst-case scenario, it is even more effective on weaker attacks. Accordingly, this paper adopts *observation-domain spoofing*. The spoofing in the observation domain is advantageous because the authentic signal is inherently nulled, presenting a subtle attack.

The first method of implementing observation-domain spoofing is *position offset spoofing*. With position offset spoofing, a position offset $\delta\boldsymbol{r}$ is added to the authentic measured position $\delta\boldsymbol{r}_a$ to generate a spoofed position $\boldsymbol{r}_s = \boldsymbol{r}_a + \delta\boldsymbol{r}(t)$. This is accomplished by altering the pseudorange and carrier phase measurements from each satellite so that they correspond to the spoofed position with the desired additive position offset $\delta\boldsymbol{r}$. The spoofed pseudorange $\rho_s^i(t)$ and carrier phase $\phi_s^i(t)$ measurements for the ith satellite are constructed as follows

$$\rho_s^i(t) = \rho_a^i(t) + \delta\rho^i(\delta\boldsymbol{r}(t)) \tag{5}$$

$$\phi_s^i(t) = \phi_a^i(t) + \delta\phi^i(\delta\boldsymbol{r}(t)) \tag{6}$$

where $\rho_a^i(t)$ and $\phi_a^i(t)$ are the authentic pseudorange and carrier phase for the ith satellite and $\delta\rho^i(\cdot)$ and $\delta\phi^i(\cdot)$ are the nonlinear (but easily linearizable) mapping functions, based on the geometry for the particular ith satellite, that map the position offset $\delta\boldsymbol{r}(t)$ to the corresponding pseudorange and carrier phase offsets.

The second method of implementing observation-domain spoofing is *timestamp spoofing*. With timestamp spoofing, the measurements at a particular time are reassigned to have an alternate measurement timestamp. The required modifications to the spoofed pseudorange $\rho_s^i(t)$ and carrier phase $\phi_s^i(t)$ measurement from each satellite induced by the new timestamp may be expressed as

$$\rho_s^i(t) = \rho_a^i(t + \delta t(t)) + \Delta\rho^i(t, \delta t(t)) \tag{7}$$

$$\phi_s^i(t) = \phi_a^i(t + \delta t(t)) + \Delta\phi^i(t, \delta t(t)) \tag{8}$$

where $\delta t(t)$ is the timestamp shift applied. The authentic observables from time $t + \delta t(t)$ are fed to the estimator as if they had occurred at time t. The functions $\Delta\rho^i(t, \delta t(t))$ and $\Delta\phi^i(t, \delta t(t))$ adjust the timestamp-shifted observables to account for the transmitting spacecraft's orbital motion and clock evolution over the interval from t to $t + \delta t(t)$.

In position offset spoofing, because $\phi_s(t) = \phi_a(t) + \delta\phi(\delta\boldsymbol{r}(t))$, all vehicle motion reflected in $\phi_a(t)$ is also present in $\phi_s(t)$. This includes all high-frequency motion due to the road irregularities and other minor movements. A detection technique designed to detect small-amplitude, high-frequency discrepancies in $\phi(t)$ via the WFARC would not actually see such discrepancies unless $\delta\phi(\delta\boldsymbol{r}(t))$ also included simulated high-frequency content.

By contrast, timestamp spoofing borrows spoofed phase and pseudorange measurements from a different time instant, ensuring that high-frequency variations in these quantities will be different from those predicted by the *a priori* state based on IMU propagation. This is more representative of an actual spoofing attack scenario in which the attacker cannot predict the high-frequency vehicle motion. Moreover, by reducing the timestamp shift $\delta t(t)$, one can realize ever-subtler attacks that are increasingly hard to detect, allowing exploration of worst-case-for-detectability spoofing.

Timestamp spoofing is also worst-case in a different sense. Because each spoofed observable is made to be consistent with the GNSS constellation geometry, the spoofing detector is presented with observables whose implied geometry is consistent with actual transmitter locations, not, for example, with a single transmitting spoofer. Moreover, because there are no irregularities in the observables that might result from an over-the-air or direct-injection attack in which the authentic signals are not perfectly nulled and so conflict with the spoofing signals, the spoofing detector is presented with observables whose variations, both in amplitude and frequency content, are entirely plausible. Thus, timestamp spoofing is representative of a case in which a well-financed attacker is able to place a single-satellite-full-single-ensemble spoofer capable of full authentic-signal nulling along the line-of-sight from the target vehicle to each overhead GNSS satellite, as illustrated in Fig. 3.

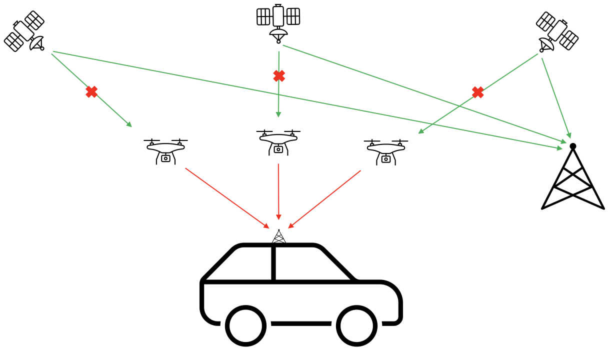


Fig. 3: Timestamp spoofing is representative of a worst-case spoofing attack in which the attacker positions a fleet of drone-borne spoofers such that spoofing signals (1) perfectly null the corresponding authentic signals, and (2) emanate from positions along the line of sight connecting each overhead GNSS satellite to the target vehicle. The reference receiver shown to the right, whose GNSS observables are used for precise CDGNSS processing, is presumed to be unaffected by the spoofing attack.

RESULTS

The following section presents an analysis of the proposed test statistic in both the non-spoofing case and against a worst-case attack. Results with the industrial- and consumer-grade IMU are presented.

Characterization of the Null Hypothesis

This spoofing detector is premised on a hypothesis test between statistical models for the authentic and counterfeit GNSS signals. The statistics of the null hypothesis must be fully characterized so that a statistical baseline is established, against which carrier phase errors induced by spoofing in the same setting can be compared. The null hypothesis of dynamic ground vehicle scenarios includes natural effects such as blockage and multipath, which is the predominant source of error.

To analyze the null hypothesis, the WFARC was calculated in the nominal case through the entirety of the TEX-CUP dataset containing no spoofing. Because multipath is dependent on the surrounding environment, two categories were separately considered: shallow urban and deep urban. Measurements were separated into these categories manually by identifying segments of the dataset where the vehicle resided in shallow urban and deep urban areas.

Fig. 4 shows the complementary cumulative distribution function (CCDF) of the WFARC in shallow and deep urban environments for the nominal case with industrial- and consumer-grade IMUs. The test statistic in the deep urban case has a much longer tail, which is expected because of the extreme multipath and blockage in deep urban areas. The cyan line represents the largest value of the WFARC in the shallow urban environment and the purple line represents the largest value of the WFARC in the deep urban environment. These will be the thresholds used to detect spoofing. Because the test statistic in the null hypothesis is never larger than these values, it corresponds to having a false alarm probability of zero. A chi-squared test can be used to lower these thresholds but comes at the cost of having a fixed false positive rate. Fig. 5 shows the time history of the WFARC over the TEX-CUP dataset.

It is important to note that the WFARC while using the consumer grade IMU is generally smaller than the WFARC while using the industrial grade IMU. This is expected because the consumer grade IMU is of lesser quality, thus having more variance with each measurement. The *a priori* state estimate from IMU tight coupling has a larger uncertainty because the estimator has less confidence in the IMU measurements, leading corrupted measurements to be more believable. Once again, in the null hypothesis, spikes in the WFARC are caused by multipath and blockage.

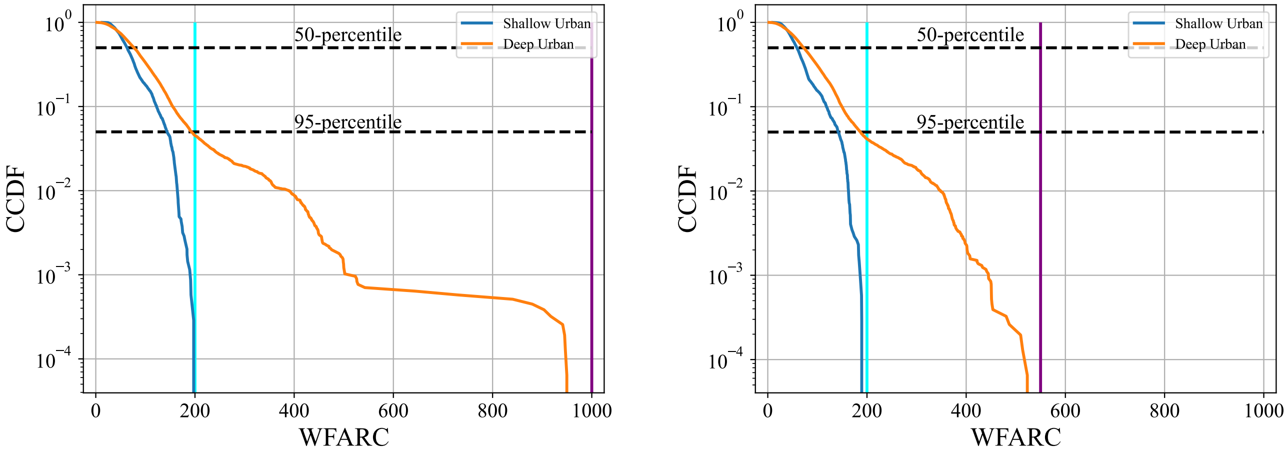


Fig. 4: The complementary cumulative distribution function (CCDF) of the WFARC over the entire TEX-CUP dataset with the LORD MicroStrain 3DM-GX5-25 (industrial grade) IMU on the left and with the Bosch BMX055 (consumer grade) IMU on the right. The test statistic is separated into two categories: shallow urban and deep urban. The cyan line represents the largest value of the WFARC in the shallow urban environment and the purple line represents the largest value of the WFARC in the deep urban environment. The deep urban environment has a significantly longer tail compared ot the shallow urban environment.

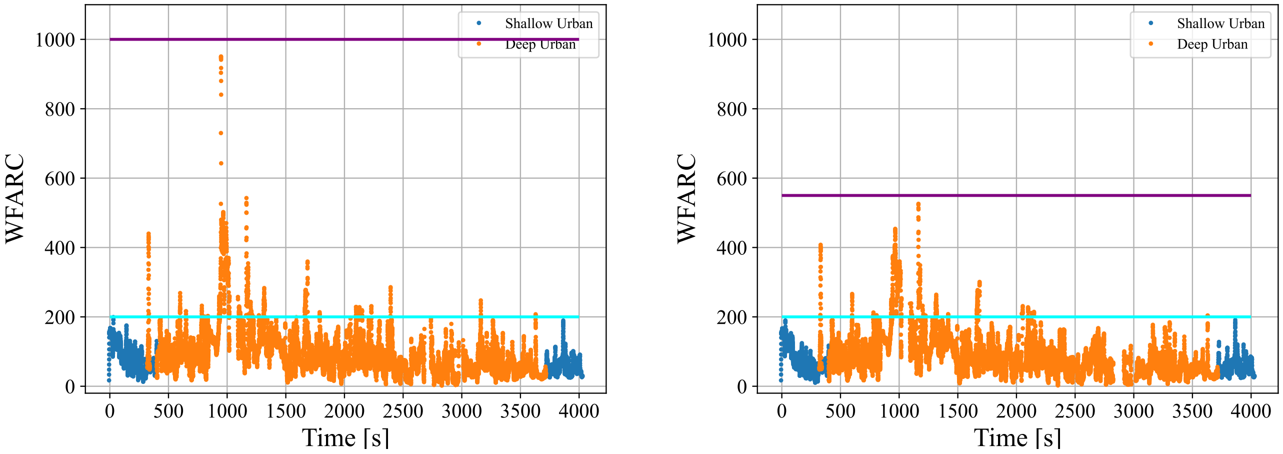


Fig. 5: A time history of the WFARC over the entire TEX-CUP dataset with the industrial grade IMU on the left and the consumer grade IMU on the right.

Performance Against a Worst-Case Spoofing Attack

The following is an example of a worst-case spoofing attack in a shallow urban environment. In this scenario, the spoofing attack begins while the vehicle is stopped at a stoplight and continues as the vehicle begins to move. The WFARC in this scenario are shown in Fig. 6 with both industrial- and consumer-grade IMUs. The vehicle starts moving at the 163 second mark. The spoofing attack begins at the 163 second mark just before first movement and ends at the 175 second mark. As the vehicle begins to move, the position errors will grow gradually because the vehicle slowly begins to accelerate forward, inducing a position error. Three different time shift attacks in the same scenario are shown in this figure. The shift of .15 seconds is the least subtle attack while the .05 second attack is the most challenging attack because the faults are much smaller. As the vehicle begins to move, the estimator recognizes inconsistencies between the spoofed GNSS measurements and the IMU because of the tight coupling. The rise in the WFARC above the thresholds shows this disagreement that is attributed to spoofing.

With the industrial grade IMU and using the shallow urban threshold, all three time shifts spoofing attacks were identified within a second. The estimator knows that the IMU data are different than the GNSS measurements from the WFARC, much more than anything multipath would induce in the shallow urban environment. If the vehicle was in the deep urban environment, the .05 second shift spoofing attack would just be attributed to multipath. The sensitivity of the test is dependent on multipath environment.

All three attacks were identified while using the consumer-grade IMU within two seconds. If the deep urban threshold was applied, only the least challenging attack would have been identified. In all cases, the WFARC is significantly smaller compared to when the industrial IMU is used. Once again, this is because the estimator has more confidence in the spoofed GNSS measurements than the lower quality IMU. Interestingly, there is a spike in the WFARC after the spoofing attack is over. This happens because the estimator is showing trauma from the spoofing attack– the abrupt return of the true GNSS measurements were significantly different from what the previously ingested spoofed measurements were predicting.

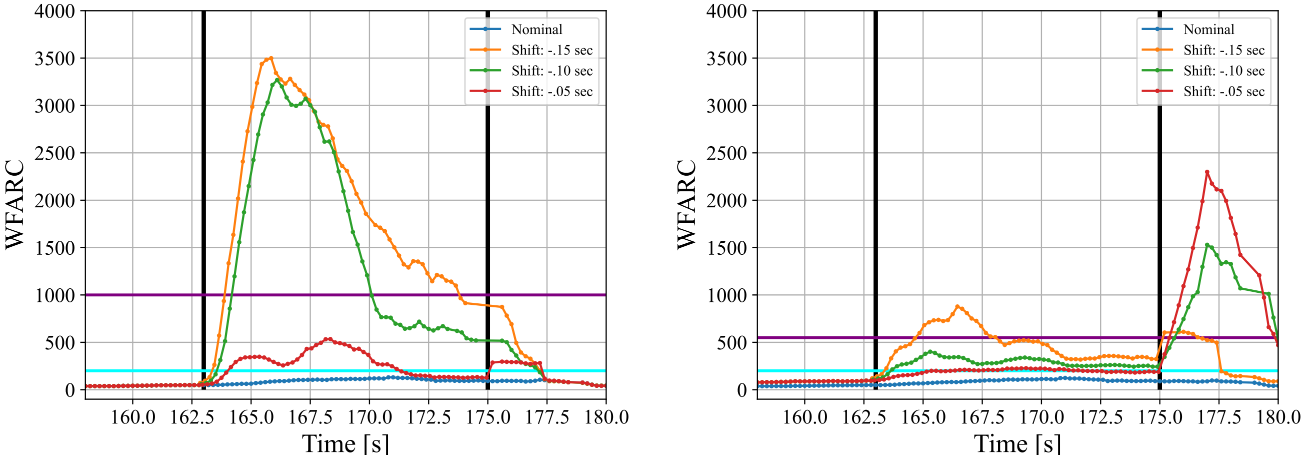


Fig. 6: WFARC during a worst-case spoofing attack in the shallow urban environment. The plot on the left is with the LORD MicroStrain 3DM-GX5-25 (industrial grade) IMU and the plot on the right is with the Bosch BMX055 (consumer grade) IMU. Spoofing begins at the 163 second mark and ends at the 175 second mark. The vehicle is stopped at a red light, but then starts moving at the 163 second mark, introducing a small drag off from the true position. Plotted here are 3 different time shift attacks in the same scenario. The shift of .15 seconds is the least subtle attack while the .05 second attack is the most challenging attack. From the analysis if the null hypothesis, the cyan line denotes the shallow urban threshold and the purple line denotes the deep urban threshold.

The corresponding position errors in each attack is shown in Fig. 7. The worst-case attack (time shift of -.05 seconds) only introduces a .5 meter offset over 10 seconds, indicative of an extremely subtle attack. Even the least subtle attack (time shift of -.15 seconds) only introduces a 2 meter offset after 10 seconds, which is much more challenging than the attacks simulated in the related work.

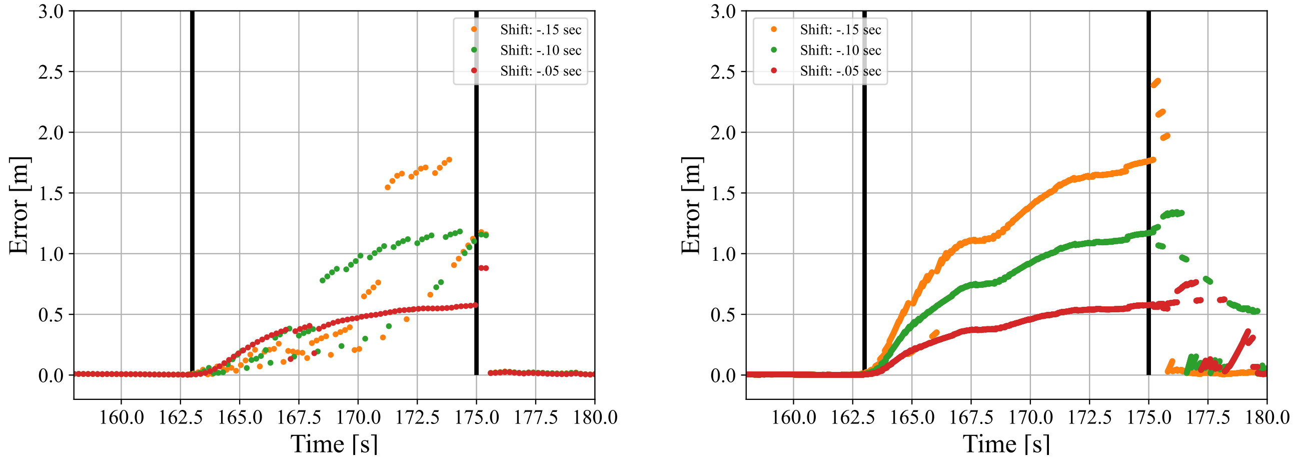


Fig. 7: The position errors induced from the different spoofing attacks. The plot on the left is with the LORD MicroStrain 3DM-GX5-25 (industrial grade) IMU and the plot on the right is with the Bosch BMX055 (consumer grade) IMU.

CONCLUSION

A powerful single-antenna carrier-phase-based test to detect GNSS spoofing attacks on ground vehicles equipped with a low-cost IMU was developed, implemented, and validated. Artificial worst-case spoofing attacks were injected into a dataset collected by a vehicle-mounted sensor suite in Austin, Texas and detected within two seconds. This was accomplished by using a spoofing detection technique that capitalized on the carrier phase fixed-ambiguity residual cost produced by a well-calibrated CDGNSS solution that is tightly coupled with a low-cost IMU. The finer movements of the vehicle, such as slight steering movements and road vibrations, are the necessary unpredictable dithering a spoofer is not able to replicate. The differences between IMU-predicted and measured carrier phase values offer the basis for an exquisitely sensitive GNSS spoofing detection statistic. This paper developed the null-hypothesis empirical distributions for the test statistic in both shallow and deep urban areas, and uses these distributions to demonstrate that high-sensitivity spoofing detection is possible despite integer folding and urban multipath. Additionally, the effectiveness of consumer- and industrial-grade IMUs for spoofing detection was compared. The type of tightly-coupled IMU-GNSS estimator whose by-products the proposed detection technique exploits is not currently available on commercial passenger vehicles, but can be expected to be adopted in future automated vehicles, since it provides all-weather dm-level absolute positioning.

3 Physics-Based Anomaly Detection

3a Lateral-Direction Localization Attack in High-Level Autonomous Driving: Domain-Specific Defense Opportunity via Lane Detection

ABSTRACT

Localization in high-level Autonomous Driving (AD) systems is highly security critical. While the popular Multi-Sensor Fusion (MSF) based design can be more robust against single-source sensor spoofing attacks, it is found recently that state-of-the-art MSF algorithms can still be vulnerable to GPS spoofing alone due to practical factors, which can cause various road hazards such as driving off road or onto the wrong way. In this work, we perform the first systematic exploration of the novel usage of lane detection (LD) to defend against such attacks. We first systematically analyze the potentials of such a domain-specific defense opportunity, and then design a novel LD-based defense approach, LD^3, that aims at not only detecting such attacks effectively in the real time, but also safely stopping the victim in the ego lane upon detection considering the absence of onboard human drivers.

We evaluate LD^3 on real-world sensor traces and find that it can achieve effective and timely detection against existing attack with 100% true positive rates and 0% false positive rates. Results also show that LD^3 is robust to diverse environmental conditions and is effective at steering the AD vehicle to safely stop within the current traffic lane. We implement LD^3 on two open-source high-level AD systems, Baidu Apollo and Autoware, and validate its defense capability in both simulation and the physical world in end-to-end driving. We further conduct adaptive attack evaluations and find that LD^3 is effective at bounding the deviations from reaching the attack goals in stealthy attacks and is robust to latest LD-side attack.

INTRODUCTION

Recently, high-level Autonomous Driving (AD) vehicles [40], e.g., Level-4 ones, are gradually becoming part of the transportation system by providing commercial services such as self-driving taxis [41], [42], buses [43], [44], and trucks [45], [46]. In particular, AD companies such as Waymo and Baidu are already offering commercial RoboTaxi services without safety drivers [41], [47], and more others are performing tests on public roads [48], [49]. To achieve high driving automation, the *high-level AD system* (the "*brain*") in such a vehicle needs to localize itself with *centimeter-level* accuracy on the map [10], [50], [51] to ensure safe and correct driving. Thus, today's industry-grade high-level AD systems predominantly adopt a Multi-Sensor Fusion (MSF) based localization design, which combines sensor inputs, typically GPS, LiDAR, and IMU, for overall higher accuracy and robustness in practice [52]–[57].

Due to the reliance on sensor inputs, AD localization is inherently vulnerable to sensor spoofing attacks, in particular GPS spoofing [58], [59], a long-existing security problem that is fundamentally difficult in both prevention and detection in practice [58], [60]. Although the MSF-based design is generally more robust against such single-source sensor attacks, recent work [58] find that state-of-the-art MSF algorithms are still vulnerable to strategic GPS spoofing attacks due to non-deterministic and practical factors such as sensor noises and algorithm inaccuracies. To leverage such non-deterministic vulnerabilities, the authors devise a lateral-direction localization attack named *FusionRipper* to *opportunistically* inject lateral deviations in the MSF localization outputs, which will be translated into lateral deviations in the physical world by the AD control. Such lateral-direction localization attack is especially safety-critical in the AD context due to the potential consequences of road departure [61].

BACKGROUND AND THREAT MODEL

High-level AD Localization and MSF

Today's high-level (e.g., Level-4 [40]) AD systems widely adopt a modular design with functional components such as localization, perception, prediction, planning, and control [55]–[57], [62], [63]. Among them, localization is one of the most important modules that provides global positioning on the map for other modules such as planning and control to make safety-critical driving decisions. Since high-level AD systems need to navigate on the roads complete autonomously without any drivers, a localization with *centimeter-level* accuracy is required to localize the AD vehicle on the traffic lane [10], [50], [51]. High-level AD systems are typically equipped with various positioning sensors with diverse properties. For example, GPS provides global positioning with high availability, however, it often contains large positioning noises due to satellite signal transmission interferences and multi-path effect [64]; on the other hand, LiDAR localization algorithms (LiDAR locators) are able to accurately position the vehicle on a prebuilt LiDAR reflectance map using point cloud matching [52], [53], [65], [66]. However, LiDAR locator performance can be severely degraded under adverse weather conditions or with an outdated LiDAR map. Thus, to achieve both high accuracy and robustness, high-level AD systems predominantly adopt a *Multi-Sensor Fusion* (MSF) based localization design to *leverage the strengths and compensate the weaknesses of different sensors* such as GPS, LiDAR, and IMU [52]–[57].

Lateral-Direction Localization Attack

For AD localization, a direct threat is the attacks targeting the localization sensors such as GPS spoofing [34], [59], [67]–[71], in which the attacker transmits fake satellite signals to the victim GPS receiver and thus cause it to resolve positions manipulated by the attacker. However, due to the high robustness provided by sensor fusion, MSF is often considered as a promising defense strategy for GPS spoofing [67], [72]–[74]. Contrary to the common belief, prior work [58] proposes an *opportunistic* lateral-direction localization attack method, called *FusionRipper*, which can use GPS spoofing alone to inject *lateral deviations* in the MSF localization outputs and thus cause the AD vehicle to drive off-road or onto the wrong way. *FusionRipper* is consist of two attack stages: *vulnerability profiling* and *aggressive spoofing*. In the vulnerability profiling stage, it spoofs the GPS inputs of MSF localization with a small constant distance d (e.g., 0.5 m) in the lateral direction, waiting to discover a vulnerable attack window. Whenever the AD vehicle's physical deviation is larger than certain threshold (e.g., 0.3 m), *FusionRipper* launches the aggressive spoofing stage, where a scaling factor f (e.g., 1.2) will be continuously applied to the spoofing distance in each second to quickly introduce large lateral deviations in the MSF localization outputs. *FusionRipper* has shown high attack effectiveness on the representative MSF algorithms, including the one in the industry-grade Baidu Apollo AD system [62]. To best of our knowledge, *FusionRipper* [58] is the only localization attack that is able to defeat the MSF based localization algorithm in high-level AD systems.

Threat Model

Attacker's capability. In this work, we assume the attacker can launch practical lateral-direction localization attacks through external means such as GPS spoofing, which can cause lateral deviations in the localization outputs. Specifically, we focus on the lateral-direction attacks since such attacks (1) can cause the AD vehicle to violate the traffic norm that a vehicle should be driving within its designated lane boundaries and should not have unexpected lane straddling behaviors, and (2) pose a direct threat to the AD vehicle and road safety, e.g., it can cause the AD vehicle to drive off highway cliff or onto the wrong way and being hit by other vehicles that failed to yield in time.

In particular, we do not consider simultaneous attacks that target both AD localization and lane detection at the same time, since such simultaneous attack neither already exists, nor can be easily achieved today (detailed discussions in §).

AD control assumption. Same as *FusionRipper* [58] and also as a common design in academia [75] and industry [62], [63], we assume the AD systems are designed to drive at the center of traffic lane and constantly correct the deviations to the center. Since AD controllers constantly correct such deviations at a high frequency, e.g., 100 Hz [62], the lateral deviations in the AD localization will thus be directly reflected as physical world deviations, but to the opposite direction.

LANE DETECTION FOR HIGH-LEVEL AD LOCALIZATION DEFENSE: OPPORTUNITY ANALYSIS

Motivation and novelty. Currently, no software-based defense solutions have been proposed to address the latest GPS spoofing-based lateral-direction localization attack in high-level AD systems (§). The closest ones are the recent physical-invariants based detectors proposed for small robotic vehicles such as drones and rovers, e.g., SAVIOR [76] and CI [77], which estimate the physical dynamics of drones and rovers to validate the GPS signal. Although they show high effectiveness for such small robotic vehicles under large attack deviation goals, their effectiveness in AD vehicle context is fundamentally more limited since (1) existing vehicle dynamics models have difficulties in modelling high-speed and curvy-road settings [78], [79]; and (2) in the AD context, the attack deviation goals can be much smaller (thus harder to detect) while still being safety-critical. As we concretely evaluate later in §, direct adaptation of such existing physical-invariant based approach to the AD context suffers from very high false positives and is actually close to random guessing.

In comparison to small robotic vehicles, the AD context may also have its unique defense opportunities for such lateral-direction localization attacks. *Lane Detection (LD)* [80], [81], a technology commonly used in low-level AD systems for lane centering [82], [83], is such an example that can be used to measure the vehicle's lateral position within the current lane in real time, which is directly related to the lateral-direction attack goal (lane departure). Although effective in low-level AD systems (e.g., Level-2 ones such as Tesla Autopilot [83] that still count on human drivers to take over anytime), *LD is currently not used for high-level AD localization purpose (e.g., Level-4 ones such as Waymo that do not assume onboard human drivers)*. This is because what LD can provide is by nature only *local* positioning (i.e., relative positioning within ego lane), while high-level AD requires *global* positioning (i.e., in world coordinates on a map) for safe and correct driving decision-making without human drivers. Although there exist camera-based global localization methods using lane markings [84], [85], they are not generally adopted in state-of-the-art high-level AD localization [52]–[57] as they are far from reaching the required centimeter-level accuracy [10], [50], [51].

While less suitable for global localization accuracy purposes in high-level AD, in this paper we propose to be the first to explore novel use of LD for *defense purposes* in high-level AD localization. To concretely understand the potential of such a domain-specific defense opportunity, we analyze LD's defense properties in the following 5 general aspects.

1) General to lateral-direction localization attack. As mentioned above, LD can provide real-time information directly related to the *attack goal* of lateral-direction localization attacks. Thus, LD by nature has the potential to provide general defense capabilities to not only the existing attack designs such as those in §, but also their potential adaptive versions or other new attack designs in the future, as long as the attack goal is to cause lateral deviations.

2) Technology maturity. Benefit from the growing prosperity of Deep Neural Networks (DNNs), LD is already a mature technology that has been used for lane centering in low-level AD systems and vehicles, e.g., OpenPilot [82], Tesla Autopilot [83], GM Cadillac, Honda Accord, Toyota RAV4, Volvo XC90, etc. In fact, the existing camera-based LD solutions are quite robust to the dynamic environmental conditions. For example, Tesla Autopilot can effectively recognize lane lines even during a night storm [86]. Apart from DNN advancement, the camera auto-exposure and vehicle headlights also improve the usability of LD. Later in §, we also evaluate our defense on datasets with various environmental conditions and show that it is robust to low visibility conditions.

3) Defense deployability. Since today's high-level AD vehicles are all equipped with cameras for road object detection, using them for an LD-based defense solution is thus readily deployable without the need to install any new hardware. Moreover, many state-of-the-art LD models are publicly available [81], [87], including those used in industry-grade lane centering systems [82]; some high-level AD systems are also using LD for camera calibrations [62].

4) Defense coverage. For LD to be effective, the road at least needs to have lane line markings, which may not be available in local road segments such as intersections. Interestingly, due to real-world sensor noises and algorithm inaccuracies, the attacks to MSF localization are *fundamentally opportunistic*. For example, despite having a high overall attack success rate, latest lateral-direction localization attack cannot predict when and where a large deviation can be injected to the MSF outputs [58]. Due to such opportunistic property, the attacker *cannot deterministically cause a desired lateral deviation to appear and only appear in regions without lane line markings*. Such an attack property is fundamental to the MSF localization designs popularly used in high-level AD systems, since with such a design the attack effectiveness is fundamentally dependent on sensor noises and algorithm inaccuracies of other sources, which are neither observable nor controllable by a tailgating attacker [58].

Motivated by this insight, we analyze all attack traces evaluated in the *FusionRipper* paper [58] and our own evaluation later (§), and find that LD can indeed provide a decent practical defense coverage: among all attack starting points in the traces, only *0.8% (15/1813)* achieved the attack goal in road regions without lane line markings. Thus, an LD-based defense, if effective, can already provide protection for the 99.2% of the possible attack attempts. In addition, autonomous trucks, which are an important high-level AD application, are generally not subject to such limitation since they mainly operate on the "middle mile" (i.e., highways) [88], [89], where lane line markings are generally always available.

5) Independence to existing localization attack. To defend against existing attacks, a desired defense property is that the lane line markings perceived by LD are not already used in MSF localization. This is because if such information is already used, existing attacks might have already exploited their vulnerable periods (e.g., natural detection inaccuracies), making the additional use of such information for defense less likely to be effective. In representative MSF localization designs, the

LiDAR locator is the only one among the MSF inputs (§) that is possible to utilize lane line markings as features. Thus, we perform an experimental analysis to understand the dependency between state-of-the-art LiDAR locators [52], [63] and lane line markings in Appendix . Our results show that today's LiDAR localization algorithms have a *statistically-strong independence* of the lane line markings, very likely because lane markings is much less useful for global localization on a map compared to more unique road features such as buildings, roadside layouts, and traffic signs. This thus suggests that LD can indeed provide independent defense information to existing attacks. However, such independence property will disappear in adaptive attack settings (i.e., consider attacking LD after the defense is deployed). Thus, we require our defense design to be fully-aware of such adaptive attack surface (§), and also evaluate it later (§).

NOVEL LD-BASED DEFENSE DESIGN: LD^3

Considering the multi-dimensional defense opportunities above, in this paper we are motivated to design the first domain-specific lane detection-based defense approach against lateral-direction AD localization attack, named *LD^3 (Lane Detection based Lateral-Direction Localization attack Defense)*. In this section, we first describe the associated design challenges and then present the design details.

Design Challenges

Although LD comes with various defense opportunities, systematically leveraging it for AD localization defense purpose still needs to address the following main design challenges:

C1: Non-trivial design details for attack detection. Although at the high level LD can provide information directly related to lateral-direction attacks (i.e., lateral deviation to lane departure), at the detailed defense design level there are still many technical challenges we need to address, for example (1) incompatibility of the coordinate systems, i.e., LD is by default in *local* positioning coordinate system (i.e., within the ego lane), while the attack is in *global* coordinate system (i.e., the world coordinates); (2) choice of the attack-influenced information level for attack detection, e.g., directly at the spoofed GPS signal level or at the attack-influenced MSF output level; and (3) sufficient robustness to natural LD inaccuracies in practice, e.g., missing or incorrect detection, for minimizing possible false positives in attack detection.

C2: Need for AD-specific attack response design. Since high-level AD vehicles are travelling at high speed and by design cannot assume on-board human driver ready for take-over at any time (already the case in some commercial AD services [41], [47]), it is necessary to further design an attack response step that can (1) minimize the safety risks during response, and (2) assume no dependence on human assistance. For small robotic vehicles such as drones and rovers, prior works have considered using state estimation models to replace the attacked physical sensor after attack detection [90], [91]. However, such methods still count on human operators to take over as soon as possible since such state estimations cannot replace physical sensors for a prolonged duration due to drifting [90], not to mention that such models are suffering from much more severe motion model accuracy limitations when applied to the AD context (§). Thus, a new design is needed to achieve our AD-specific response goal above.

C3: Adaptive attack from LD side. While LD is currently independent to existing high-level AD localization attacks due to the lack of use (§), our defense-purpose use of it in LD^3 is inherently introducing a new attack surface from the LD side. In fact, recent works have already discovered concrete lateral-direction attacks against LD in production AD context [92]. To systematically account for such inherent adaptive attack surface, our defense design thus needs to consider the more challenging setup where both the attack detection and response designs cannot simply assume the LD side is trustworthy (and use it as the benign reference accordingly) when its outputs are inconsistent with the AD localization side.

Design Overview

In this section, we explain each design component in LD^3 and how they address the above design challenges. Fig. 8 shows an overview of LD^3 fitted in a typical high-level AD system.

Attack detection at MSF output level. As shown in Fig. 8, the attack detection step is performed in the localization module to constantly check the consistency between the LD outputs and original localization output and raise anomalies use popular anomaly detectors such as CUmulative SUM (CUSUM). To address the incompatibility of their coordinate systems mentioned in *C1*, we convert both into a unified lateral deviation representation w.r.t. the *lane centerline* since that's

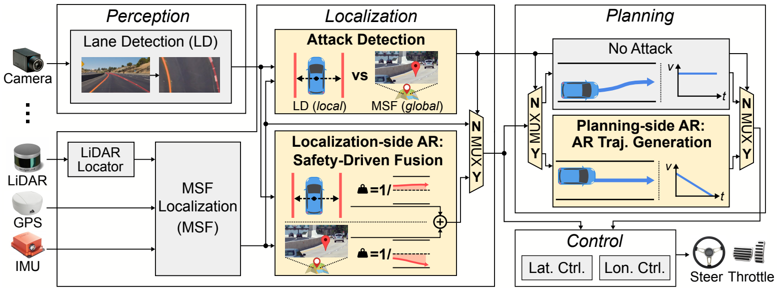


Fig. 8: Overview of LD^3 design integrated in a typical high-level AD system. New components are highlighted in yellow.

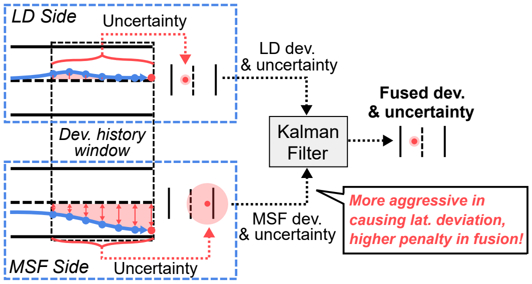


Fig. 9: Illustration of safety-driven fusion in the attack response (§).

directly related to the lateral-direction attack goal. Regarding the choice of the attack-influenced information level for attack detection, we choose to detect at the MSF output level rather than at the GPS output level since (1) in normal conditions, GPS positions can naturally have large noises while MSF outputs are at centimeter-level accuracy [52]. Thus, performing the detection at the MSF level can better reduce false positives; and (2) detecting at the MSF output level also allows LD^3 taking advantage of the *opportunistic* property of *FusionRipper*, for which the attacker cannot predict where and when will MSF exhibit large deviations. This thus can make it much more difficult for the attacker to easily bypass the detection by targeting locations without lane line markings. We also have designs for addressing false positives from common lane detection inaccuracies.

Attack response via safe in-lane stopping. As discussed in *C2*, we need a new AD-specific design for the Attack Response (AR) step. There are several common choices in human driving if the vehicle navigation is malfunctioning, for example maintaining driving in the current lane waiting for the system to recover, or pulling over to the road side. However, these cannot apply to the context of AD localization attacks, since without knowing the accurate real-time location, we cannot even know how to safely and correctly drive in the current lane or to road side. We also cannot blindly count on the LD outputs to drive due to the need to account for the adaptive attack surface on the LD side (*C3*). Thus, we consider the safest AR choice is to try to safely stop in the current lane, which has the minimal reliance on the attack-time localization accuracy for maximizing safety in the AR period. More importantly, on the attack side, since this minimizes the attackable duration after detection, it can fundamentally *bound* the attack-achievable deviation in AR period.

Safety-driven fusion for adaptive LD attack. Although the in-lane stopping AR strategy can already bound the attack-achievable deviation, it is still highly desired if we can minimize the attacker's impact on the localization accuracy during the AR period, since to safely stop, there is still a long stopping distance that the ego vehicle has to travel, especially when the speed is high (e.g., over 50 meters at 60 mph [93]). To account for the adaptive LD attack surface (*C3*), the key challenge is how to decide which side (LD or MSF) to trust when they are conflicting with each other in AR. Motivated by the safety-first principle in production AD design [94], we propose a novel *safety-driven fusion* design, which systematically decides the contributions from different fusion inputs based on their tendencies to cause unsafe driving; the higher such tendency is, the smaller their contributions will be to the final fusion output. In our problem context, such a tendency is judged by the deviation aggressiveness to cause lane departure, which will thus by default penalize the attacked side no matter it is from LD or MSF, leading to less attack-introduced deviation. To bypass this penalty and more effectively influence the fused results, the attacked side has to be less aggressive in lateral deviations. However, given the limit on the attackable duration imposed by the in-lane stopping AR strategy, the attack-achievable deviation during AR will still be reduced. Thus, under our AR design that bounds the attackable duration, such safety-driven fusion design can further fundamentally reduce the attacker's capability in causing safety damages during AR even in adaptive settings.

Attack Detection Design

As described above, we choose to perform the attack detection at the MSF output level, which is thus designed as a post-processing step in the localization module as shown in Fig. 8.

As mentioned in *C1*, the MSF and LD outputs are in different coordinate systems. Therefore, we first need to convert them to a unified coordinate system such that they are comparable. For MSF outputs, we obtain an *MSF-based lateral deviation to the lane centerline* (D_i^{MSF}) by querying the MSF position in the semantic map [95], which is a standard utility on high-level AD systems storing the road geometry information of the area that the AD vehicle is allowed to drive. For the LD outputs, we can calculate the lateral deviation to the centerline based on the left and right lane line polynomial functions (detailed in Appendix). However, real-world lane markings can be complicated and confusion sometimes. For example,

Algorithm 1 Attack detection by checking the consistency between MSF and LD

```
Notations: $MSF$: MSF position output; $LD$: lane detection output; $S$, $b$, $\tau$: CUSUM statistic, weight, anomaly threshold; $D$: deviation to lane centerline; $lw_{\text{map}}$: lane width from semantic map
Initialize: $S_0 \leftarrow 0$
 1: for each new lane detection output $LD_i$ do                  ▷ e.g., runs at 20Hz
 2:     $MSF_i \leftarrow$ latest MSF position     ▷ MSF is often more frequent than LD
 3:     $D_i^{\text{MSF}} \leftarrow$ MapLaneDev($MSF_i$)                ▷ MSF dev. (Appendix F)
 4:     $lw_{\text{map}} \leftarrow$ MapLaneWidth($MSF_i$)           ▷ lane width (Appendix F)
 5:     $D_i^{\text{LD}} \leftarrow$ LdDev($LD_i$, $lw_{\text{map}}$)         ▷ LD dev. to centerline (Alg. 4)
 6:     $S_i \leftarrow \max(0, S_{i-1} + |D_i^{\text{MSF}} - D_i^{\text{LD}}| - b)$ ▷ calc. CUSUM statistic
 7:     if $S_i > \tau$ then
 8:         attacked ← true                  ▷ report under attack if over threshold
 9:         break
10:     end if
11: end for
12: ⇒ switch to attack response
```

it is common to find that one of the lane lines missing or incorrectly detected in regions with lane splitting and merging. Therefore, we design two optimizations to calculate a more robust lateral deviation from the LD outputs leveraging the lane width from the semantic map (detailed in Appendix), which is a problem-specific improvement opportunity since in the main LD usage domain, low-level AD systems, such semantic maps are not generally available. Since LD^3 relies on the existence of lane line markings, we disable the attack detection prior to entering these regions based on the information from the semantic map.

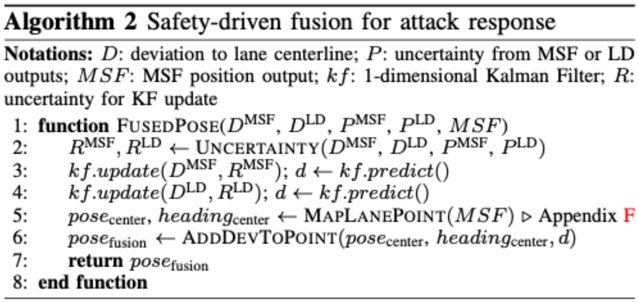
After obtaining the MSF- and LD-based lateral deviations, we can then use their deviation consistency to determine if MSF localization is under attack. To do so, we apply the widely-used CUSUM anomaly detector (line 6–10), which has shown high detection effectiveness in prior works [76], [96]. The CUSUM detector calculates a statistic $S_i = max(0, S_{i-1}{+}|r_i|{-}b); S_0 = 0$, where $r_i = D_i^{\text{MSF}} - D_i^{\text{LD}}$ is the residual between the MSF and LD lateral deviations, b is a weight to prevent the CUSUM statistic from monotonically increasing in the benign scenarios. We consider as under attack if S_i is over a certain threshold τ. Once an attack is detected, we then switch to the Attack Response stage.

Attack Response Design

As described in §8, we consider safe in-lane stopping as the safest AR choice. As shown in Fig. 8, the AR is composed of two components to safely drive the vehicle before stop: (1) AR trajectory generation on the planning side, and (2) safety-driven fusion of MSF and LD on localization side.

Planning-side AR: AR trajectory generation. The planning module in the high-level AD system periodically generates planned trajectories, for which the controllers take as speed and lateral position reference to produce throttling and steering commands. Thus, to enforce the AR goal, the planning module needs to generate an AR trajectory with a stopping motion. Since our AR goal is to stop in the ego lane, we design the AR trajectory to be aligned with the lane centerline. To reduce the speed, we then set a slowing-down speed profile on the AR trajectory based on a safe deceleration value used in high-level AD systems. Generally, a deceleration <4.6 $\mathrm{m/s^2}$ is considered as safe for maintaining steady control [97]. Thus, to calculate the speed profile of the AR trajectory, we apply 4 $\mathrm{m/s^2}$ as the deceleration, which is also defined in Baidu Apollo as the maximum allowed deceleration to ensure safety [62]. Note that since the original planning algorithms are typically designed under the assumption that the localization accuracy is high (i.e., cm-level [10]), we find directly re-using such algorithms in AR will result in unstable control since the planned trajectories are too sensitive to the larger localization errors and uncertainties after fusing the LD and MSF sides when one side is under attack. Thus, we directly set the planned trajectory as the centerline of the ego lane to achieve more stable control.

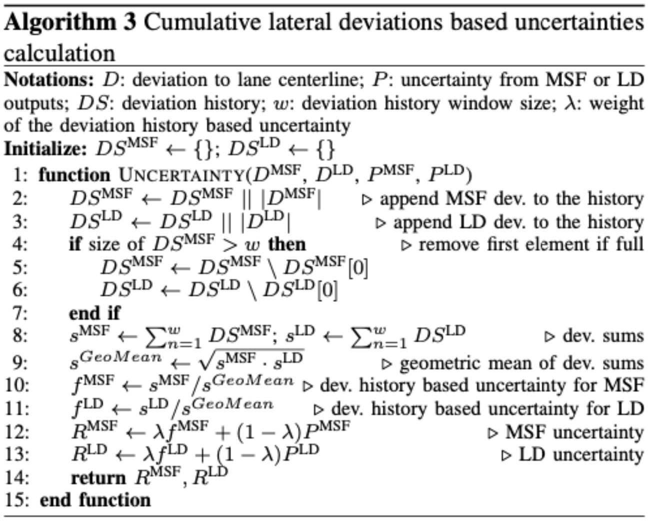
Localization-side AR: safety-driven fusion. As described in §, we need to design a safety-driven fusion algorithm on the localization side that can systematically fuse LD and MSF outputs while taking less contributions from the side that is more aggressive in causing lateral deviations. To achieve this, we leverage a classic fusion algorithm design, *Kalman Filter* (KF) based fusion, which can systematically determine the contributions of each fusion source using *uncertainties* [98], [99]. In the original design, the uncertainty score calculation are based on the noise-level measurements reported by the sources themselves, which thus are not suitable in attack settings since such measurements are also fundamentally under the attacker's control.



Algorithm 2 Safety-driven fusion for attack response

Notations: D: deviation to lane centerline; P: uncertainty from MSF or LD outputs; MSF: MSF position output; kf: 1-dimensional Kalman Filter; R: uncertainty for KF update

```
function FusedPose($D^{\text{MSF}}$, $D^{\text{LD}}$, $P^{\text{MSF}}$, $P^{\text{LD}}$, $MSF$)
    $R^{\text{MSF}}, R^{\text{LD}} \leftarrow$ Uncertainty($D^{\text{MSF}}$, $D^{\text{LD}}$, $P^{\text{MSF}}$, $P^{\text{LD}}$)
    $kf.update(D^{\text{MSF}}, R^{\text{MSF}})$; $d \leftarrow kf.predict()$
    $kf.update(D^{\text{LD}}, R^{\text{LD}})$; $d \leftarrow kf.predict()$
    $pose_{\text{center}}, heading_{\text{center}} \leftarrow$ MapLanePoint($MSF$)  ▷ Appendix F
    $pose_{\text{fusion}} \leftarrow$ AddDevToPoint($pose_{\text{center}}, heading_{\text{center}}, d$)
    return $pose_{\text{fusion}}$
end function
```



Algorithm 3 Cumulative lateral deviations based uncertainties calculation

Notations: D: deviation to lane centerline; P: uncertainty from MSF or LD outputs; DS: deviation history; w: deviation history window size; λ: weight of the deviation history based uncertainty

Initialize: $DS^{\text{MSF}} \leftarrow \{\}$; $DS^{\text{LD}} \leftarrow \{\}$

```
 1: function Uncertainty($D^{\text{MSF}}$, $D^{\text{LD}}$, $P^{\text{MSF}}$, $P^{\text{LD}}$)
 2:     $DS^{\text{MSF}} \leftarrow DS^{\text{MSF}} \,||\, |D^{\text{MSF}}|$          ▷ append MSF dev. to the history
 3:     $DS^{\text{LD}} \leftarrow DS^{\text{LD}} \,||\, |D^{\text{LD}}|$              ▷ append LD dev. to the history
 4:     if size of $DS^{\text{MSF}} > w$ then                  ▷ remove first element if full
 5:         $DS^{\text{MSF}} \leftarrow DS^{\text{MSF}} \setminus DS^{\text{MSF}}[0]$
 6:         $DS^{\text{LD}} \leftarrow DS^{\text{LD}} \setminus DS^{\text{LD}}[0]$
 7:     end if
 8:     $s^{\text{MSF}} \leftarrow \sum_{n=1}^{w} DS^{\text{MSF}}$; $s^{\text{LD}} \leftarrow \sum_{n=1}^{w} DS^{\text{LD}}$        ▷ dev. sums
 9:     $s^{GeoMean} \leftarrow \sqrt{s^{\text{MSF}} \cdot s^{\text{LD}}}$           ▷ geometric mean of dev. sums
10:     $f^{\text{MSF}} \leftarrow s^{\text{MSF}} / s^{GeoMean}$  ▷ dev. history based uncertainty for MSF
11:     $f^{\text{LD}} \leftarrow s^{\text{LD}} / s^{GeoMean}$     ▷ dev. history based uncertainty for LD
12:     $R^{\text{MSF}} \leftarrow \lambda f^{\text{MSF}} + (1-\lambda) P^{\text{MSF}}$                ▷ MSF uncertainty
13:     $R^{\text{LD}} \leftarrow \lambda f^{\text{LD}} + (1-\lambda) P^{\text{LD}}$                    ▷ LD uncertainty
14:     return $R^{\text{MSF}}, R^{\text{LD}}$
15: end function
```

To systematically realize our safety-driven fusion design between LD and MSF, we thus still leverage such uncertainties-based fusion framework but design novel uncertainty score calculation based on their tendencies to cause lane departure. Above lists the pseudocode for the uncertainty calculation. As shown (line 2–7), we store the historical lateral deviations from MSF and LD in two fixed-size windows. To obtain the uncertainties, we first calculate the cumulative deviations in these two windows, and then calculate their proportions to the geometric mean of them (line 8–11). We choose geometric mean over arithmetic mean since it can better penalize the source with a larger cumulative deviation. To increase the design flexibility, we include both our cumulative lateral deviation based uncertainty and the uncertainty from MSF/LD algorithms in the final uncertainty and use a weight λ to adjust their fractions (line 12–13).

With the uncertainties, we apply standard KF update/predict operations to fuse the MSF and LD lateral deviations. We then add the fused lateral deviation to the closest centerline point along the lateral direction based on the lane heading to instantiate a fused localization in the global coordinate system. Fig. 9 illustrates an example of the safety-driven fusion process.

DEFENSE EFFECTIVENESS EVALUATION

In this section, we evaluate LD^3 against the state-of-the-art lateral-direction attack targeting high-level AD localization.

Evaluation Methodology

Targeted AD system and attack. Since LD^3 is designed for high-level AD systems, we choose the industry-grade full-stack Baidu Apollo AD system [62] as a representative prototyping target. Specifically, Baidu Apollo adopts an MSF-based localization highly representative in both design (KF-based MSF) and implementation (state-of-the-art localization accuracy [52]). Note that although our evaluation here uses Baidu Apollo, the LD^3 design itself is generalizable to other industry-grade high-level AD systems; for example, later in § we also implemented it in Autoware for end-to-end physical

TABLE I: Details of the 562 total attack traces used in our evaluation and the *FusionRipper* attack effectiveness.

	Attack Trace #	Road Type	Avg. Speed	FusionRipper Attack			
				Attack Goal Dev	Best d	Best f	Success Rate
ka-local31	174	Local	10.9m/s	1.3m	0.5	1.2	99.4%
ka-local33	170	Local	9.5m/s	1.3m	0.3	1.3	98.3%
ka-highway36	182	Highway	26.3m/s	1.9m	0.3	1.3	100%
ka-highway18	36	Highway	24.8m/s	1.9m	0.3	1.3	100%

evaluation. For targeted attacks, we evaluate against the recent *FusionRipper* attack [58] since it is (1) the state-of-the-art and only lateral-direction localization attack that can break MSF localization; and (2) directly applicable to the above representative MSF implementation.

Real-world sensor traces and *FusionRipper* attack effectiveness. Since our evaluation target is the *FusionRipper* attack, we follow the same evaluation methodology as in their paper [58] and conduct our evaluation on real-world sensor traces from the KAIST complex urban dataset [100]. Specifically, we look for traces with camera data as required by LD^3, select the ones that the Apollo MSF can stably operate without attack [58], and apply *FusionRipper* from each consecutive timestamp as in [58]. In total, we obtain 562 attack traces summarized in Table I. These traces cover diverse driving scenarios, e.g., different road types (344 on local roads and 218 on highways), driving speeds (9.5 to 26.3 m/s), time-of-day (e.g., 36 in the morning, 182 around sunset time), and road conditions (e.g., 170 with snow on road).

For each trace, we follow the same method to identify the most effective attack parameters as in the *FusionRipper* paper. Note that in the table our attack goal deviation is larger than the original *FusionRipper* paper since they focus on the minimum urban lane width (i.e., 2.7 m) while we set the attack goal in a more realistic setting by measuring the lane widths in the dataset. This does not affect the attack effectiveness; as shown, the overall attack success rate is over 98%, which is consistent with the *FusionRipper* paper. In our evaluation, we exclude the scenarios without lane markings (e.g., when the vehicle is in an intersection) since it is out of the applicable domain for LD^3. As analyzed in §, the lack of coverage of such scenarios do not eliminate the defense value since only 0.8% of the attacks can possibly succeed in such scenarios and such successes are out of the attacker's control.

Lane detection and AD control effects under attack. Since LD^3 does not assume any specific requirement on the lane detector, we are free to use any state-of-the-art lane detector or even an ensemble of lane detectors. In our evaluation, we opt to the LD model used in OpenPilot [82], which is already used commercially for Automated Lane Centering. The KAIST traces include time-synchronized left and right camera frames from a front-facing stereo camera. In our evaluation, we regard the left and right cameras as independent cameras and run the LD model and calculate lateral deviations separately on them. We then aggregate their results to obtain an averaged lateral deviation on the LD side.

Since KAIST traces are collected under benign driving, we need to model the LD outputs when the AD localization is under attack. Same as the *FusionRipper* paper [58], we assume the lateral deviations in the MSF localization will be directly reflected as physical world deviations to the *opposite* direction (§). We then model the attack-influenced LD outputs by adding the physical world deviations to the lateral deviations calculated from the benign LD outputs. Later in §, we evaluate LD^3 in both end-to-end simulation and physical-world environments without such an assumption.

Baseline: SAVIOR. As a baseline, we evaluate the attack detection effectiveness of the closest alternative software-based method based on latest prior works for small robotics vehicles such as drones and rovers: physical-invariant based defenses [76], [77]. Specifically, we select SAVIOR [76] as a representative design since it adopts more principled state estimation models and thus shows superior detection performance over prior designs such as CI [77]. The detailed setup for SAVIOR evaluation can be found in Appendix .

Evaluation metrics. As LD^3 involves two defense stages with different defense goals, we separate the evaluation into attack detection and response evaluations. For attack detection evaluate, we plot the *ROC curves* to systematically show the TPRs and FPRs under different CUSUM parameters b and τ (§). In addition to ROC curves, we also report the maximum MSF lateral deviation before the attack is detected by LD^3. This *detection deviation* is a metric to indicate the detection timeliness, e.g., a detection deviation smaller than the lane straddling deviation (i.e., deviation to touch the lane line) means that the attack is detected early in time before it can cause any meaningful adversarial consequences. For attack response evaluation, we focus on the lateral deviations since our AR goal is to steer the vehicle to stop within the lane boundaries. In particular, we report two lateral deviation metrics with one measuring the *maximum deviation* before the vehicle fully

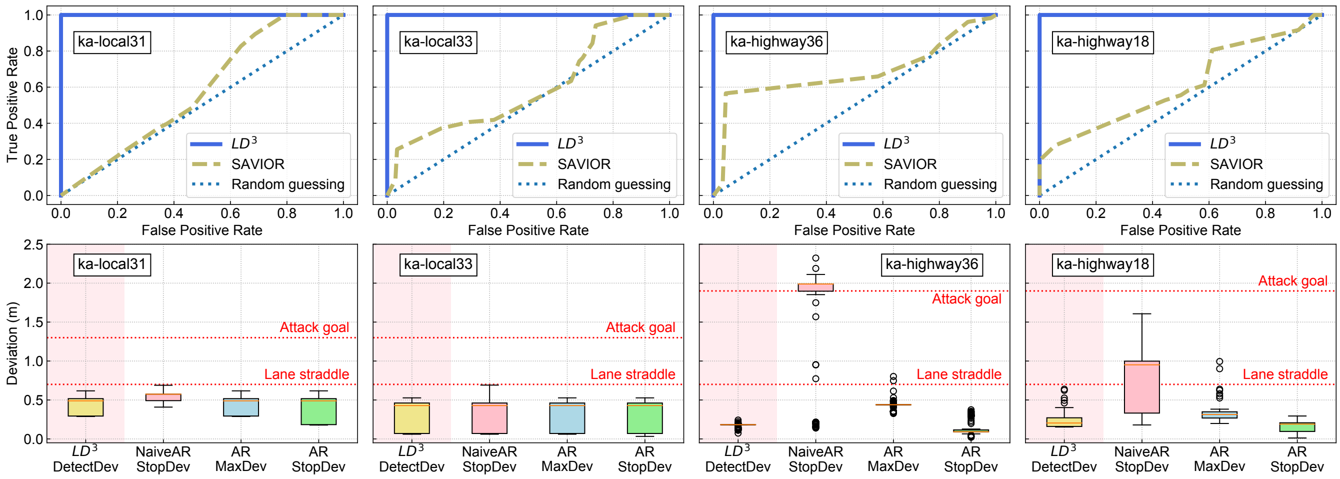


Fig. 10: (Top) Attack detection ROC curves; (Bottom) Detection and Attack Response (AR) deviations in the LD^3 evaluation.

stops, and another one measuring the final *stopping deviation*. In practice, the latter is more important since it will be the permanent deviation after the vehicle stops.

Attack Detection Effectiveness

Attack detection rates. The top figures in Fig. 10 show the detection ROC curves of LD^3 against *FusionRipper*. As shown, LD^3 can achieve effective detection with 100% TPRs and 0% FPRs on all 4 traces. During the searching for best CUSUM parameters, we find that in benign drivings, the differences between MSF and LD lateral deviations are always bounded within certain range (<0.6 m). However, in attacked drivings, *FusionRipper* will cause larger lateral deviations on the MSF side, which will also be reflected on the LD side in the opposite direction. Such a difference between benign and attacked drivings makes the attack easily detectable by LD^3. Fig. 11 shows an example of the benign and attacked MSF and LD lateral deviations and their CUSUM statistics. In the attacked case, *FusionRipper* launches the vulnerability profiling stage from t=1544686730 and discovers a vulnerable window at t=1544686765, where MSF starts to exhibit larger lateral deviations. Because of the distinctive MSF/LD consistency levels between the benign and attack cases, it is thus straightforward to set a CUSUM threshold to differentiate them.

Baseline comparison. As shown, SAVIOR's detection performance is only slightly better than random guessing and far from being an ideal detector. Such a poor detection performance would render SAVIOR unpractical since it will introduce lots of false positives in normal driving.

The reason behind the poor detection performance in the AD context is twofold. First, compared to drones and rovers, the physical dynamics of the vehicle are much harder to model due to the complex physical moving characteristics, e.g., tire-road frictions, aerodynamic forces, road bank angles, etc. [101]. For example, prior study [79] finds that the error of kinematic bicycle model increases very fast at high speeds (e.g., 25 $\mathrm{m/s}$) or on curvy roads (e.g., steering angle at 4°). In comparison, the bicycle model used in SAVIOR is reported having an average position error of 0.33 m *within 0.8 sec* under low-speed settings (e.g., 13.8 m/s) in [78] and its error keeps accumulating as time progresses; comparably, the same bicycle model incurs an average error of 1.076 m on *ka-local31* within 1 sec, where the trace contains many turns and curvy roads.

Second, the attack deviation goals in the AD context can be much smaller but still being safety-critical. While SAVIOR is effective at detecting attacks on small robotics vehicles such as drones with large deviation goals (e.g., $\sim$50 m [76]), attacks targeting high-level AD systems requires much smaller deviation and thus harder to detect. For example, even lateral deviations <0.5 m are enough to cause lane departure on narrow urban roads (e.g., 2.7 m wide [102]).

Attack detection deviations. To evaluate attack detection deviations, we choose a CUSUM weight $b = 0.6$ and threshold $\tau = 0.1$, which can achieve best attack detection effectiveness on all traces. For example, the detection deviation in Fig. 11 is 0.36 m under these CUSUM parameters. The bottom figures in Fig. 10 show the distributions of maximum deviations *FusionRipper* has reached before being detected by LD^3 (box plots with pink background). As shown, LD^3 can promptly detect the attack before it can even cause lane straddling, and the average detection deviations are all below 0.5 m. However, there do exist two attack cases in *ka-highway18* that the detection deviations are close to the lane straddling deviation. This

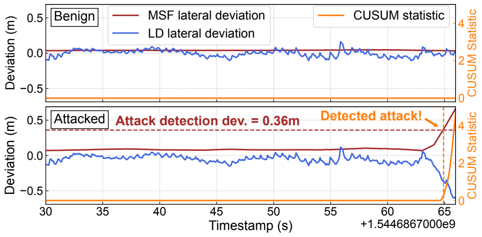


Fig. 11: Benign and attacked MSF/LD lateral deviations and CUSUM statistics.

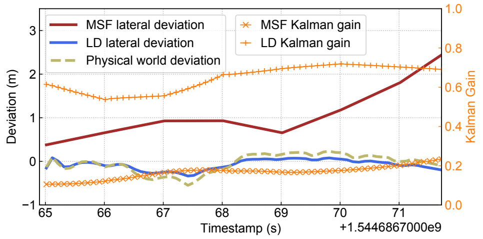


Fig. 12: MSF/LD and physical world deviations and Kalman gains during AR period.

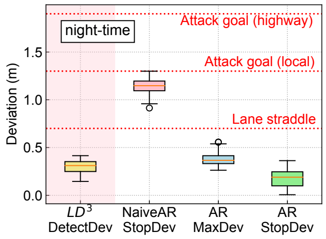


Fig. 13: Detection and AR deviations on the night-time trace.

is because in these attack cases, the lateral deviation at the MSF side raises very rapidly between detection intervals such that the deviation has already reached a large number (e.g., 0.63 m) before LD^3 has a chance to perform the detection. Nevertheless, none of the attack cases are detected after *FusionRipper* starts to cause lane straddling and all of them are far away from reaching the attack goal deviation.

Attack Response Effectiveness

The distributions of the maximum deviations and final stopping deviations are shown in the bottom figures in Fig. 10 (box plots without background colors). During the AR periods, none of the attack cases have a maximum or stopping deviation over the attack goal deviation (1.3 m for local and 1.9 m for highway). Despite 4 attack cases on the highway traces have maximum deviations exceed the lane straddling deviation (0.7 m), their stopping deviations are all corrected back to be within the lane boundaries. This shows that our AR design (§) is effective at keeping the vehicle within the lane boundaries when it stops, which can prevent the much more dangerous situation where it stops out of the ego lane.

Moreover, comparing between local and highway, the highway traces often have *larger maximum deviations* and *smaller stopping deviations*. This is because the driving speeds when the attacks are detected on the highway traces (27.3 $\mathrm{m/s}$ on avg.) are much higher than that on the local traces (3.8 $\mathrm{m/s}$ on avg.). This leads to a much longer AR period on highways ($\sim$7 sec on avg.) than that on local roads ($<$1 sec on avg.). As a result, *FusionRipper* can keep causing larger lateral deviations after the attack is detected, but in the meanwhile, our AR design can also correct more given the longer AR period.

Fig. 12 shows an example of the MSF/LD and physical world deviations during the AR period on *ka-highway36*, where the maximum deviation and stopping deviation are 0.52 m and 0.19 m, respectively. In this example, since *FusionRipper* keeps increasing the deviation on the MSF side, the safety-driven fusion (§) penalizes the lateral deviations on the MSF side with higher uncertainties and thus results in smaller Kalman gains, which indicate the weights of the inputs in KF update. Consequently, the fusion process prioritizes the lateral deviations on the LD side, which are similar to the physical world deviations, and the lateral controller thus can steer the vehicle towards the right direction.

Comparison with naive AR design. A naive AR design, named NaiveAR, applies the maximum deceleration to stop but still keeps using the MSF outputs for steering. Such design is similar to the *in-lane stop* planning scenario that Baidu Apollo adopts to handle emergencies [103]. To evaluate this, we record the MSF lateral deviations at the end of AR periods and regard them as the stopping deviations based on the control assumption (§). The stopping deviations of NaiveAR are shown in Fig. 10. Because of the longer AR periods on the highway, the stopping deviations under NaiveAR are significantly higher than that using our complete AR design, especially on the highway traces. In particular, since the lateral deviations on *ka-highway36* increase very quickly, over 75% of the attacked cases still reaches a lateral deviation higher than the attack goal deviation, which consequently leads to >75% attack success rate for *FusionRipper* on *ka-highway36* despite the attacks are correctly detected. On the other hand, with the complete AR design, none of the attack cases can be even deviate out of the lane boundaries.

Evaluation under Limited Visibility

Trace collection and defense evaluation setup. We collect a *night-time* driving trace at around 11 p.m. our local time using an Advanced Driver-Assistance System (ADAS) device named EON [104], which is the official device to run OpenPilot [82]. Specifically, we record the localization and LD outputs during the trace collection for the defense evaluation. The trace is

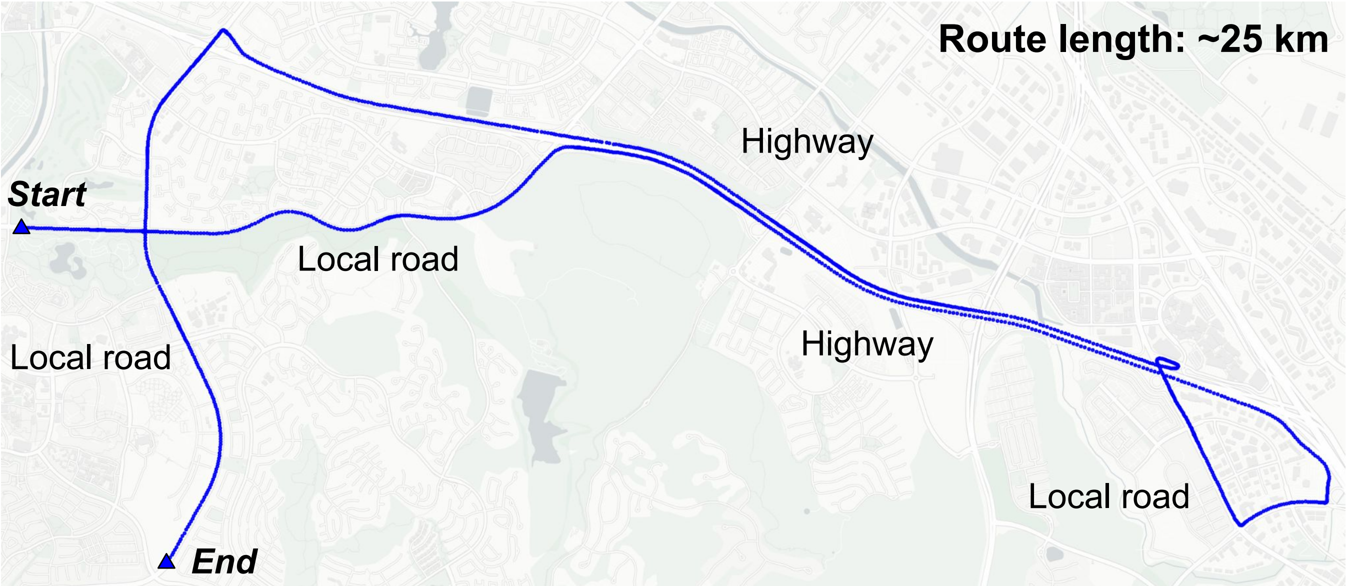


Fig. 14: Route of the night-time sensor trace collected using EON [104] in §.

$\sim$25 km in length with 3 local road and 2 highway segments as shown in Fig. 14. Since EON does not provide LiDAR data, we are not able to run MSF and *FusionRipper* attack. To model the attack effect, we apply the lateral deviations from the *most aggressive* attack trace in *ba-local* trace used in the *FusionRipper* paper [58] to the localization outputs, which only takes 10 sec from the start of attack to reaching a 2 m lateral deviation. This is similar to the prior works where they directly apply the attack traces in the target systems for attack detection evaluation [76], [77]. Specifically, we apply the attack trace consecutively to all road segments excluding the intersections, which results in 98 attacked and 98 corresponding benign segments in total.

Defense effectiveness. Similar to results on KAIST traces (§), LD^3 can achieve effective attack detection with 100% TPR and 0% FPR on the night-time trace. The attack detection and AR deviations are shown in Fig. 13. As shown, even under such low-light condition, LD^3 can still timely detect the attack with an average detection deviation of 0.29 m. Consistent with findings in §, the stopping deviation on the real vehicle trace is only 0.17 m on average, which means LD^3 is effective at stopping the vehicle within the lane boundaries. In comparison, NaiveAR has a stopping deviation much higher than LD^3, where one attack segment (maximum deviation is 1.33 m) exceeds the goal deviation for local roads.

END-TO-END EVALUATIONS

In this section, we implement LD^3 on 2 open-source full-stack AD systems, Baidu Apollo [62] and Autoware [63], and evaluate LD^3 under end-to-end drivings in both simulation and the physical world. The demo videos are available on our project website at **https://sites.google.com/view/ld3-defense**.

Evaluation in AD Simulator

Experimental setup. We implement LD^3 in Baidu Apollo v5.0.0 [62] following the design in Fig. 8. Specifically, we reuse the SCNN model [81] for LD, which is currently used only for camera calibration in Baidu Apollo. We run the complete Baidu Apollo AD system with all functional modules enabled in a production-grade AD simulator, LGSVL [105]. Since LGSVL does not provide LiDAR locator maps required for MSF, we instead run Baidu Apollo localization in the Real-Time Kinematic mode, which directly takes the ground truth positions from LGSVL. To simulate the *FusionRipper* attack effect, we add the lateral deviations from the same attack trace used in § to the localization outputs.

We evaluate the benign and attacked drivings with LD^3 in 4 driving scenarios on two LGSVL maps: Single Lane Road (SLR) and San Francisco (SF). Specifically, the SLR map is a long straight road, and we create a low-speed (SLR-Low) and high-speed (SLR-High) driving scenario on it by adjusting the maximum cruising speed in Apollo planning. The SF map is a 1:1 re-creation of a portion of the San Francisco city, from which we select a straight (SF-Straight) and a curvy

TABLE II: Maximum deviations to lane center and attack consequences under different defense settings in the 4 simulation scenarios in §. Each setting was run for 10 times with randomized attack starting times. Benign driving with LD^3 is also presented and was run for 10 times. The maximum deviations are represented as (mean, std) in meters.

Simulation scenario	Lane straddle dev	Attacked						Benign	
		LD^3		LD^3-NaiveAR		No Defense		LD^3	
		Max dev	Consequence	Max dev	Consequence	Max dev	Consequence	Max dev	Consequence
SLR-Low	0.83	0.47, 0.08	Stop in lane	1.69, 0.06	Stop w/ lane straddle	7.94, 0.05	Fall off road	0.07, 5e-5	Reach destination
SLR-High	0.83	0.69, 0.06	Stop in lane	1.64, 0.16	Stop w/ lane straddle	7.93, 0.04	Fall off road	0.07, 5e-5	Reach destination
SF-Straight	1.00	0.67, 0.23	Stop in lane	1.02, 0.01	Hit curb	1.84, 0.16	Hit tree or barrier	0.14, 7e-4	Reach destination
SF-Curvy	0.75	0.43, 0.14	Stop in lane	0.90, 0.12	Hit lane divider	0.97, 0.14	Hit lane divider	0.31, 0.01	Reach destination

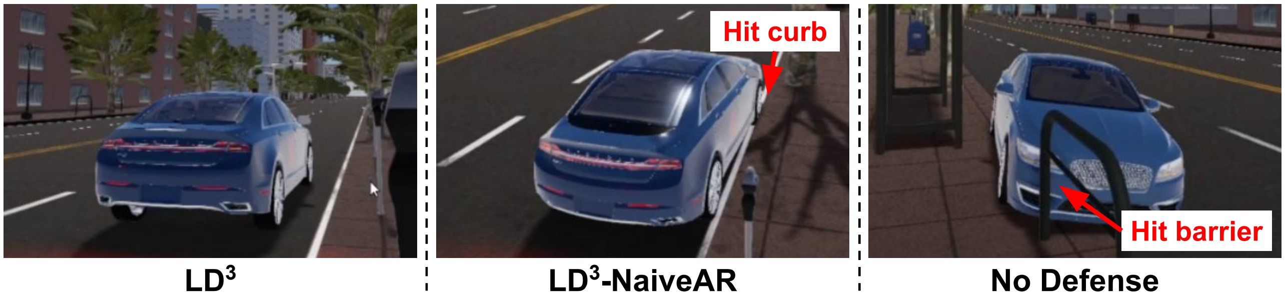


Fig. 15: Simulation snapshots of the vehicle stopping locations under the 3 defense settings in SF-Straight.

road (SF-Curvy). In our evaluation, we also include the LD^3 variant with the naive AR design (§) and a setting without any defenses. We repeat the simulation for 10 times with different attack starting times for each combination of simulation scenarios and defense settings.

Results and demos. Our simulation results show that the attack detection rates for both LD^3 and LD^3-NaiveAR are all 100% in the 10 runs, and none of the benign drivings are falsely detected as under attack. Table II shows the maximum lateral deviation achieved in the whole simulation (including both attack detection and response periods) in each scenario/defense setting and the corresponding vehicle stopping location. As shown, with LD^3, the average maximum deviations are smaller than lane straddling deviation in all 4 scenarios and the vehicle can always safely stop in the lane. In comparison, due to the blind trust of the localization outputs in the AR period, LD^3-NaiveAR has much higher maximum deviations than LD^3 and the vehicle's stopping locations are either lane straddling or already crashing into the road curb/barrier. Nevertheless, the No Defense setting is even worse than LD^3-NaiveAR, where the vehicle is simply deviated to fall off the road in SLR-Low and SLR-High. Snapshots of the vehicle stopping locations in SF-Straight are shown in Fig. 15. The demos of the 4 simulation scenarios and 3 defense settings are available on our project website.

Evaluation on AD Development Chassis of Real Vehicle Size and Closed-loop Control

Experimental setup. We experiment on an AD chassis as shown in Fig. 16, which is specifically designed for Level-4 AD system prototyping and testing. The chassis is of a real vehicle size, capable of closed-loop control, and fully equipped with Level-4 AD sensors including LiDAR, GPS, IMU, cameras, RADARs, and ultrasonic sensors. Since AD vehicle testing is not allowed to be on public roads by default, we reserve a parking lot in our institute for the experiments. Specifically, we mark a straight traffic lane with 3.5 m width (the most common lane width in KAIST dataset and our night-time driving trace) in the parking lot and create the corresponding semantic map for Autoware.

We ported LD^3 to the Autoware AD system [63], which is currently supported by the AD chassis. To facilitate the attack, we apply the same *FusionRipper* attack trace used in § and § to the localization outputs in Autoware. Unlike OpenPilot and Baidu Apollo, the lane detector in Autoware can only detect lane lines in pixels rather than in the world coordinates. Therefore, we directly obtain the ground truth lane line information from the map using the unmodified localization outputs, since LD is already a mature technology (§) and has been shown to be quite accurate in § and §. We enable the relevant components in Autoware including localization, global/local plannings, and control. During the experiments, the AD chassis is completely driven by Autoware unless taken over by us from a remote controller in emergency situations. We evaluate three defense settings: (1) *w/ LD^3 w/ attack*, (2) *w/o LD^3 w/ attack*, and (3) *w/ LD^3 w/o attack*. For each, we experiment

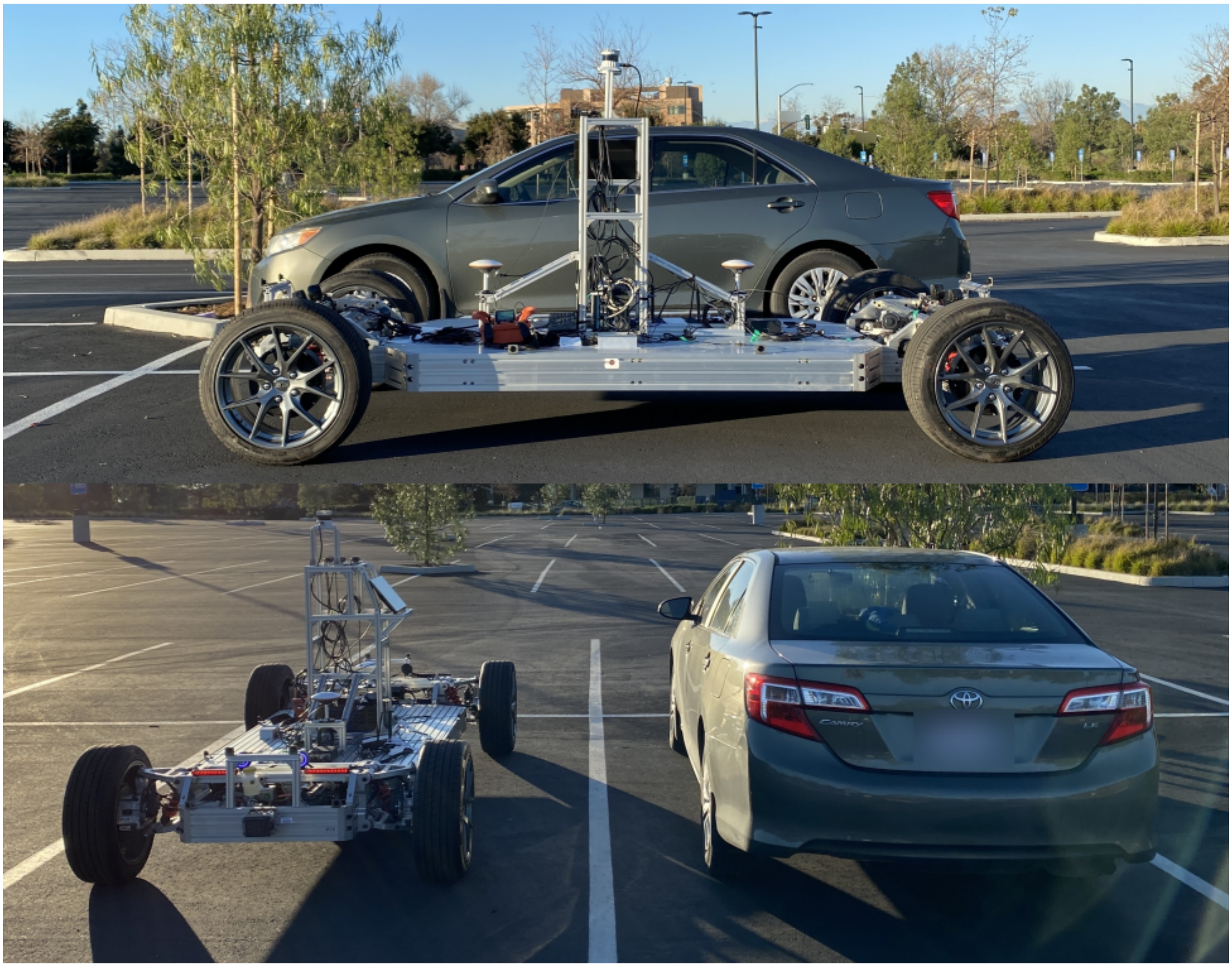


Fig. 16: Side-by-side views of the AD development chassis used in § and a Toyota Camry.

TABLE III: The detection, maximum, and stopping deviations in the three settings at two different driving speeds. We repeat the experiments for *w/* LD^3 *w/ attack* for 3 times and report the (mean, std) deviations. We do not repeat the other two settings as they are quite stable.

Speed	w/ attack				w/o attack	
	w/ LD^3			w/o LD^3	w/ LD^3	
	Det dev	Max dev	Stop dev	Max/Stop dev	Max dev	Stop dev
4 m/s	0.07m, 0.01m	0.36m, 3e-3m	0.05m, 0.05m	2.59m	0.13m	8e-3m
2 m/s	0.02m, 2e-3m	0.27m, 0.04m	0.01m, 1e-3m	2.23m	0.11m	7e-3m

TABLE IV: Maximum physical deviations can be achieved without being detected under various LD fluctuation assumptions. The percentages indicate the probabilities of such fluctuations.

Trace	LD fluctuation (μ, σ)	Max physical world deviation		
		0 (100%)	μ (50%)	$\mu + 3\sigma$ (0.3%)
ka-local31	0.12m, 0.08m	0.7m	0.82m	1.06m
ka-local33	0.14m, 0.10m	0.7m	0.84m	1.14m
ka-highway36	0.29m, 0.10m	0.7m	0.99m	1.29m
ka-highway18	0.20m, 0.11m	0.7m	0.90m	1.23m

in driving speeds of 2 m/s (4.5 mph) and 4 m/s (9 mph) for safety concerns. We prolong the AR stage by using deceleration <3 m/s^2 in both cases to better showcase the driving behaviors during AR. Specifically, we repeat the experiments for 3 times for *w/ LD^3 w/ attack*. Since the other two are always quite stable, we thus do not record more iterations for those experiments.

Results and demos. Table III shows the detection, maximum, and stopping deviations under the three settings. As shown, LD^3 on average can detect the attack when the vehicle's physical deviation is still small and start the AR stage. Within the AR period, the average maximum deviations are 0.36 m and 0.27 m at speeds of 4 m/s and 2 m/s, respectively, and the final stopping deviations are always within 0.1 m. In comparison, without LD^3, the vehicle keeps deviating and we have to manually press the emergency button on the remote to prevent it from crashing into the curb. Such a distinctive driving behaviors with and without LD^3 are consistent with our trace-based (§) and simulation results (§). Without the attack, the vehicle's trajectories well align with the road centerline (i.e., the reference trajectory Autoware plans to enforce) and eventually complete the route and stop at the center of the lane. We also record demo videos of the vehicle driving behaviors under the three settings (videos are available on our website).

EVALUATION AGAINST ADAPTIVE ATTACKS

In this section, we take a step further to examine LD^3's capability under potential adaptive attacks, including (1) an idealized stealthy attack that can evade the detection, and (2) the latest LD-side attack, which is the inherent new attack surface introduced by LD^3 approach (§).

Stealthy Attack Evaluation

In this evaluation, we analyze the maximum lateral deviations that a hypothetical stealthy attack can achieve by assuming stronger and unrealistic attack capabilities.

Evaluation methodology. Based on the CUSUM anomaly detection formulation (§), the attack should satisfy $S_{i-1} + |D_i^{\text{MSF}} - D_i^{\text{LD}}| - b < \tau$ in order to prevent detection. Assuming the last CUSUM statistic $S_{i-1} = 0$, the maximum MSF lateral deviation without being detected is thus $D_{i,max}^{\text{MSF}} = D_i^{\text{LD}} + \tau + b$, which is also the maximum physical world deviation given the control assumption (§). Since τ and b are fixed in the defense, the attacker can carefully select a timing where the LD has a large lateral deviation fluctuation to the actual vehicle location due to detection noises, and apply the MSF lateral deviation to the same direction as the LD's fluctuation direction to achieve a large physical world deviation. Therefore, the attacker's capability on capturing a particular LD fluctuation window determines the maximum physical world deviations she can achieve without being detected. Thus, we evaluate the maximum physical world deviations by assuming various levels of LD fluctuations that the attacker can capture.

Assumptions on attack capabilities. In this evaluation, we assume the attacker has very unrealistic attack capabilities in order to achieve such a stealthy attack. In particular, the attacker should have a *white-box knowledge* on (1) where exactly on the road that the LD will have a large fluctuation and how much it is, and (2) the attack detection method and parameters used in the target AD system. Moreover, the attacker should also have *precise* and *instantaneous* control over the lateral deviations in the MSF localization outputs in order to execute such attack when large fluctuations appear.

Results. Table IV shows the maximum physical world deviations that the stealthy attack can achieve under different LD fluctuation assumptions. Specifically, we calculate LD fluctuation distributions in each trace and assume that the attacker knows where a certain level of fluctuation happens. Without any such assumptions, the attacker can at most inject $\tau + b = 0.7$ m lateral deviation, which is just about to touch the lane boundaries. On the other hand, the attacker can *at most* cause 0.99 m and 1.29 m lateral deviations on the 4 traces if she can capture an average and a 3-σ LD fluctuation, respectively. Note that the probabilities of such fluctuations to appear are 50% and 0.3% according to the normal distribution. In conclusion, even under very unrealistic attack assumptions, the maximum lateral deviations are still less than the local road attack goal (1.3 m) for *FusionRipper*, which shows that LD^3 is quite effective at bounding the lateral deviations. Moreover, it also highlights that LD is indeed a mature technology (§) suitable for defense given its high stability.

LD-side Adaptive Attack Evaluation

Evaluation methodology. We explore the defense capability of LD^3 against the latest LD attack in production low-level AD systems, named Dirty Road Patch (DRP) attack [92], which is designed to affect the detected lane line shapes to mislead the automated lane centering system to drive the vehicle out of the lane boundaries. In LD, the lane line shapes are represented as polynomial functions, which are used in LD^3 to calculate the vehicle's lateral deviations (Appendix). From the 40 attack traces used in the original DRP attack paper, we extract the attacked lane line polynomials in each frame and calculate an averaged LD deviation trace. In LD^3 design, LD attacks cannot disrupt the driving behaviors before the attack is detected since only MSF outputs are used for navigation at this moment. To cause vehicle deviations, the LD attack has to trigger

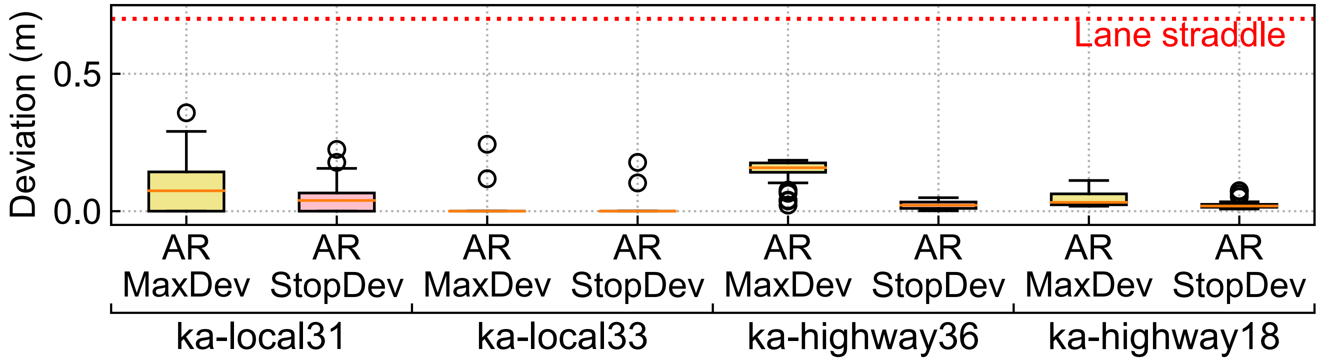


Fig. 17: Maximum and stopping deviations during the AR period under the LD attack.

the detection in the first place and affect the *fused* localization in the AR period (§) in order to affect the vehicle control. Therefore, we focus on the AR period in our evaluation. To model the DRP attack effect, we apply the deviation trace (start from the detection deviation 0.7 m) to the LD side in the KAIST traces. Since the MSF side is benign and should generally well-align with the physical positions of the vehicle, we set the MSF outputs in the AR period with the same deviation as the fused localization, but to the opposite direction based on the control assumption (§).

Results. Fig. 17 shows the maximum and stopping deviations in KAIST traces. As shown, none of them is able to even cause lane straddling. On average, the maximum and stopping deviations in the AR period are only 0.08 m ($\delta = 0.08$ m) and 0.02 m ($\delta = 0.03$ m), respectively. Such a result indicate that LD^3 is quite robust to adaptive attack to the LD side as well. This is because the safety-driven fusion (§) in LD^3 can effectively penalize the more aggressive source in the driving context, which in this case is the attacked LD outputs, and prevent the fused localization from being influenced by it.

LIMITATIONS DISCUSSION

Defense coverage of lane detection. In this work, we are the first to explore the novel usage of LD for defense. However, as a defense relying on LD, a potential limitation is the lane line marking coverage. However, as we analyzed in §, the non-deterministic nature of attacks to MSF localization greatly alleviate such a limitation, where an LD-based defense has the potential to defend against the majority (99.2%) of the attack attempts. In addition, for important AD applications such as autonomous trucks, they are naturally not subject to such limitation as they mostly operate on highways [88], [89]. At design level, since high-level AD systems come with semantic maps with accurate road geometry information, LD^3 knows exactly where are the regions without lane line markings and can temporarily disable the defense in such regions (§). To address this limitation, a potential future improvement is to also consider other road markings available in such regions, e.g., stop lines [106] and crosswalk markings [107] in intersections, to help localize the vehicle and to detect MSF deviations. Nevertheless, it is unclear how prevalent such road markings are and how mature and robust the existing perception algorithms are to recognize such road markings.

Simultaneous attacks to MSF and LD. Since LD^3 leverages LD to detection lateral-direction attack on MSF, attacks that simultaneously target MSF and LD can thus potentially bypass our detection. In fact, such a vulnerability is a general limitation for CPS security research that uses sensor cross-checking/fusion for defense purposes [18], [108]–[112]. However, in practice, the defense value of LD^3 highly depends on *whether such a simultaneous attack already exists or can be easily achieved*. For MSF and LD, neither of them holds today, since (1) although individual attacks on MSF or LD exist, no existing work shows that they can be effectively *coordinated and synchronized* to achieve simultaneous attack effect control, and (2) it is far from trivial to achieve this with existing individual attack vectors. Specifically, among the attack vectors on camera [84], [92], [113]–[118], only three works [92], [115], [118] actually evaluated and shown attack effectiveness on LD in realistic AD settings. All these three works consider adding malicious patterns to the ground (e.g., via road patch or stickers) as the attack vector. However, considering the non-deterministic nature of the existing high-level localization attacks (§), it would be hard, if not impossible, for the attacker to figure out where to place the attack pattern beforehand, not to mention how to carefully synchronize the malicious pattern with the localization-side attack to effectively bypass LD^3. Therefore, we consider such simultaneous attack design neither already exists nor can be easily achieved, and leave the systematic exploration of its feasibility as a future research direction.

RELATED WORK

AD localization attacks. Since AD systems rely on sensors for localization, prior works have proposed sensor spoofing/jamming attacks targeting GPS, LiDAR, IMU, camera, RADAR [34], [58], [67], [68], [70], [71], [113], [114], [119], [120]. Among them, only *FusionRipper* [58] has shown to be able to break the MSF localization on high-level AD systems and cause lateral deviations in MSF outputs. Thus, in this work we target *FusionRipper* and show that LD^3 can effectively detect *FusionRipper* and can steer the vehicle to safely stop in ego lane.

Physical-invariant based defenses. Recently, researchers propose physical-invariant based defenses, CI [77] and SAVIOR [76], to detect sensor attacks such as GPS spoofing by cross-checking sensor measurements with system state estimations based on the physical invariants, i.e., the relationships between system states and control inputs. However, as shown in §, the direct adaptation of existing physical-invariant based approach is largely limited because of the complexity of physical dynamics and much smaller attack deviation goals in the AD context. In addition, none of them has proposed attack response designs, which is especially important for AD systems (§). Nevertheless, such physical-invariant based attack detection methods are complimentary to LD^3 and can be incorporated into our design for attack detection if the accuracy of state-estimation model can be further improved.

Attack response/recovery. According to a survey on the broader Cyber-Physical Systems security, existing defenses mostly focus on attack detection and very few works studied attack responses [121]. Particularly, Choi et al. [90] and Zhang et al. [91] recently propose *attack recovery* methods, which apply similar state estimations as above to replace attacked sensors in the attack recovery period. Thus, they suffer from the same model accuracy limitations in the AD context. Moreover, they intend to maintain normal operations of the system for a short duration until the system is taken-over by the human driver, which does not exist on high-level AD vehicles when deployed commercially [41], [47]. Additionally, attack responses in high-level AD systems require more careful design on AR trajectories (§) to safely navigate the vehicle.

CONCLUSION

In this work, we perform the first systematic exploration of the novel usage of lane detection (LD) to defend against lateral-direction attacks in high-level AD localization. We design the first domain-specific LD-based defense approach, LD^3, that is capable of both real-time attack *detection* and *response*. Our evaluation on real-world AD sensor traces show that LD^3 is much more effective than directly-adapted physical-invariant based defenses at attack detection with accurate and timely detection. We also show that LD^3 can safely stop the vehicle in the current lane upon detection. We implement LD^3 on two open-source high-level AD systems and evaluate its effectiveness under end-to-end driving with closed-loop control in both simulation and the physical world. We also evaluate against two adaptive attacks and find that LD^3 is robust to an idealized stealthy attack that aims to evade detection and the latest LD-side attack targeting the response stage.

Converting LD outputs to lateral deviations

The LD output consists of the detected left and right lane lines, which are represented as polynomial functions in the bird's eye view [62], [82]. An example of the polynomial functions is shown in Fig. 18. For these polynomial functions, the absolute values at $x = 0$ represent the vehicle's distances to the lane lines, d_{left} and d_{right}. Therefore, we can calculate the lateral deviation to the lane centerline by

$$lw/2 - d_{\text{left}} \quad \text{or} \quad d_{\text{right}} - lw/2, \tag{9}$$

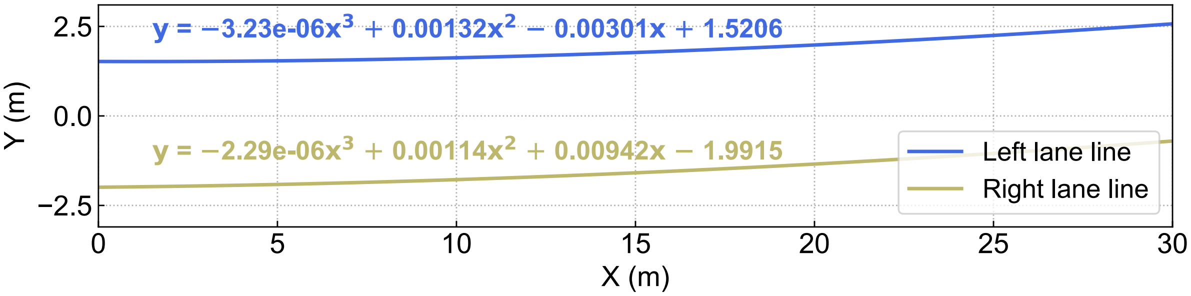


Fig. 18: Example of left/right lane line polynomial functions.

Algorithm 4 Calculation of LD deviation to lane centerline

Notations: lw_{map}: lane width from map; $poly(\cdot)$: polynomial function fitted on detected lane line; d: distance to lane line; D: deviation to lane centerline

```
function LDDev($LD$, $lw_{\text{map}}$)
    $d_{\text{left}} \leftarrow |LD.poly_{\text{left}}(0)|$ if $LD.poly_{\text{left}}$ else $\infty$    ▷ dist. to left line
    $d_{\text{right}} \leftarrow |LD.poly_{\text{right}}(0)|$ if $LD.poly_{\text{right}}$ else $\infty$    ▷ dist. to right line
    if $LD.poly_{\text{left}}$ and $d_{\text{left}} < d_{\text{right}}$ then    ▷ left line is correct
        $D \leftarrow lw_{\text{map}}/2 - d_{\text{left}}$    ▷ dev. to centerline; $+$: left, $-$: right
    else if $LD.poly_{\text{right}}$ and $d_{\text{right}} < d_{\text{left}}$ then    ▷ right line is correct
        $D \leftarrow d_{\text{right}} - lw_{\text{map}}/2$
    else    ▷ if none of the lane lines are correctly detected
        $D \leftarrow$ last calculated $D$    ▷ re-use last dev. to centerline
    end if
    return $D$
end function
```

where lw is the lane width. We calculate the lateral deviation as a *signed* number to differentiate the deviations to the left (positive) and to the right (negative).

Although we can also obtain the lane width from the lane line polynomials (i.e., $lw_{\text{poly}} = d_{\text{left}} + d_{\text{right}}$), as mentioned in §, it is not uncommon that one of the lane lines is missing or incorrectly detected in real world driving, e.g., when the current lane splits into a through lane and a left or right turn lane. In such cases, directly using the distances from the polynomial functions would result into a wrong lateral deviation. To address this, we include two optimizations in the lateral deviation calculation: (1) instead of estimating the lane width from the polynomial functions, we query the current lane width from the semantic map, and (2) prioritize the lane line with a smaller distance to the vehicle by using it to calculate the lateral deviation in Eq. 9. This is because for the lane splitting scenario mentioned above, the incorrectly-detected lane line often has a much larger distance compared to the correctly-detected one. A special handling is that when both lane lines are incorrectly detected, which is very rare in SCNN [81] and never occur in OpenPilot LD model [82], we will reuse the previously calculated lateral deviation.

Evaluation of LiDAR Localization Dependency on Lane Line Markings

Evaluation methodology. To evaluate the dependency of LiDAR localization on lane line markings, we first create two traces of modified LiDAR data: one without lane line markings (denote as *no-marking*) and another with incorrect lane line markings (denote as *wrong-marking*). Next, we execute the LiDAR locators on the original LiDAR trace as well as on the two modified traces. *If a LiDAR locator does not rely on the lane line markings, we should observe a high similarity between the original and the modified executions.*

Specifically, LiDARs scan the surrounding environment and output Point Cloud Data (PCD), which stores the 3D positions and intensities of the reflected laser points. Since the lane line markings will exhibit distinctively higher intensities than the other road surface due to their color differences, we create the *no-marking* PCDs by changing their intensities to the same as other road surface area. To do that, we first apply the commonly-used RANSAC plane segmentation [122] on the PCDs to find all points that belong to the ground plane, i.e., the road surface, and then set the intensities of these ground points to their median value. This thus effectively makes the lane line markings indistinguishable from the other road surface. The creation of *wrong-marking* PCDs is slightly more complicated. After recognizing all ground points, we identify the lane line marking points depending on whether their intensities are above a certain threshold. For each lane line marking point, we search a corresponding ground point that is *laterally offset by half-lane-width* and set their intensities the same as the original lane line marking points. Finally, we clear the original lane line marking points by setting their intensities to the median ground point intensities. Since the lane line markings are moved by half-lane-width, the *wrong-marking* PCDs should have the largest lateral LiDAR localization impact if the lane line markings have any effect on the LiDAR locator. Fig. 19 shows such an example of the original PCD and the one with *no-marking* and *wrong-marking*.

Experimental setup. We evaluate on 2 LiDAR locators, one from Baidu Apollo (BA-LiDAR locator) [52] and another from Autoware (AW-LiDAR locator) [63]. Details of the LiDAR locators can be found in Appendix . Since MSF localization takes not only position measurements but also position uncertainties from LiDAR locator as inputs, we calculate both the *position accuracies* and *uncertainty correlation* with the original and no/wrong-marking PCDs to show the similarity. We evaluated on the same 5 local road and highway traces in *FusionRipper* [58] from two datasets. For each trace, we exclude

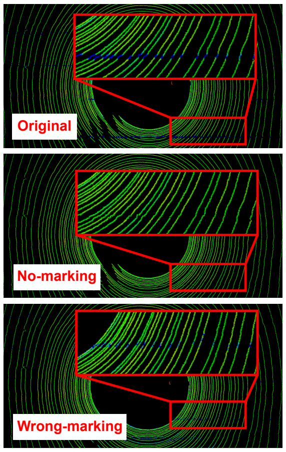


Fig. 19: PCDs with the original, removed, and incorrect lane line markings.

TABLE V: The uncertainty correlation coefficients (r) and position accuracies (RMSE) of LiDAR locators using the original, lane line markings removed (denote as *no-marking*), and incorrect lane line markings PCDs (denote as *wrong-marking*). Results with statistically strong correlation are highlighted in **bold**; we omit the p-values as they are all statistically significant. The numbers are averaged across all sensor traces used in *FusionRipper* [58].

	Uncertainty Correlation (r)		Position Accuracy (RMSE)		
	Original vs No-marking	Original vs Wrong-marking	Original	No-marking	Wrong-marking
BA-LiDAR Locator	**0.89**	**0.64**	0.064 m	0.065 m	0.063 m
AW-LiDAR Locator	**1.0**	**1.0**	0.076 m	0.076 m	0.076 m

TABLE VI: Semantic map APIs required for LD^3.

Map API	Description
`MapLaneDev(pose)`	Query the deviation from `pose` to the closest lane centerline
`MapLaneWidth(pose)`	Query the width of the closest lane to `pose`
`MapLanePoint(pose)`	Query the closest point and lane heading on the closest lane centerline to `pose`
`MapIsIntersection(pose)`	Query if `pose` is located in an intersection

intersections since they do not have lane line markings. Among them, since *ba-local* does not provide ground truth positions, we calculate the position accuracy based on the LiDAR locator with the original lane line markings.

Results. Table V shows the experiment results. For the position accuracy, we report the Root Mean Squared Error (RMSE) between the LiDAR locator positions and the ground truth positions or the ones with the original lane line markings. For the correlation, we use the commonly-used Pearson's correlation, and a correlation coefficient >0.5 is considered strongly correlated [123]. As shown, for both LiDAR locators, the uncertainty correlation coefficients between the original and modified PCDs are all well above the threshold for strong correlation, and their position accuracies are also all at centimeter-level. Particularly, since AW-LiDAR locator does not use lane line markings at the design level (Appendix), the traces consequently show perfect correlations and identical position accuracies no matter how we modify the lane line markings. Such a result suggests that the existing LiDAR locators used in high-level AD systems are indeed largely ignore the lane line marking information when localizing the vehicle on the map, which might because global localization focuses more on the unique features on the road, such as buildings, roadside layouts, and traffic signs. As a result, this indicates that lane line markings are largely independent of the ones that are already used in high-level AD localization and thus pose a great potential for defense purposes.

Details of the LiDAR Locators

At design level, Baidu Apollo LiDAR locator (BA-LiDAR locator) [52] considers point cloud intensities in its position calculation. Thus, the modifications of lane line intensities do have the potential to affect the BA-LiDAR locator performance. On the other hand, Autoware LiDAR locator (AW-LiDAR locator) [63] only uses the position data in the PCD and completely ignores the intensities. This means that AW-LiDAR locator does not consider lane line markings at the design level. Since AW-LiDAR locator implements the Normal Distributions Transform (NDT) algorithm [65], which does not output position uncertainty by default, thus we follow a common adaptation for NDT to use the point cloud matching fitness score as the uncertainty [124].

SAVIOR Evaluation Setup

To evaluate SAVIOR, we follow the similar methodology as the ground rover evaluation in the SAVIOR paper [76], i.e., using the kinematic bicycle model [78] and an Extended Kalman Filter (EKF) to predict the system state (i.e., position in x, y coordinates) given the vehicle control commands (i.e., steering and acceleration). Although the vanilla bicycle model does not have tunable parameters, we follow a similar implementation as SAVIOR by adding coefficients to the bicycle model equations [125]. Same as SAVIOR, we use the *nlgreyest* system identification tool from Matlab [126] to find the coefficients that can best fit the sensor and control trace. During the evaluation, we continuously calculate the residuals between the GPS measurements and the predicted positions from the EKF, and feed the residuals to a CUSUM anomaly detector for attack detection. An execution that triggers the CUSUM detector will be considered as under attack.

Since the KAIST dataset [100] does not store the control commands when the traces were collected, which are required for the SAVIOR evaluation, we replay the KAIST sensor traces as inputs to Baidu Apollo v5.0.0 [62] to collect the control module outputs, i.e., steering and throttle commands. In particular, the control module calculates such commands based on the localization and a planned trajectory, which is a sequence of trajectory points that the vehicle should follow. However, the planned trajectory is runtime information optimized by the planning module during driving, which is not available in the dataset. Since the ground truth positions in the KAIST traces represent the trajectory points followed by the AD vehicle, we thus convert the ground truth positions into planned trajectories according to the format in Baidu Apollo and use them as one of the control inputs. With the planned trajectories, we then feed the benign localization and attacked localization outputs to obtain the benign control commands and attack influenced control commands respectively.

In addition to the KAIST traces, we also evaluate SAVIOR on a dataset that contains the original control commands to validate SAVIOR's detection performance in an ideal setting. However, similar performance is observed in that dataset to the ones on KAIST traces. More details of this are in Appendix .

Evaluating SAVIOR on Dataset with Control Commands

Since SAVIOR requires vehicle control commands, for which we collected by replaying the KAIST traces in Baidu Apollo in our evaluation (Appendix), one might argue that SAVIOR may perform much better if given the originally collected vehicle control commands. Therefore, we evaluate SAVIOR on the comma2k19 dataset [127], which contains the original vehicle control commands when the traces were collected. Since the comma2k19 dataset does not provide LiDAR data, we thus cannot run the MSF attack. To evaluate SAVIOR, we apply the most aggressive GPS spoofing parameters in the MSF attack ($d = 2.0$, $f = 2.0$) to the GPS data and examine SAVIOR's capability at detecting such obvious GPS spoofing attempts. As shown in Fig. 20, SAVIOR's detection performance is close to the one on the *ka-highway36* (Fig. 10) and is still far from a perfect detector.

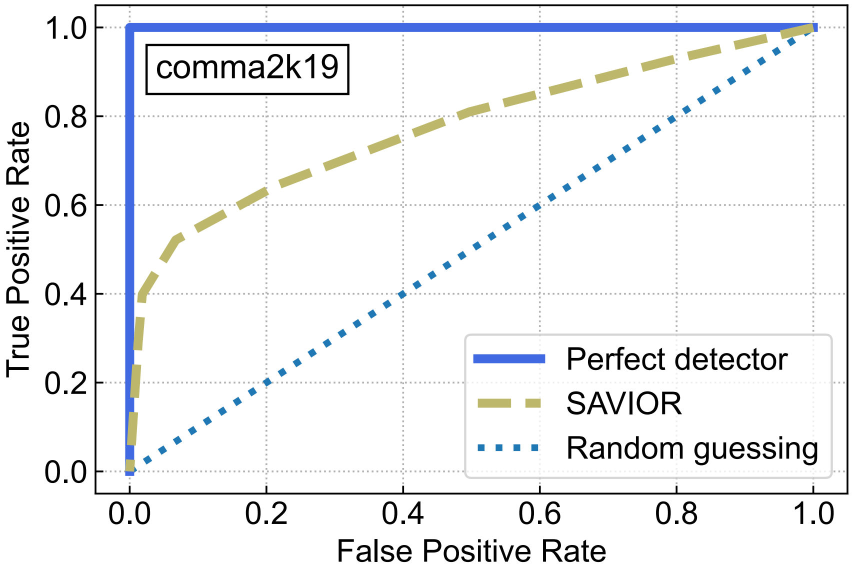


Fig. 20: Attack detection ROC curve of SAVIOR on comma2k19 [127] with original vehicle control commands.

Semantic Map APIs Required in LD^3

As mentioned in §, the LD^3 design queries the semantic map in high-level AD systems to obtain the lateral deviation to lane centerline or lane width at specific positions. Table VI lists the map APIs for LD^3 and their descriptions. Note that all these map APIs are available in typical high-level AD systems, e.g., Baidu Apollo [62] and Autoware [63].

4 Machine Learning PNT Model Security

4a Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack

ABSTRACT

Automated Lane Centering (ALC) systems are convenient and widely deployed today, but also highly security and safety critical. In this work, we are the first to systematically study the security of state-of-the-art deep learning based ALC systems in their designed operational domains under physical-world adversarial attacks. We formulate the problem with a safety-critical attack goal, and a novel and domain-specific attack vector: dirty road patches. To systematically generate the attack, we adopt an optimization-based approach and overcome domain-specific design challenges such as camera frame inter-dependencies due to attack-influenced vehicle control, and the lack of objective function design for lane detection models.

We evaluate our attack on a production ALC using 80 scenarios from real-world driving traces. The results show that our attack is highly effective with over 97.5% success rates and less than 0.903 sec average success time, which is substantially lower than the average driver reaction time. This attack is also found (1) robust to various real-world factors such as lighting conditions and view angles, (2) general to different model designs, and (3) stealthy from the driver's view. To understand the safety impacts, we conduct experiments using software-in-the-loop simulation and attack trace injection in a real vehicle. The results show that our attack can cause a 100% collision rate in different scenarios, including when tested with common safety features such as automatic emergency braking. We also evaluate and discuss defenses.

BACKGROUND

Overview of DNN-based ALC Systems

Fig. 21 shows an overview of a typical ALC system design [82], [128], [129], which operates in 3 steps:

Lane Detection (LD). Lane detection (LD) is the most critical step in an ALC system, since the driving decisions later are mainly made based on its output. Today, production ALC systems predominately use front cameras for this step [130]–[133]. On the camera frames, an LD model is used to detect lane lines. Recently, DNN-based LD models achieve the state-of-the-art accuracy [81], [134], [135] and thus are adopted in the most performant production ALC systems today such as Tesla Autopilot [131]. Since lane line shapes do not change much across consecutive frames, recurrent DNN structure (e.g., RNN) is widely adopted in LD models to achieve more stable prediction [82], [136], [137]. LD models typically first predict the lane line points, and then post-process them to lane line curves using curve fitting algorithms [81], [134], [138], [139].

Before the LD model is applied, a Region of Interest (ROI) filtering is usually performed to the raw camera frame to crop the most important area out of it (i.e., the road surface with lane lines) as the model input. Such ROI area is typically around the center and much smaller than the original frame, to improve the model performance and accuracy [140].

Lateral control. This step calculates *steering angle decisions* to keep the vehicle driving at the center of the detected lane. It first computes a desired driving path, typically at the center of the detected left and right lane lines [141]. Next, a control loop mechanism, e.g., Proportional-Integral-Derivative (PID) [142] or Model Predictive Control (MPC) [143], is applied to calculate the optimal steering angle decisions that can follow the desired driving path as much as possible considering the vehicle state and physical constraints.

Vehicle actuation. This step interprets the steering angle decision into actuation commands in the form of *steering angle changes*. Here, such actuated changes are limited by a maximum value due to the physical constraints of the mechanical control units and also for driving stability and safety [141]. For example, in our experiments with a production ALC with 100 Hz control frequency, such limit is 0.25° per control step (every 10 ms) for vehicle models [144]. As detailed later in §, such a steering limit prevents ALC systems from being affected too much from successful attack in one single LD frame, which introduces a unique challenge to our design.

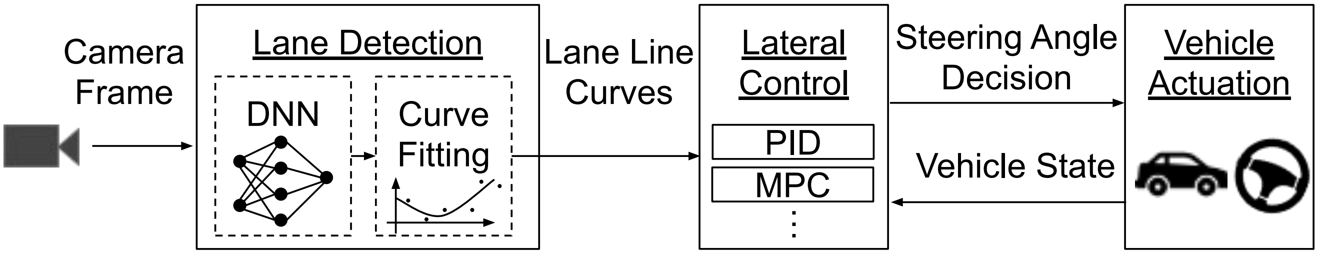


Fig. 21: Overview of the typical ALC system design.

TABLE VII: Required deviations and success time for successful attacks on ALC systems on highway and local roads.

Road Type	Required Lateral Deviation	Required Success Time
Highway	0.735 meters	<2.5 seconds (average driver reaction time to road hazard)
Local road	0.285 meters	

Physical-World Adversarial Attacks

Recent works find that DNN models are generally vulnerable to adversarial examples, or adversarial attacks [145], [146]. Some works further explored such attacks in the physical world [147]–[158]. While these prior works concentrate on DNN models for image classification and object detection tasks, we are the first to systematically study such attacks on production DNN-based ALC systems, which requires to address several new and unique design challenges as detailed later in §.

ATTACK FORMULATION AND CHALLENGE

Attack Goal and Incentives

In this paper, we consider an attack goal that directly breaks the design goal of ALC systems: causing the victim vehicle a lateral deviation (i.e., deviating to the left or right) large enough to drive out of the current lane boundaries. Meanwhile, since ALC systems assume a fully-attentive human driver who is prepared to take over at any moment [159], [160], such deviation needs to be achieved fast enough so that the human driver cannot react in time to take over and steer back. Table VII shows concrete values of these two requirements for successful attacks on highway and local roads respectively, which will be used as evaluation metrics later in §. In the table, the required deviations are calculated based on representative vehicle and lane widths in the U.S., and the required success time is determined using commonly-used average driver reaction time to road hazards.

Targeted scenario: Free-flow driving. Our study targets the most common driving scenario for using ALC systems: *free-flow* driving scenarios [161], in which a vehicle has at least 5–9 seconds clear headway [162] and thus can drive freely without considering the front vehicle [161].

Safety implications. The attack goal above can directly cause various safety hazards in the real world: (1) *Driving off road*, which is a direct violation of traffic rules [163] and can cause various safety hazards such as hitting road curbs or falling down the highway cliff. (2) *Vehicle collisions*, e.g., with vehicles parked on the road side, or driving in adjacent or opposite traffic lanes on a local road or a two-lane undivided highway. Even with obstacle or collision avoidance, these collisions are still possible for two reasons. First, today's obstacle and collision avoidance systems are not perfect. For example, a recent study shows that the AEB (Automatic Emergency Braking) systems in popular vehicle models today fail to avoid crashes 60% of the time [164]. Second, even if they can successfully perform emergency stop, they cannot prevent the victim from being hit by other vehicles that fail to yield on time. Later in §, we evaluate the safety impacts of our attack with a simulator and a real vehicle.

Threat Model

We assume that the attacker can obtain the same ALC system as the one used by the victim to get a full knowledge of its implementation details. This can be done through purchasing or renting the victim vehicle model and reverse engineering it, which has already been demonstrated possible on Tesla Autopilot [165]. Moreover, there exist production ALC systems that are open sourced [82]. We also assume that the attacker can obtain a motion model [166] of the victim vehicle, which will be used in our attack generation process (§). This is a realistic assumption since the most widely-used motion model (used by us in §) only needs vehicle parameters such as steering ratio and wheelbase as input [166], which can be directly found from vehicle model specifications. We assume the victim drives at the speed limit of the target road, which is the most common case for free-flow driving. In the attack preparation time, we assume that the attacker can collect the ALC inputs (e.g., camera frames) of the target road by driving the victim vehicle model there with the ALC system on.

Design Challenges

Compared to prior works on physical-world adversarial attacks on DNNs, we face 3 unique design challenges:

C1. Lack of legitimately-deployable attack vector in the physical world. To affect the camera input of an ALC system, it is ideal if the malicious perturbations can appear legitimately around traffic lane regions in the physical world. To achieve high legitimacy, such perturbations also must not change the original human-perceived lane information. Prior works use small stickers or graffiti in physical-world adversarial attacks [154], [165], [167]. However, directly performing such activities to traffic lanes in public is illegal [168]. In our problem setting, the attacker needs to operate in the middle of the road when deploying the attack on traffic lanes. Thus, if the attack vector cannot be disguised as legitimate activities, it becomes highly difficult to deploy the attack in practice.

C2. Camera frame inter-dependency due to attack-influenced vehicle actuation. In real-world ALC systems, a successful attack on one single frame can barely cause any meaningful lateral deviations due to the steering angle change limit at the vehicle actuation step (§). For example, for the vehicle models with 0.25° angle change limit per control loop (§), even if a successful attack on a single frame causes a very large steering angle decision at MPC output (e.g., 90°), it can only cause at most 1.25° actuated steering angle changes before the next frame comes, which can only cause up to *0.3-millimeter* lateral deviations at 45 mph ($\sim$72 km/h).

Thus, to achieve our attack goal in §, the attack must be *continuously effective on sequential camera frames* to increasingly reach larger actuated steering angles and thus larger lateral deviations per frame. In this process, due to the dynamic vehicle actuation applied by the ALC system, the attack effectiveness for later frames are directly dependent on that for earlier frames. For example, if the attack successfully deviates the detected lane to the right in a frame, the ALC system will steer the vehicle to the right accordingly. This causes the following frames to capture road areas more to the right, and thus directly affect their attack generation. There are prior works considering attack robustness across sequential frames, e.g., using EoT [149], [150] and universal perturbation [169], but none of them consider frame inter-dependencies due to attack-influenced vehicle actuation in our problem setting.

C3. Lack of differentiable objective function design for LD models. To systematically generate adversarial inputs, prior works predominately adopt optimization-based approaches, which have shown both high efficiency and effectiveness [145], [167], [170], [171]. However, the objective function designs in these prior works are mainly for image classification [150], [167] or object detection [151], [152], [154] models, which thus aim at decreasing class or bounding box probabilities. However, as introduced in §, LD models output detected lane line curves, and thus to achieve our attack goal the objective function needs to aim at changing the *shape* of such curves. This is substantially different from decreasing probability values, and thus none of these existing designs can directly apply.

Closer to our problem, prior works that attack end-to-end autonomous driving models [155]–[158] directly design their objective function to change the final steering angle decisions. However, as described in §, state-of-the-art LD models do not directly output steering angle decisions. Instead, they output lane line curves and rely on the lateral control step to compute the final steering angle decisions. However, many steps in the lateral control module, e.g., the desired driving patch calculation and the MPC framework, are generally not differentiable to the LD model input (i.e., camera frames), which makes it difficult to effectively optimize.

CONCLUSION

In this work, we are the first to systematically study the security of DNN-based ALC in its designed operational domains under physical-world adversarial attacks. With a novel attack vector, dirty road patch, we perform optimization-based attack generation with novel input generation and objective function designs. Evaluation on a production ALC using real-world traces shows that our attack has over 95% success rates with success time substantially lower than average driver reaction time, and also has high robustness, generality, physical-world realizability, and stealthiness. We further conduct experiments

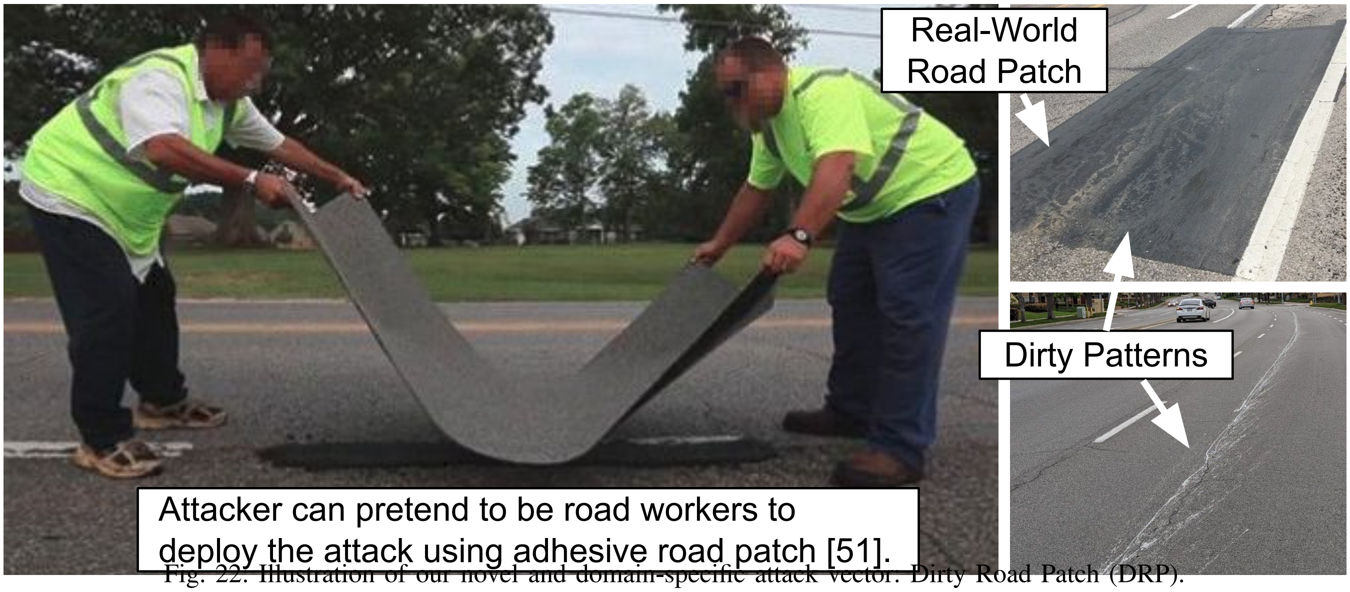

Real-World
Road Patch
Dirty Patterns
Attacker can pretend to be road workers to
deploy the attack using adhesive road patch [51].

Fig. 22: Illustration of our novel and domain-specific attack vector: Dirty Road Patch (DRP).

using both simulation and a real vehicle, and find that our attack can cause a 100% collision rate in different scenarios. We also evaluate and discuss possible defenses. Considering the popularity of ALC and the safety impacts shown in this paper, we hope that our findings and insights can bring community attention and inspire follow-up research.

DIRTY ROAD PATCH ATTACK DESIGN

In this paper, we are the first to systematically address the design challenges above by designing a novel physical-world attack method on ALC, called *Dirty Road Patch (DRP) attack.*

Design Overview

To address the 3 design challenges in §, our DRP attack method has the following novel design components:

Dirty road patch: Domain-specific & stealthy physical-world attack vector. To address challenge **C1**, we are the first to identify *dirty road patch* as an attack vector in physical-world adversarial attacks. This design has 2 unique advantages. First, road patches can appear to be legitimately deployed on traffic lanes in the physical world, e.g., for fixing road cracks. Today, deploying them is made easy with adhesive designs [172] as shown in Fig. 22. The attacker can thus take time to prepare the attack in house by carefully printing the malicious input perturbations on top of such adhesive road patches, and then pretend to be road workers like those in Fig. 22 to quickly deploy it when the target road is the most vacant, e.g., in late night, to avoid drawing too much attention.

Second, since it is common for real-world roads to have dirt or white stains such as those in Fig. 22, using similar dirty patterns as the input perturbations can allow the malicious road patch to appear more normal and thus stealthier. To mimic the normal dirty patterns, our design only allows color perturbations on the gray scale, i.e., black-and-white. To avoid changing the lane information as discussed in §, in our design we (1) require the original lane lines to appear exactly the same way on the malicious patch, if covered by the patch, and (2) restrict the brightness of the perturbations to be strictly lower than that of the original lane lines. To further improve stealthiness, we also design parameters to adjust the perturbation size and pattern, which are detailed in §.

So far, none of the popular production ALC systems today such as Tesla, GM, etc. [132], [133], [159], [173]–[178] identify roads with such dirty road patches as driving scenarios that they do not handle, which can thus further benefit the attack stealthiness.

Motion model based input generation. To address the strong inter-dependencies among the camera frames (**C2**), we need to dynamically update the content of later camera frames according to the vehicle actuation decisions applied at earlier ones in the attack generation process. Since adversarial attack generation typically takes thousands of optimization iterations [179], [180], it is practically highly difficult, if not impossible, to drive real vehicles on the target road to obtain such dynamic

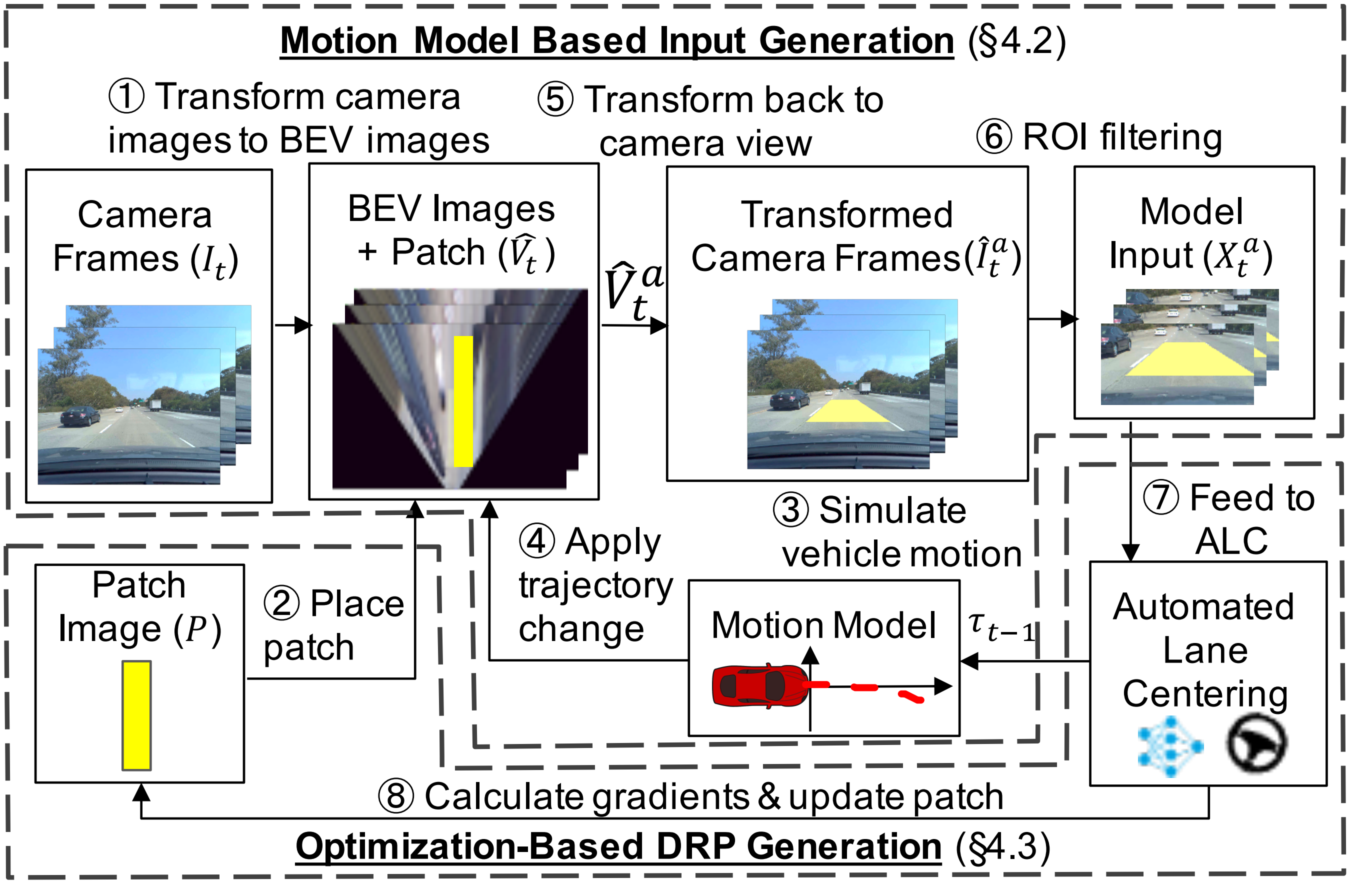

Motion Model Based Input Generation (§4.2)
① Transform camera images to BEV images
⑤ Transform back to camera view
⑥ ROI filtering
Camera Frames (I_t)
BEV Images + Patch ($\hat{V}_t$)
$\hat{V}_t^a$
Transformed Camera Frames($\hat{I}_t^a$)
Model Input (X_t^a)
③ Simulate vehicle motion
④ Apply trajectory change
⑦ Feed to ALC
Patch Image (P)
② Place patch
Motion Model
τ_{t-1}
Automated Lane Centering
⑧ Calculate gradients & update patch
Optimization-Based DRP Generation (§4.3)

Fig. 23: Overview of our DRP (Dirty Road Patch) attack method. ROI: Region of Interest; BEV: Bird's Eye View.

frame update in every optimization iteration. Another idea is to use vehicle simulators [105], [181], but it requires the attacker to first create a high-definition 3D scene of the target road in the real world, which requires a significant amount of hardware resource and engineering efforts. Also, launching a vehicle simulator in each optimization iteration can greatly harm the attack generation speed.

To efficiently and effectively address this challenge, we combine *vehicle motion model* [166] and *perspective transformation* [182], [183] to dynamically synthesize camera frame updates according to a driving trajectory simulated in a lightweight way. This method is inspired by Google Street View [184] that synthesizes 360° views from a limited number of photos utilizing perspective transformation. Our method only requires one trace of the ALC system inputs (i.e., camera frames) from the target road without attack, which can be easily obtained by the attacker (§).

Optimization-based DRP generation. To systematically generate effective malicious patches, we adopt an optimization-based approach similar to prior works [145], [167]. To address challenge **C3**, we design a novel lane-bending objective function as a differentiable surrogate that aims at changing the derivatives of the desired driving path before the lateral control module, which is equivalent to change the steering angle decisions at the lateral control design level. Besides this, we also have other domain-specific designs in the optimization problem formulation, e.g., for a differentiable construction of the curve fitting process, malicious road patch robustness, stealthiness, and physical-world realizability.

Fig. 23 shows an overview of the malicious road patch generation process, which is detailed in the following sections.

Motion Model based Input Generation

In Fig. 23, step ①–⑦ belong to the motion model based input generation component. As described earlier in §, the input to this component is a trace of ALC system inputs such as camera frames from driving on the target road without attack. In ①, we apply *perspective transformation*, a widely-used computer vision technique that can project an image view from a 3D coordinate system to a 2D plane [182], [183]. Specifically, we apply it to the original camera frames from the driver's view to

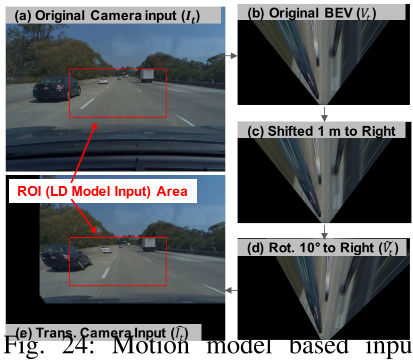

(a) Original Camera input (I_t)
(b) Original BEV (V_t)
(c) Shifted 1 m to Right
(d) Rot. 10° to Right ($\hat{V}_t$)
ROI (LD Model Input) Area
(e) Trans. Camera Input ($\hat{I}_t$)

Fig. 24: Motion model based input generation from original camera input.

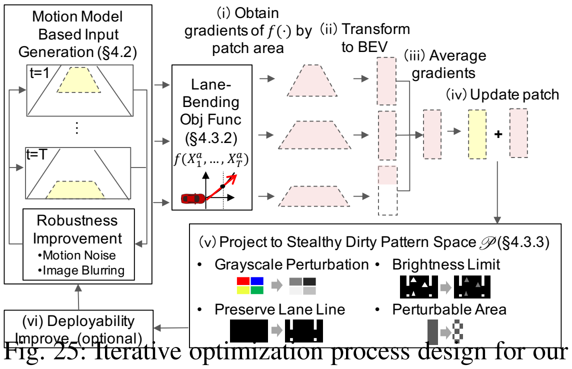

Motion Model Based Input Generation (§4.2)
t=1
t=T
Robustness Improvement
•Motion Noise
•Image Blurring
Lane-Bending Obj Func (§4.3.2)
$f(X_1^a, \dots, X_T^a)$
(i) Obtain gradients of $f(\cdot)$ by patch area
(ii) Transform to BEV
(iii) Average gradients
(iv) Update patch
+
(v) Project to Stealthy Dirty Pattern Space $\mathscr{P}$(§4.3.3)
• Grayscale Perturbation
• Brightness Limit
• Preserve Lane Line
• Perturbable Area
(vi) Deployability Improve. (optional)

Fig. 25: Iterative optimization process design for our optimization-based DRP generation.

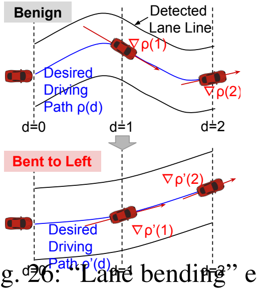


Fig. 26: "Lane bending" effect of our objective function by maximizing $\nabla\rho(d)$ at each curve point.

obtain their Bird's Eye View (BEV) images. This transformation is highly beneficial since it makes our later patch placement and attack-influenced camera frame updates much more natural and thus convenient. We denote this as $V_t := \mathrm{BEV}(I_t)$, where I_t and V_t are the original camera input and its BEV view respectively at frame t. This process is inversible, i.e., we can also obtain I_t with $\mathrm{BEV}^{-1}(V_t)$.

Next, in ②, we obtain the generated malicious road patch image P from the optimization-based DRP generation step (§) and place it on V_t to obtain the BEV image with the patch, denoted as $\widehat{V}_t := \Lambda(V_t, P)$. To achieve consistent patch placements in the world coordinate across frames, we calculate the *pixel-meter relationship*, i.e., the number of pixels per meter, in BEV images based on the driving trace of the target road. With this, we can place the patch in each frame precisely based on the driving trajectory changes across frames.

Next, we compute the vehicle moving trajectory changes caused by the placed malicious road patch, and reflect such changes in the camera frames. We represent the vehicle moving trajectory as a sequence of vehicle states $S_t := [x_t, y_t, \beta_t, v_t], (t = 1, ..., T)$, where x_t, y_t, β_t, v_t are the vehicle's 2D position, heading angle, and speed at frame t, and T is the total number of frames in the driving trace. Thus, the trajectory change at frame t is $\delta_t := S_t^a - S_t^o$, where S_t^a and S_t^o are vehicle states with and without attack respectively.

To calculate δ_t caused by the attack effect at the frame $t-1$, we need to know the attack-influenced vehicle state S_t^a. To achieve that, we use a *vehicle motion model* to simulate the vehicle state S_t^a by feeding the steering angle decision τ_{t-1} from the lateral control step in the ALC system (§) given the attacked frame at $t-1$ and the previous vehicle state S_{t-1}^a, denoted as $S_t^a := \mathrm{MM}(S_{t-1}^a, \tau_{t-1})$. A vehicle motion model is a set of parameterized mathematical equations representing the vehicle dynamics and can be used to simulate its driving trajectory given the speed and actuation commands. In this process, we set the vehicle speed as the speed limit of the target road as described in our threat model (§). In our design, we adopt the kinematic bicycle model [78], which is the most widely-used motion model for vehicles [78], [185], [186].

With δ_t, in ④ we then apply affine transformations on the BEV image $\widehat{V}_t$ to obtain the attack-influenced one $\widehat{V}_t^a$, denoted as $\widehat{V}_t^a := T(\widehat{V}_t, \delta_t)$. Fig. 24 shows an example of the shifting and rotation $T(\cdot)$ in the BEV, which synthesizes a camera frame with the vehicle position shifted by 1 meter and rotated by 10° to the right. Although it causes some distortion and missing areas on the edge, the ROI area (red rectangle), i.e., the LD model input, is still complete and thus sufficient for our purpose. Since the ROI area is typically focused on the center and much smaller than the raw camera frame (§), our method can successfully synthesize multiple complete LD model inputs from only 1 ALC system input trace.

Next, in ⑤, we obtain the attack-influenced camera frame at the driver's view $\widehat{I}_t^a$, i.e., the direct input to ALC, by projecting $\widehat{V}_t^a$ back using $\widehat{I}_t^a := \mathrm{BEV}^{-1}(\widehat{V}_t^a)$. Next, in ⑥, the ROI filtering is used to extract the model input $X_t^a := \mathrm{ROI}(\widehat{I}_t^a)$. X_t^a and vehicle state S_t^a are then fed to ALC system in ⑦ to obtain the steering angle decision τ_t, denoted as $\tau_t := \mathrm{ALC}(X_t^a, S_t^a)$. Step ③–⑦ are then iteratively applied to obtain $\widehat{I}_{t+1}^a, \widehat{I}_{t+2}^a, ...$ one after one until all the original frames are updated to reflect the moving trajectory changes caused by P. These updated attack-influenced inputs are then fed to the optimization-based DRP generation component, which is detailed next.

Optimization-Based DRP Generation

In Fig. 23, step ⑧ belongs to the optimization-based road path generation component. In this step, we design a domain-specific optimization process on the target ALC system to systematically generate the malicious dirty road patch P.

DRP attack optimization problem formulation. We formulate the attack as the following optimization problem:

$$\begin{aligned}
\min \quad & \mathscr{L} && (10)\\
\text{s.t.} \quad & X_t^a = \text{ROI}(\text{BEV}^{-1}(T(\Lambda(V_t, P), S_t^a - S_t^o))) \quad (t = 1, ..., T) && (11)\\
& \tau_t^a = \text{ALC}(X_t^a, S_t^a) \qquad (t = 1, ..., T) && (12)\\
& S_{t+1}^a = \text{MM}(S_t^a, \tau_t^a) + \epsilon_t \qquad (t = 1, ..., T-1) && (13)\\
& S_1^a = S_1^o && (14)\\
& P = \text{BLUR}(\text{FILL}(B) + \Delta) && (15)\\
& \Delta \in \mathscr{P} && (16)
\end{aligned}$$

where the $\mathscr{L}$ in Eq. 10 is an objective function that aims at deviating the victim out of the current lane boundaries as fast as possible (detailed in §). Eq. 11–14 have been described in §. In Eq. 15, the patch image $P \in \mathbb{R}^{H \times W \times C}$ consists of a base color $B \in \mathbb{R}^C$ and the perturbation $\Delta \in \mathbb{R}^{H \times W \times C}$, where W, H, and C are the patch image width, height, and the number of color channels respectively. We select an asphalt-like color as the base color B since the image is designed to mimic a road patch. Function FILL: $\mathbb{R}^C \rightarrow \mathbb{R}^{H \times W \times C}$ fills B to the entire patch image. Since we aim at generating perturbations that mimic the normal dirty patterns on roads, we restrict Δ to be within a stealthy road pattern space $\mathscr{P}$, which is detailed in §. We also include a noise term ϵ_t in Eq. 13 and an image blurring function $\text{BLUR}(\cdot)$ in Eq. 15 to improve the patch robustness to vehicle motion model inaccuracies and camera image blurring, which are detailed in §.

Optimization Process Overview: Fig. 25 shows an overview of our iterative optimization process design. Given an initial patch image P, we obtain the model input $X_1^a, ..., X_T^a$ from the motion model based input generation process. In step (i), we calculate the gradients of the objective function with respect to $X_1^a, ..., X_T^a$, and only keep the gradients corresponding to the patch areas. In step (ii), these gradients are projected into the BEV space. In step (iii), we calculate the average BEV-space gradients weighted by their corresponding patch area sizes in the model inputs. This step involves an approximation of the gradient of $\text{BEV}^{-1}(\cdot)$. Next, in step (iv), we update the current patch with Adam [187] using the averaged gradient as the gradient of the patch image. In step (v), we then project the updated patch into the stealthy road pattern space $\mathscr{P}$. This updated patch image is then fed back to the motion model based input generation module, where we also add robustness improvement such as motion noises and image blurring. We terminate this process when the attack-introduced lateral deviations obtained from the motion model are large enough.

Lane-Bending Objective Function Design: As discussed in §, directly using steering angle decisions as $\mathscr{L}$ makes the objective function non-differentiable to $X_1^a, ..., X_T^a$. To address this, we design a novel lane-bending objective function $f(\cdot)$ as a differentiable surrogate function. In this design, our key insight is that at the design level, the lateral control step aims at making steering angle decisions that follows a *desired driving path* in the middle of the detected left and right lane line curves from the lane detection step (§). Thus, changing the steering angle decisions is equivalent to changing the derivatives of (or "bending") such desired driving path curve. This allows us to design $f(\cdot)$ as:

$$f(X_1^a, ..., X_T^a) = \sum_{t=1}^{T} \sum_{d \in D_t} \nabla \rho_t(d; \{X_j^a | j \leq t\}, \theta) + \lambda ||\Omega_t(X_t^a)||_p \tag{17}$$

where $\rho_t(d)$ is a parametric curve whose parameters are decided by (1) both the current and previous model inputs $\{X_j^a | j \leq t\}$ due to frame inter-dependencies (§), and (2) the LD DNN parameters θ. D_t is a set of curve point index $d = 0, 1, 2, ...$ for the desired driving path curve at frame t. λ is the weight of the p-norm regularization term, designed for stealthiness (§). We then can define $\mathscr{L}$ in Eq. 10 as $f(\cdot)$ and $-f(\cdot)$ when attacking to the left and right. Fig. 26 illustrates this surrogate function when attacking to the left. As shown, by maximizing $\nabla \rho_t(d)$ at each curve point in Eq. 17, we can achieve a "lane bending" effect to the desired driving path curve. Since the direct LD output is lane line points (§) but $\rho_t(\cdot)$ require lane line curves, we further perform a differentiable construction of curve fitting process.

Designs for Dirty Patch Stealthiness: To mimic real-world dirty patterns like in Fig. 22, we have 4 stealthiness designs in stealthy road pattern space $\mathscr{P}$ in Eq. 16:

Grayscale perturbation. Real-world dirty patterns on the road are usually created by dust or white stains (Fig. 22), and thus most commonly just appear white. Thus, we cannot allow perturbations with arbitrary colors like prior works [154]. Thus, our design restricts our perturbation Δ in the grayscale (i.e., black-and-white) by only allowing increase the Y channel in the YCbCr color space [188], denoted as $\Delta_Y \geq 0$.

Preserving original lane line information. We preserve the original lane line information by drawing the same lane lines as the original ones on the patch (if covered by the patch). Note that without this our attack can be easier to succeed, but

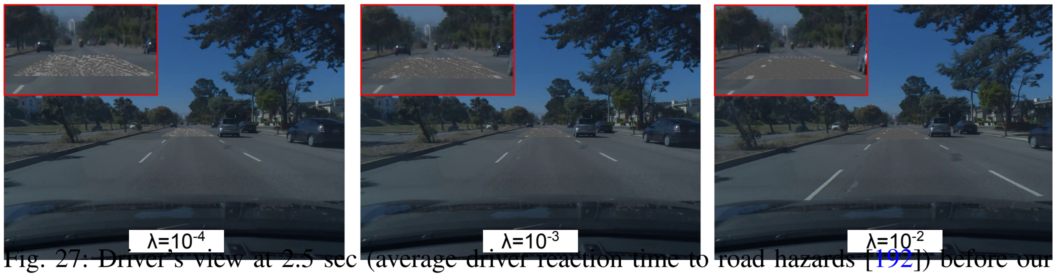


Fig. 27: Driver's view at 2.5 sec (average driver reaction time to road hazards [192]) before our attack succeeds under different stealthiness levels in local road scenarios. Inset figures are the zoomed-in views of the malicious road patches.

Fig. 28: Real-world dirty road patterns.

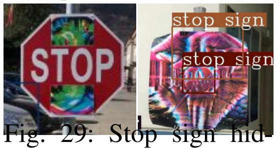


Fig. 29: Stop sign hiding and appearing attacks [154].

as discussed in §, it is much more preferred to preserve such information so that the attack deployment can more easily appear as legitimate road work activities and the deployed patch is less likely to be legitimately removed.

Brightness limits. While the dirty patterns are restricted to grayscale, they are still the darker, the stealthier. Also, to best preserve the original lane information, the brightness of the dirty patterns should not be more than the original lane lines. Thus, we (1) add the p-norm regularization term in Eq. 17 to suppress the amount of Δ_Y, and (2) restrict $B_Y + \Delta_Y < \text{LaneLine}_Y$, where B_Y and LaneLine_Y are Y channel values for the base color and original lane line color respectively.

Perturbation area restriction. Besides brightness, also the fewer patch areas are perturbed, the stealthier. Thus, we define Perturbable Area Ratio (PAR) as the percentage of pixels on P that can be perturbed. Thus, when PAR=30%, 70% pixels on P will only have the base color B.

Designs for Improving Attack Robustness, Deployability, and Physical-World Realizability: We also have domain-specific designs for improving (1) **attack robustness**, which addresses the driving trajectory/angle deviations and camera sensing inaccuracies in real-world attacks; (2) **attack deployability**, which designs an optional *multi-piece patch attack* mode that allows deploying DRP attack with multiple small and quickly-deployable road patch pieces; and (3) **physical-world realizability**, which addresses the color and pattern distortions due to physical-world factors such as lighting condition, printer color accuracy, and camera color sensing capability.

ATTACK METHODOLOGY EVALUATION

In this section, we evaluate the effectiveness, robustness, generality, and realizability of our DRP attack methodology.

Targeted ALC system. In our evaluation, we perform experiments on the production ALC system in OpenPilot [82], which follows the state-of-the-art DNN-based ALC system design (§). OpenPilot is an open-source production Level-2 driving automation system that can be easily installed in over 80 popular vehicle models (e.g., Toyota, Cadillac, etc.) by mounting a dashcam. We select OpenPilot due to its (1) *representativeness*, since it is reported to have close performance to Tesla Autopilot and GM Super Cruise and better than many others [189]–[191], (2) *practicality*, from the large quantity and diversity of vehicle models it can support [82], and (3) *ease to experiment with*, since it is the only production ALC system that is open sourced. In this paper, we mainly evaluate on the lane detection model in OpenPilot v0.7.0, which is released in Dec. 2019.

Evaluation dataset. We perform experiments using the comma2k19 dataset [127], which contains over 33 hours driving traces between California's San Jose and San Francisco in a Toyota RAV4 2017 driven by human drivers. These traces are collected using the official OpenPilot dashcam device, called EON. From this dataset, we manually look for short free-flow driving periods to make road patch placement convenient. In total, we obtain 40 eligible short driving clips, 10 seconds each, with half of them on the highway, and half on local roads. For each driving clip, we consider two attack scenarios: attack to the left, and to the right. Thus, in total we evaluate 80 different attack scenarios.

Attack Effectiveness

Evaluation methodology and metrics. We evaluate the attack effectiveness using the evaluation dataset described above. For each attack scenario, we generate an attack road patch, and use the motion model based input generation method in § to simulate the vehicle driving trajectory influenced by the malicious road patch. To judge the attack success, we use the attack

TABLE VIII: Attack success rate and time under different stealthiness levels. Larger λ means stealthier. Average success time is calculated only among the successful cases. Pixel $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_{inf}$ are the average pixel value changes from the original road surface in the RGB space and normalized to $[0, 1]$.

Stealth. Level λ	Succ. Rate	Succ. Time (s)	Pixel $\mathcal{L}_1$	Pixel $\mathcal{L}_2$	Pixel $\mathcal{L}_{inf}$
10^{-2}	97.5%	0.903	0.018	0.045	0.201
10^{-3}	100%	0.887	0.033	0.066	0.200
10^{-4}	100%	0.886	0.071	0.109	0.200

goal defined in § and concrete metrics listed in Table VII, i.e., achieving over 0.735m and 0.285m lateral deviations on highway and local road scenarios respectively within the average driver reaction, 2.5 sec. We measure the achieved deviation by calculating the lateral distances at each time point between the vehicle trajectories with and without the attack, and use the earliest time point to reach the required deviation to calculate the success time.

Since ALC systems assume a human driver who is prepared to take over, it is better if the malicious road patch can also look stealthy enough at 2.5 sec (driver reaction time) before the attack succeeds so that the driver won't be alerted by its looking and decide to take over. Thus, in this section, we also study the stealthiness of the generated road patches. Specifically, we quantify their perturbation degrees using the average pixel value changes from the original road surface in $\mathcal{L}_1, \mathcal{L}_2$ and $\mathcal{L}_{\text{inf}}$ distances [193], [194] and also a user study.

Experimental setup. For each scenario in the evaluation dataset, we manually mark the road patch placement area in the BEV view of each camera frame based on the lane width and shape. To achieve consistent road patch placements in the world coordinate across a sequence of frames, we calculate the number of pixels per meter in the BEV images and adjust the patch position in each frame precisely based on the driving trajectory changes across consecutive frames. The road patch sizes we use are 5.4 m wide, and 24–36 m long to ensure at least a few seconds of visible time at high speed. The patches are placed 7 m far from the victim at the starting frame. For stealthiness levels, we evaluate the $\mathcal{L}_2$ regularisation coefficient $\lambda = 10^{-2}, 10^{-3}$, and 10^{-4}, with PAR set to 50%. According to Eq. 17, larger λ value means more suppression of the perturbation, and thus should lead to a higher stealthiness level. For the motion model, we directly use the vehicle parameters (e.g., wheelbase) of Toyota RAV4 2017, the vehicle model that collects the traces in our dataset.

Results. As shown in Table VIII, our attack has high effectiveness ($\geq$97.5%) under all the 3 stealthiness levels. Fig. 27 shows the malicious road patch appearances at different stealthiness levels from the driver's view at 2.5 seconds before our attack succeeds. As shown, even for the lowest stealthiness level ($\lambda = 10^{-4}$) in our experiment, the perturbations are still smaller than some real-world dirty patterns such as the left one in Fig. 28. In addition, the perturbations for all these 3 stealthiness levels are a lot less intrusive than those in previous physical-world adversarial attacks in the image space [154], e.g., in Fig. 29. Among the successful cases, the average success time is all under 0.91 sec, which is substantially lower than 2.5 sec, the required success time. This means that even for a fully attentive human driver who is always able to take over as soon as the attack starts to take effect, the average reaction time is still far from enough to prevent the damage. .

Stealthiness user study. To more rigorously evaluate the attack stealthiness, we conduct a user study with 100 participants, and find that (1) even for the lowest stealthiness level at $\lambda = 10^{-4}$, only *less than 25%* of the participants decide to take over the driving before the attack starts to take effect. This suggests that the majority of human drivers today do not treat dirty road patches as road conditions where ALC systems cannot handle; and (2) at 2.5 seconds before the attack succeeds, the attack patches with $\lambda = 10^{-2}$ and 10^{-3} appear to be *as innocent as normal clean road patches to human drivers*, with only less than 15% participants deciding to take over.

From these results, the stealthiness level with $\lambda = 10^{-3}$ strikes an ideal balance between attack effectiveness and stealthiness: it does not increase driver suspicion compared to even a benign clean road patch at 2.5 seconds before our attack succeeds, while having no sacrifice of attack effectiveness as shown in Table VIII. We thus use it as the default stealthiness configuration in our following experiments.

Comparison with Baseline Attacks

Evaluation methodology. To understand the benefits of our current design choices over possible alternatives, we evaluate against 2 baseline attack methods: (1) *single-frame EoT attack*, which still uses our lane-bending objective function but optimizes for the EoT (Expectation over Transformation) of the patch view (e.g., different positions/angles) in a single camera frame, and (2) *drawing-lane-line attack*, which directly draws straight solid white lane line instead of placing dirty road patches. EoT is a popular design in prior works to improve attack robustness across sequential frames [149], [150].

TABLE IX: Attack success rates of the DRP attack and 2 baseline attacks under different patch area lengths.

Attack	Patch Area Length 12m	18m	24m	36m
DRP	66.25%	82.50%	90.75%	100%
Single-frame EoT	0.00%	8.75%	21.25%	50.00%
Drawing-lane-line	2.50%	13.75%	31.25%	53.75%

Thus, comparing with such a baseline attack can evaluate the benefit of our motion model based input generation design (§) in addressing the challenge of frame inter-dependencies due to attack-influenced vehicle actuation (C2 in §).

The drawing-lane-line attack is designed to evaluate the type of ALC attack vector identified in the prior work by Tencent [165], which uses straightly-aligned white stickers to fool Tesla Autopilot on road regions *without lane lines*. In our case, we perform evaluations in road regions *with lane lines*, and use a more powerful form of it (directly drawing solid lane lines) to understand the upper-bound attack capability of this style of perturbation for ALC systems.

Experimental setup. For single-frame EoT attack, we apply random transformations of the patch in BEV via (1) lateral and longitudinal position shifting. We apply up to ± 0.735m and ± 0.285m for highway and local respectively, which are their maximum in-lane lateral shifting from the lane center; and (2) viewing angle changes. we apply up to $\pm 5.8°$ changes, the largest average angle deviations under possible real-world trajectory variations). For each scenario, we repeat the experiments for each frame with a complete patch view (usually the first 4 frames), and take the most successful one to obtain the upper-bound effectiveness. Other settings are the same as the DRP attack, e.g., $\lambda = 10^{-3}$.

For the drawing-lane-line attack, we use the same perturbation area (i.e., the patch area) as the others for a fair comparison. Specifically, we sample points every 20cm at the top and bottom patch edges respectively, and form possible attacking lane lines by connecting a point at the top with one at the bottom. We exhaustively try all possible top and bottom point combinations and take the most successful one. The attacking lane lines are 10cm wide (a typical lane marking width [195]) with the same white color as the original lane lines.

Results. Table IX shows the results under different patch area lengths. As shown, the DRP attack always has the highest attack success rate than these two baselines (with a $\geq$46% margin). When the patch area length is shorter and thus the perturbation capability is more limited, such advantage becomes larger; when the length is 12m, the success rates of single-frame EoT attack and the drawing-lane-line attack drops to 0% and 2.5%, while that for DRP is still 66%. This shows that our motion model based input generation can indeed benefit attack effectiveness, as it can more accurately synthesize subsequent frame content based on attack-influenced vehicle actuation, instead of the blind synthesis in EoT. Also note that the single-frame EoT attack still uses our domain-specific lane-bending objective function design. The drawing-lane-line attack only has 2.5% success rate when the length is 12m; the length used in the Tencent work is actually even shorter (<5m) [165]. This shows that in the road regions *with lane lines*, simply adding lane-line-style perturbations, especially a short one, can barely affect production ALC systems. Instead, an attack vector with larger perturbation area, e.g., in DRP attack, may be necessary.

Attack Robustness, Generality, and Deployability Evaluations

Robustness to run-time driving trajectory and angle deviations. As described in §, the run-time victim driving trajectories and angles will be different from the motion model predicted ones in attack generation time due to run-time driving dynamics. To evaluate attack robustness against such deviations, we use (1) 4 levels of vehicle position shifting at each vehicle control step in attack evaluation time, and (2) 27 vehicle starting positions to create a wide range of approaching angles and distances to the patch, e.g., from (almost) the leftmost to the rightmost position in the lane. Our attack is shown to maintains a high effectiveness ($\geq$ 95% success rate) even when the vehicle positions at the attack evaluation time has 1m shifting on average from those at the attack generation time at each control step.

Attack generality evaluation. To evaluate the generality of our attack against LD models of different designs, ideally we hope to evaluate on LD models from other production ALC besides OpenPilot, e.g., from Tesla Autopilot. However, OpenPilot is the only one that is currently open sourced. Fortunately, we find that the LD models in some older versions of OpenPilot actually have different DNN designs, which thus can also serve for our purpose. We evaluate on 3 versions of LD models with large DNN architecture differences, and find that our attack is able to achieve $\geq$90% success rates against all 3 LD models, with an average attack transferability of 63%.

Attack deployability evaluation. We evaluate the attack deployability by estimating the required efforts to deploy the attack road patch. We perform experiments using our multi-piece patch attack mode design (§), and find that the attack success rate can be as high as *93.8% with only 8 pieces of quickly-deployable road patches*, each requiring only 5-10 sec for 2 people to deploy based on videos of adhesive patch deployment [196].

Physical-World Realizability Evaluation

While we have shown high attack effectiveness, robustness, and generality on real-world driving traces, the experiments are performed by synthesizing the patch appearances digitally, which is thus still different from the patch appearances in the physical world. As discussed in §, there are 3 main practical factors that can affect the attack effectiveness in physical world: (1) the lighting condition, (2) printer color accuracy, and (3) camera sensing capability. Thus, in this section we perform experiments to understand the physical-world attack realizability against these 3 main practical factors.

Evaluation methodology: miniature-scale experiments. To perform the DRP attack, a real-world attacker can pretend to be road workers and place the malicious road patch on public roads. However, due to the access limit to private testing facilities, we cannot do so ethically and legally on public roads with a real vehicle. Thus, we try our best to perform such evaluation by designing a *miniature-scale experiment*, where the road and the malicious road patch are first physically printed out on papers and placed according to the physical-world attack settings but in miniature scale. Then the real ALC system camera device is used to get camera inputs from such a miniature-scale physical-world setting. Such miniature-scale evaluation methodology can capture all the 3 main practical factors in the physical-world attack setting, and thus can sufficiently serve for the purpose of this evaluation.

Experimental setup. As shown in Fig. 30, we create a miniature-scale road by printing a real-world high-resolution BEV road texture on multiple ledger-size papers and concatenating them together to form a long straight road. In the attack evaluation time, we create the miniature-scale malicious road patch using the same method, and place it on top of the miniature-scale road following our DRP attack design. The patch is printed with a commodity printer: RICOH MP C6004ex Color Laser Printer. We mount EON, the official OpenPilot dashcam device, on a tripod and face it to the miniature-scale road. The road size, road patch size, and the EON mounting position are carefully calculated to represent OpenPilot installed on a Toyota RAV4 driving on a standard 3.6-meter wide highway road at 1:12 scale. We also create different lighting conditions with two studio lights. The patch size is set to represent a 4.8me wide and 12m long one in the real world scale. The other settings are the same as in §.

Evaluation metric. Since the camera is mounted in a static position, we evaluate the attack effectiveness directly using the steering angle decision at the frame level instead of the lateral deviation used in previous sections. This is equivalent from the attack effectiveness point of view since the large lateral deviation is essentially created by a sequence of large steering angle decisions at the frame level. Specifically, we first find the camera frame that has the same relative position between the camera and the patch as that in the miniature-scale experimental setup. Then we compare its *designed* steering angle at the attack generation time and its *observed* steering angle that the ALC system in OpenPilot intends to apply to the vehicle in the miniature-scale experiment. Thus, the more similar these two steering angles are, the higher realizability our attack has in the physical world.

Results. Fig. 31 shows a visualization of the lane detection results of the benign and attacked scenarios in the miniature-scale experiment using the OpenPilot's official visualization tool. As shown, in the benign scenario, both detected lane lines align accurately with the actual lane lines, and the desired driving path is straight as expected. However, when the malicious road patch is placed, it bends the detected lane lines significantly to the left and causes the desired driving path to be curving to the left, which is exactly the designed attack effect of our lane-bending objective function (§). In this case, the designed steering angle is 23.4° to the left at the digital attack generation time, and the observed one in the physical miniature-scale experiment is 24.5° to the left, which only differs by 4.7%. In contrast, in the benign scenario the observed steering angle for the same frame is 0.9° to the right.

Robustness under different lighting conditions. We repeat this experiment under 12 lighting conditions ranging from 15 lux (corresponding to sunset/sunrise) to 1210 lux (corresponding to midday of overcast days). The results show that the same attack patch above is able to *maintain a desired steering angle of 20-24° to the left under all 12 lighting conditions*, which are all significantly different from the benign scenario (0.9° to the right).

Robustness to different viewing angles. We evaluate the attack robustness from 45 different viewing angles created by different distances to the patch and lateral offsets to the lane center. Our results show that our attack always achieves over

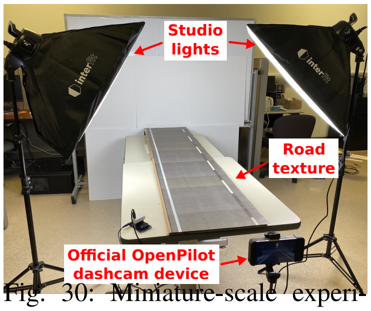


Fig. 30: Miniature-scale experiment setup. Road texture/patch are printed on ledger-size papers.

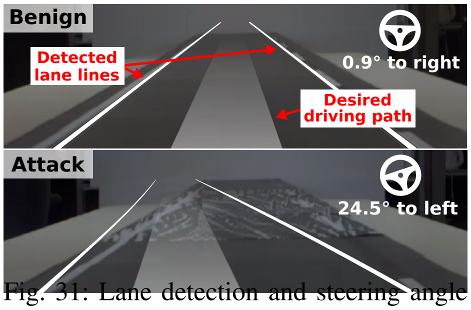


Fig. 31: Lane detection and steering angle decisions in benign and attacked scenarios in the miniature-scale experiment.

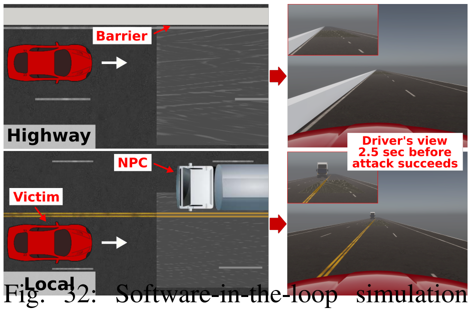


Fig. 32: Software-in-the-loop simulation scenarios and driver's view 2.5 sec before attack succeeds.

23.4° to the left from all viewing angles. We record videos in which we dynamically change viewing angles in a wide range while showing real-time lane detection results under attack, available at **https://sites.google.com/view/cav-sec/drp-attack/**.

SOFTWARE-IN-THE-LOOP SIMULATION

To understand the safety impact, we perform software-in-the-loop evaluation of our attack on LGSVL, a production-grade autonomous driving simulator [105]. We overcame several engineering challenges in enabling this setup and open-sourced via our website [197].

Evaluation scenarios. We construct 2 attack scenarios for highway and local road settings respectively, as shown in Fig. 32. For the former, we place a concrete barrier on the left, and for the latter, we place a truck driving on an opposite direction lane. The attack goals are to hit the concrete barrier or the truck.

Experimental setup and evaluation metrics. We perform evaluation on OpenPilot v0.6.6 with the Toyota RAV4 parameters. We follow the methodology in § to obtain and apply the color mapping in our simulation environment. The patch size is 5.4m wide and 70m long, and we place it in the simulation environment by importing the generated patch image into Unity. The other parameters are the same as §. To evaluate the attack effectiveness from different victim approaching angles, for each scenario we evaluate the same patch from 18 different starting positions, created from the combinations of 2 longitudinal distances to the patch (50 and 100 m) and 9 lateral offsets (from -95% to 95%) as shown in Fig. 33. The patch is visible at all these starting positions. We repeat 10 times for each starting position in each scenario.

Results and video demos. Our attack achieves 100% success rates from all 18 starting positions in both highway and local road scenarios. Fig. 33 shows the averaged vehicle trajectories from each starting positions. As shown, the vehicle always first drives toward the lane center since the ALC system tries to correct the initial lateral deviations. After that, the patch starts to take effect, and causes the vehicle to deviate to the left significantly and hit the barrier or truck. We record demo videos at **https://sites.google.com/view/cav-sec/drp-attack/**. In the highway scenario, after the victim hits the concrete barrier, it bounces away quickly due to the abrupt collision. For local road, the victim crashes to the front of the truck, causing both the victim and truck to stop. This suggests that the safety impacts of our attack can be severe.

SAFETY IMPACT ON REAL VEHICLE

While the simulation-based evaluation above has shown severe safety impacts, it does not simulate other driver assistance features that are commonly used with ALC at the same time in real-world driving, for example Lane Departure Warning (LDW), Adaptive Cruise Control (ACC), Forward Collision Warning (FCW), and Automatic Emergency Braking (AEB). This makes it unclear whether the safety damages shown in § are still possible when these features are used, especially the safety-protection ones such as AEB. In this section, we thus use a real vehicle to more directly understand this.

Evaluation methodology. We install OpenPilot on a Toyota 2019 Camry, in which case OpenPilot provides ALC, LDW, and ACC, and the Camry's stock features provide AEB and FCW [82]. We then use this real-world driving setup to perform experiments on a rarely-used dead-end road, which has a double-yellow line in the middle and can only be used for U-turn. The driver's view of this road is shown on the left of Fig. 34. In our miniature-scale experiment in §, the attack realizability from the physically-printed patch to the LD model output has already been validated under 12 different lighting conditions. Thus, in this experiment we evaluate the safety impact by directly injecting an attack trace at the LD model output level.

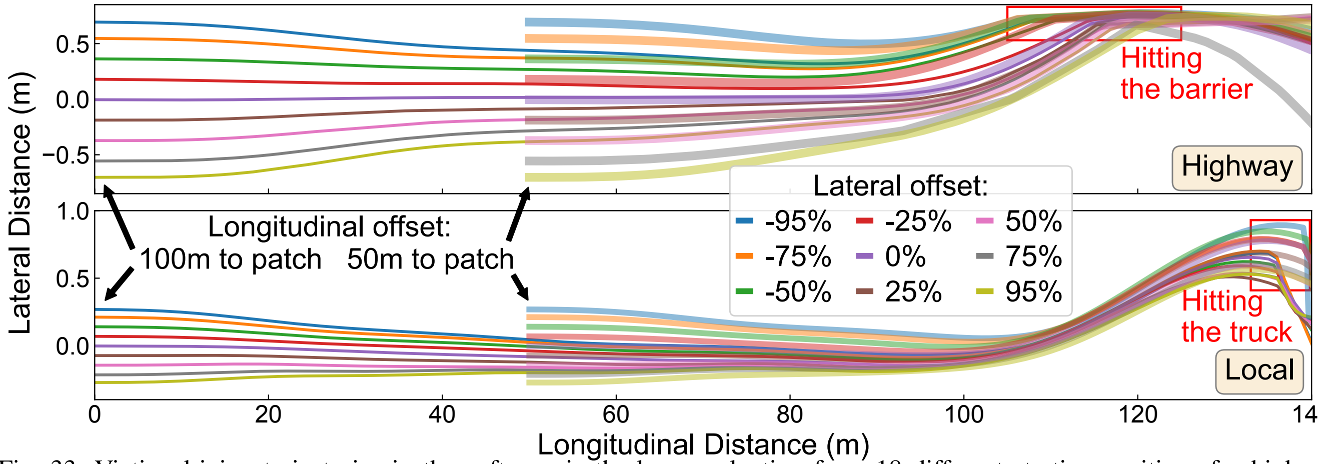


Fig. 33: Victim driving trajectories in the software-in-the-loop evaluation from 18 different starting positions for highway and local road scenarios. Lateral offset values are percentages of the maximum in-lane lateral shifting from lane center; negative and positive signs mean left and right shifting.

This can also avoid blocking the road for sticking patches to the ground and cleaning them up, which may affect other vehicles.

To create safety-critical driving scenarios, we place cardboard boxes adjacent to but outside of the current lane as shown in Fig. 34, which can mimic road barriers and obstacles in opposite direction as in § while not causing damages to the vehicle and driver safety. Similar setup is also used in today's vehicle crash tests [198]–[201]. To ensure that we do not affect other vehicles, we place the cardboard boxes only when the entry point of this dead-end road has no other driving vehicles in sight, and quickly remove them right after our vehicle passes them as required by the road code of conduct [202]–[204].

Experiment setup. We perform experiments in day time with and without attack, each 10 times. The driving speed is kept at ∼28 mph (∼45 km/h), the min speed for engaging OpenPilot on our Camry. The injected attack trace is from our simulation environment (§) at the same driving speed.

Results. Our experiment results show that our attack causes the vehicle to hit the cardboard boxes in all the 10 attack trials (100% collision rate), including 5 front and 5 side collisions. The collision variations are caused by randomness in the dynamic vehicle control and the timing differences in OpenPilot engaging and attack launching. In contrast, in the trials without attack, OpenPilot can always drive correctly and does not hit or even touch the objects in any of the 10 trials.

These results thus show that driver assistance features such as LDW, ACC, FCW, and AEB are not able to effectively prevent the safety damages caused by our attack on ALC. We examine the attack process and find that LDW is not triggered since it relies on the same lane detection module as ALC and thus are affected simultaneously by our attack. ACC does not take any action since it does not detect a front vehicle to follow and adjust speed in these experiments. FCW is triggered 5 times out of the 10 collisions, but it is only a warning and thus cannot prevent the collision by itself. Moreover, in our experiments FCW is triggered only *0.46 sec* before the collision on average, which is far too short to allow human drivers to react considering the 2.5-second average driver reaction time to road hazard (§).

In our Camry model, FCW and AEB are turned on together as a bundled safety feature [205]. However, while we have observed some triggering of FCW, we were not able to observe any triggering of AEB among the 10 attack trials, leading to a *100% false negative rate*. We check the vehicle manual [205] and find that this may be because the AEB feature (called *pre-collision braking* for Toyota) is used very conservatively: it is triggered only when the possibility of a collision is *extremely high*. This observation is also consistent with the previously-reported high failure rate (60%) for AEB features on popular car models today [164]. Such conservative use of AEB can reduce false alarms and thus avoid mistaken sudden emergency brakes in normal driving, but also makes it difficult to effectively preventing the safety damages caused by our attack — in our experiments, it was not able to prevent any of the 10 collisions. The video recordings for these real-vehicle experiments are available at **https://sites.google.com/view/cav-sec/drp-attack/**.

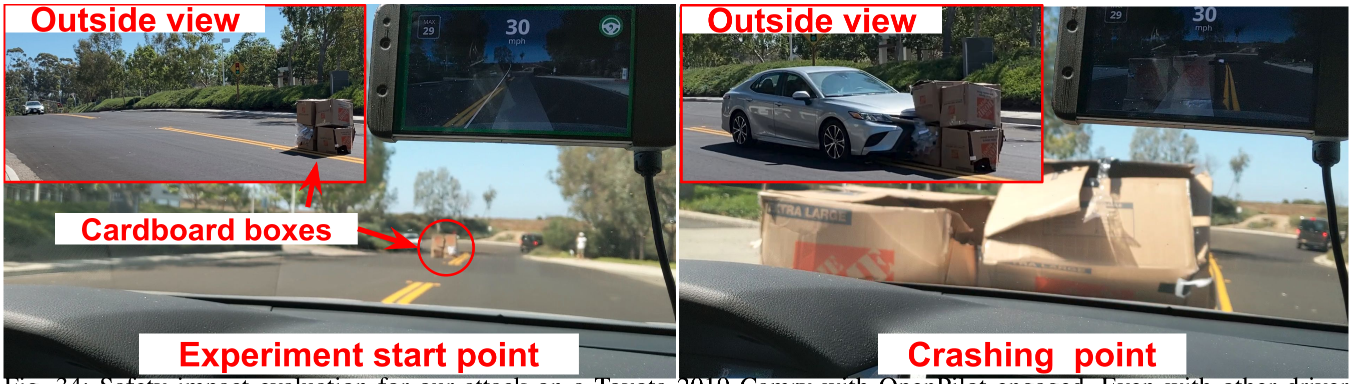


Fig. 34: Safety impact evaluation for our attack on a Toyota 2019 Camry with OpenPilot engaged. Even with other driver assistance features such as Automatic Emergency Braking (AEB), our attack still causes collisions in all the 10 trials.

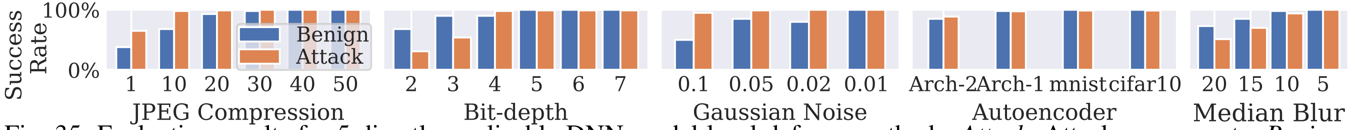


Fig. 35: Evaluation results for 5 directly-applicable DNN model level defense methods. *Attack*: Attack success rate. *Benign*: Percentage of scenarios where the ALC can still behave correctly (i.e., not driving out of current lane) with defense applied.

LIMITATIONS AND DEFENSE DISCUSSION

Limitations of Our Study

Attack deployability. As evaluated in §, our attack can achieve a high success rate (93.8%) with only 8 pieces of quickly-deployable road patches, each requiring only 5-10 sec to deploy for 2 people. To further increase stealthiness, the attacker can pretend to be road workers like in Fig. 22 to avoid suspicion, and pick a deployment time when the target road is the most vacant, e.g., at late night. Nevertheless, lower deployment efforts is always more preferred for attackers to reduce risks. One potential direction to further improve this is to explore other common road surface patterns besides dirty patterns, which we leave as future work.

Generality evaluation. Although we have shown high attack generality against LD models with different designs (§), all our evaluations are performed on only one production ALC in OpenPilot. Thus, it is still unclear whether other popular ALC, e.g., Tesla Autopilot and GM Cruise, are vulnerable to our attack. Unfortunately, to the best of our knowledge, the OpenPilot ALC is the only production one that is open sourced. Due to the same reason, we are also unable to evaluate the transfer attacks from OpenPilot to these other popular ALC systems. Nevertheless, since the OpenPilot ALC is representative at both design and implementation levels (§), we think our current discovery and results can still generally benefit the understanding of the security of production ALC today. Also, since DNNs are generally vulnerable to adversarial attacks [145]–[154], [167], [179], if these other ALC systems also adopt the state-of-the-art DNN-based design, at least at design level they are also vulnerable to our attack.

End-to-end evaluation in real world. In this work, we evaluate our attack against various possible real-world factors such as lighting conditions, patch viewing angles, victim approaching angles/distances, printer color accuracy, and camera sensing capability (§, §), and also evaluate the safety impact using software-in-the-loop simulation (§) and attack trace injection in a real vehicle (§). However, these setups still have a gap to real-world attacks as we did not perform direct end-to-end attack evaluation with real vehicles in the physical world. Such a limitation is caused by safety issues (vehicle-enforced minimum OpenPilot engagement speed at 28 mph, or 45 km/h) and access limits to private testing facilities (for patch placement). In the future, we hope to overcome this by finding ways to lower the minimum engagement speed and obtain access to private testing facilities.

Defense Discussion

Machine Learning Model Level Defenses: In the recent arms race between adversarial machine learning attacks and defenses, numerous defense/mitigation techniques have been proposed [180], [206]–[212]. However, so far none of them studied LD models. As a best effort to understand the effectiveness of existing defenses on our attack, we perform evaluation on 5 popular defense methods that only require model input transformation without re-training: JPEG compression [213], bit-depth reduction [208], adding Gaussian noise [214], median blurring [208], and autoencoder reformation [215], since they are directly applicable to LD models. Our experiments use the same dataset and success metrics as in §. Meanwhile, we also evaluate a *benign-case success rate*, defined as the percentage of scenarios where the ALC can still behave correctly (i.e., not driving out of current lane) when the defense method is applied.

Fig. 35 shows the evaluation results. As shown, for each defense method we also vary the parameters to explore the trade-off between attack success rate and benign-case success rate. As shown, while all methods can effectively decrease the attack success rate with certain parameter configurations, the benign-case success rates are also decreased at the same time. In particular, when the benign-case success rates are still kept at 100%, the attack success rates are still 99 to 100% for all methods. This shows that *none of these methods can effectively defend against our attack without harming ALC performance in normal driving scenarios*. This might be because these defenses are mainly for disrupting digital-space human-imperceptible perturbations, and thus are less effective for physical-world realizable attacks with human-perceptible (but seemingly-benign) perturbations.

These results show that directly-applicable defense methods cannot easily defeat our attack. Thus, it is necessary to explore (1) novel adaptions of more advanced defenses such as adversarial training to LD, or (2) new defenses specific to LD and our problem setting, which we leave as future work.

Sensor/Data Fusion Based Defenses: Besides securing LD models, another direction is to fuse camera-based lane detection with other independent sensor/data sources such as LiDAR and High Definition (HD) map [216]. For example, LiDAR can capture the tiny laser reflection differences for lane line markings, and thus is possible to perform lane detection [217]. However, while LiDARs are commonly used in high-level (e.g., Level-4) AD systems such as Google Waymo [218] that provide self-driving taxi/truck, so far they are not generally used in production low-level (e.g., Level-2) AD such as ALC, e.g., Tesla, GM Cadillac, Toyota RAV4, etc. [131]–[133], [173]. This is mainly because LiDAR is quite costly for vehicle models sold to individuals (typically ≥$4,000 each for AD [219]). For example, Elon Musk, the co-founder of Tesla, claims that LiDARs are "*expensive sensors that are unnecessary (for autonomous vehicles)*" [220].

Another possible fusion source is lane information from a pre-built HD map of the targeted road, which can be used to cross-check with the run-time detected lane lines to detect our attack. However, this requires ALC providers to collect and maintain accurate lane line information for each road, which can be time consuming, costly, and also hard to scale. To the best of our knowledge, ALC systems in production Level-2 AD systems today do not use HD maps in general. For instance, Tesla explicitly claims that it does not use HD map for Autopilot driving since it is a *"non-scalable approach"* [221].

Nevertheless, considering that Level-4 AD systems today are able to build and heavily utilize HD maps [222], [223], we think leveraging HD maps is still a more feasible solution than requiring production Level-2 vehicle models to install LiDARs. If such a map can be available, a follow-up research question is how to effectively detect our attack without raising too many false alarms, since mismatched lane information can also occur in benign cases due to (1) vehicle position and heading angle inaccuracies when localized on the HD map, e.g., due to sensor noises in GPS and IMU [50], [52], and (2) normal-case LD model inaccuracies.

RELATED WORK

Autonomous Driving (AD) system security. For AD systems, there are mainly two types of security research: *sensor security* and *autonomy software security*. For *sensor security*, prior works studied spoofing/jamming on camera [113]–[115], LiDAR [113], [119], [224], RADAR [114], ultrasonic [114], and IMU [225], [226]. For *autonomy software security*, prior works have studied the security of object detection [119], [151], [152], [154], [227] and tracking [228], localization [229], traffic light detection [230], and end-to-end AD models [155]–[158]. Our work studies autonomy software security in production ALC. The only prior effort is from Tencent [165], but it neither attacks the designed operational domain for ALC (i.e., roads with lane lines), nor generates perturbations systematically by addressing the design challenges in §.

Physical-world adversarial attacks. Multiple prior works have explored image-space adversarial attacks in the physical world [147]–[154], [231]. In particular, various techniques have been designed to improve the physical-world robustness, e.g., non-printability score [148], [151]–[153], low-saturation colors [154], and EoT [149]–[152], [154]. In comparison, prior efforts concentrate on image classification and object detection, while we are the first to systematically design physical-world adversarial attacks on ALC, which require to address various new and unique design challenges (§).

CONCLUSION

In this work, we are the first to systematically study the security of DNN-based ALC in its designed operational domains under physical-world adversarial attacks. With a novel attack vector, dirty road patch, we perform optimization-based attack generation with novel input generation and objective function designs. Evaluation on a production ALC using real-world traces shows that our attack has over 95% success rates with success time substantially lower than average driver reaction time, and also has high robustness, generality, physical-world realizability, and stealthiness. We further conduct experiments using both simulation and a real vehicle, and find that our attack can cause a 100% collision rate in different scenarios. We also evaluate and discuss possible defenses. Considering the popularity of ALC and the safety impacts shown in this paper, we hope that our findings and insights can bring community attention and inspire follow-up research.

4b On Robustness of Lane Detection Models to Physical-World Adversarial Attacks

ABSTRACT

Deep Neural Network (DNN)-based lane detection is widely utilized in autonomous driving technologies. At the same time, recent studies demonstrate that adversarial attacks on lane detection can cause serious consequences on particular production-grade autonomous driving systems. However, the generality of the attacks, especially their effectiveness against other state-of-the-art lane detection approaches, has not been well studied. In this work, we report our progress on conducting the first large-scale empirical study to evaluate the robustness of 4 major types of lane detection methods under 3 types of physical-world adversarial attacks in end-to-end driving scenarios. We find that each lane detection method has different security characteristics, and in particular, some models are highly vulnerable to certain types of attack. Surprisingly, but probably not coincidentally, popular production lane centering systems properly select the lane detection approach which shows higher resistance to such attacks. In the near future, more and more automakers will include autonomous driving features in their products. We hope that our research will help as many automakers as possible to recognize the risks in choosing lane detection algorithms.

INTRODUCTION

Lane detection is an essential technology for realizing autonomous driving. Like most other computer vision areas, lane detection has been significantly benefited from the recent advances of deep neural networks (DNNs) as camera is the most frequently used sensor [80]. In the 2017 TuSimple Lane Detection Challenge [232], DNN-based lane detection shows substantial performance as all top 3 teams opt for DNN-based lane detection. However, DNN-based approach has a well-known vulnerability against adversarial attacks [145], [146]. Recent works show that Automated Lane Centering (ALC) systems, Level-2 driving automation, are vulnerable to physical-world adversarial attacks such as malicious patches [233] and stickers [234]. In this work, we report our recent progress on conducting the first large-scale empirical study to evaluate the robustness of major lane detection methods against physical-world adversarial attacks in autonomous driving. These prior works are limited to showing the effectiveness of their attacks on particular lane detection methods. For example, DRP attack [233] is evaluated only on a curve fitting-based lane detection in OpenPilot [82]. The attack shown in [234] is demonstrated only on a segmentation-based lane detection in Tesla Model S. Thus, the effectiveness of attacks on other lane detection methods and the security properties of these lane detection models against adversarial attacks have not been well studied.

In this paper, We first taxonomize state-of-the-art DNN-based lane detection models into 4 major categories (§) We then introduce state-of-the-art physical-world adversarial attacks against ALC systems (§). In §, we construct a methodology to fairly evaluate the robustness of the 4 major types of lane detection models in the end-to-end evaluation. To simulate end-to-end scenarios, we develop a bridge between lane detection methods and the vehicle lateral control implemented in

OpenPilot [82], an open-source production ALC system. In §, we evaluate the robustness of 4 major types of lane detection approaches against 3 types of physical-world adversarial attacks by answering 3 research questions. Throughout this study, we find that each type of lane detection model has different security properties against adversarial attacks: several models are even vulnerable to a naive attack which just draws a white line on the road. Surprisingly, but probably not coincidentally, popular production ALC systems, Tesla Model S and OpenPilot [82], properly select the lane detection approach which shows higher resilience to the drawing-lane-line attack. We then discuss the conclusion and further directions of our study in §.

BACKGROUND

DNN-based Lane Detection

We taxonomize state-of-the-art DNN-based lane detection methods into 4 approaches. Similar taxonomy is also adopted in prior works [235], [236].

Segmentation approach. Segmentation approach handles lane detection as a segmentation task, which classifies whether each pixel is on a lane line or not. Since this approach achieved the state-of-the-art performance in the 2017 TuSimple Lane Detection Challenge [232] (all top-3 winners adopt the segmentation approach [81], [87], [237]), it has been applied in many recent lane detection methods [238]–[240]. This segmentation approach is also used in the industry. A reverse-engineering study reveals that Tesla Model S adopts this segmentation-based approach [234]. The major drawback of this approach is its higher computational and memory cost than the other approaches. Due to the nature of the segmentation approach, it needs to predict the classification results for every pixel, the majority of which is just background. Additionally, this approach requires a postprocessing step to extract the lane line curves from the pixel-wise classification result.

Row-wise classification approach. This approach [236], [241]–[243] leverages the domain-specific knowledge that the lane lines should locate the longitudinal direction of driving vehicles and should not be so curved to have more than 2 intersections in each row of the input image. Based on the assumption, this approach formulates the lane detection task as multiple row-wise classification tasks, i.e., only one pixel per row should have a lane line. Although it still needs to output classification results for every pixel similar to the segmentation approach, this divide-and-conquer strategy enables to reduce the model size and computation while keeping high accuracy. For example, UltraFast [241] reports that their method can work at more than 300 FPS with a comparable accuracy 95.87% on the TuSimple Challenge dataset [232]. On the other hand, SAD [239], a segmentation approach, works at 75 frames per second with 96.64% accuracy. This approach also requires a postprocessing step to extract the lane lines similar to the segmentation approach.

Curve-fitting approach. The curve-fitting approach [244], [245] fits the lane lines into parametric curves (e.g., polynomials and splines). This approach is applied in an open-source production driver assistance system, OpenPilot [82]. The main advantage of this approach is lightweight computation, allowing OpenPilot to run on a smartphone-like device without GPU. To achieve high efficiency, the accuracy is generally not high as other approaches. Note that prior work mentions that this approach is biased toward straight lines because the majority of lane lines in the training data are straight [244].

Anchor-based approach. Anchor-based approach [235], [246], [247] is inspired by region-based object detectors such as Faster R-CNN [248]. In this approach, each lane line is represented as a straight proposal line (anchor) and lateral offsets from the proposal line. Similar to the row-wise classification approach, this approach takes advantage of the domain-specific knowledge that the lane lines are generally straight. This design enables to achieve state-of-the-art latency and performance. LaneATT [235] reports that it achieves a higher F1 score (96.77%) than the segmentation approaches (95.97%) [81], [239] on the TuSimple dataset.

Physical-world Attacks on Automated Lane Centering

After researchers found DNN models generally vulnerable to adversarial examples or adversarial attacks [145], [146], the following work further explored such attacks in the physical world [147], [150], [152]. Recent studies demonstrate that ALC systems, Level-2 driving automation, are also vulnerable to physical-world adversarial attacks.

Dirty Road Patch Attack [233]. Dirty Road Patch (DRP) attack is proposed as a domain-specific adversarial attack to DNN-based ALC systems [233]. DRP attack pretends to be a benign but dirty road patch. The dirty surface pattern is generated by a white-box optimization-based method to work as an adversarial example to lane detection models. To mimic a road patch, the DRP attack has stealthiness constraints such as the gray-scale color restriction and perturbable area ratio. While it has high attack success rates, DRP attack requires white-box access to the target system and relatively heavy deployment effort.

Drawing-Lane-Line Attack. As the nature of lane detection, drawing a line on the road can be an effective attack vector. A recent work [234] demonstrates that they can mislead Tesla Model S to the adjacent lane by putting several small stickers on the road without the original lane line. Phantom attack [115] also demonstrates that they can mislead Tesla Model S by projecting fake lane lines from a drone in the nighttime. The drawing-lane-line attack is not as effective as the DRP attack based on our experience, but its vulnerability to this attack is more severe because of its ease of deployability.

METHODOLOGY

To fairly evaluate the security properties of the 4 major types of lane detection approaches, we design evaluation methodology in end-to-end driving scenarios under 3 types of physical-world adversarial attacks.

Attack Implementation

We implemented 3 types of state-of-the-art physical-world adversarial attacks based on prior attacks against ALC systems discussed in .

White-Box DRP Attack We implement the DRP attack [233]. While the original DRP attack uses the lane bending objective function, we apply a newly-designed attack objective discussed in § to conduct a fair comparison with other attacks and to deal with the output space different from the original DRP attack, which outputs the detected lane lines in the bird's-eye view. All lane detection methods evaluated in this study detected lane lines in the driver's view.

Black-Box DRP Attack To make the DRP attack work in a black-box setup, we apply a query-based black-box attack approach [249] to extend the DRP attack to a black-box attack. We replace the gradient calculation in the original white-box DRP attack with the gradient estimation technique NES [249].

Black-Box Drawing-Lane-Line Attack We explore the most effective line with a metaheuristic strategy according to prior work [234]. We parameterize the drawing lane line as the start point, endpoint, and line width and optimize the parameters with the tree-structured Parzen estimator [250] implemented in Hyperopt [251]. As the objective of the original attack [234] is only applicable to the segmentation approach, we optimize our original attack objective introduced in § to conduct a fair comparison with other attacks.

Attack Objective

To fairly evaluate the attack capability of each attack, we formulate an attack objective function that can be commonly used for all 4 types of lane detection models. We named it the *expected road center function*, which averages all detected lane lines weighted with their probabilities. Intuitively, the average of all lane lines is expected to represent the road center. If the expected center locates at the center of the input image, its value will be 0.5 in the normalized image width. We maximize the expected road center to attack to the right and minimize it to attack to the left. Detailed calculation of the expected road center for each method is in our preprint paper [252]. When attacking multiple frames, we average the objective of each frame over all attacking frames.

End-to-End Simulation

To evaluate the system-level consequence in autonomous driving, we simulate vehicle trajectories under attacks with the same methodology used in [233]. We combine a vehicle motion model [166] and perspective transformation [182], [183] to dynamically synthesize camera frame updates according to a driving trajectory. This approach enables us to evaluate the attacks on the real-world driving traces in a lightweight way. To control a vehicle based on the lane detection results, we develop a bridge between the lane detection model and the vehicle lateral control implemented in OpenPilot [82], an open-source production ALC system. It calculates the desired driving path based on detected lane lines and makes a steering plan to follow the desired driving path with Model Predictive Control (MPC) [128]. In our implementation, the desired driving path is the center of the left and right lane lines.

Attack Goal. To judge the attack success in the end-to-end simulation, we follow the criteria proposed in the DRP attack [233]. We use the attack goal achieving over 0.735 m lateral deviation on the highway within the average driver reaction, 2.5 sec. 0.735 m is the required distance to touch the lane line when a vehicle driving at the center of a 3.6m-wide highway lane. The lateral deviation is calculated between the generated trajectories with attack and without attack. Since the original human driving in the dataset sometimes does not drive at the center of the road, we compare the case with attack and without attack to more precisely measure the attack effect. For each scenario, we consider two attack success criteria: *Targeted goal* is the case that the vehicle deviates over 0.735 m to the attacking direction. *Untargeted goal* is the case that the vehicle deviates over 0.735 m to either the left or right.

We also quantify the ability to drive in a benign scenario. We define a metric called *benign failure rate*, which is whether the human driving and the simulated trajectory deviate by more than 0.735 m. Although the benign failure rate is expected to be always zero because ALC systems should be able to handle normal scenarios, some failure cases occur due to several reasons such as motion model inaccuracy and unstable human driving, e.g., not driving at the center of the road.

EXPERIMENTS

We conduct a large-scale evaluation of 4 major types of lane detection approaches against 3 adversarial attacks: white-box DRP, black-box DRP, black-box drawing-lane-line attacks. This evaluation is designed to answer the following questions:

RQ1: Is black-box attack as effective as white-box attack?
RQ2: Which attack vector is more effective to attack?
RQ3: Are attacks transferable to other models?
*

Evaluation Setup

We evaluate the robustness of 4 major types of lane detection approaches against 3 adversarial attacks: white-box DRP, black-box DRP, black-box drawing-lane-line attacks. For each approach, we select a representative model for each approach as shown in Table X with the selection reasons. The pretrained weights of all models are obtained from the authors' or publicly available websites[1]. All pretrained weights are trained on the TuSimple Challenge training dataset [232].

We collect 20 free-flow[2] highway driving traces from the comma2k19 dataset [127]. For each driving trace, we consider two attack scenarios: attack to the left, and to the right. Thus, in total, we evaluate 40 different attack scenarios. For the lateral control, we use OpenPilot v0.7.0. For the longitudinal control, we used the velocity in the original trace. For the motion model, we use the parameters of Toyota RAV4 2017 (e.g., wheelbase), which is used to collect the traces of the comma2k19 dataset. We manually adjust the input image size and field-of-view to be similar to the TuSimple dataset. We use a 5.4 m x 36 m patch size, which is the same as the one used in the DRP attack [233]. The patch is placed at 7 m away from the vehicle at the first frame. When the patch covers lane lines, we draw lane lines on the patch to keep the original lane line information. When generating the attack, we use the first 20 frames (1 second). When evaluating the attack, we use all 50 frames (2.5 seconds), the average driver's reaction time. More details of each attack implementation and parameters are in our preprint paper [252].

Evaluation on End-to-End Driving Scenario

To evaluate the system-level effects in autonomous driving, we conduct an end-to-end evaluation with the methodology introduced in §. Table XI shows the results of the end-to-end evaluation. As shown, PolyLaneNet demonstrates the highest robustness as it has the lowest attack success rates in all attack scenarios. We can observe a typical trade-off of accuracy and robustness. As in Table X, PolyLaneNet is reported as a lesser performance model. However, in terms of the robustness, PolyLaneNet has the best robustness among 4 major lane detection models.

[1] We obtained the pretrained models from:
LaneATT https://github.com/lucastabelini/LaneATT
SCNN https://github.com/harryhan618/SCNN_Pytorch
UltraFast https://github.com/cfzd/Ultra-Fast-Lane-Detection
PolyLaneNet https://github.com/lucastabelini/PolyLaneNet

[2] Vehicle has at least 5-9 seconds headway.

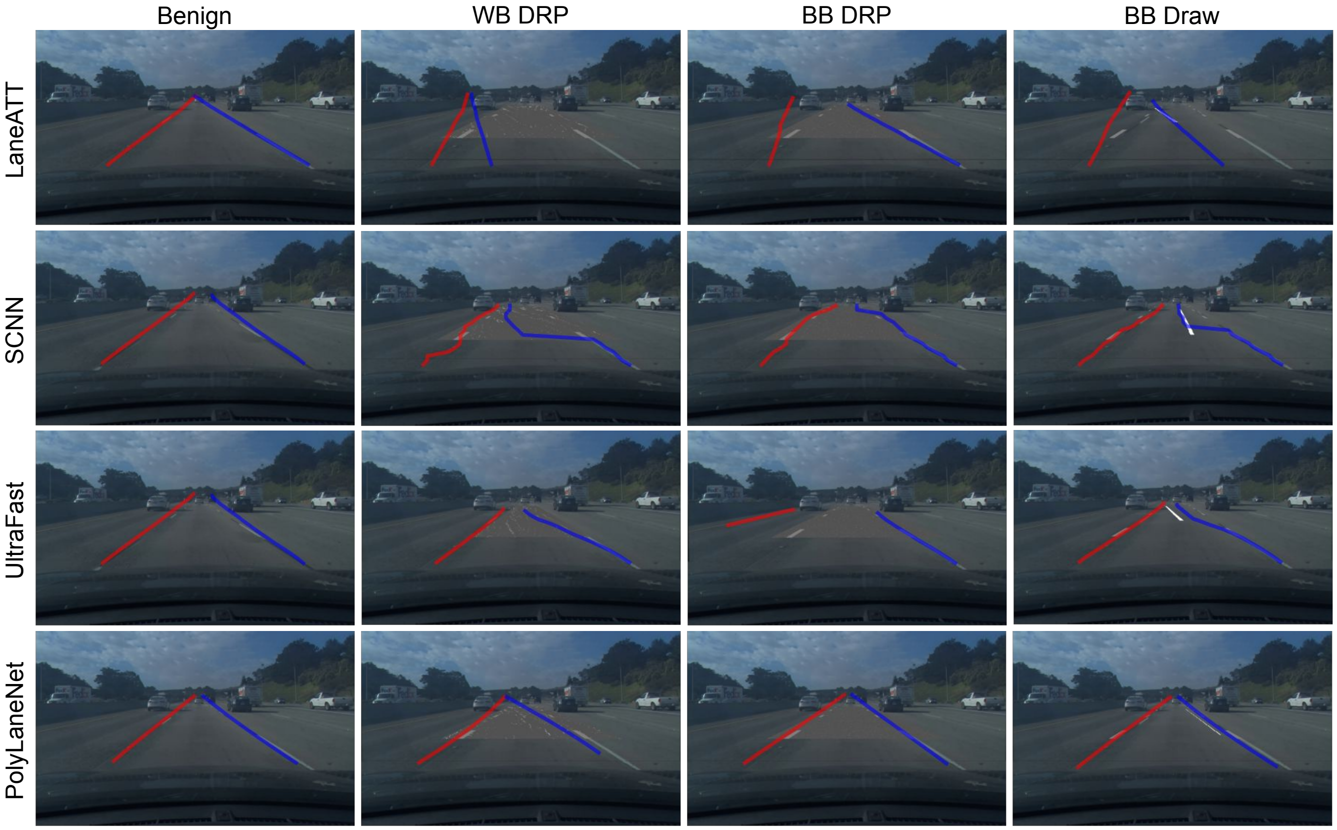


Fig. 36: Examples of the end-to-end benign and 3 different attack scenarios on Comma2k19. Each image is taken at the 4th frame (0.2 sec after the start of the attack). The red and blue lines are the detected left and right lines respectively.

TABLE X: Target lane detection methods and its selection reason. *Acc.* is the accuracy of the TuSimple Challenge dataset [232] in the reference papers.

Approach	Selected Method	Acc.	Selection Reason
Segmentaion	SCNN [81]	96.53%	TuSimple Challenge winner's model
Row-wise classif.	UltraFast (ResNet18) [241]	95.87%	Highest accuracy among those whose official code is available.
Curve-fitting	PolyLaneNet (b0) [244]	88.62%	Highest accuracy among those whose official code is available.
Anchor-based	LaneATT (ResNet34) [235]	95.63%	Highest accuracy among those whose official code is available.

*

RQ1: Is black-box attack as effective as white-box attack?

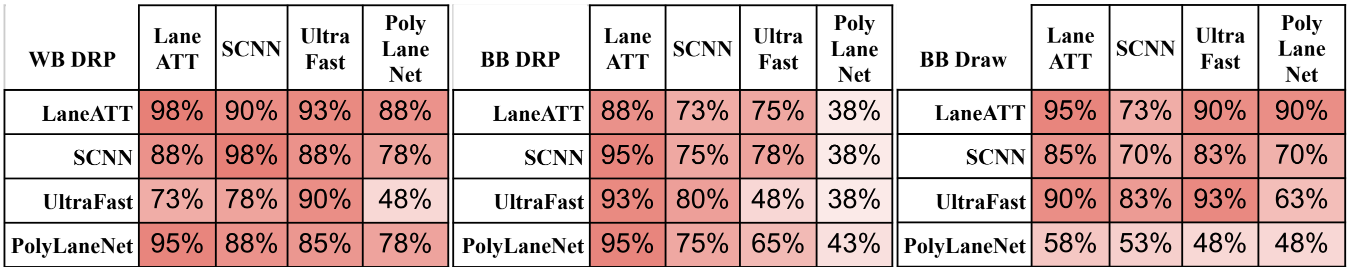
Generally, the white-box attack has more attack capability as it can leverage more specifications of target models. However, recent studies report that black-box attacks can outperform white-box attacks [253] because gradient descent-based methods tend to suffer from local minima. In our evaluation, the same phenomenons are observed: the black-box drawing-lane-line attack outperforms to the white-box DRP attack on LaneATT and UltraFast. The black-box DRP attack has generally lower attack capability than the white-box DRP attack. We think that the stealthiness constraints of the DRP attack (e.g., gray-scale color and perturbable area ratio) could be too complex to be effectively optimized by the NES-based gradient estimation. Meanwhile, the black-box DRP attack has a high attack success rate (88% for the untargeted goal) against LaneATT. Our results demonstrate that the black-box attacks have close or even effectiveness as the white-box attacks.

*

RQ2: Which attack vector is more effective to attack?

TABLE XI: Attack success rate under the end-to-end benign and 3 different attack for targeted and untargeted goals. *Benign* is the benign failure rate defined in §. The **bold** and underlined letters mean the highest and lowest attack success rates, respectively.

		Targeted Goal			Untargeted Goal		
	Benign	WB DRP	BB DRP	BB Draw	WB DRP	BB DRP	BB Draw
LaneATT	20%	**78%**	**53%**	**90%**	**98%**	**88%**	**95%**
SCNN	**30%**	**78%**	43%	58%	**98%**	75%	70%
UltraFast	25%	75%	50%	83%	90%	48%	93%
PolyLaneNet	5%	48%	25%	30%	78%	43%	48%



WB DRP	Lane ATT	SCNN	Ultra Fast	Poly Lane Net
LaneATT	98%	90%	93%	88%
SCNN	88%	98%	88%	78%
UltraFast	73%	78%	90%	48%
PolyLaneNet	95%	88%	85%	78%

BB DRP	Lane ATT	SCNN	Ultra Fast	Poly Lane Net
LaneATT	88%	73%	75%	38%
SCNN	95%	75%	78%	38%
UltraFast	93%	80%	48%	38%
PolyLaneNet	95%	75%	65%	43%

BB Draw	Lane ATT	SCNN	Ultra Fast	Poly Lane Net
LaneATT	95%	73%	90%	90%
SCNN	85%	70%	83%	70%
UltraFast	90%	83%	93%	63%
PolyLaneNet	58%	53%	48%	48%

Fig. 37: Transfer success rate of all pairs of models for the untargeted goal in the end-to-end scenarios. Each row indicates the source model that generates the attack and each column indicates the target model.

The black-box drawing-lane-line attack has the highest attack effectiveness on LaneATT and UltraFast. For PolyLaneNet, the black-box drawing-lane-line attack is more effective than The black-box DRP attack, while the white-box DRP attack is the most effective. SCNN has less vulnerable to the black-box DRP attack and the black-box drawing-lane-line attack, as the black-box drawing-lane-line attack has a higher attack success rate on the targeted goal, but the black-box DRP attack has a higher attack success rate on the untargeted goal. In summary, each lane detection approach has different sensitivity to the drawing-lane-line attack vector. For LaneATT, it could be due to the structure of anchor proposals as discussed in §. However, LaneATT is the only anchor-based method that the source code is available so far. Further research is required to confirm if the vulnerability to the drawing-lane-line attack is derived from a particular design of LaneATT or a fundamental problem of the anchor-based approach. For UltraFast, it shows different sensitivity to the drawing lane line attack compared to SCNN, even though both UltraFast and SCNN predicts the lane line for each pixel. Due to the divide-and-conquer strategy of UltraFast, it may rely too much on local features, i.e., SCNN may judge the lane lines based on more global features such as semantics on the road (e.g., lane lines should be roughly parallel with other lanes). Due to the ease of attack deployability, the vulnerability to the drawing-lane-line attack is severe. For autonomous driving, we should choose relatively robust models against naive attacks like drawing-lane-line attacks.

*

RQ3: Are attacks transferable to other models?

As shown in Fig. 37, the attack success rate is mostly less than the attack generated with the target model (diagonal cells). However, the transfer success rates still keep high attack success rates. Moreover, the attack generated with LaneATT has high transferability to PolyLaneNet in the drawing-lane-line attack: The transfer success rate is 90% attack success rate in the untargeted Goal. The results indicate that PolyLaneNet also has a vulnerability to the drawing-lane-line attack, but the robustness of PolyLaneNet makes it more difficult to generate attacks. Hence, the attacks to one lane detection model are likely to have high transferability to another model, and it is sometimes helpful to find the vulnerability of lane detection models which are robust to normal adversarial attacks.

DISCUSSIONS

While the vulnerability of DNN models against adversarial attacks is widely reported, we may have optimistic expectations that it is almost impossible to exploit it in autonomous driving due to the low deployability mentioned in [233] and the lack of demo in reasonable settings. For example, the demo in [234] is in the intersection without lane lines, which are generally out of the operational domain of ALC. Thus, to our knowledge, I have not observed that any production autonomous driving systems have defense mechanisms against adversarial attacks. However, it does not mean that we can select a lane detection method just based on benign performance. Several lane detection approaches may have high sensitivity against naive attacks like drawing-lane-line attacks. Surprisingly, the production ALC systems we mention select the lane detection approach which shows higher resilience against the drawing-lane-line attack: Tesla Models S adopts the segmentation approach and OpenPilot [82] adopts the Curve-fitting approach. We think this is not coincident but due to the careful design choice of the company. In the near future, more and more automakers will install autonomous driving features in their products. We would like to facilitate more research to build robust lane detection methods so that as many automakers as possible are aware of the risks involved in algorithm selection.

Possible Defenses: So far, effective DNN model-level defenses against adversarial attacks are not reported. According to [233], none of the input transformation-based defenses can effectively defend against their attack without harming performance in normal scenarios. Another possible defense is cross-checking with other data sources. For example, Level-4 autonomous driving systems typically obtain driving lane information from HD maps. However, this method incurs large additional costs as it needs quite accurate localization and continuous maintenance of the HD map.

CONCLUSION AND FUTURE DIRECTION

In this work, we report the recent progress on conducting the first large-scale empirical study to evaluate the robustness of 4 major types of lane detection methods under state-of-the-art 3 physical-world adversarial attacks in autonomous driving scenarios. We find that each lane detection method has different security properties. Particularly, several models show high vulnerability to the drawing-lane-line attack. Thus, it is essential to be aware of the robustness to such naive attacks as Tesla and OpenPilot choose relatively robust methods against the drawing-lane-line attack. We hope that our research will help as many automakers as possible to recognize the risks in choosing lane detection algorithms. In future work, we plan to evaluate a more wide variety of lane detection models and adversarial attacks, especially effective black-box attacks. We also plan to explore more research questions such as dataset transferability. Although Comma2k19 and TuSimple datasets are similar driver's view images, there can be some domain shifts between them. The evaluation of the attack applicability and transferability on different datasets is also a considerable aspect of the robustness of lane detection models. Based on the insight of this study, we would like to work on the development of effective defense methods and robust model training that can improve the robustness of lane detection models in practical autonomous driving.

4c Entropy Based Metric to Assess the Accuracy of PNT Information

ABSTRACT

Entropy measures uncertainty present within the data. Highly Automated Vehicles (HAVs) can navigate safely and efficiently if location information of occluded dynamic objects is available. It is assumed that dynamic objects have GPS receivers, and location information can be acquired through a fast communication link. However, GPS info can be easily modified or suffers from high error because the transmission link is not secure or due to Non Line of Sight (NLOS) between transmitter and receiver. To solve this problem, an entropy metric is introduced to ascertain the value of the supplied information and reject information with a high amount of error present within the data. This work focuses on pedestrians as dynamic objects and uses Finite State Machine (FSM) based hierarchical control to navigate HAVs. It is shown that the entropy metric can improve the efficiency of the control of HAVs.

INTRODUCTION

Entropy is a tool frequently used in statistics to assess the average level of uncertainty present in the samples of a random variable. Highly Automated Vehicles (HAVs) have a level of autonomy 5 as per SAE standard, in an urban environment, HAVs are highly dependent on position information of occluded dynamic objects. Position information of occluded dynamic objects is critical for safe and efficient navigation for all the stakeholders. When collected through sensors, this information is vulnerable to hacking and high errors due to urban canyons. These errors can cause accidents and unnecessary delays in the navigation of HAVs. An entropy-based metric can be used to assess the value and reliability of data received. This work uses a simulation setup with occluded pedestrians, and control of HAVs to assess whether using such a metric would benefit the operation of HAVs. For explanation, three such scenarios are presented in section 1.2 and illustrated in Fig. 38. The entropy of a random variable gives the value present within the information. In the case of highly likely events, information is less valuable and vice versa. Consider a discrete random variable $X = \{x_1, x_2, ..., x_n\}$ and Probability Mass Function (PMF) as P(X). Then entropy can be explicitly written as shown in (18)

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_b P(x_i). \tag{18}$$

Where b is the base of the logarithm, we have used $b = 10$, which is used when the probability of the event occurring is $1/10$. If $H(x)$ has a high value, we can classify it as poor quality, meaning information with high error and data can be rejected.

Literature Review

Entropy has been used to control automated vehicles to assess the reliability of data received on several occasions. In [254], a method is presented for data fusion to track, recognize, and monitor Intelligent Transportation Systems (ITS). In this problem, Robust data alignment is done for successful data fusion. A cost criterion based on entropy is proposed for outlier rejection. In [255] multiple targets must be tracked with multiple dynamic sensing agents. Mobile sensing agents plan their motion so that tracking can be efficient and accurate. An entropy-based cost function is utilized to reject information with an unacceptable error range.

In [256] a scenario is established where three types of vehicles exist on a highway. Namely, fully equipped, partially equipped, and not-equipped. A fully-equipped vehicle has local sensors and communication capability. In contrast, a partially-equipped vehicle can only communicate, and a not-equipped vehicle cannot communicate and does not have local sensors. The objective is to maintain a tracking list of all the vehicles present in the scenario with a tracking list in the partially and fully equipped vehicles. A Kalman Filter (KF) is used for data fusion and correction, and a covariance matrix generated through KF is used to measure the system's entropy. Data is rejected and considered unreliable if the entropy is beyond a threshold. This is used to determine location of occluded pedestrians in ”occupancy grid”

In the present work, the pedestrian case is chosen because the safety of all the stakeholders is critical. As local sensors of ego vehicles cannot detect occluded pedestrians, communication setups have to be established. In our simulation setup, we use the information of occluded pedestrians through LTE/4G.

In case pedestrians are occluded, several methods are utilized to detect a pedestrian. In one method, using off-camera on the streets in an industrial area is used to detect the pedestrians through WiFi [257]. However, this solution is costly and not scalable. Another method by [258] utilizes software-defined radio to transmit the position of pedestrians that, in turn, is also very expensive as dedicated DSRC modules are too expensive. In [259] 802.11 b/g/n communication method is used so that HAVs can receive position information of occluded pedestrians, and the data is fused with LIDAR to get accurate results. However, this solution is also too expensive and not scalable. Currently, we do not have a vast network of WiFi routers in the road infrastructure. [260] used 3G/WLAN to communicate pedestrian information to the vehicle. However, it was not fast enough that the problem could become scalable and choke the network if the number of vehicles or pedestrians increased. With 4G/LTE and 5G, pedestrians can send their position information, and the communication modules are affordable. We suggest this communication protocol for sharing pedestrian location information with the vehicle. We have assumed all pedestrians have cell phones with GPS sensors available for pedestrian localization. The problem with GPS sensors is that it suffers from Non-Line of Sight (NLOS) in urban areas due to high-rise buildings and can have a high amount of position, navigation, and timing errors, which can have errors up to 50 meters. Also, GPS transmitting frequency can be easily generated, and fake information can be generated to misguide the sensor, resulting in accidents or unnecessary collision avoidance measures from the ego vehicle.

We have developed a process to detect occluded dynamic objects in our proposed method. We assume that the typical profile of errors in GPS position information is known, and we will use baseline controllers to compare with our design controller of ego vehicle to prove our system is more efficient and safe using performance metrics.

Simulation Case Study

To explain Vulnerable Road Users (VRU). We are presenting three cases in Fig. 38 taken from ISO standard 22737 section 3.1 [261]. Fig. 38(a) is inspired from section 3.1.1 and Fig. 38(b) and Fig. 38(c) is inspired from section 3.1.2. in (b), a bicyclist comes from an alleyway and appears suddenly in front of the ego vehicle. Similarly, in (c), a bicyclist appears from an uncontrolled intersection. Collectively pedestrians and bicyclists are classified as VRUs. The implemented simulation setup is shown in Fig. 38(a). A static pedestrian who is falsely reporting its position with high errors. An occluded dynamic pedestrian jaywalking and a parked vehicle on one road lane. Such a setup is created to show that using entropy and extra

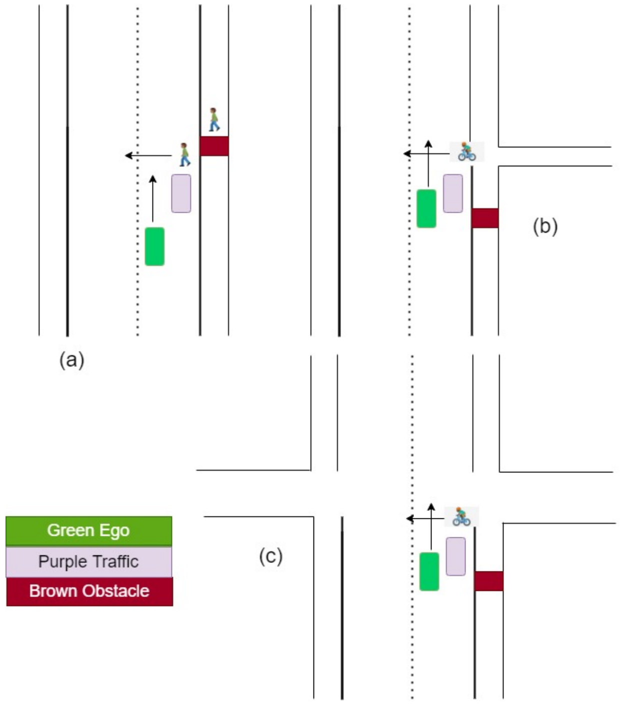


Fig. 38: Scenarios: (a) pedestrian on street, (b) Bike from alleyway, (c) Bike in uncontrolled intersection

information from the environment can result in the safe and efficient control of HAVs. The goal is to test our proposed method with multiple baselines to show our method's effectiveness using some performance metrics. Goals achieved in this case study are listed below

- Introduced an entropy-based metric that will assess the reliability of position information received
- Designed some metrics to measure the performance of HAVs in terms of energy consumed and minimum distance to a dynamic object maintained by HAVs
- Designed Finite State Machine based hierarchical control of ego vehicle

In the next section entire architecture of how communication setup is created and how entropy is performed to assess the position accuracy of the data and the metrics utilized to measure the performance of the process is defined, and results are provided in the next section with a concluding remarks in the last section.

METHODOLOGY

In this section, the whole process of how occluded pedestrian is detected and avoided successfully is explained. The complete process is shown in Fig. 39. Each block of the process will be explained in subsequent sections, and finally, brief information about the baseline controllers used for comparison is presented.

Environment

Ego vehicle is modeled with NVIDIA PhysX model which is similar to dynamic bicycle model. Streets and intersections are built in the environment. Position, velocity, and acceleration information can be extracted from the environment, and steering, throttle, and brake to control the vehicle. Pedestrians with a point mass model are also added within the environment, with speed and heading control available for the pedestrian. Their position can be extracted from the environment. The environment can be seen in Fig. 38(a).

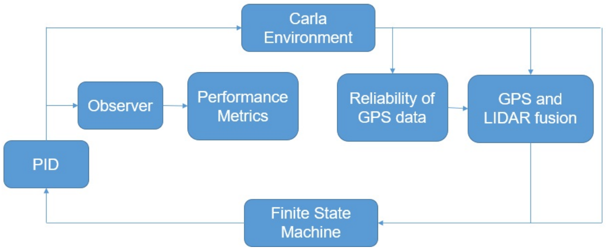


Fig. 39: Block Diagram of the Process

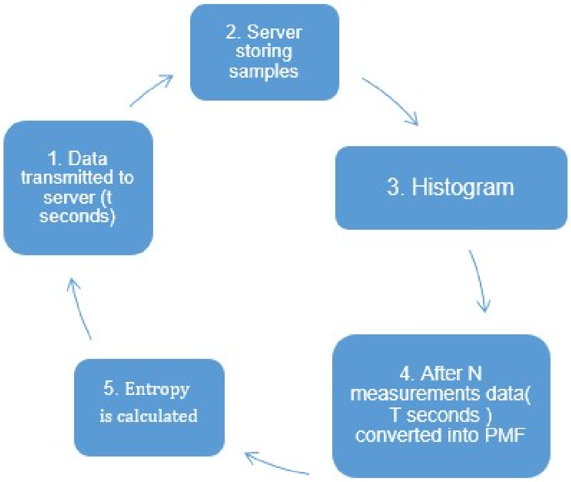

2. Server storing samples
3. Histogram
1. Data transmitted to server (t seconds)
4. After N measurements data(T seconds) converted into PMF
5. Entropy is calculated

Fig. 40: Flow diagram for entropy calculation

Reliability of GPS data

It is assumed that all the pedestrians and vehicles have GPS receivers and 4G/5G transceivers present. Pedestrians are transmitting their position information to a cloud server. The cloud server will then send the data to vehicles in the vicinity of pedestrians to modify their trajectories to avoid collisions. In Urban scenarios GPS can have high error [262]. The primary reason for this error is high-rise buildings that block the transmitter and receiver's Line of Sight (LOS). A process is designed to assess the reliability of the amount of error present in the data. The process is explained in Fig. 40. In the first step, N samples are collected. N has to be small so that process can be executed in real-time. After that, L_2 norm of each two-dimensional sample is calculated. Afterward, a histogram is calculated as shown in (19). where s_j is each GPS sample collected from a pedestrian. In Fig. 40, it is to be noted that t seconds is different from T seconds as to evaluate histogram, we need some samples; therefore, in our work, T=5t.

$$P(s_j) = \frac{Histogram(s_j)}{\sum_{i=1}^{n} Histogram(s_i)}. \tag{19}$$

Then in the next step entropy is calculated as shown in (20) .

$$H(S) = -\sum_{j=1}^{n} P(s_j) log_b P(s_j). \tag{20}$$

The process was applied on real-time data taken from the Dept of public health, city of Cincinnati [263] with two types of error added random walk and Gaussian Noise with shifted mean and high variance. Results in Fig. 41 show that a

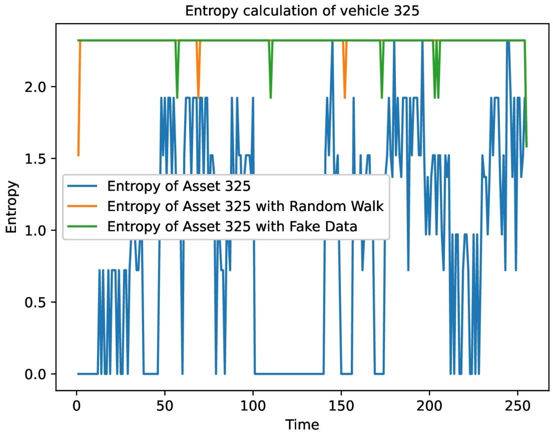


Fig. 41: Entropy of GPS data from Dept of Cincinnati

threshold (approximately 1.89) can be established to ascertain the variance of the data. Moreover, data can be rejected if the data has a high variance. The blue curve is the original data collected, while the green curve is when Gaussian error $X \sim \mathcal{N}(\mu = 0,\, \sigma^2 = 10)$ is added into the data to show the effectiveness of the method. Similarly, in orange curve random walk error $X \sim \mathcal{N}(\mu = 0,\, \sigma^2 = 3N)$ is added into the original data. where $N = \{1, 2, 3, ...\}$. In Fig. 41 it can be seen that a threshold of approximately 1.8 can be used, and if the entropy value is below this threshold, data will be rejected.

GPS and LIDAR Fusion

To fuse data two step process is followed taken from [264]. Scan Matching is performed as shown in (21)

$$\min_{x_i^{sm}(t)} (\frac{1}{2} e_{ij}^T(t) P_{rel} e_{ij}(t) + \frac{1}{2} e_i^T(t) P_{gps} e_i(t)). \tag{21}$$

$$e_{ij}(t) = z_{ij}(t) - \max(\left\| x_i^{sm}, x_j^{gps} \right\|_2). \tag{22}$$

$$e_i(t) = x_i^{sm} - x_j^{gps}. \tag{23}$$

where x_i^{sm} is realigned position of ego vehicle and x_j^{gps} is the GPS measurement from pedestrian j. While $z_{ij}(t)$ is the relative position measurement from LIDAR where i is the ego vehicle and j is the jth pedestrian and P_{rel}, P_{gps} are covariance matrices.

The benefit of using scan matching is that not only LIDAR and GPS data is fused, but the error is also corrected. The next step in the process is Kalman Filtering (KF). The benefit of using KF is twofold we get the next predicted state of the pedestrian while the amount of error in the position is corrected. The model used to predict the next state of a pedestrian is based on the point mass model.

The next challenge in the process is that there are three possible detections because of communication errors and local sensor (LIDAR) limitations. The three cases and how they are handled to create a final list of position information of all dynamic objects presented are shown in Fig. 42. A similar approach is followed in all three cases. First, LIDAR detections are matched with GPS data using the vicinity rule. The vicinity rule means that in the Horizontal Alert Level (HAL) of LIDAR detection, it is assumed that GPS detection will be available and then processed through KF to get the final predicted positions. The rest of the data is processed through KF to generate a final list of detections.

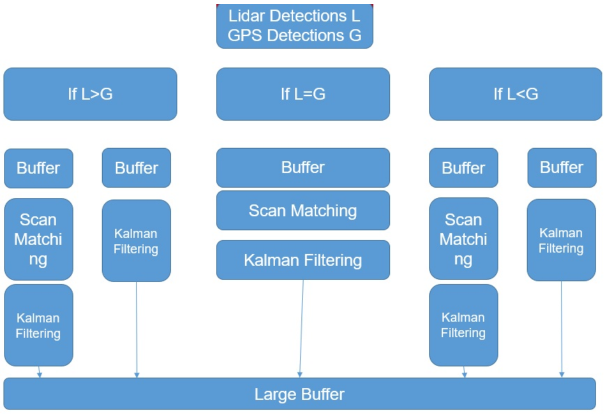


Fig. 42: GPS and LIDAR data Fusion

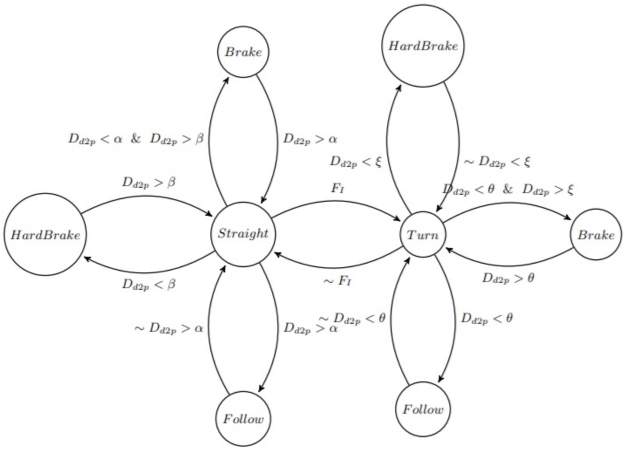


Fig. 43: Hierarchical Finite State Machine

Finite State Machine based Hierarchical Control

Finite State Machines (FSM) is a mathematical model through which a finite number of mathematical computations can be performed through a finite number of interconnected states. FSM is widely used to control machines requiring a sequence of operations. A machine can be in one of a finite number of states at any given time. State transitions happen based on indicators from the environment indicating the machine's state. Hierarchical FSM control is utilized to control the vehicle, and at a low-level PID control is implemented. The FSM controller is shown in Fig. 43. It has two higher-level states straight and turn. With a straight node, we have further three states: follow, brake, emergency brake, and similar three states in higher node turn but with different indicators to switch between states. The detail and function of each indicator are shown in table XII.

TABLE XII: Flags and their Interpretation

Symbol	Definition
F_I	Intersection Flag
α	Distance to start decelerating
D_{d2p}	Distance to closest pedestrian
β	Distance to start hard decelerating
θ	Distance to start decelerating while turning
ξ	Distance to start hard decelerating while turning

Performance Metrics

To measure the performance of the designed process, three metrics are used. The first metric consists of using the vehicle's lateral position and angle, implying the collision avoidance measure used to avoid the pedestrian. The metric is shown in (24)

$$J1 = \sum_{x} x^T Q x + u_1^T R u_1. \tag{24}$$

Q and R are positive definite and positive semi-definite matrices, respectively. Where x is the state vector with states, $[y \ \theta]$ where y is the vehicle's lateral position with respect to the origin and θ is the angle with respect to the ego vehicle. Furthermore, u_1 is the input vector with steering angle γ given to the vehicle.

This metric $J1$ is derived by implementing the observer model using the lateral position and angle of the dynamic model as shown in [265]. $J1$ is a convex function and can measure the energy consumed by the vehicle while performing evasive action to avoid the vehicle.

The second metric is based on the distance to the closest pedestrian maintained by the vehicle at all times and is shown in (25)

$$J2 = \frac{1}{D_{d2p}} + k\xi. \tag{25}$$

Where D_{d2p} is the distance of the vehicle with the closest pedestrian and k is a very high gain, and ξ is a binary value which is one when the vehicle collides with the pedestrian.

This metric is derived such that when the vehicle gets closer to the pedestrian, it penalizes the metric heavily as it is inversely proportional to the distance to the vehicle and gives a very high penalty if the vehicle collides with the pedestrian. A binary term is added with a high gain to indicate a collision. The additional binary term is added because the distance is measured from the vehicle's center to the center of the pedestrian. It is to be noted that the binary term is piece-wise defined, and when a pedestrian is in close vicinity of the vehicle, it is considered a collision.

Finally, the last metric measures deceleration within the vehicle when it is performing evasive action to avoid the vehicle. The metric is shown in (26)

$$J3 = \sum h(u_2)^T Q h(u_2). \tag{26}$$

where $h(u_2)$ is shown in (27) and u_2 is the acceleration of the vehicle when it is performing an evasive action to avoid the pedestrian

$$h(u_2) = (-1 + sign(u_2))u_2. \tag{27}$$

Baseline Controllers for Comparison

To show a better performance. Three baseline controllers were used, the first one is vanilla PID control with longitudinal control to avoid dynamic objects, and the other two have similar architecture as in Fig. 39. Nevertheless, some blocks were removed. In the second baseline, the block labeled reliability of GPS data is removed. In the third baseline controller, both the blocks' reliability of GPS data and fusion of LIDAR and GPS functionalities were removed, and the performances were compared. Vanilla PID is utilized only in metric J2 because the FSM-based controller implemented has PID at a low-level hierarchical control. Consequently, if the proposed method performs better after removing the blocks, it will perform better if only vanilla PID is compared. Vanilla PID is used in J2 to demonstrate that vehicle collides with a pedestrian in the absence of the proposed controller.

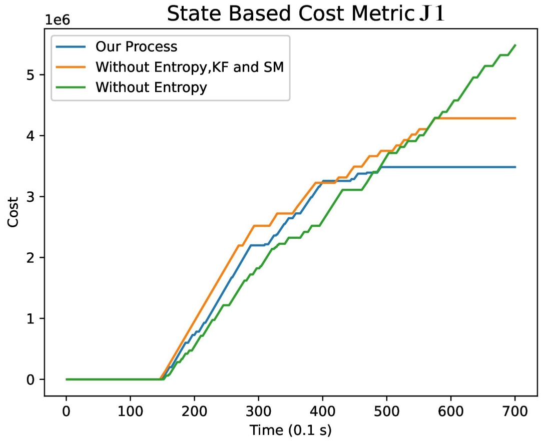


Fig. 44: State based Cost Metric

RESULTS AND DISCUSSION

The simulation environment used to test the designed process is CARLA [266] open-source environment. The scenario is described in Fig. 38 part (a). The pedestrian moving has a GPS error of $X \sim \mathcal{N}(\mu = 0, \sigma^2 = 3)$ and the pedestrian which is stopped on the sidewalk has an error of $X \sim \mathcal{N}(\mu = 0, \sigma^2 = 10)$. These errors are chosen because GPS errors in the literature are modeled using the Gaussian error model [267], and variance is chosen such that it remains under the acceptable range. This reference [268] shows that the error variance is less than 10 meters. The required speed is 20 m/s because the maximum speed limit in US streets is 45mph, approximately equal to 20 m/s. The vehicle has to start decelerating if the distance to the pedestrian is less than seven meters and has to go into hard deceleration mode if the distance to the closest pedestrian is less than four meters. The results with each metric are shown in Fig. 44, Fig. 45 and Fig. 46.

The metrics are measured only when the vehicle takes action to avoid a collision with the pedestrian. In Metric $J1$ that is shown in section , we can see the results in Fig. 44. Until 15 seconds, the pedestrian is not detected. No energy is consumed to avoid the pedestrian. However, after that, it can be seen that until 40 seconds, baseline two and baseline three algorithms mentioned in section are still consuming energy because the pedestrian on the sidewalk is reporting its position in the collision range of the vehicle and our process can reject this data using entropy metric as described earlier. In the second figure, Fig. 45 the baseline controller one collides with the pedestrian while other baselines and our complete process can avoid the pedestrian. The reason for this is that the setup provides occluded pedestrian's position beforehand, and it has knowledge well before there is a possibility of a collision, and the vehicle performs preemptive action. Moreover, the last metric $J3$ also shows similar results in Fig. 46 because of the same reasons described above. Vanilla PID is not used in Fig. 46 and Fig. 44 because low-level control of FSM includes PID, and the proposed method performs better when scan matching and KF are implemented within the system. Consequently, if the system outperforms when only specific features are turned off, it will certainly perform better than vanilla PID. In Fig. 45 it is explicitly used to show that the ego vehicle collides with the pedestrian and which is a safety hazard. Our process outperforms baseline controllers and can avoid pedestrians based on the metrics. However, it can be observed that baseline methods outperform the proposed method in some instances. This is because the entropy metric is not always perfect, and sometimes erroneous information results in extra energy consumption of energy, but the overall proposed method outperforms baseline methods. Another noteworthy thing is that having extra information prevents collision. Suppose such a system, as described above, had been developed and deployed. Arizona's fatal crash of a semi-automated vehicle with a pedestrian could have been avoided.

CONCLUSION

In this paper, a communication setup was provided to receive position information of the occluded so that collisions can be avoided. This solution is scalable because the communication protocol and access can be easily obtained as almost every pedestrian carries a GPS sensor and LTE/4G module in their cell phones. However, creating a solution in such a way creates a problem if the information has a significant amount of errors present within the information, which will reduce vehicle

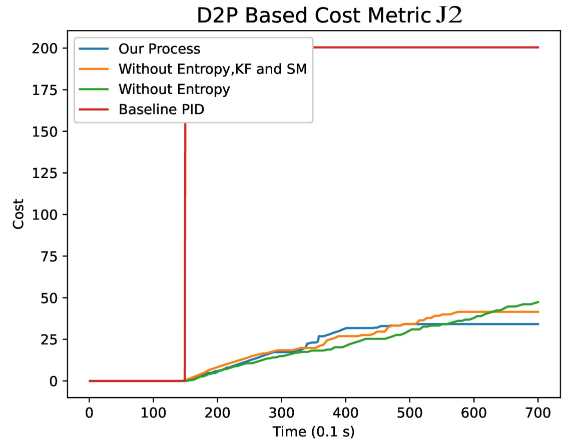


Fig. 45: Distance to Pedestrian based Cost Metric

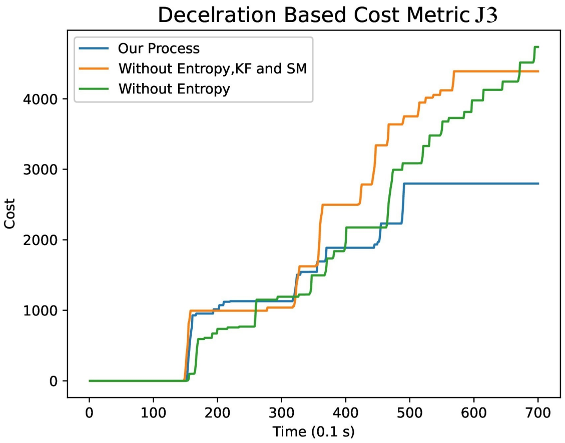


Fig. 46: Deceleration based Cost Metric

efficiency significantly. So to solve this problem, an entropy-based threshold is introduced to solve this problem. Then, the fusion process of LIDAR data with GPS data to remove other inaccuracies was introduced. Metrics were presented to show the performance of the proposed method. Results show that the proposed method can make HAVs safer for all the stakeholders present within the environment and make HAVs more efficient in terms of energy consumption to evade pedestrians. Since most pedestrians have LTE/4G modules and GPS devices within their cellular phones, Hardware Infrastructure is already present. Nothing new needs to be established. The method proposed is practical in the real world.

5 Receiver Data Validation by PNT Solutions from Other Sources

5a Robust Navigation for Urban Air Mobility via Tight Coupling of GNSS with Terrestrial Radionavigation and Inertial Sensing

ABSTRACT

This paper presents a method of tightly coupling carrier-phase-differential GNSS (CDGNSS) with terrestrial radionavigation system (TRNS) signals and data to build a robust positioning, velocity, and timing (PVT) solution for urban air mobility (UAM). UAM will require precise and robust PVT solutions that are resilient to interference and jamming. CDGNSS offers absolute positioning with high availability and sub-decimeter accuracy but cannot serve as the sole source of PVT for UAM because of its vulnerability to interference: a single potent GNSS jammer could deny UAM service across an entire city were GNSS the sole means UAM navigation. TRNS signals are stronger than those of GNSS and offer additional frequency diversity. Their multipath errors, although larger than for GNSS at street level due to the low elevation angles with which TRNS signals propagate from terrestrial transmitters, are manageably small at altitudes where UAM vehicles will operate. Thus, TRNS offers an attractive backup to GNSS for UAM. This paper explores two techniques for fusion of TRNS and CDGNSS: loosely- and tightly-coupled. The loosely-coupled technique fuses information from the two sensing modalities at the level of full PVT solutions. The tightly-coupled technique explored here fuses GNSS carrier phase and pseudorange measurements with TRNS pseudorange, Doppler, and calibrated pressure sensor measurements, together with inertial sensor measurements, to produce a unified PVT solution. Innovations-based measurement exclusion is applied to reduce the impact of GNSS and TRNS multipath errors and of pressure anomalies due, e.g., to ground effect at take-off and landing. Both loosely- and tightly-coupled techniques are tested on an aerial vehicle platform in an environment where both GNSS and TRNS signals are available. Error growth of the tightly-coupled technique during extended intervals of GNSS denial is studied to determine whether UAM service could continue uninterrupted when only inertial and TRNS measurements remain available.

INTRODUCTION

Central to the transportation revolution that will be driven by urban air mobility (UAM) is the problem of robust and secure navigation. Urban environments offer more challenges, such as interference and multipath, when compared to open-sky conditions. As the only positioning system that offers absolutely-referenced meter-level accuracy with global coverage, GNSS will no doubt play a significant role in this revolution. If strengthened against jamming and spoofing, carrier-phase-differential GNSS (CDGNSS), coupled with low-cost inertial sensing, will be nearly sufficient for position, velocity, and timing (PVT) needs. But nearly sufficient is insufficient: it is not enough for a UAM PVT solution to offer decimeter-accurate positioning with 99% availability, or even 99.9% availability. UAM will demand that its navigation systems offer dm-accurate positioning with integrity risk on the order of 10^{-7} for a meter-level alert limit and availability with several more 9s than 99.9% [10], [269]–[271].

This paper's technique is best viewed as one part of a comprehensive navigation solution concept called deep-layered navigation (DLN) in which synergistic but independent navigation systems are layered to increase accuracy and robustness (47). DLN is the navigation analog of the "defense in depth" concept in information security, where multiple layers of security controls and checkpoints are emplaced throughout a system such that even when some layers are breached, security is maintained. Likewise, in the safety-of-life UAM navigation context, multiple layers of navigation systems, all interoperable and mutually-reinforcing but substantially independent, are an essential defense against the whims of Mother Nature and the foibles of human nature.

At DLN's core sits redundant inertial navigation, which is virtually impervious to radio frequency (RF) interference, poor weather, signal blockage, and data ambiguity. The outermost layer—the default navigation system and first line of defense—is a specialized variant of inertially-aided CDGNSS, recently developed in [14], [272], that has been substantially secured against spoofing and substantially hardened against the multipath and signal blockage conditions of the urban *ground vehicle* environment, which can be considered a worst-case realization of the urban air vehicle environment. But despite its coupling with inertial sensing, the technique developed in [14] cannot tolerate extended GNSS outages. A secondary source of absolute PVT is required to bound the growth of position errors.

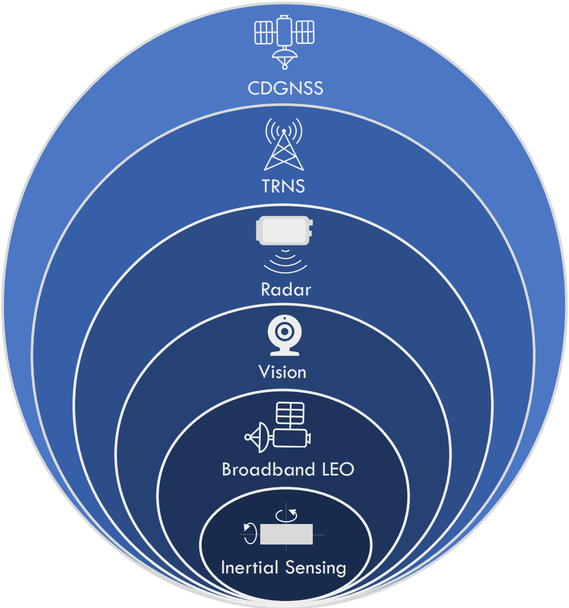


Fig. 47: Diagram of deep layered navigation. Overlapping layers of navigation to provide a PVT solution with high availability even when some of the layers are not available.

TRNS is the second line of defense in this paper's DLN concept, and the primary focus of the paper. Its pseudorange measurements to surrounding beacons are independent of, but fully interchangeable with, GNSS measurements. TRNS beacons provide much stronger signals compared to GNSS, operate at a different frequency, and offer a full absolutely-referenced backup PVT solution to GNSS. In particular, this paper explores tight coupling with NextNav's Metropolitan Beacon System (MBS). MBS is particularly attractive for UAM because its signals carry not only wideband (multipath-resistant) synchronization sequences for ranging but also corrections data for barometric altitude determination and, for CDGNSS.

This paper's development of a tightly-coupled GNSS-TRNS-inertial PNT system is a prelude to upcoming work on comprehensive deep-layered navigation for UAM, including additional layers based on radar localization, visual odometry, and LEO-satellite-provided GNSS.

Related Work

Augmentation of GNSS with terrestrial signals has been explored and shown to provide an added benefit over exclusive use of GNSS [273], [274]. But these techniques were demonstrated only on ground vehicles, and the sensor integration with CDGNSS was not tightly-coupled: the terrestrial signals were not incorporated in a way that permitted aiding of the ambiguity resolution process critical to CDGNSS. Loose coupling between GNSS and TRNS has been used to augment GNSS and provide an increase in both accuracy and availability on aerial vehicles [275]–[277]. But these methods fuse standard GNSS, not CDGNSS, with TRNS, and thus lack the decimeter accuracy that will be desirable, if not required, for UAM. Relative ranging measurements from ultra wide band (UWB) systems have been used to constrain integer ambiguities in CDGNSS and improve accuracy even with degraded GNSS reception [278]. Although UWB systems provide adequate performance, the limited range of the UWB signal makes it an unfavorable choice for UAM. Fusion of GNSS and signals of opportunity has been explored for aerial vehicles with promising results [279], [280]. But for a safety-of-life application like UAM, it is likely that signals of opportunity will be viewed less favorably than a dedicated TRNS as a secondary means of navigation.

Contributions

This paper makes four primary contributions. First, it presents and demonstrates the first use of tightly-coupled CDGNSS, TRNS, and inertial sensing to provide a secure and robust PVT solution. Second, it develops a novel innovations-based measurement exclusion technique which mitigates the impact of GNSS and TRNS multipath errors and pressure anomalies. Third, it offers a comparative analysis of loose and tight coupling on an aerial vehicle in an environment where only TRNS signals are available. Fourth, this paper preforms a study of error growths during periods of GNSS denial to determine whether PVT requirements for UAM could be met despite extended intervals of GNSS denial.

MEASUREMENT MODELS

This section presents the TRNS measurement models for the tightly- and loosely-coupled estimation strategies. In the loosely-coupled case, the TRNS measurement is the position determined by the TRNS receiver. In the tightly-coupled case, the TRNS measurements are pseudorange and altitude derived from barometric pressure. The measurements are fused with inertial sensing data and GNSS position and attitude solutions (loosely-coupled) or raw GNSS observables (tightly-coupled). Models for the inertial and GNSS measurements may be found in [14] and [32].

Loose Coupling

The position measurement $\mathbf{z}_{\mathrm{lc}}(k)$ at time k from the TRNS receiver is modeled as containing the true position $\mathbf{x}_{\mathrm{lc}}(k)$, a bias from the true position $\mathbf{b}_{\mathrm{lc}}(k)$, and white Gaussian measurement noise $\mathbf{w}_{\mathrm{lc}}(k)$:

$$\mathbf{z}_{\mathrm{lc}}(k) = \mathbf{x}_{\mathrm{lc}}(k) + \mathbf{b}_{\mathrm{lc}}(k) + \mathbf{w}_{\mathrm{lc}}(k) \tag{28}$$

The position bias $\mathbf{b}_{\mathrm{lc}}(k)$ is modeled as a mean-reverting Ornstein–Uhlenbeck (OU) process with white Gaussian noise $\mathbf{v}_{\mathrm{lc}}(k)$:

$$\mathbf{b}_{\mathrm{lc}}(k+1) = \alpha_{\mathrm{lc}}\mathbf{b}_{\mathrm{lc}}(k) + \mathbf{v}_{\mathrm{lc}}(k) \tag{29}$$

The term α_{lc} determines how quickly the process reverts back to zero. The mean reversion is governed by the time step $T(k) = t_{k+1} - t_k$ and the time constant τ_{lc} of the exponential decay:

$$\alpha_{\mathrm{lc}} = e^{-T(k)/\tau_{\mathrm{lc}}} \tag{30}$$

Tight Coupling

The TRNS measurements utilized for tight coupling are the TRNS pseudoranges and barometric-pressure-based altitude measurements. The TRNS pseudorange to transmitter i at time k, is modeled as including the true range to the TRNS transmitter $\rho_i(k)$, the receiver clock offset between the receiver clock and GPS time $\delta t_{\mathrm{rx}}(k)$, the transmitter clock offset $\delta t_i(k)$, multipath error $m_i(k)$, and white Gaussian measurement noise $w_i(k)$:

$$P_i(k) = \rho_i(k) + [\delta t_{\mathrm{rx}}(k) - \delta t_i(k)]c + m_i(k) + w_i(k) \tag{31}$$

The receiver clock is modeled as a temperature-compensated crystal oscillator (TCXO) with the two-parameter clock model given in [281]. The receiver clock offset $\delta t_{\mathrm{rx}}(k)$ evolves in time based on the clock offset rate $\dot{\delta t}_{\mathrm{rx}}(k)$, time step $T(k)$, and noise $\boldsymbol{v}(k)$:

$$\begin{bmatrix} \delta t_{\mathrm{rx}}(k+1) \\ \dot{\delta t}_{\mathrm{rx}}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T(k) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta t_{\mathrm{rx}}(k) \\ \dot{\delta t}_{\mathrm{rx}}(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \tag{32}$$

$$\boldsymbol{v}(k) = \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \tag{33}$$

The receiver clock offset noise $\boldsymbol{v}(k)$ is assumed to be zero mean with a variance that is a function of two parameters, h_0 and h_{-2}, related to the asymptotes of the Allan variance of the receiver clock [281]:

$$\mathbb{E}\left[\boldsymbol{v}(k)\boldsymbol{v}^{\mathsf{T}}(k)\right] = S_{\mathrm{g}} \begin{bmatrix} \frac{T^3(k)}{3} & \frac{T^2(k)}{2} \\ \frac{T^2(k)}{2} & T(k) \end{bmatrix} + S_{\mathrm{f}} \begin{bmatrix} T(k) & 0 \\ 0 & 0 \end{bmatrix} \tag{34}$$

$$S_{\mathrm{g}} = 2\pi^2 h_{-2} \tag{35}$$

$$S_{\mathrm{f}} = \frac{h_0}{2} \tag{36}$$

The TRNS transmitter clocks are disciplined by a GNSS receiver to keep them close to GPS time, but a small offset remains. The offset between the ith transmitter's clock and GPS time, $\delta t_i(k)$, is modeled as a mean-reverting OU processes with α_{tx} being modeled as in (30) but having a unique time constant τ_{tx} to match the behavior of the transmitter clock offsets.

$$\delta t_i(k+1) = \alpha_{\text{tx}}\delta t_i(k) + v_i(k) \tag{37}$$

$$\alpha_{\text{tx}} = e^{-T(k)/\tau_{\text{tx}}} \tag{38}$$

The TRNS receiver measures altitude using a barometric pressure sensor that is calibrated over the NextNav MBS network using information from NextNav's weather stations. The calibration does not occur continuously; rather, it is only updated once per day. The long update interval allows for pressure biases to accumulate due to short-term changes in temperature, pressure, or other changes in weather. The altitude measurement at time k, $z_{\text{p}}(k)$, is modeled as containing the true altitude $x_{\text{p}}(k)$, a bias $b_{\text{p}}(k)$, and white Gaussian measurement noise $w_{\text{p}}(k)$:

$$z_{\text{p}}(k) = x_{\text{p}}(k) + b_{\text{p}} + w_{\text{p}}(k) \tag{39}$$

The measurement bias is modeled as an OU process with white Gaussian noise $v_{\text{p}}(k)$ and with α_{p} is modeled according to (30) with a unique time constant τ_{p} that is selected to model the pressure sensor errors:

$$b_{\text{p}}(k+1) = \alpha_{\text{p}} b_{\text{p}}(k) + v_{\text{p}}(k) \tag{40}$$

$$\alpha_{\text{p}} = e^{-T(k)/\tau_{\text{p}}} \tag{41}$$

ESTIMATOR

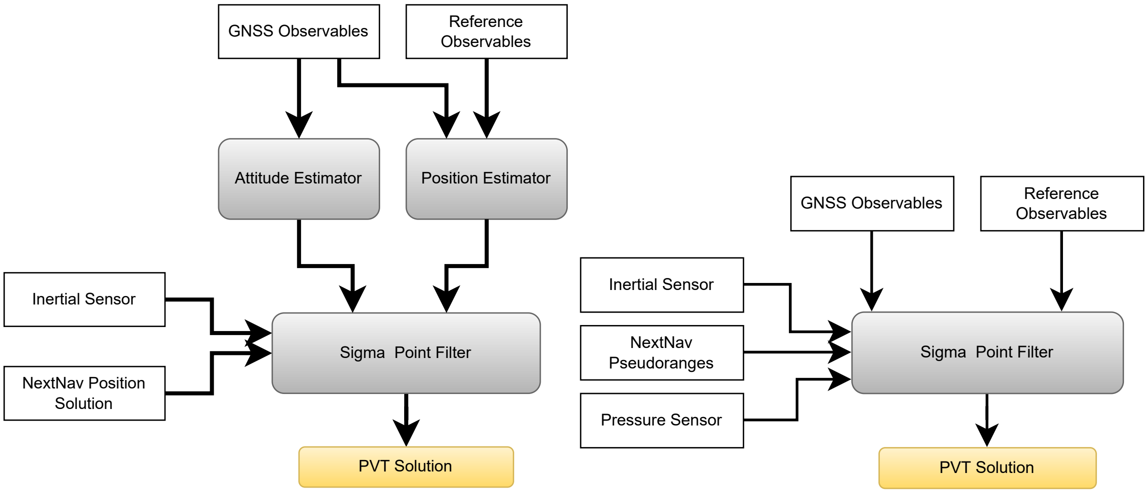


Fig. 48: Left: In the loosely-coupled configuration, the GNSS-based position attitude estimates are fused with inertial data and the TRNS-based position estimate to produce a PVT solution. Right: In the tightly-coupled configuration, raw measurements from each source are combined in a single estimator to produce a PVT solution.

This paper incorporates the measurement models of the TRNS measurements into the Radionavigation Laboratory's (RNL) precise positioning engine, called PpEngine [14], [16], [32]. The positioning engine was constructed in two versions. The first version loosely couples the GNSS and TRNS measurements with the inertial sensor measurements (Fig. 48). The second tightly couples the GNSS and TRNS measurements with the inertial sensor measurements (Fig. 48).

It will be convenient to define the following reference frames:

u: The *IMU frame* is centered at and aligned with the IMU accelerometer triad.
b: The *body frame* has its origin at the phase center of the aerial test vehicle's primary GNSS antenna. Its x axis points towards the phase center of the secondary antenna, its y axis is aligned with the boresight vector of the primary antenna, and its z axis completes the right-handed triad.
w: The *world frame* is a fixed geographic East-North-Up (ENU) frame, with its origin at the phase center of the reference GNSS antenna, which is located at a fixed base station with known coordinates.

Both the loosely- and tightly-coupled estimator states are an augmented version of the original PpEngine state containing the aerial vehicle's position in the world frame $\mathbf{r}_k^{\mathrm{w}}$, velocity in the world frame $\mathbf{v}_k^{\mathrm{w}}$, attitude expressed as Euler error angles between the body frame and the world frame $\mathbf{e}$, accelerometer bias in the IMU frame $\mathbf{b}_{\mathrm{a}}^{\mathrm{u}}$, and gyro bias in the IMU frame $\mathbf{b}_{\mathrm{g}}^{\mathrm{u}}$:

$$\mathbf{X} = \left\{\mathbf{r}_k^{\mathrm{w}}, \mathbf{v}_k^{\mathrm{w}}, \mathbf{e}, \mathbf{b}_{\mathrm{a}}^{\mathrm{u}}, \mathbf{b}_{\mathrm{g}}^{\mathrm{u}}\right\} \tag{42}$$

Loosely-Coupled Estimator

The loosely-coupled implementation of PpEngine determines the aerial vehicle's state (43) by cascading multiple estimators (Fig. 48). The CDGNSS position of the primary GNSS antenna is determined using the double differenced pseudoranges and carrier phase measurements from the aerial vehicle's primary GNSS antenna and the reference receiver [16]. A separate position measurement is determined by the NextNav TRNS receiver, and its fixed position relative to the phase center of the aerial vehicle's primary GNSS antenna. Next, the two dimensional attitude of the aerial vehicle is estimated using the double differenced pseudoranges and carrier phase measurements between the two GNSS antennas and the known baseline between the antennas [16]. In addition, a position and attitude estimate is determine using the measurements from the inertial sensor. These separate position and attitude solutions and their estimated covariance are cascaded into a second level estimator to produce the complete position and attitude solution (43) with the state augmented to include the TRNS position bias $\mathbf{b}_0^{\mathrm{w}}$.

$$\mathbf{X}_{\mathrm{lc}} = \left\{\mathbf{r}_k^{\mathrm{w}}, \mathbf{v}_k^{\mathrm{w}}, \mathbf{e}, \mathbf{b}_{\mathrm{a}}^{\mathrm{u}}, \mathbf{b}_{\mathrm{g}}^{\mathrm{u}}, \mathbf{b}_{\mathrm{lc}}^{\mathrm{w}}\right\} \tag{43}$$

Tightly-Coupled Estimator

The tightly-coupled version of PpEngine incorporates all the sensor measurements directly into a single estimator (Fig. 48). The single estimator is used to determine the state [14], [282]. The estimator state (44) was augmented by including estimates of the TRNS receiver clock offset δt_{rx}, TRNS receiver clock offset rate $\dot{\delta t}_{\mathrm{rx}}$, transmitter clock offsets for n of transmitters $\delta t_i, \ldots, \delta t_n$, and pressure sensor offset $b_{\mathrm{p}}^{\mathrm{w}}$.

$$\mathbf{X}_{\mathrm{tc}} = \left\{\mathbf{r}_k^{\mathrm{w}}, \mathbf{v}_k^{\mathrm{w}}, \mathbf{e}, \mathbf{b}_{\mathrm{a}}, \mathbf{b}_{\mathrm{g}}, \delta t_{\mathrm{rx}}, \dot{\delta t}_{\mathrm{rx}}, \delta t_1, \ldots, \delta t_n, b_{\mathrm{p}}^{\mathrm{w}}\right\} \tag{44}$$

Estimating the clock parameters for both the local clock and transmitter clock from the pseudorange measurements is not fully observable. However, the local clock offset can be separated from the transmitter clock offsets stylistically. The errors that are constant across all transmitters will show up in the local clock offset. Any errors that are unique to one transmitter will appear in the clock offset of that transmitter.

The transmitter clock offsets are being estimated despite not being fully observable, because the addition of a second TRNS receiver would allow for the transmitter clock offset to be observable. If two UAM vehicles each had their own TRNS receiver the vehicles would be able to jointly estimate the TRNS transmitter clock parameters. This would make the TRNS transmitter clock offsets observable for any UAM network operating two or more vehicles.

Innovations Testing

The tightly-coupled PpEngine preforms outlier exclusion to reject TRNS measurements that may be affected by multipath, and pressure measurements anomalies. The outliers are detected by preforming a hypothesis test on the TRNS pseudorange innovations, and the pressure sensor innovations. The normalized innovation squared (NIS) (45) at time k is found by normalizing the innovation $\boldsymbol{\nu}(k)$ with the innovation covariance $\mathbf{S}(k)$.

$$\mathrm{NIS} = \boldsymbol{\nu}^{\mathsf{T}}(\mathrm{k})\mathbf{S}^{-1}(\mathrm{k})\boldsymbol{\nu}(\mathrm{k}) \tag{45}$$

The NIS is distributed as Chi squared with nz degrees of freedom, where nz is the number of measurements at epoch k. The NIS can be used as a test statistic in a hypothesis test with $\nu > 0$ is the threshold that yields the chosen probability of false alarm P_F.

$$\mathrm{NIS} \underset{\mathrm{H}_0}{\overset{\mathrm{H}_1}{\gtrless}} \nu^* \tag{46}$$

An outlier is detected when the NIS surpasses the threshold value ν^*. Further processing then identifies and excludes the outlier measurement from the update.

DATA COLLECTION

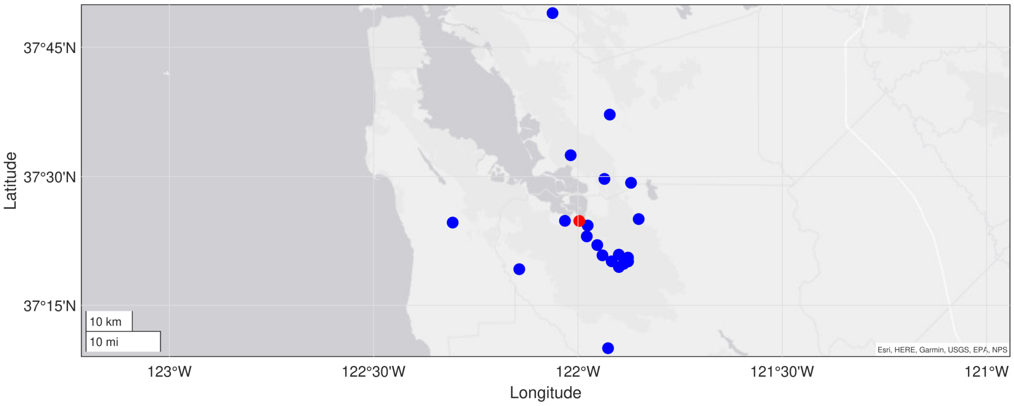


Fig. 49: Map of the location of the TRNS transmitters visible to the TRNS receiver during flight and the flight location in red

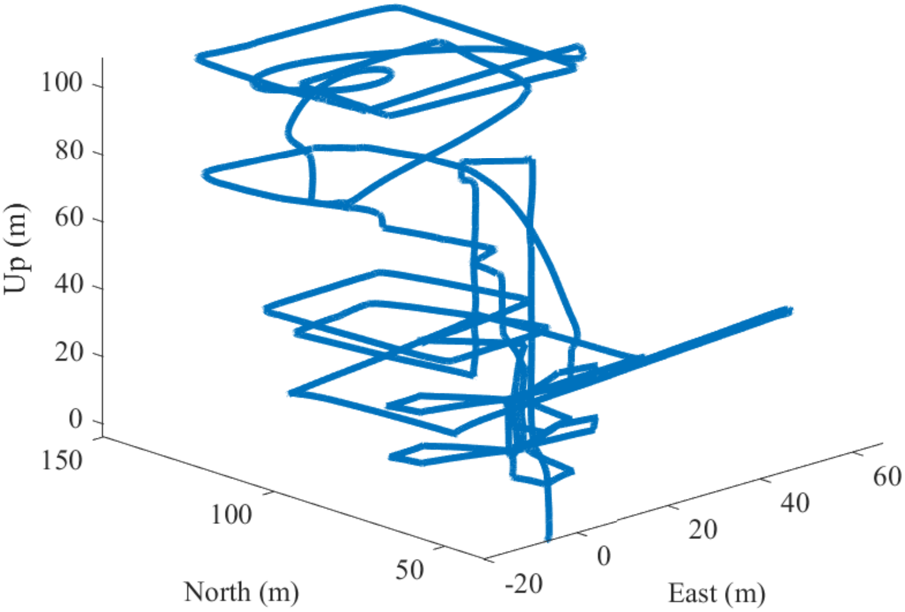


Fig. 50: Flight path of the aerial vehicle during data collection flight. The position is shown in meters from the phase center of the reference receiver antenna in the ENU frame.

A data set was collected using the UT RNL aerial test vehicle (Fig. 51). The aerial test vehicle is a DJI Matrice 300 multi-rotor vehicle with a sensors mounted on the vehicle. The sensors used in this data set are a dual antenna GNSS L1

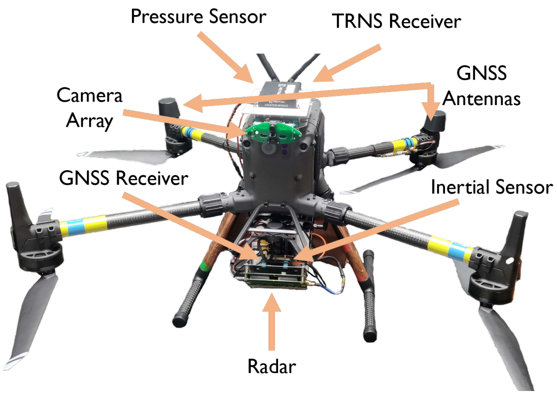


Fig. 51: RNL's aerial test vehicle. It carries two single frequency GNSS antennas, a radar sensor, TRNS receiver, pressure sensor, and three cameras.

receiver, a NextNav TRNS receiver, and a Bosch BMX055 consumer grade inertial measurement unit (IMU). In addition, a GNSS reference receiver was placed near the planned flight location. The dataset contains dual antenna GNSS L1 double differenced pseudoranges, carrier phase measurements, TRNS position solution, TRNS pseudoranges, calibrated barometric pressure sensor measurements, and inertial sensor measurements. The reference and aerial test vehicle GNSS receivers utilized the RNL's *RadioLynx*, GNSS front end with a 5 MHz bandwidth, and was processed with the PpRx software-defined GNSS receiver [29]–[33].

A data collection flight was preformed in an area where NextNav TRNS signals are available (Fig. 49). The flight path was limited to 80 meters by 100 meters in the east and north directions due to constraints on communication range with the aerial vehicle. FAA regulations limited the altitude ceiling to 121 meters above ground level. The full flight path shown in (Fig. 50). The recorded dataset includes approximately 900 seconds of data at different altitudes and varying vehicle velocities.

RESULTS

The following section presents an analysis of the TRNS pseudoranges and an evaluation of the loosely- and tightly-coupled implementations of the position estimator.

Analysis of TRNS Pseudoranges

The TRNS pseudoranges were analyzed using the collected dataset and the tightly-coupled version of PpEngine. The estimated TRNS transmitter clock offsets (Fig. 52) were observed to determine the behavior of the TRNS transmitter clocks. The estimated transmitter clock offsets were found to be non zero and unique to each transmitter. The largest difference between transmitter clock offsets indicates that the transmitter clocks differ by as much as 30 nanoseconds. These estimated TRNS transmitter clock offsets are not fully observable meaning part of the estimate may be due to the TRNS receiver clock offset.

The TRNS pseudorange innovations (Fig. 53) were examined and found to be nearly zero-mean and white, demonstrating that the pseudorange errors were well modeled. This shows that despite the lack of full observability of the clock offsets, the location of the TRNS receiver can still be estimated.

Artificial GNSS Outage

A long scale GNSS outage experiment was preformed to represent a worst case scenario of a full loss of GNSS availability. The experiment was preformed for both loosely- and tightly-coupled estimators. The loosely-coupled solution (Fig. 54) shows that during a long duration GNSS outage a bias between the CDGNSS position and TRNS position arises. During this outage the estimated error standard deviation was 1 meter in both the east and north directions.

The experiment was repeated utilizing the tightly-coupled estimator (Fig. 54). During the outage the estimated error standard deviation was 2 meters in both the east and north directions. The tightly-coupled estimator has a similar offset from the

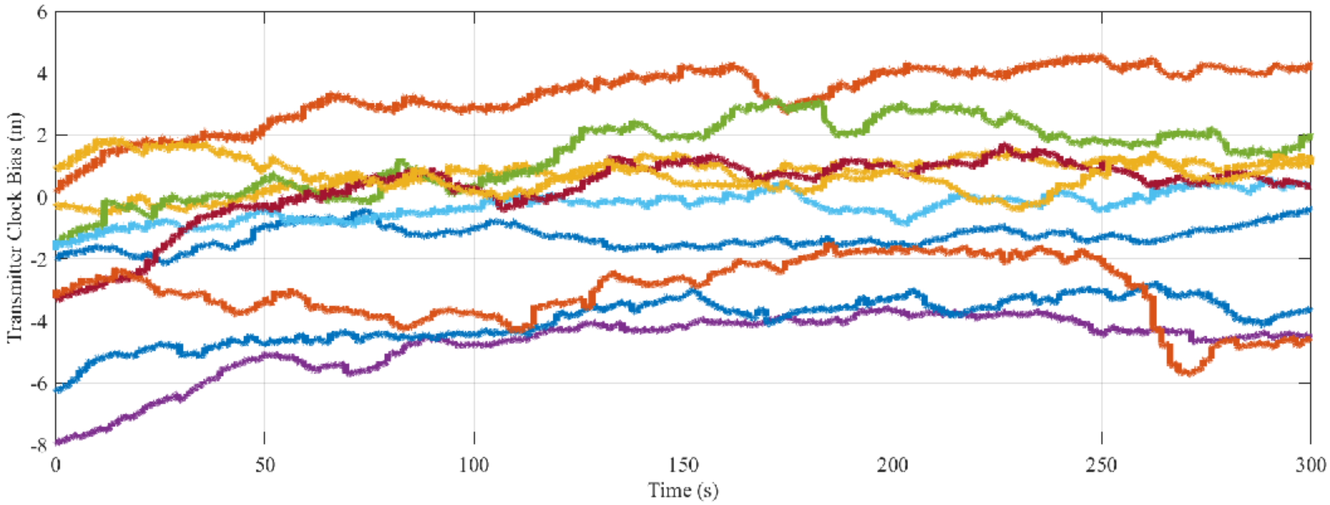


Fig. 52: Estimated TRNS transmitter clock offset of the TRNS transmitters in meters

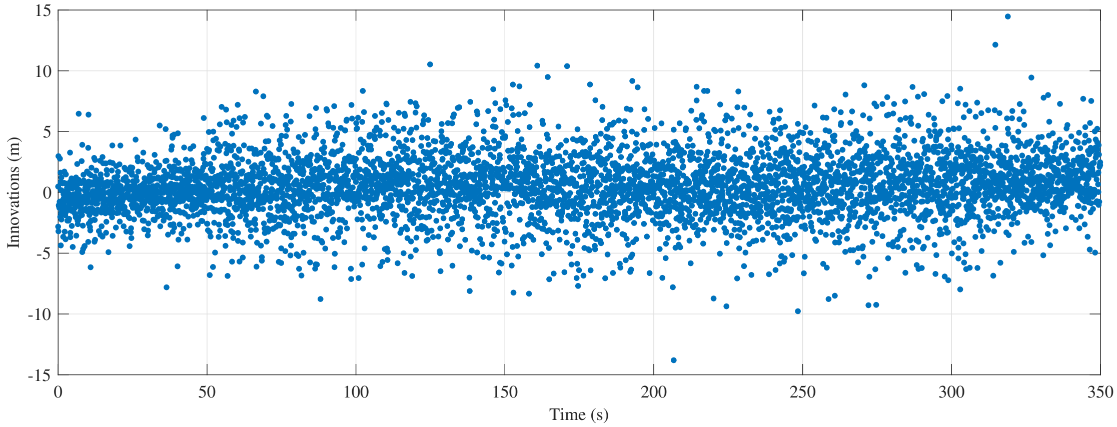


Fig. 53: TRNS pseudorange innovations in meters

CDGNSS position solution showing that there is an unknown source of error in the TRNS network causing the offset of the TRNS position solution.

Intermittent Artificial GNSS Outages

A UAM vehicle will likely only face interference or a loss of GNSS availability for a portion of the flight. A test of intermittent GNSS availability was preformed by artificially suppressing GNSS measurements from entering the estimator for a duration of 30 seconds with a 60 second period. The estimator was first tested with a series of intermittent GNSS outages without the addition of the TRNS measurements (Fig. 55).

The error grows in the position during the outage is exponential. The exponential error growth is cause by drift in the consumer grade inertial sensor. During the short duration outage the 1-σ uncertainty grew to 28 meters in the east and north directions, and 8.4 meters in the Up direction. The maximum error during the test was found to be 166.7 meters, demonstrating the need for the addition of TRNS to constrain the error growth during GNSS outages.

The intermittent GNSS outages test was then preformed with the addition of TRNS measurements. The same interval was used so the tightly-coupled and loosely-coupled estimators could be directly compared to one another. For both tightly- and

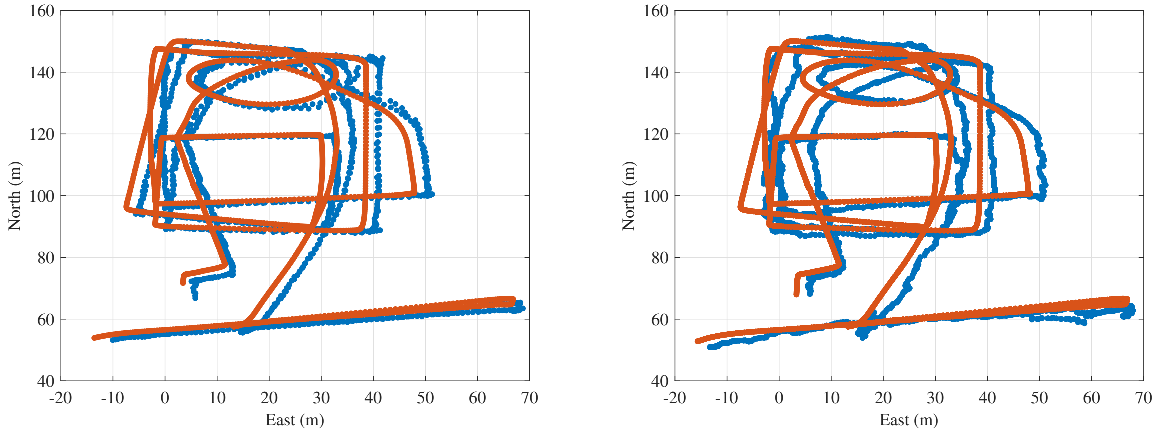


Fig. 54: Left: The blue is the loosely-coupled TRNS position solution during period of artificial GNSS outage. The orange is the CDGNSS position. Right: The blue is the tightly-coupled TRNS position solution during period of artificial GNSS outage. The orange is the CDGNSS position.

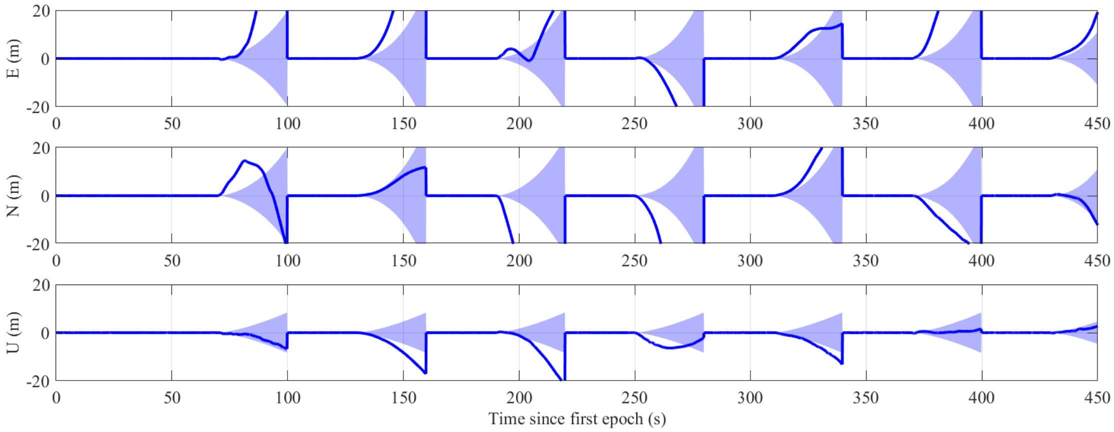


Fig. 55: Intermittent periods of artificial GNSS and TRNS outage. The solid blue shows the position errors in meters in the ENU frame due to the inertial sensor. The shaded region is the 1-σ uncertainty.

loosely-coupled estimators the addition of TRNS measurements served to constrain the error growth during GNSS outages (Fig. 56).

The loosely-coupled estimator 1-σ error uncertainty grew to 1 meter in each direction. In contrast, the tightly-coupled estimator 1-σ error uncertainty grew to 1.5 meter in the east and north directions while only growing to 1 meter in the Up direction. The tightly-coupled estimator had a larger uncertainty due to the unknown TRNS transmitter clock offsets. Both estimators had very similar behavior in the Up direction due to the poor vertical dilution of precision causing the position estimate in the Up direction to be heavily weighted by the pressure measurements.

CONCLUSION

TRNS provides a backup PNT source that can be utilized during GNSS outages to constrain the error growth of an inertial sensor. this paper demonstrates the first use of tightly-coupled CDGNSS, TRNS, and inertial sensing to provide a robust PVT solution on an aerial vehicle. An innovations-based measurement exclusion technique which mitigates the impact of TRNS multipath errors and pressure anomalies. A comparative analysis of loose and tight coupling on an aerial vehicle during simulated GNSS outages. It was found that during short duration GNSS outages the 1-σ position error in the east and north directions were 1 meter and 1.5 meters for the loosely- and tightly-coupled estimators respectively. Additional

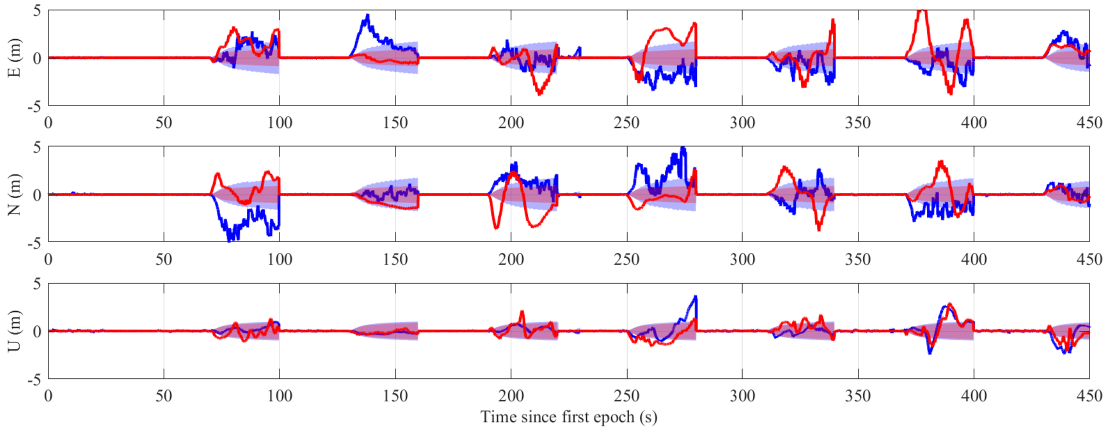


Fig. 56: Position errors in meters in the ENU frame during intermittent periods of artificial GNSS outages. The solid line is the error of the position estimate and the shaded region is the 1-σ uncertainty. The blue shows the tightly-coupled estimator while the red shows the loosely-coupled estimator.

positioning systems will need to be added to provide overlapping layers to the positioning in order to achieve decimeter accurate position during a GNSS outage. The results in this paper provide a first step in building such a system.

5b A Proposal for Securing Terrestrial Radio-Navigation Systems

ABSTRACT

The security of terrestrial radio-navigation systems (TRNS) has not yet been addressed in the literature. This proposal builds on what is known about securing global navigation satellite systems (GNSS) to address this gap, re-evaluating proposals for GNSS security in light of the distinctive properties of TRNS. TRNS of the type envisioned in this paper are currently in their infancy, unburdened by considerations of backwards compatibility: security for TRNS is a clean slate. This paper argues that waveform- or signal-level security measures are irrelevant for TRNS, preventing neither spoofing nor unauthorized use of the service. Thus, only security measures which modify navigation message bits merit consideration. This paper proposes orthogonal mechanisms for navigation message encryption (NME) and authentication (NMA), constructed from standard cryptography primitives and specialized to TRNS: message encryption allows providers to offer tiered access to navigation parameters on a bit-by-bit basis, and message authentication disperses the bits of a message authentication code across all data packets, posing an additional challenge to spoofers. The implementation of this proposal will render TRNS more secure and resilient than traditional civil GNSS.

INTRODUCTION

Global Navigation Satellite Systems (GNSS) have provided excellent positioning solutions in open, outdoor environments, enabling a wide range of navigation and timing applications. However, the indoor environment remains largely out of reach to these weak signals. The requirement for accurate and assured indoor positioning limits the effectiveness of GNSS in high-stakes, safety-of-life applications like enhanced E911, as well as in a new generation of commercial applications like warehouse automation and asset tracking.

Terrestrial radionavigation systems (TRNS), such as the commercial systems Locata [283] and NextNav [284], are emerging to address these needs. These systems are marketed to provide position, navigation, and timing (PNT) solutions in environments where GNSS signals are degraded or denied. TRNS consist of networks of synchronized terrestrial transmitters, or *pseudolites*, which operate analogously to GNSS satellites. These pseudolites broadcast signals powerful enough to reach the interiors of typical buildings, permitting the acquisition of terrestrial PNT service by urban or indoor users. A TRNS may serve to augment GNSS signals, improving solution geometry and availability in dense urban areas [285], [286], or it may serve as a primary navigation aid in the indoor environment [287].

The TRNS architecture [283], [284] and its sensitivity to wide-band radio-frequency interference (RFI) [288], [289] have been investigated in the literature. There have not, however, been any public proposals for how to secure TRNS–or even any substantive discussion of security considerations.

Broadly, the security of TRNS parallels that of other historical radio-navigation systems, and thus security considerations for TRNS can draw from lessons learned in the vibrant body of research on GNSS signal security. The important distinctions are threefold: first, the vastly different dynamic range of terrestrial versus space-based transmissions; second, the largely indistinguishable angular distribution of spoofed and authentic signals; and third, the possibility of multi-lateral (i.e. network) sensing of transmissions within the space bounded by the pseudolites. Of particular note is the way in which the adversary's receive power advantage renders exotic signal-level security techniques like spreading code authentication [290], [291] or deterministic code-phase dithering irrelevant: the adversary can always produce a pristine signal replica.

Contributions. This paper makes two contributions. First, it analyzes the security considerations of TRNS with these three differences in mind. Second, it offers a concrete proposal for how to secure TRNS, with a focus on data-level security in recognition of the futility of waveform- or signal-level security. This concrete proposal has two non-obvious aspects: MAC leavening, whereby a modest number of message authentication bits spread throughout the transmitted packets provide a significant improvement in security, and multi-level encryption, which has not been used before in PNT security and makes the adoption of this proposal more enticing for commercial service providers.

Organization of this paper. Section analyzes the security considerations of TRNS. Section gathers results from past proposals for GNSS security, and discusses the relevance of each technique for TRNS. Section details this paper's proposal for securing TRNS with navigation message encryption and/or navigation message authentication. Section concludes the paper.

SECURITY CONSIDERATIONS FOR TRNS

From the perspective of a radio-navigation system, there are essentially two types of adversaries: parties wishing to obtain service without authorization (stow-aways), and parties wishing to deny, degrade, or deceive authorized users of the service (jammers or spoofers). This divides radio-navigation security into two domains, termed Encryption (denying stow-aways) and Authentication (detecting spoofing). (N.B. that cryptographic encryption techniques are a useful tool in both domains). The focus of this work on terrestrial commercial systems prompts the adoption of the term "subscriber" to refer to an authorized user.

Dynamic Range

The greater dynamic range of terrestrial signals is a fundamental difference in the following sense: with GNSS, a spoofer cannot easily gain an advantage in received signal strength by moving closer to the transmitter, because this would require climbing thousands of kilometers above the ground. Instead, the adversary who wishes to obtain a pristine signal must build a large antenna. In TRNS, however, the adversary can "walk right up to" the pseudolite, obtaining a signal as clear as they could wish. Furthermore, because a subscriber cannot anticipate how much path loss may be present, it cannot anticipate how strong a signal ought to be after de-spreading. These asymmetries enable an adversary to obtain pristine signal replicas at low cost and high reliability, by locating a receive antenna close to the pseudo-lite. This renders spreading code encryption (SCE) (after the fashion of the GPS P(Y) code) largely irrelevant for TRNS: an adversary can always build a network of receivers to obtain both the pseudolites' spreading codes and position.

Radio-Frequency Interference

Radio-navigation systems, both GNSS and TRNS alike, are susceptible to RFI caused by jammers and spoofers. Fig. 57 gives an overview of RFI. Spoofing is of particular interest among all the RFI threats, as it stealthily fools a victim receiver without leaving obvious telltale signs. As a matched spectrum interference, spoofing signal is statistically correlated with the authentic signal, and a spoofer can achieve maximum spoofing efficacy by arbitrarily adjust this signal's power, code phase, carrier phase, and signal structure. Spoofing can be broadly classified into the following types of attack:

1) Self-consistent spoofing: This attack synthesizes false code phases and beat carrier phases, such that a desired position/timing fix is induced at the victim receiver without triggering an alarm from an unusual code/carrier divergence.

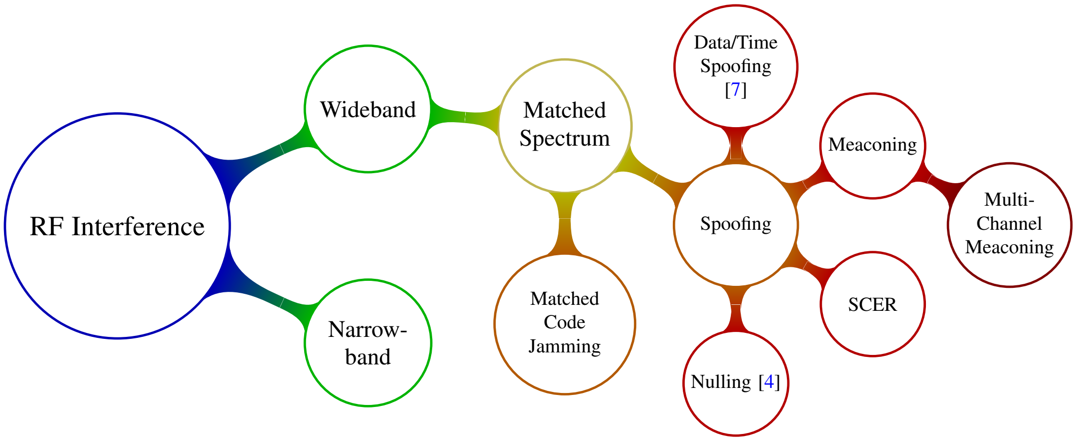


Fig. 57: A taxonomy of RF interference (i.e. an attaxonomy).

2) Data/Time spoofing: This attack generates a signal that has counterfeit data bits but is otherwise in near-perfect code-phase alignment with the authentic signal within the tracking channel of the victim receiver.
3) Security Code Estimation and Replay (SCER): This attack generates a counterfeit signal with a delay, by tracking individual signals and attempting to estimate each signal's unpredictable security code chips or navigation data bits on the fly.
4) Meaconing: This attack records the ensemble of authentic signals and replays them to create a desired position/timing offset. This can be done by either rebroadcasting the authentic signals recorded from a remote antenna at the intended position, or inducing independent delay variations in each authentic signal using phased-array signal processing [292].

Spoofing

The threat from GNSS spoofing has been a concern within the GNSS community, ever since a portable spoofer was developed and successfully tested against a COTS receiver [293]. A number of live-signal spoofing tests in a controlled environment which followed thereafter also affirmed the effect [34], [36], [294]. This threat continues to be relevant today, with recent rumors of spoofing "in the wild" seen in specific spots such as Black Sea [295], Syria [296] and China [297], or affecting multiple victim receivers which coincidentally move along the same track [298]. With recent advancements in RF microelectronics, together with open-source GNSS signal generation software, building a functional GNSS spoofer will become more accessible to the masses in the near future [299]. The spoofing threat is also relevant to TRNS because a functional TRNS spoofer can be modified from a GNSS spoofer, given sufficient resources and knowledge of the TRNS signal architecture.

TRNS has differentiated itself by having a high SNR and a limited-access standard, which is perceived to be able to counter against conventional spoofers that rely on high signal power and accurate prediction of spreading code and/or navigation data bit to mount a successful attack. However, these characteristics do not make TRNS foolproof against all spoofing threats. In fact, TRNS system has to tackle additional challenges due to high signal strength, wider signal dynamic range, proximity of threats to transmitters, as well as a potential reliance on GNSS for network synchronization. TRNS therefore faces a longer list of vulnerabilities from its signal and physical characteristics than GNSS.

Unlike GNSS signals that have signal strength below noise floor, the spreading code sequence of TRNS can be exposed without the use of high-gain antenna due to its high SNR. Reference [300] shows that the time slot usage, transmitters' PRN and navigation data bit of the Metropolitan Beacon System (MBS) from NextNav can be derived by analyzing the power spectrum of the MBS signal. This makes the cost of SCER attack on TRNS lower than that on GNSS, since the embedded security codes of TRNS can be more easily observed and hence estimated. In addition, even if TRNS adopts a restricted access standard and requires the use of secure tamper-resistant receiver to store the secret key like military GNSS signals, it is still susceptible to record-and-replay attacks.

TRNS provides a wide-area positioning service using a network of synchronized terrestrial transmitters. To ensure high accuracy in the PNT solution, stringent synchronization and frequency stability requirements are placed on all pseudolites, which may be satisfied either by: (1) the use of dedicated low-latency fiber-optic connection across the entire network, which will incur significant setup cost and will limit the deployment sites, or (2) the use of GNSS-disciplined atomic clocks, which reduces infrastructure cost and offers greater flexibility in the placement of the pseudolites. While option 2 may be preferable to providers, it exposes TRNS to an additional attack surface through its reliance upon GNSS. In addition, the relative accessibility of the pseudolites compared to the Earth-orbiting GNSS satellites indicates that TRNS is more susceptible to direct attacks, either by physical or cyber tampering, or by co-locating a high-power interference transmitter to overwhelm its signal.

LESSONS LEARNED FROM GNSS SECURITY

TRNS inherits from traditional radio-navigation a bevy of well-known attacks. For the same reason, TRNS can benefit from the products of a vibrant research effort over the past 20 years to secure GNSS. Not all the techniques that have been proposed for securing GNSS are applicable to TRNS— but it is equally true that the obligation of GNSS operators to backwards compatibility has prevented them from fully exploiting these developments. The time is right to incorporate what has been learned about GNSS security into TRNS. The purpose of this section is to review some of the most powerful security techniques that have been proposed for GNSS and to identify those ideas that are compatible with TRNS.

GNSS spoofing defenses proposed in recent literature can be broadly classified into two categories: (1) cryptographic techniques that utilize unpredictable but verifiable signal modulation in the GNSS spreading code or navigation data, and (2) non-cryptographic techniques such as signal processing techniques, geometric techniques, or drift monitoring techniques. A comprehensive review of GNSS spoofing defenses is presented in [4]. While these techniques have been proven to be effective for GNSS, there are challenges to their implementation for TRNS.

Non-cryptographic Defenses

The preliminary ideas of GNSS spoofing defenses fall within the realm of non-cryptographic defenses, as they do not require any changes to GNSS signal-in-space (SIS). These techniques are categorized based on their method of differentiating spoofing signals from authentic signals, by looking for consistency in the signal characteristics, signal geometry, or PNT solution.

Geometric techniques exploit the RF signals' geometric diversity to verify the authenticity of the signal source. This includes angle-of-arrival (AOA) discrimination techniques [8], [9], [301], [302] or Doppler frequency difference of arrival (FDOA) [303] discrimination using multiple antennas. Other geometric techniques advocate the use of single antenna, and discriminate spoofed and authentic signals either with a known perturbation profile [304] or random motion profile [305], or using multiple feeds from a single antenna [306]. The assumptions made by these techniques are: (1) the spoofing signals generally arrive from below or near the horizon [306], (2) the observations from spoofing signals is not aligned with the actual geometry between the satellites and the victim receiver [8], [301], and (3) there are strong correlation of signal characteristics of different satellites from the spoofing signals [9], [302], [304], [305]. However, it is not costly for a sophisticated spoofer to co-locate dedicated spoofing sources at each of the TRNS pseudolites, thereby defeating all the assumptions made by these techniques. In addition, the need for hardware modification or additional hardware might not be suitable for applications that either use an existing hardware for mass-market adoption, or have SWaP-C constraints.

Drift monitoring techniques, on the other hand, look for unusual changes in the output of the receiver, such as position or clock fix, by coupling with external sensors. These include the use of an oscillator to check for inconsistency in the clock bias or clock drift [307], or the use of visual/inertial/radar odometry to place constraints on the reasonable error growth of a position fix [111], [308]. The applicability of these techniques is limited by the SWaP-C constraints of the applications, and the authentication performance is limited by the accuracy of these sensors.

Signal processing techniques look for sudden deviations in the received signal characteristics to indicate an onset of a spoofing attack. These techniques detect changes in the received carrier amplitude or the RF front-end's AGC set-point, or a distortion in the complex correlation function [309]. Signal processing techniques can be implemented in software, unlike the previous categories of techniques discussed which require additional hardware. These techniques are effective for GNSS which has signal strength below the noise floor and narrow signal dynamic range. However, this is not applicable to TRNS, which generally has high SNR and a wide signal dynamic range for quick acquisition in both dense-urban and indoor environments. A potential spoofer will have a wide margin to change the total received power and create a distortion-free correlation function using the spoofing signal, and these indicators will not be picked up by the PD detector proposed by [309].

Cryptographic Defenses

The main objective of cryptographic spoofing defenses is to ensure information security. Cryptographic techniques include encryption, which enforces the secrecy of data from unauthorized access, and authentication which verifies the origin of the data. They provide three features: (1) authentication, by verifying the origin of information, (2) confidentiality, by protecting the information from disclosure to non-authorized parties, and (3) integrity, by detecting any unauthorized information modification. These features increase the resilience of the signal against spoofing.

Several GNSS cryptographic spoofing defenses have been proposed and/or implemented in both civil and limited-access GNSS signals. These spoofing defenses add cryptographic features in small segments or in entire portion to either the fast-rate spreading code or the low-rate navigation data. These cryptographic techniques can be classified into the following groups: (1) navigation message encryption (NME), which encrypts the whole navigation data message before being modulated onto the spreading code, (2) spreading code encryption (SCE), which encrypts the whole spreading code sequence, (3) navigation message authentication (NMA), which adds unpredictable digital signature into the navigation data using asymmetric cryptography, and (4) spreading code authentication (SCA), which inserts unpredictable watermark sequences within the open spreading code.

The straightforward, blanket encryption of a navigation signal may be attractive as a means both to deny service to stow-aways and to authenticate the signal to subscribers. However, there are sigificant caveats in both applications. The first regards the use of symmetric cryptography.

One may apply symmetric encryption to the entire navigation message (NME) and/or the spreading code (SCE, *a la* the GPS P(Y) code). The premise is that a spoofer who does not know the symmetric key cannot produce a valid spoofing signal, or equivalently that a receiver can be confident in a signal that appears in the output of a correlator tuned to the secret spreading sequence (with similar reasoning for NME). However, a symmetric approach to authentication is extremely fragile, because a leaked symmetric key can be used for spoofing. For this reason, military deployment of SCE involves tamper-resistant hardware and costly, elaborate procedures for secure distribution and management of the secret symmetric keys. This approach is untenable for civil or commercial radio-navigation.

NMA and SCA, in contrast, avoid the fragility of symmetric key management by adopting asymmetric cryptography, using either delayed release approach or public-private key pair. In SCA, short segments of unpredictable spreading code sequences (termed as "watermarks") are interleaved with long segments of predictable spreading codes in fixed or random positions [290]. The receiver uses the predictable sequences to track the broadcast signal, and stores the unpredictable segments in the buffer while waiting for the information about the watermarks. Once this information arrives, the receiver can synthesize the unknown spreading sequence with the correct watermarks embedded in the right position, and correlates this code segment with the relevant segment from its recorded signal to verify signal authenticity. This technique requires modifications to the GNSS signal generation. Hence, it will be difficult or impossible to be implemented on existing GNSS which requires backward compatibility. However, TRNS, which comes with a green-field waveform, can consider the implementation of SCA into its waveform design.

A growing literature advocates the use of NMA for civil GNSS signal authentication, with proposed implementations for GPS [290], [310], [311], Galileo [312], [313], QZSS [314] and SBAS [315]–[317]. NMA is already implemented in the Galileo Open Service, which will start its Open Service Navigation Message Authentication (OSNMA) signal-in-space transmission in the first quarter of 2020 and have full service available in 2021 [318]. This technique uses either an asymmetric private-key/public-key approach such as the *elliptic curve digital signature algorithm* (ECDSA) [311], or a delayed symmetric key release approach such as *timed efficient stream loss-tolerant authentication* (TESLA) [310]. Unlike SCA, this technique can be implemented into existing GNSS signal, provided that there are available unused bits in the navigation message to store the digital signature. However, the leftover bits in the navigation message are usually limited. A trade-off has to be made between the cryptographic strength of the NMA scheme, which is determined by the size of the key and the digital signature, and the authentication latency, which is determined by the frequency of digital signature validation. TRNS has more flexibility in incorporating NMA into their waveform design, and can offer low *time-to-first-authenticated-fix* (TTFAF) while maintaining strong cryptographic security.

In contrast to GNSS, TRNS comes with a clean-slate waveform design, and is not constrained by the need of backward compatibility. This offers TRNS providers flexibility in their application of the latest cryptographic defense techniques—many of which were originally proposed for GNSS. The next section proposes one implementation of NME and NMA for a TRNS.

TRNS SECURITY DESIGN

As discussed in Sec. , this paper addresses TRNS vulnerabilities to two types of adversaries: spoofers and unauthorized users.

With regard to a spoofing adversary, a subscriber is said to have assured PNT from its TRNS network if either (1) the subscriber's pseudorange measurements are not substantially affected by the spoofing signal, or, (2) the spoofing attack is flagged as such. The security proposal outlined in this section aspires not only to aid a protected TRNS subscriber in meeting one of these conditions, but also to enable provision of tiered subscriber segments *a la* selective availability.

Broadly, there are two types of spoofing attacks: one in which the adversary forges a valid signal (navigation message and spreading code) that has not been previously generated by an authentic transmitter, and the other in which the adversary simply re-broadcast a signal previously broadcasted by an authentic transmitter. Authentication mechanisms are designed to thwart the first kind of attack via SCA and/or NMA. Crucially, neither SCA nor NMA can defend against the second type of spoofing attack [319]. This section focuses on design of an NMA scheme for TRNS that also provides some benefits of SCA.

At this point, the reader might point out that the GPS P(Y) code in fact uses SCE to prevent the first kind of spoofing attack. This is true. In the special case where the subscriber (e.g., a SAASM receiver) has *a priori* access to the spreading code (i.e., the plaintext) and the symmetric key, but the spoofer does not, SCE can provide authentication. However, this is untenable in the case of TRNS because a general TRNS subscriber cannot be trusted as benign. As such, this section does not propose SCE/NME for anti-spoofing.

With regard to unauthorized usage, it is important to concede that it is not possible to prevent the usage of TRNS signals as a signal-of-opportunity, whereby unauthorized users estimate the position and clock states of the authentic transmitters by means other than the navigation message. With that said, unauthorized use as a signal-of-opportunity is much more involved than the case where the navigation message is plainly available. Accordingly, this section proposed the use of NME to limit terrestrial PNT service to authorized users.

Selective Navigation Message Encryption

This sub-section considers an adversary that is not a valid subscriber of the TRNS service, but nevertheless wishes to exploit the service. Data confidentiality provided by symmetry key encryption is sufficient to defeat this type of adversary. Beyond the traditional GNSS NME scheme, which envisions a single segment of authorized users, this paper proposes a scheme that can be customized for multiple tiers of subscribers. For example, the highest tier subscribers may decrypt the full navigation message and access the most accurate transmitter position and clock states, whereas lower tier subscribers may only decrypt a few most significant bits of such information.

Fig. 58 provides an overview of the proposed encryption scheme. This scheme is based on the counter mode (CTR) of the block cipher operation, which is a standard method to generate a pseudo-random keystream from a short shared secret. The use of this method requires two components: a shared secret key and a unique initial value (IV). The rest of this sub-section describes a method that involves tiered distribution of secret keys and the provision of a unique IV.

Each tier of subscription grants access to some subset of the pre-shared secrets (PSS) and corresponding encryption bit masks (EBM) used by the system. Subscribers download these secrets in batches via a secure secondary channel and store them in their receivers' non-volatile memory. At each encryption period (e.g. day of the month), a unique value of $\mathrm{PSS} = (\mathrm{PSS1}, \mathrm{PSS2})$, is retrieved from storage. PSS1 takes the role of a symmetric key. PSS2 is concatenated with the pseudolite ID (TxID) and time of day (ToD), e.g. GPS or UTC time, to form a unique IV, from which the block cipher E generates the key stream (KS).

$$\mathrm{KS} = \mathrm{E}(\mathrm{PSS1}, (\mathrm{TxID} \parallel \mathrm{ToD} \parallel \mathrm{PSS2}))$$

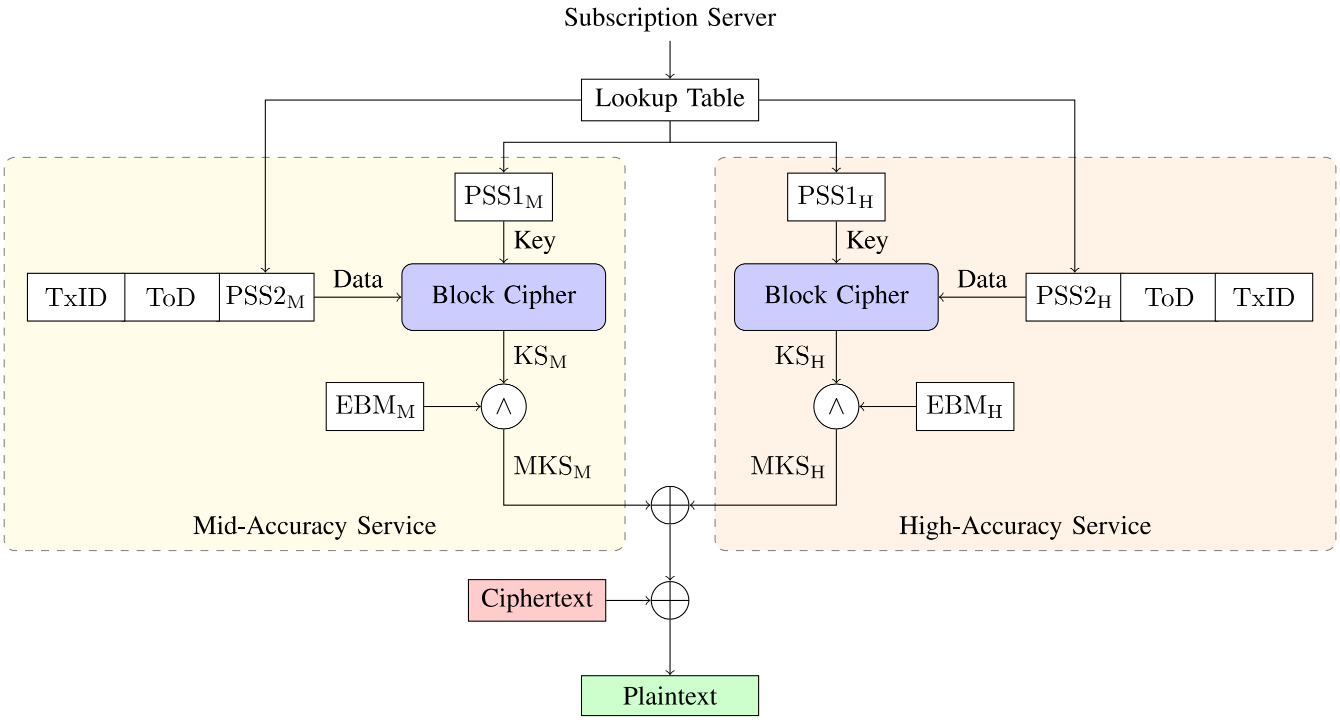


Fig. 58: Proposed TRNS NME scheme from the perspective of a high-accuracy service receiver. Note that in the high-accuracy receiver, both mid- and high-accuracy key streams are computed in order to decrypt the entire message.

Note that while $\mathrm{PSS2}$ is a not publicly-known in this scenario, this is not necessary a requirement. The most important consideration here is that the same key-IV pair must never be re-used. For example, if ToD were chosen to be "seconds since midnight", then the same key-IV pair would repeat every 24 hours until a new PSS pair is retrieved. Accordingly, it must be ensured that ToD does not repeat faster than the key-swapping period.

A suitable block cipher to be used is AES-128 (Advanced Encryption Scheme, using block size of 128 bits), which offers an equivalent symmetric-key strength of 128 bits. This symmetric-key strength of 128 bits is recommended by U.S. National Institute of Standards and Technology (NIST) guidelines for cryptographic security beyond 2030. The IV to the block cipher has to match its block size. The key stream KS is combined with the EBM to form the masked key stream MKS. The EBM enables tiered usage of NME.

$$\mathrm{MKS} = \mathrm{KS} \wedge \mathrm{EBM}$$

The masked key stream is then XOR with the ciphertext C to reveal the plaintext P.

$$P = \mathrm{MKS} \oplus C$$

Each masked key stream applies to a different set of message bits. A high-accuracy subscriber, for instance, will be provided with the full suite of pre-shared secrets, enabling it to reconstruct each of the masked key streams and thus to decrypt the entire message. A mid-accuracy subscriber will only be able to reconstruct the masked key streams protecting the most significant bits of each of the navigation parameters encoded in the message. Access is further limited to the period of a subscription by limiting which days' pre-shared secrets are provided to which receivers. (Naturally, such a scheme cannot prevent subscribers from sharing secrets with non-subscribers, beyond what protection is possible through e.g. software obfuscation. Such insider attacks may call for remedies of a legal, rather than technical, nature.)

It must be noted that the stream cipher structure (i.e. XOR-based encryption) is not suitable to ensure the authenticity of data. That is, it does not prove that an incoming navigation message to a TRNS receiver originates from an authentic TRNS pseudolite, because it is *malleable*: an attacker can take a valid encrypted packet $(\mathrm{E}(M) \,\|\, \mathrm{CRC}(\mathrm{E}(M)))$ and XOR it with $(X \,\|\, \mathrm{CRC}(X))$ for any bit string X, producing a new valid encrypted packet which decrypts to $M \oplus X$.

More generically, the symmetric structure of this cipher is not suitable to prevent real-time forgery of encrypted signals by a spoofer who might also, secretly, be a subscriber with access to the symmetric keys. This type of spoofing attack will be mitigated with NMA in the next subsection.

Combined Data and Signal Authentication

This sub-section presents an NMA method based on the TESLA protocol [320] that additionally provides limited signal authentication against a *half duplex* re-broadcast-type spoofing attack.

Notionally, NMA requires asymmetric cryptography to generate and verify digital signatures, and thereby to perform data origin authentication. Naïve alternatives using symmetric cryptography suffer from the validator-can-spoof problem: anyone who can validate such a "signature" can also forge one. However, asymmetric cryptography is substantially more costly in both computation and communication overhead than symmetric cryptography when compared at an equivalent level of security (i.e. $\log_2$ of the number of operations in the best-known attack). For instance, ECDSA produces signatures whose length in bits is roughly four times the equivalent security level.

The TESLA protocol introduced a key innovation that bypassed this dilemma and enabled the use of lightweight symmetric cryptography for NMA. TESLA involves a form of asymmetry based on the delayed release of symmetric keys. This protocol has emerged as a strong contender among broadcast authentication proposals for GNSS [318]. The communication overhead of TESLA in bits per authentication epoch is roughly twice the equivalent security level.

1) Data Authentication: This sub-section considers an adversary attempting to spoof the subscribers of a TRNS. Importantly, such an adversary may be a highest-tier subscriber, and hence have access to all symmetric encryption keys. As such, all navigation message and spreading code bits, encrypted or otherwise, are known to the adversary.

The authentication design proposed in this paper relies on the vanilla TESLA protocol for data-level authentication. Fig. 59 describes the key chain and message authentication code generation per the TESLA protocol. The TESLA protocol progresses in a reverse direction along a one-way key chain generation, starting with the root key K_n obtained from the control segment (i.e. subscription server) and ending with the public key K_0 to be dispersed to all subscribers via secondary channels for bootstrapping. Each downstream key K_{i-1} is derived from the upstream key K_i using a one-way hash function H_{A1}, and subsequently disclosed in the ith broadcast message.

$$K_{i-1} = \mathrm{H}_{\mathrm{A1}}(K_i)$$

The specific key corresponding to each epoch K_i is then passed into a different hash function H_{A2} to generate the input key K_i' for a hash-based message authentication code (HMAC) function. The authentication code MAC_i is computed from the concatenation M_i of all messages in the ith epoch. The reason for having a second hash function before HMAC is subtle; interested readers should refer to [320, Sec. 3.4].

Note that authentication is orthogonal to encryption: the scheme works equally well in deployments with no encryption at all; in this case, the input M_i to the HMAC is the plaintext. In either case, the input to the HMAC is whichever bit string is known to all receivers once forward error correction has been removed.

$$\begin{aligned}\mathrm{MAC}_i &= \mathrm{trunc}(\text{HMAC}(K_i', M_i)) \\ &= \mathrm{trunc}(\text{HMAC}(\mathrm{H}_{\mathrm{A2}}(K_i), M_i))\end{aligned}$$

Fig. 60 shows the process of authentication in an NMA-enabled receiver, which operates in two phases. During the warm-start phase, the receiver obtains the first packet $P_i = [M_i, \mathrm{MAC}_i, K_{i-1}]$ from the broadcast. As MAC_i cannot be verified instantaneously without the corresponding K_i, the packet is stored in the receiver's memory until the arrival of K_i. However, the first received key K_{i-1} can still be validated. This is done by applying K_{i-1} through the prescribed chain of one-way hash functions, and by matching the terminal key from the chain with the public key K_0 obtained from the server. At the

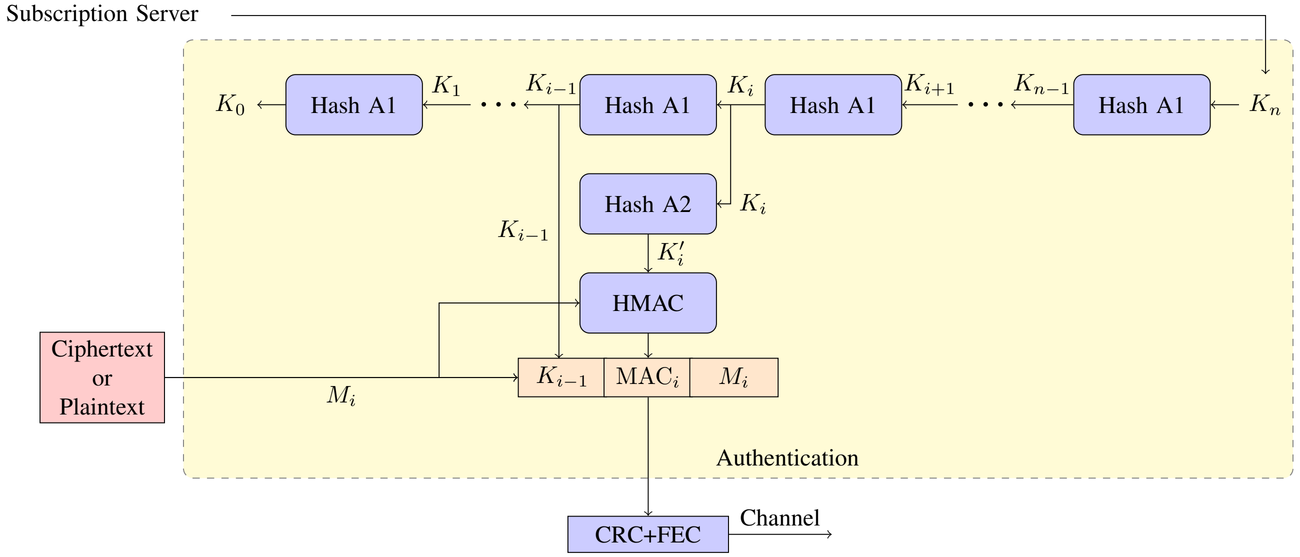


Fig. 59: Authentication processes at the TRNS pseudolite, which include one-way key chain generation, MAC generation, and broadcast packet formation.

next epoch, K_i arrives and the receiver can transit into the steady-state phase, where it can perform both key and MAC validation. The MAC generated from passing M_i and K_i into the HMAC function is compared with the broadcasted MAC_i. The broadcasted MAC is deemed to be authentic if it matches the locally generated MAC. In addition, the broadcasted K_i goes through a shorter one-way key hash chain to obtain an output key. K_i is considered authentic if the output key matches with the previously-validated key K_{i-1}. An authentication event (AE) occurs when both components of the MAC-key pair are deemed to be valid by the NMA scheme.

TESLA's security draws from the cryptographic strength of the keyed-hash MAC (HMAC) construction and the one-way key hash chain, both of which depend on the strength of the underlying hash function, the length of the key, and the size of the MAC tag. To meet the equivalent key symmetric-key strength of 128 bits for cryptographic security beyond 2030 [321], SHA-256 is recommended as the hash function to be used, and the key size is required to be at least 128 bits. NIST also recommends the size of the MAC tag to be at least 32 bits, to minimize the occurrence of MAC tag forgery [322]. Hence, the authentication overhead is at least 160 bits per AE. In addition, [323] mentions that the collision resistance of the hash chain decreases linearly with its length. The length of the key generation chain should therefore either be appropriately limited, or be circumvented by increasing the key length at the cost of a higher authentication overhead.

2) Signal Authentication: The proposed NMA scheme—that is, the TESLA-based MAC-and-key mechanism described thus far—only serves to verify the origin of the data. Hence, the data fields relevant to the PNT calculation, such as the pseudolites' positions and timing offsets, are authenticated. However, NMA does not prevent attacks wherein the spoofer re-broadcasts an authentic TRNS signal.

One type of re-broadcast attack, known as *security code estimation and replay* (SCER), requires the spoofer to measure and estimate the current broadcast symbol, and then generate and transmit a forged signal with the desired delay. There is known to be no absolute defense against SCER spoofing in a uni-directional radionavigation system. However, a mitigating factor is that SCER attacks are somewhat challenging to execute because of the need for the spoofer to *full duplex*.

In a lower-cost *half-duplex* attack, the spoofer transmits either intermittently or in an open-loop fashion, generating the spoofing waveform using only information collected while not transmitting. Removing the requirement for nanosecond-latency real-time bit estimation removes substantial engineering challenges in mounting this attack. However, such a spoofer faces a dilemma when dealing with the unpredictable segments of the broadcast message: it can continue with its open-loop transmission and make random guesses about the unpredictable bits, thereby running a high risk of triggering an alarm from NMA; or it can modulate its transmission amplitude to leave an open window for the true signal to pass through. This is significant, because this modulation is potentially detectable by a clever receiver, which will raise an alarm. To avoid

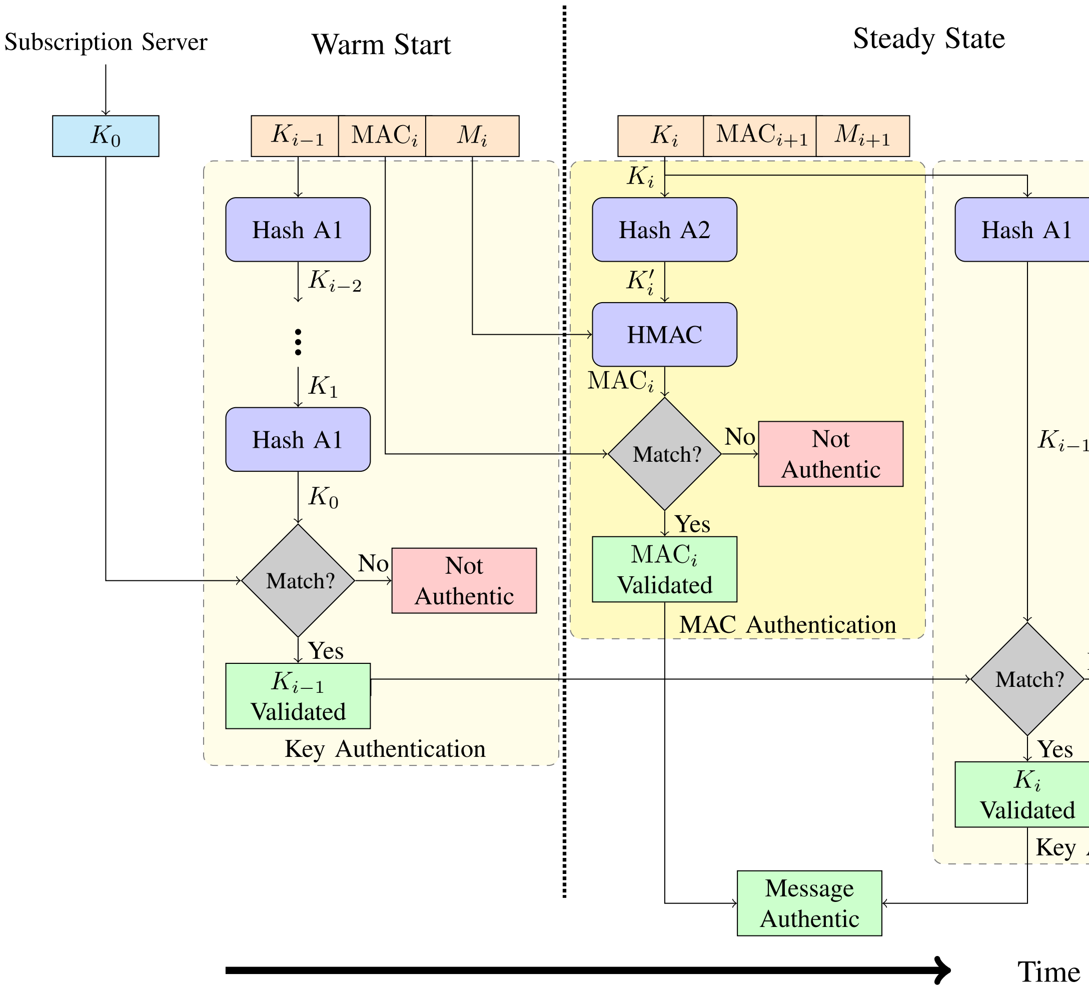


Fig. 60: Authentication processes within the TRNS receiver, which includes key validation during bootstrapping, and both key and MAC validation during steady-state phase.

detection, the spoofer must limit the rate of change of amplitude and phase variables that it is introducing in between these open windows. Thus, while the half-duplex spoofer would like to introduce controlled delays (and hence position offsets) into the victim's delay-locked loop, each open window forces it to smoothly transition these delay variables back to zero. This limits the size of possible undetectable offsets. The rest of this section extends the TESLA-based NMA scheme to maximize the number of open windows that the half-duplex adversary must deal with, thus providing limited signal authentication.

Since the adversary considered here is potentially a highest-tiered subscriber, everything but MAC_i and K_i are already known to the adversary. If the unknown bits are packaged together at the end of an epoch, as is conventional in data networks, the half-duplex adversary is very effective: the only open windows the receiver can expect are those covering the (infrequent) MAC and key packets; otherwise, the attacker is free to transmit faulty timings provided that they send valid data.

The key idea introduced in this paper is to leaven the unpredictable MAC_i bits into the navigation message packets such that the time duration between any two open windows is as short as possible. This process is shown in Figs. 61 and 62. The watermark bits are placed at predictable positions in the navigation message stream so that the receiver can still access the relevant fields for PNT calculation. The exact locations of these watermark bits are non-critical, as they will be spread throughout the transmitted waveform by the interleaver. However, the watermarks should be spaced out by at least the constraint length of the convolutional code in order to maximize the number of affected code bits.

The requirement to introduce controlled delays and transition them to zero before the next open window, together with maximal frequency of open windows, limits the adversary's ability to spoof large position incursions.

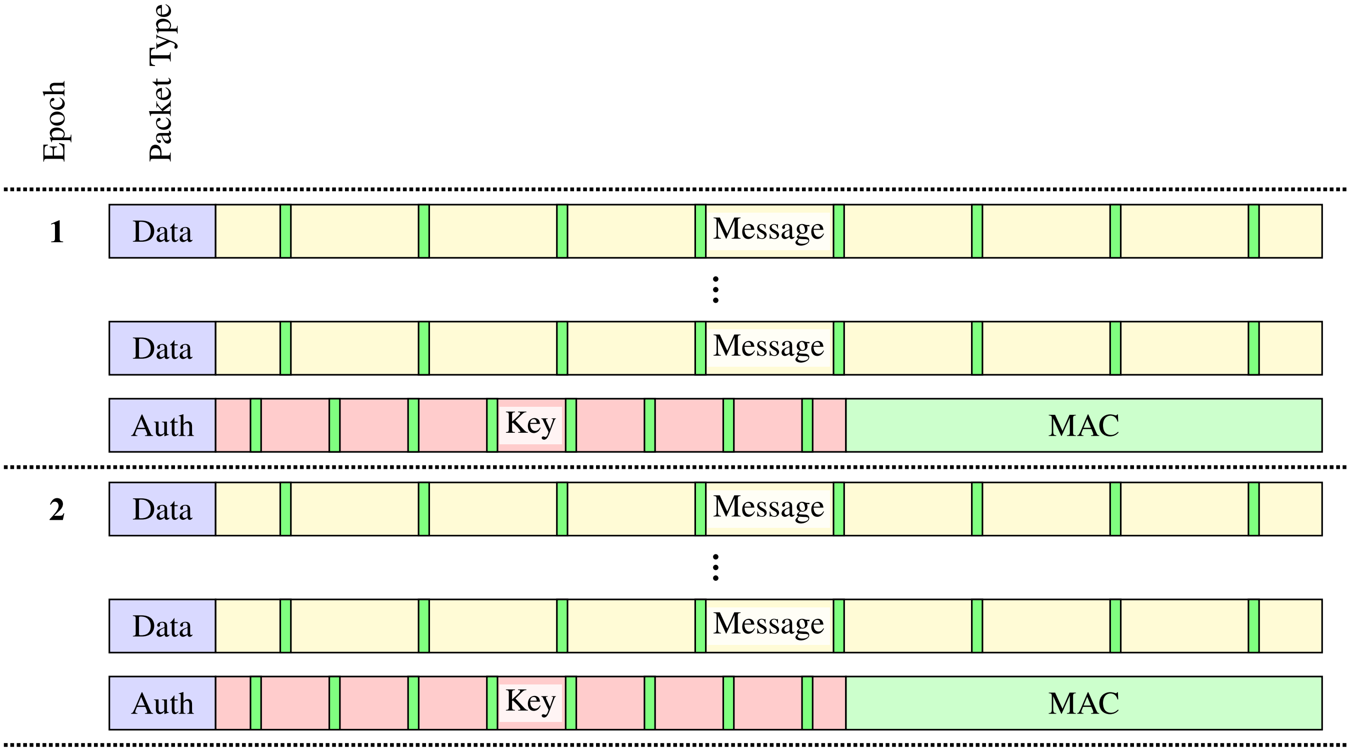


Fig. 61: NMA for a TRNS navigation stream. Error detection, forward error correction, and encryption are not shown. Authentication packets terminate each authentication epoch, and contain the TESLA key for the previous epoch (red), together with a message authentication code (green) computed from the preceding packets in the current epoch. "Watermark" MAC bits (green stripes) are inserted at fixed positions to frustrate half-duplex spoofing attacks. Note that while authentication can proceed without all MAC bits, it cannot proceed without all key bits. For this reason, HMAC output bits (green) may be truncated to trade reduced security for reduced authentication overhead, but key bits (red) cannot be truncated.

The duration between open windows is minimized if all of the MAC_i and K_i bits are uniformly distributed across the navigation message. However, note that while authentication can proceed without all MAC bits, it cannot proceed without all key bits. Leavening key bits in the navigation message would increase the likelihood of failed authentication due to a packet error containing a key bit. Accordingly, the proposed protocol leavens only the HMAC output bits to trade reduced security for reduced authentication overhead. Another consequence of a packet error would be incomplete recovery of the navigation message bits, which would also preclude authentication. Fortunately, a receiver may re-construct lost navigation message bits before computing the MAC if these bits are known to be repeated verbatim on a set schedule, and at least one was successfully decoded.

Although this elaboration of the proposed NMA scheme provides a degree of signal authentication, it is not foolproof against all types of spoofing attacks. It aims for the lesser goal of defeating half-duplex attacks and forcing attackers to turn to more costly alternatives like SCER. Unfortunately, SCA fares no better against SCER attacks than the proposed MAC-leavened NMA scheme. As such, use of exotic signal-level authentication schemes provide no additional advantage.

CONCLUSION

This paper outlines the unique vulnerabilities of a generic T-PNT system due to its terrestrial infrastructure, high signal strength with wide dynamic range for deep-urban and indoor coverage, and a potential reliance on GNSS for network synchronization. Despite these challenges, this paper draws upon the flexibility offered by a clean-slate TRNS waveform to propose cryptographic schemes that offers more than protection against both low-cost spoofers and unauthorized users. An NME scheme is introduced, which not only limits T-PNT service to authorized users, but also can be customized for multiple subscriber tiers by implementing selective decryption. In addition, a novel TESLA-based NMA scheme that leavens

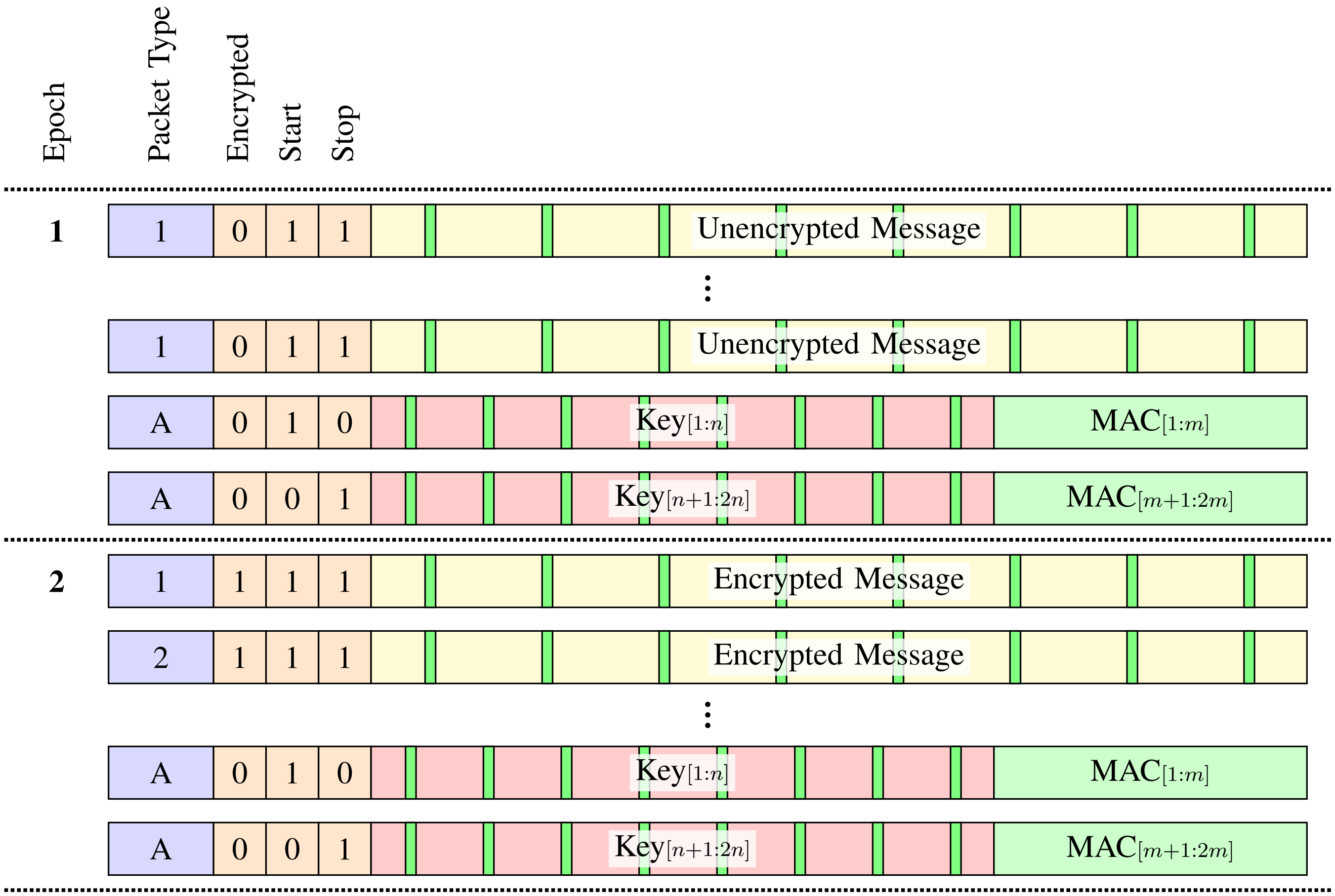

Epoch
Packet Type
Encrypted
Start
Stop
1
1 0 1 1 Unencrypted Message
⋮
1 0 1 1 Unencrypted Message
A 0 1 0 $\text{Key}_{[1:n]}$ $\text{MAC}_{[1:m]}$
A 0 0 1 $\text{Key}_{[n+1:2n]}$ $\text{MAC}_{[m+1:2m]}$
2
1 1 1 1 Encrypted Message
2 1 1 1 Encrypted Message
⋮
A 0 1 0 $\text{Key}_{[1:n]}$ $\text{MAC}_{[1:m]}$
A 0 0 1 $\text{Key}_{[n+1:2n]}$ $\text{MAC}_{[m+1:2m]}$

Fig. 62: NMA for a short-packet TRNS navigation stream. Packets may be fragmented (e.g. Start, Stop) as required. The schedule of packet types, analogous to almanac pages in GPS, determines the time-to-first-fix. To improve authentication robustness, a receiver may re-construct lost packets before computing the MAC if these packets are known to be repeated verbatim on a set schedule, and at least one was successfully decoded. Note that a spoofer attempting a downgrade attack (spoofing a zero bit in the "Encrypted" field) will trigger authentication alarms.

unpredictable MAC bits into the navigation message packets is presented, which provides both data authentication and a certain degree of signal authentication against half-duplex spoofing attacks. While the proposed schemes are not fool-proof against all types of spoofing attacks and unauthorized use (e.g. as signals of opportunity), they offer robust and accurate PNT service only to TRNS subscribers with selective availability and enhanced data security.

5c Autonomous Signal-Situational-Awareness in a Terrestrial Radionavigation System

ABSTRACT

This paper aims to augment terrestrial radionavigation systems (TRNS) with autonomous signal-situational-awareness capability, allowing TRNS operators to detect spoofing and meaconing attacks within their systems. Such a capability is necessary to address a vulnerability to certain replay attacks that remains even when TRNS signals are secured by navigation message encryption and authentication. Two signal authentication techniques are developed to detect a weak spoofing signal in the presence of static and dynamic multipath. Both are shown to be effective in simulations of the varied operating environments that TRNS will encounter. With autonomous signal-situational-awareness, TRNS gain a defensive capability that GNSS cannot easily match: a comprehensive defense against most man-in-the-middle attacks on position, navigation, and timing services.

INTRODUCTION

Global navigation satellite systems (GNSS) struggle to provide positioning, navigation, and timing (PNT) coverage in deep-urban and indoor environments. Current and upcoming terrestrial radionavigation systems (TRNS) like Locata [283] and NextNav [324] seek to extend PNT coverage by placing powerful ranging beacons throughout an urban environment. GNSS and TRNS face shared challenges in signal authentication and anti-spoofing that arise from fundamental properties of radio systems. Thus, the extensive scholarship on GNSS vulnerabilities [2], [7] largely applies to TRNS.

As recently outlined in [325], however, TRNS have unique security challenges: (1) the dynamic range of TRNS signal power is vastly wider than that of GNSS, allowing would-be spoofers access to high signal-to-noise ratio (SNR) signals and complicating spoofing mitigation based on simultaneous demodulation of spoofed and authentic waveforms [326]; (2) the angular distributions of spoofed, authentic, and multipath signals significantly overlap, rendering angle-of-arrival techniques based on multi-element antennas [9], [327] less effective; and (3) TRNS transmitters are physically accessible.

Nevertheless, TRNS also have inherent security advantages. Chief among these is that TRNS transmitters also function as receivers and can thus (1) accurately characterize the surrounding signal landscape's nominal statistics and thereafter (2) search for anomalies that reveal the presence of interfering signals. Current development of commercial TRNS clean-slate designs offers an opportunity to exploit this advantage of TRNS for enhanced security.

Related Work in Signal-Processing-Based Spoofing Detection Techniques: Several spoofing detection techniques proposed in the GNSS literature apply advanced signal processing algorithms to extract signal characteristics for source verification. Unlike other non-cryptographic spoofing defenses, these techniques can be readily implemented on existing GNSS receivers via a firmware upgrade. They can be divided into two classes, one that detects the inception of a spoofing attack, and another that performs a brute-force search for all signals in the landscape for post-inception detection.

Included in the first class are techniques that look for a sudden deviation in the received signal characteristics (carrier amplitude, beat carrier phase, code phase, carrier-to-noise density ratio, or received power) to detect the onset of a spoofing attack [328]–[330]. Also included are techniques based on Signal Quality Monitoring (SQM) that identify anomalous distortion in the complex correlation function [309], [331]. Multiple signal metrics can be derived by combining observations of both the received power and the correlation function distortion [5].

The second class of techniques performs a brute-force acquisition search for the presence of known signals using Complex Ambiguity Function (CAF) monitoring [332]. This approach avoids the problem of missed detection due to the transient nature of initial spoofing drag-off.

These techniques generally work for GNSS, as it has signal strength below the noise floor and a narrow dynamic range of signal power. In contrast, TRNS generally have high SNR—for quick acquisition in both dense-urban and indoor environments—and a wide signal power dynamic range. Analogous to variations in the received signal strength from low-elevation GNSS satellites in an urban environment, without detailed knowledge of its deep-fading channel model, a mobile receiver cannot straightforwardly predict the received signal strength of the authentic signal emanating from a particular TRNS beacon. Relatedly, when masquerading its signal as common multipath, a potential spoofer will have a wide margin to adjust its power in its attempt to overtake a victim receiver's tracking loops. In general, because TRNS operate in a quantitatively distinct regime of parameter space compared to GNSS (see [333, Subsec. 5.2.3]) it is challenging for mobile TRNS receivers to directly apply existing GNSS-targeted techniques for spoofing detection.

On the other hand, TRNS infrastructural monitors can fully exploit signal processing techniques for spoofing detection. Assuming that each has multiple correlators and secure clock synchronization [334], such monitors can narrowly characterize all signals in their nominal operating environments, after which signal anomalies in the surveilled landscape become apparent. This paper capitalizes on this feature of TRNS to propose two signal authentication techniques customized for TRNS monitors. It will be shown that a spoofing signal—even one with SNR below that of the authentic signal—can be detected despite the presence of static and dynamic multipath.

Related Work in TRNS Security: The present work complements the cryptographic security proposal presented in [325]. Briefly, [325] proposes a multi-tiered navigation message encryption (NME) + message authentication code (MAC)-based navigation message authentication (NMA) scheme. One can think of [325] as offering a basic level of navigation security via cryptographic methods. No TRNS should be fielded without such basic measures.

However, the techniques proposed in [325] are not sufficient to secure TRNS because the exposed spreading codes of a high-SNR TRNS signals makes them vulnerable to replication in a security code estimation and replay (SCER) [335] or meaconing attack. More generally, NME+NMA cannot fully protect TRNS against low-latency replay attacks. Even exotic signal-level security techniques like spreading code authentication (SCA) [291] or deterministic code-phase dithering [336] can be rendered ineffective by a spoofer's ability to access high-power authentic signals in a TRNS network.

Contributions: To address the gap in TRNS defenses against low-latency signal replay attacks, this paper proposes an autonomous signal-situational-awareness (SSA) overlay capability within a TRNS network. SSA is intended to augment basic TRNS cryptographic security. While some spoofers will remain undetectable, SSA gives TRNS operators a significantly improved chance of catching threats and alerting users without resorting to costly full-duplex techniques (those requiring bi-directional communication with users). Note that SSA is not possible for current GNSS space vehicles in medium Earth orbit, which can neither receive each other's signals nor detect low-power ground-based spoofers. This work seeks to place TRNS SSA on a solid theoretical and practical footing. First, signal authentication techniques for SSA are developed based on the prior work in [5] and [6]. Second, simulations with a theoretical model of multipath and spoofing signals are used to quantify the effectiveness of autonomous SSA under some of the myriad operating conditions encountered by generic TRNS.

SIGNAL AUTHENTICATION

This paper adopts a Bayesian binary hypothesis testing framework for distinguishing between the null hypothesis H_0 for the spoof-free case, and the alternate hypothesis H_1 for the spoofing case. The TRNS pre-correlation and post-correlation signal model for single-spoofer scenarios in a multipath environment, together with the probability distributions of signal components required to characterize the detection statistic, have been outlined in [333, Sec. 5.2]. This section develops the measurement models and formulates the detection statistics for signal authentication.

Consider a TRNS monitor receiving signals from a transmitting TRNS beacon at a standoff distance d, with its post-correlation output described by [333, Eq. 5.6]. There will typically be a significant number N_M of multipath components evident in the post-correlation function $\xi_k(\tau)$ ending at time $t_k = kT$, where T is the accumulation interval, but due to the quasi-static nature of the urban environment, the variation in $\xi_k(\tau)$ will be small over $(t_{k-1}, t_k]$, $\forall k$. These variations are caused by (1) thermal noise, (2) time-varying receiver non-idealities, and (3) urban environment movement. The first two factors are modeled by additive white Gaussian noise $r_N(t)$, whose contribution to $\xi_k(\tau)$ is detailed in [333, Sec. 5.2], while the third factor is modeled as a dynamic multipath component. Revisiting [333, Eq. 5.6], each multipath component can be further segregated into static $\xi_{M_s(k,i)}$ and dynamic $\xi_{M_d(k,i)}$ components:

$$\sum_{i=1}^{N_M} \xi_{M(k,i)}(\tau) = \sum_{i=1}^{N_M} \left[\xi_{M_s(k,i)}(\tau) + \xi_{M_d(k,i)}(\tau)\right] \tag{47}$$

Let l be the number of signal taps sampling $\xi_k(\tau)$ across the lag window of interest, $\tau_w > 0$, with the centermost tap being aligned with $\tau = 0$, the location of the receiver's estimate of the authentic signal's correlation function peak, and the remaining taps being evenly spaced across τ_w. The uniform tap spacing is

$$\Delta\delta = \frac{\tau_w}{l-1}$$

and the vector of tap locations is

$$\boldsymbol{\delta} = \left[-\frac{\tau_w}{2}, -\frac{\tau_w}{2} + \Delta\delta, \cdots, \frac{\tau_w}{2} - \Delta\delta, \frac{\tau_w}{2}\right]^{\mathsf{T}} \in \mathbb{R}^l$$

with $\delta_i = -\frac{\tau_w}{2} + (i-1)\Delta\delta$ representing the ith tap location, $i = 1, \cdots, l$.

Being complex-valued, the post-correlation function can be viewed as having in-phase quadrature components: $\xi_k(\tau) = I_k(\tau) + jQ_k(\tau)$. Samples of $\xi_k(\tau)$ at the locations in $\boldsymbol{\delta}$ can be stacked into a single correlation measurement vector:

$$\boldsymbol{q}_k = \left[I_k\left(\delta_1\right), Q_k\left(\delta_1\right), \ldots, I_k(\delta_l), Q_k\left(\delta_l\right)\right]^{\mathsf{T}} \in \mathbb{R}^{2l} \tag{48}$$

A hypothesis test for signal anomaly detection can be formulated in terms of the change in the distribution of $\boldsymbol{q}_k$ due to an additional signal component or components. Let $p_0(\boldsymbol{q}_k)$ and $p_1(\boldsymbol{q}_k)$ be the distribution of $\boldsymbol{q}_k$ under the null (H_0, no spoofing present) and alternate (H_1, spoofing present) hypotheses respectively.

The measurement $\boldsymbol{q}_k$ can be further dissected into its individual components:

$$H_0 : \boldsymbol{q}_k = \bar{\boldsymbol{q}} + \boldsymbol{w}_k \tag{49a}$$

$$H_1 : \boldsymbol{q}_k = \bar{\boldsymbol{q}} + \boldsymbol{\mu}_k + \boldsymbol{w}_k \tag{49b}$$

Here, $\bar{\boldsymbol{q}}$ is the mean of $\boldsymbol{q}_k$ under H_0, and $\boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, P)$ is the measurement noise under both hypotheses, with $P = \mathbb{E}[(\boldsymbol{q}_k - \bar{\boldsymbol{q}})(\boldsymbol{q}_k - \bar{\boldsymbol{q}})^\mathsf{T}]$ being its covariance. P includes contributions due to dynamic multipath, thermal noise, and receiver non-idealities. Under H_1, there additionally enters a correlation distortion vector $\boldsymbol{\mu}_k$, which is a function of the false signal's code and carrier offsets $\Delta\tau_{Dk}$ and $\Delta\theta_{Dk}$ defined relative to the authentic signal's code and carrier phase. $\boldsymbol{\mu}_k$ will be further detailed in a later section. The amplitude of the false signal is given by $\epsilon_{Dk} > 0$. In the case of a successful detection, the false signal's amplitude and code and carrier offsets may also be estimated. For a false detection, estimates of these parameters are specious; typically, they match those of a strong dynamic multipath component.

The hypotheses H_0 and H_1 can be expressed in terms of probability distributions as follows, where $p_0(\boldsymbol{q}_k)$ is modeled as a Gaussian distribution with a mean of $\bar{\boldsymbol{q}}$ and covariance P, and $p_1(\boldsymbol{q}_k)$ has the same distribution but with an unknown deviation to the mean:

$$H_0 : \boldsymbol{q}_k \sim \mathcal{N}(\bar{\boldsymbol{q}}, P) \tag{50a}$$

$$H_1 : \boldsymbol{q}_k \sim \mathcal{N}(\bar{\boldsymbol{q}} + \boldsymbol{\mu}_k, P) \tag{50b}$$

This model conservatively assumes that P is identical under both H_0 and H_1. A spoofing signal can introduce additional time variation in $\xi_k(\tau)$ due to its own dynamic multipath, which can inflate P in the positive definite sense. However, it is impossible to know the increase in the magnitude of P *a priori*, so a less-sensitive model of having a constant P is assumed.

Suppose one subtracts the static components of $\xi_k(\tau)$. This is analogous to performing nominal signal cancellation in the correlation domain by removing the component of $\xi_k(\tau)$ due to the authentic signal and removing the static multipath $\sum_{i=1}^{N_M} \xi_{M_s(k,i)}(\tau)$. Then the correlation deviation function $\xi_{zk}(\tau) = I_{zk}(\tau) + jQ_{zk}(\tau)$ can be obtained:

$$\xi_{zk}(\tau) = \xi_{Sk}(\tau) + \sum_{i=1}^{N_M} \xi_{M_d(k,i)}(\tau) + \xi_{Nk}(\tau) \tag{51}$$

where $\xi_{Sk}(\tau)$ denotes the contribution to $\xi_k(\tau)$ due to the spoofing signal. Let

$$\begin{aligned} \boldsymbol{z}_k &\triangleq \boldsymbol{q}_k - \bar{\boldsymbol{q}} \\ &= \left[I_{zk}(\delta_1), Q_{zk}(\delta_1), \ldots, I_{zk}(\delta_l), Q_{zk}(\delta_l)\right]^\mathsf{T} \in \mathbb{R}^{2l} \end{aligned}$$

be the vector composed of samples of $\xi_{zk}(\tau)$ at the tap locations in $\boldsymbol{\delta}$. The model in (50) can now be rewritten as

$$H_0 : \boldsymbol{z}_k \sim \mathcal{N}(\mathbf{0}, P) \tag{52a}$$

$$H_1 : \boldsymbol{z}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, P) \tag{52b}$$

The model in (52) is a special case of the general Gaussian problem [337] for which the optimal detection test can be reduced to

$$L(\boldsymbol{z}_k) = \boldsymbol{z}_k^\mathsf{T} P^{-1} \boldsymbol{z}_k - (\boldsymbol{z}_k - \boldsymbol{\mu}_k)^\mathsf{T} P^{-1} (\boldsymbol{z}_k - \boldsymbol{\mu}_k) \underset{H_0}{\overset{H_1}{\gtrless}} \nu \tag{53}$$

where $L(\boldsymbol{z}_k)$ is the log likelihood ratio and $\nu > 0$ is the threshold that yields the chosen probability of false alarm P_F.

This paper tackles the detection problem using two different techniques, an *Anomaly Test* (AT), which simply measures the fit of the observation $\boldsymbol{z}_k$ to the H_0 distribution by considering only the first term of (53), and a *Generalized Likelihood Ratio Test* (GLRT), which estimates $\boldsymbol{\mu}_k$ from the observations $\boldsymbol{z}_k$ to form the full detection statistic $L(\boldsymbol{z}_k)$ for the hypothesis test. These two techniques are elaborated in the following subsections.

Anomaly Test (AT)

Consider the optimal test in (53), which can be simplified by evaluating just the likelihood of the $p_0(\boldsymbol{z}_k)$ distribution:

$$L^*_{\text{AT}}(\boldsymbol{z}_k) = \boldsymbol{z}_k^{\mathsf{T}} P^{-1} \boldsymbol{z}_k \underset{H_0}{\overset{H_1}{\gtrless}} \nu^*_{\text{AT}} \tag{54}$$

where $\nu^*_{\text{AT}} > 0$ is the threshold that yields the chosen P_F given the $p_0(\boldsymbol{z}_k)$ distribution.

This technique can be used to detect any changes from the nominal signal landscape due to the presence of RFI. Due to its low computational needs, it is favorable for round-the-clock surveillance of the signal landscape. However, it does not glean any insight into the characteristics of the spoofing signal, unlike the GLRT detector, which will be elaborated in the next subsection.

Generalized Likelihood Ratio Test (GLRT)

The set of correlation distortion parameters $\{\epsilon_{Dk}, \Delta\tau_{Dk}, \Delta\theta_{Dk}\}$ is first estimated using a modified maximum-likelihood (ML) technique proposed in [338]. The estimator derived from this ML technique can detect any anomalous signal over a wide range of spoofing-to-authentic code offsets. This subsection details the adaptation of this estimator for TRNS spoofing detection.

The complex-valued ith tap of the correlation distortion function at time index k, $\xi_{Dk}(\tau) \triangleq I_{Dk}(\tau) + jQ_{Dk}(\tau)$ is expressed in terms of its amplitude ϵ_{Dk}, code phase offset $\Delta\tau_{Dk}$ and carrier phase offset $\Delta\theta_{Dk}$ as

$$\xi_{Dk}(\delta_i) = \epsilon_{Dk} R(\delta_i - \Delta\tau_{Dk}) \exp(j\Delta\theta_{Dk}) + \xi_{Nk}(\delta_i) \tag{55}$$

The correlation distortion vector $\boldsymbol{\mu}_k$ is similarly obtained by stacking the correlation distortion function from multiple taps:

$$\boldsymbol{\mu}_k = \left[I_{Dk}\left(\delta_1\right), Q_{Dk}\left(\delta_1\right), \ldots, I_{Dk}(\delta_l), Q_{Dk}\left(\delta_l\right)\right]^{\mathsf{T}} \tag{56}$$

The estimation of the correlation distortion's code phase offset can be separated from the estimation of its amplitude and carrier phase offset by exploiting the linear relationship

$$\boldsymbol{\xi}_{Dk} = H(\Delta\tau_{Dk}, \boldsymbol{\delta})\epsilon_{Dk} \exp(j\Delta\theta_{Dk}) \tag{57}$$

where $\boldsymbol{\xi}_{Dk} = [\xi_{Dk}(\delta_1), \cdots, \xi_{Dk}(\delta_l)]^{\mathsf{T}}$ and the observation matrix $H(\Delta\tau_{Dk}, \boldsymbol{\delta})$ is

$$H(\Delta\tau_{Dk}, \boldsymbol{\delta}) = \begin{bmatrix} R(\delta_1 - \Delta\tau_{Dk}) \\ \vdots \\ R(\delta_l - \Delta\tau_{Dk}) \end{bmatrix} \tag{58}$$

A coarse search is first performed by setting the code phase estimate $\Delta\hat{\tau}_{Dk} = \delta_i$ for $i = 1, \cdots, l$ and solving for the ML estimate of $\epsilon_{Dk}\exp(j\Delta\theta_{Dk})$ for each candidate $\Delta\hat{\tau}_{Dk}$:

$$\begin{aligned} &\hat{\epsilon}_{Dk} \exp(j\Delta\hat{\theta}_{Dk}) = \\ &\left[H^{\mathsf{T}}(\Delta\hat{\tau}_{Dk}, \boldsymbol{\delta}) Q^{-1} H(\Delta\hat{\tau}_{Dk}, \boldsymbol{\delta})\right]^{-1} H^{\mathsf{T}}(\Delta\hat{\tau}_{Dk}, \boldsymbol{\delta}) Q^{-1} \boldsymbol{\xi}_{zk} \end{aligned} \tag{59}$$

where Q is the $l \times l$ Toeplitz matrix that accounts for the correlation of the complex Gaussian thermal noise among the taps [339], and $\boldsymbol{\xi}_{zk} = \xi_{zk}(\boldsymbol{\delta})$ is the vector of correlation deviation function from all signal taps. The $(a, b)^{\text{th}}$ element of Q is $Q_{a,b} = R(|a - b|\Delta\delta)$, where $\Delta\delta$ is the tap spacing.

The cost J_k corresponding to each set of estimates $\left\{\hat{a}_{Dk}, \Delta\hat{\tau}_{Dk}, \Delta\hat{\theta}_{Dk}\right\}$ is calculated as

$$J_k = \|\boldsymbol{\xi}_{zk} - H^{\mathsf{T}}(\Delta\hat{\tau}_{Dk}, \boldsymbol{\delta})\hat{\epsilon}_{Dk} \exp(j\Delta\hat{\theta}_{Dk})\|^2_Q \tag{60}$$

where the norm is defined such that $\|\boldsymbol{x}\|^2_Q = \boldsymbol{x}^{\mathsf{T}} Q^{-1} \boldsymbol{x}$. The cost J_k is proportional to the negative log-likelihood function, so the set with the minimum cost is the ML estimate.

A bisecting search is then performed to obtain a refined code phase estimate using linear interpolation. At each bisection point, new amplitude and carrier phase estimates are determined by re-evaluating (59). The process is repeated until J_k converges, and the resulting estimates are accepted as the maximum-likelihood estimate $\left\{\hat{\epsilon}_{Dk}, \Delta\hat{\tau}_{Dk}, \Delta\hat{\theta}_{Dk}\right\}$. This estimate, with an example shown in Fig. 63, can correspond to the signal characteristics of the dynamic multipath, or spoofing signal, depending on their relative signal amplitude and code offset.

The maximum-likelihood estimate of $\xi_{Dk}(\tau)$ can be computed as

$$\begin{aligned} \hat{\xi}_{Dk}(\tau) &\triangleq \hat{I}_{Dk} + j\hat{Q}_{Dk} && (61) \\ &= \hat{\epsilon}_{Dk} R(-\Delta\hat{\tau}_{Dk} + \tau) \exp(j\Delta\hat{\theta}_{Dk}) && (62) \end{aligned}$$

from which the correlation distortion vector

$$\hat{\boldsymbol{\mu}}_k = \left[\hat{I}_{Dk}\left(\delta_1\right), \hat{Q}_{Dk}\left(\delta_1\right), \ldots, \hat{I}_{Dk}\left(\delta_l\right), \hat{Q}_{Dk}\left(\delta_l\right)\right]^{\mathsf{T}}$$

is obtained to evaluate the optimal test (53).

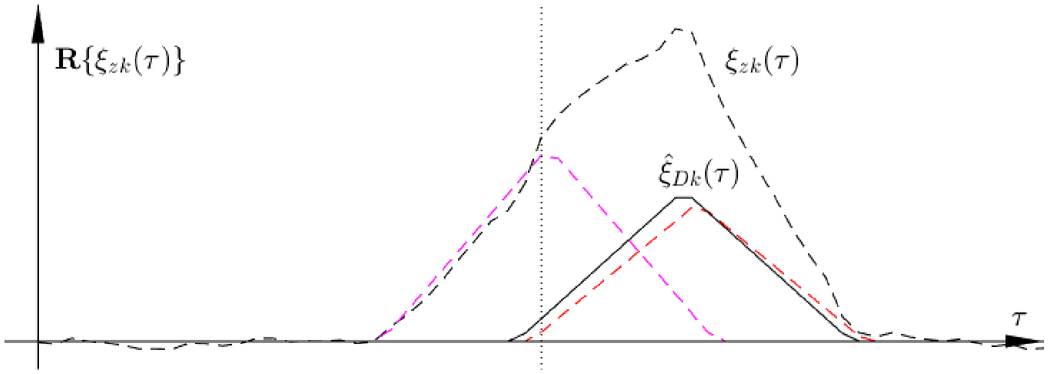


Fig. 63: The measured correlation distortion function $\xi_{Dk}(\tau)$ (dashed black) and its ML estimate $\hat{\xi}_{Dk}(\tau)$ (solid black) from an example scenario, shown in their in-phase components. The dotted black line corresponds to the delay of the authentic signal τ_A. Note that $\hat{\xi}_{Dk}(\tau)$ has a closer match to $\xi_{Sk}(\tau)$ (red) than to $\xi_{Mk}(\tau)$ (magenta), which implies that the estimated correlation distortion function is a good representation of the spoofing signal's correlation function.

Since both $p_0(\boldsymbol{z}_k)$ and $p_1(\boldsymbol{z}_k)$ are assumed to have the same covariance, (53) can be reduced to

$$L'(\boldsymbol{z}_k) = \hat{\boldsymbol{\mu}}_k^{\mathsf{T}} P^{-1} \boldsymbol{z}_k \underset{H_0}{\overset{H_1}{\gtrless}} \nu' \tag{63}$$

where $\nu' > 0$ is the threshold that yields the chosen P_F based on the distribution of $L'(\boldsymbol{z}_k)$ under H_0.

Analysis can be further simplified by letting $\boldsymbol{z}_{a,k} = R_a^{-T} \boldsymbol{z}_k$ and $\boldsymbol{\mu}_{a,k} = R_a^{-T} \boldsymbol{\mu}_k$, where R_a is the Cholesky factorization of P. The optimal test then becomes

$$L^*_{\text{GLRT}}(\boldsymbol{z}_{a,k}) = \hat{\boldsymbol{\mu}}_{a,k}^{\mathsf{T}} \boldsymbol{z}_{a,k} \underset{H_0}{\overset{H_1}{\gtrless}} \nu^*_{\text{GLRT}} \tag{64}$$

which implies a correlation-and-accumulation structure, with ν^*_{GLRT} being the threshold derived from the H_0 distribution using a chosen P_F.

This technique is sub-optimal, as the quality of the detector depends on the quality of the estimated parameters $\{\epsilon_{Dk}, \Delta\tau_{Dk}, \Delta\theta_{Dk}\}$ from ML estimation. Nonetheless, it is effective in discerning H_1 from H_0 for TRNS spoofing detection.

SIMULATIONS

The AT and GLRT spoofing detectors were tested in simulation under different scenarios. The following subsections outline the simulation setup, and the performance of the detectors under different operating conditions (different transmitter power level and receiver sensitivity range).

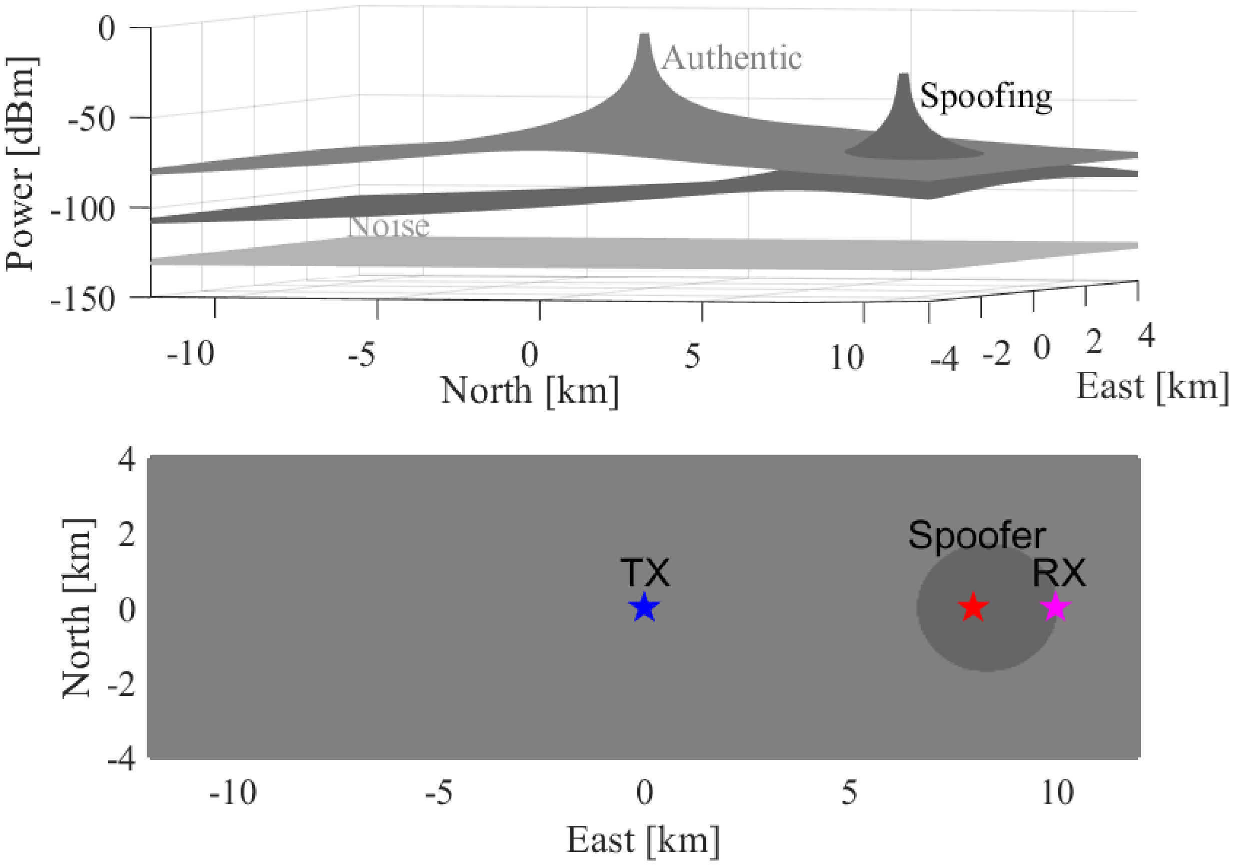


Fig. 64: Simulation setup used for all test cases. The transmitting beacon and the spoofer are located 10 km and 2 km away from the monitoring beacon respectively. The top plot shows the amplitude of the authentic and spoofing signals over a 10 km by 4 km grid, both of which are above the noise floor of the receiver.

Simulation Setup

In order to have indoor positioning capability, the transmit power and spatial distribution of TRNS beacons have to compensate for high signal energy absorption by building materials, while minimizing infrastructure cost and near-far interference. Fig. 64 represents a small subset of a dense mesh deployment of TRNS beacons with a spacing of 10 km. The worst case scenario of spoofing is considered in all test cases, where a zero-latency spoofer was placed along the line-of-sight path between a pair of transmit and listening beacons. In each run, the spoofer power was set to reflect the spoofing power ratio (i.e. ratio of the spoofing power versus the authentic signal power) at the receiving beacon. A path loss exponent α of 3 was used to reflect a generic urban environment of the TRNS beacons [340]. The monitoring receiver's antenna experiences an ambient temperature T of 290 K, and has a front-end bandwidth B of 20 MHz. 10000 runs were conducted during the calibration phase using H_0 distribution, which is made up of 1 authentic signal, 8 static multipath and 1 dynamic multipath. Each post-correlation function $\xi_k(\tau)$ is computed across a correlation window of 20 chips from 1 ms of signal accumulation. The test statistics collated during calibration were used to compute the thresholds for each detector, based on a probability of false alarm P_{FA} of 1 in 1000. 2000 runs were then conducted during the trial phase, with an additional spoofing signal in the landscape, to determine the probability of detection P_D of the spoofing signal at each spoofing power ratio. The distributions of all the signal components were outlined in [333, Subsec. 5.2.2].

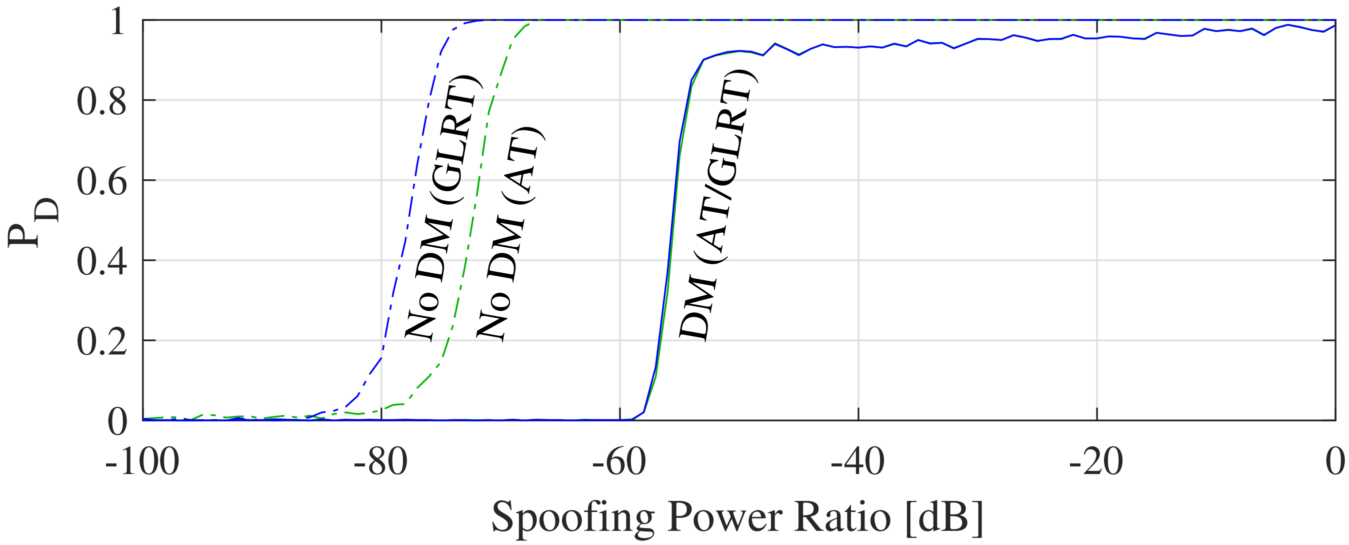


Fig. 65: Simulation results for AT and GLRT detectors, without dynamic multipath (No DM) and with dynamic multipath (DM). For discussion on the long-tail distribution on the right, see Subsection . While the DM curve of GLRT appears similar to that of AT, it exhibits differences at the 5% level in the vicinity of the threshold.

Detector Comparison

Fig. 65 shows the performance of the AT and GLRT detectors, respectively, at a beacon transmit power of 30W. This power is sufficient to compensate for propagation and absorption losses over a 10 km spacing between beacons, as anticipated by a TRNS provider like NextNav. Each detector is simulated both with and without a dynamic multipath component. In each of these 4 cases, the condition under which the detector is trained and the condition under which it is evaluated is the same. It is no surprise that the confounding influence of dynamic multipath reduces the performance of each detector. Absent dynamic multipath, the GLRT exhibits a sensitivity advantage of roughly 5 dB. Under dynamic multipath, neither detector exhibits a significant advantage: the GLRT's 50% sensitivity threshold is 0.13 dB better (i.e. lower) than that of the AT.

In the dynamic multipath cases, each detector exhibits a sharp threshold and a long tail of false negatives. The region to the left of the threshold is dominated by noise. In this regime, P_D improves with increasing spoofing power ratio as the spoofing power approaches the noise floor at the receiver. To the right is the multipath-dominated region. Here, a false negative rate of 10% narrows towards zero with increasing spoofing power ratio. This occurs because, as discussed in [333, Subsec. 5.2.2], the spoofer's simulated code phase may coincide with the window of correlator output taps that are effectively desensitized by dynamic multipath. Due to the particular parameters used, this occurs 10% of the time. At high enough spoofing power ratio, this desensitization no longer prevents detection.

Different Levels of Transmitter Power

Fig. 66 shows the detection performance of the GLRT detector under different authentic transmitter power levels. Each simulated detection has access to only 1 ms of signal. Considering transmit power levels running upwards from -70 dBW, detection performances improve at all spoofing power ratios until the detector exhibits a saturation effect at a transmit power level of 10 dBW. Note that the spacing between adjacent curves is not uniform with transmit power level.

In order to interpret the saturation and non-uniform spacing effects, one may recast these observations in terms of received power (not spoofing power ratio) versus transmitted power. However, in order to do this, one must choose a single point on the P_D curve to summarize detector performance at a particular transmit power level. In Fig. 67, this point is arbitrarily chosen to be the 50% detection threshold. That is, at any given transmit power level, Fig. 67 shows the received power corresponding to a 50% rate of detection of the spoofer by the TRNS monitor.

Fig. 67 suggests that the saturation and non-uniform spacing phenomena in Fig. 66 indicate the presence of 3 quantitatively distinct regimes, in order from right to left:

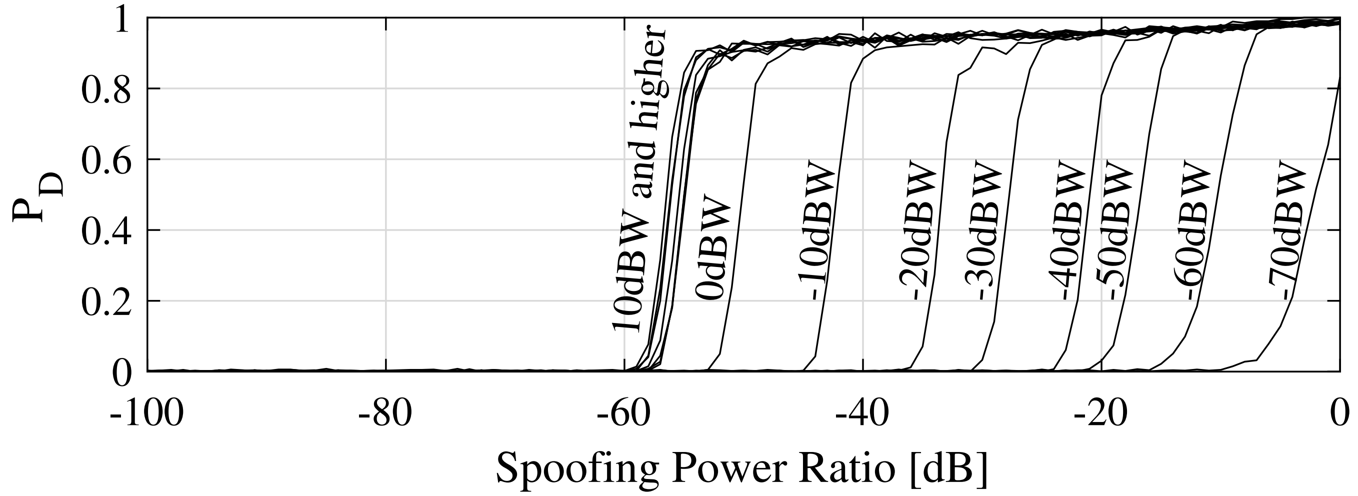


Fig. 66: Simulation results of the GLRT detector under different transmitter power level.

- Region I: Quantization noise power P_Q dominates over thermal noise P_N at the receiver, where $P_N = S_{nn}B$ is the noise power over a channel bandwidth B and S_{nn} being the noise spectral density. Furthermore, the sensitivity threshold P_I is greater than P_N.
- Region II: Thermal noise dominates over quantization noise and the detection threshold is comparable to the thermal noise level, $P_I \approx P_N$.
- Region III: Thermal noise still dominates and the spoofing signal is only detectable post-correlation ($P_I \ll P_N$).

Naturally, if $P_I > P_A$, then we are "in clover": detection is not challenging!

Receiver Front-End Details: The boundary between Regions I and II is sensitive to the behavior of the programmable gain amplifier (PGA) in the monitoring receiver. One common model for a quantizing receiver is to build a variable attenuator followed by a fixed-gain amplifier before the signal reaches the analogue-to-digital converter (ADC). In order to avoid saturating the ADC, that is, exceeding its input voltage range, the variable attenuator is commanded to reduce the power from the antenna according to the statistics of the ADC output in a feedback loop. In Fig. 66, the P_D curves begin "stacking up" when the transmit power becomes high enough to enter Region I: that is, when additional transmit power must be exactly offset by increased attenuation in the receiver. In this regime, thermal noise is negligible compared to quantization noise, which tracks with transmitter power. Thus, in Region I, the slope of the 50% detection curve in Fig. 67 is unity. Increasing the transmitter power in Region I does not improve P_D because the variable attenuator is forced to further suppress the incoming signal by the same amount, leading to no net increase in sensitivity.

In Region II, there is no suppression of the incoming signal by the variable attenuator, as all received signals are within the sensitivity range of the ADC at full PGA gain. Assuming as in Section that cancellation of the authentic signal and the static multipath components at the monitoring receiver may be considered perfect in this regime, the detector need only distinguish the spoofing signal from thermal noise and dynamic multipath. So long as the dynamic multipath remains relevant (i.e. comparably strong to the spoofed signal), it will prevent the receiver from identifying spoofing signals that are below the noise floor. resulting in a relatively flat 50% detection curve.

In Region III, both the spoofing signal and dynamic multipath have processing gain advantage over thermal noise from despreading. The detector in this regime has to only differentiate the spoofing signal from dynamic multipath, with this sensitivity decreasing with lower transmit power level, resulting in the 50% detection curve having a slope less than unity.

Receiver Sensitivity Range

Fig. 68 shows the detection performance of the GLRT detector with different ADC bit depths for two distinct transmit power levels, and Fig. 69 shows these data recast in terms of RX power at the 50% detection threshold versus authentic signal TX power. One may infer that the sensitivity threshold does not improve with bit depth at low transmit power levels. With regards to the regions discussed in Subsection , these plots reveal two trends. First, a larger ADC bit depth results in a lower

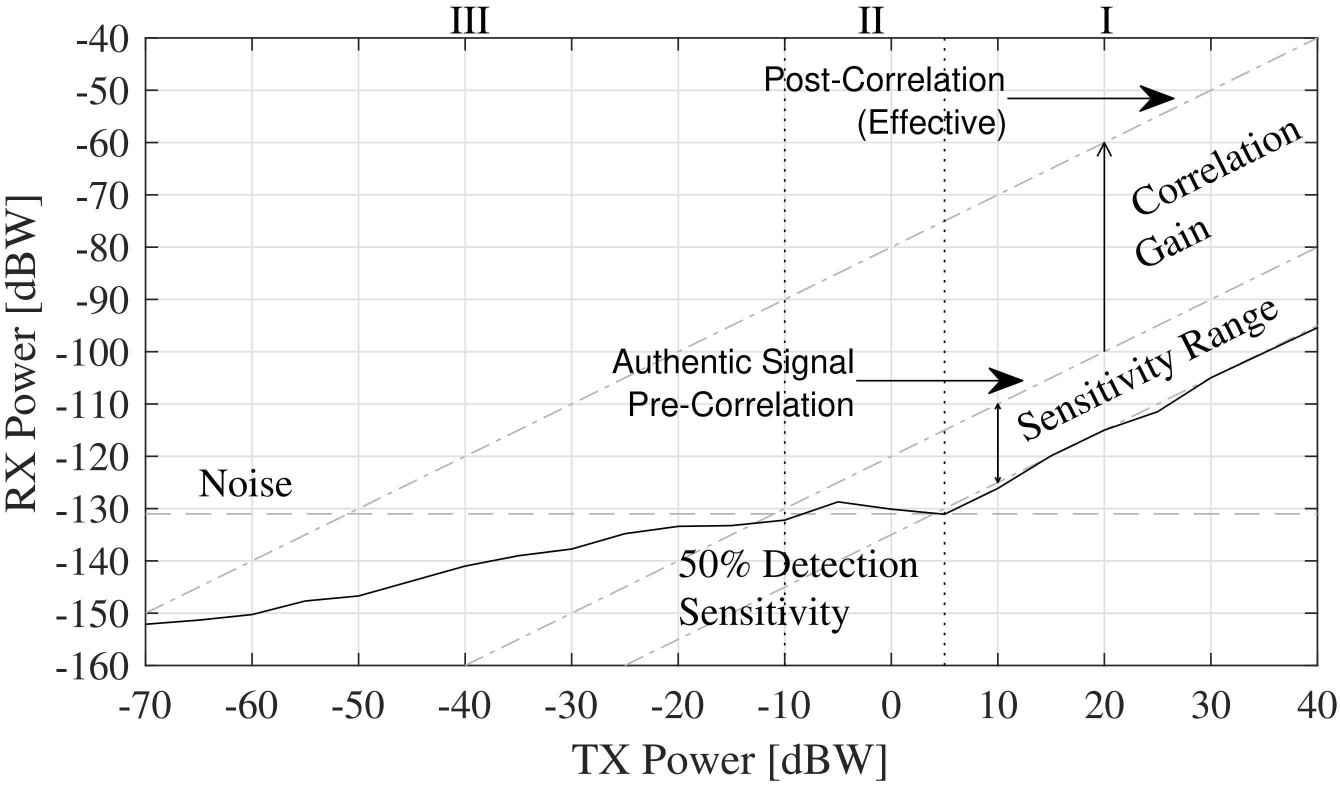


Fig. 67: Simulation results of the GLRT detector under different transmitter power, showing the 50% detection sensitivity curve with 3-bit quantization.

quantization noise level in Region I due to lower suppression by the variable attenuator. Second, the dividing line between Regions I and II moves rightward with increasing bit depth. That is, the thermal noise dominates up to a higher transmit power level. Quantization is not the performance-limiting factor in Region III.

CONCLUSION

This paper proposes the addition of signal-situational-awareness (SSA) capability to the TRNS network, to augment cryptographic NME+NMA scheme in countering against SCER and meaconing attacks. Two signal authentication techniques are proposed for SSA that allow TRNS operator to detect weak signal spoofing in the presence of multipath without the use of costly full-duplex techniques. The first technique, the *Anomaly Test*, compares the current observations against an empirical model of typical (nominal) observations, and has an advantage in simplicity and performance. The second technique searches for the spoofing signal and compares the observations against a reconstruction of the most likely spoofer: the *Generalized Likelihood Ratio Test* (GLRT) technique. The GLRT method performs as well or better than the Anomaly Test in all considered test conditions. The GLRT exhibits a sensitivity advantage of 5 dB over the Anomaly Test in the absence of dynamic multipath, which drops to 0.13 dB in the presence of dynamic multipath. In addition, the GLRT has a 50% spoofer detection threshold up to -74 dB with high transmit power level and 6-bit ADC quantization. Simulations of both detectors under various operating conditions encountered by a generic TRNS quantify their performance. Terrestrial radionavigation systems will benefit not only from techniques designed to secure traditional GNSS, but also from the exploitation of novel opportunities for signal situational awareness arising from the proximity and mutual audibility of the transmitting beacons, rendering TRNS more resilient against man-in-the-middle attacks.

5d Autonomous Signal-Situational-Awareness in a Terrestrial Radionavigation System

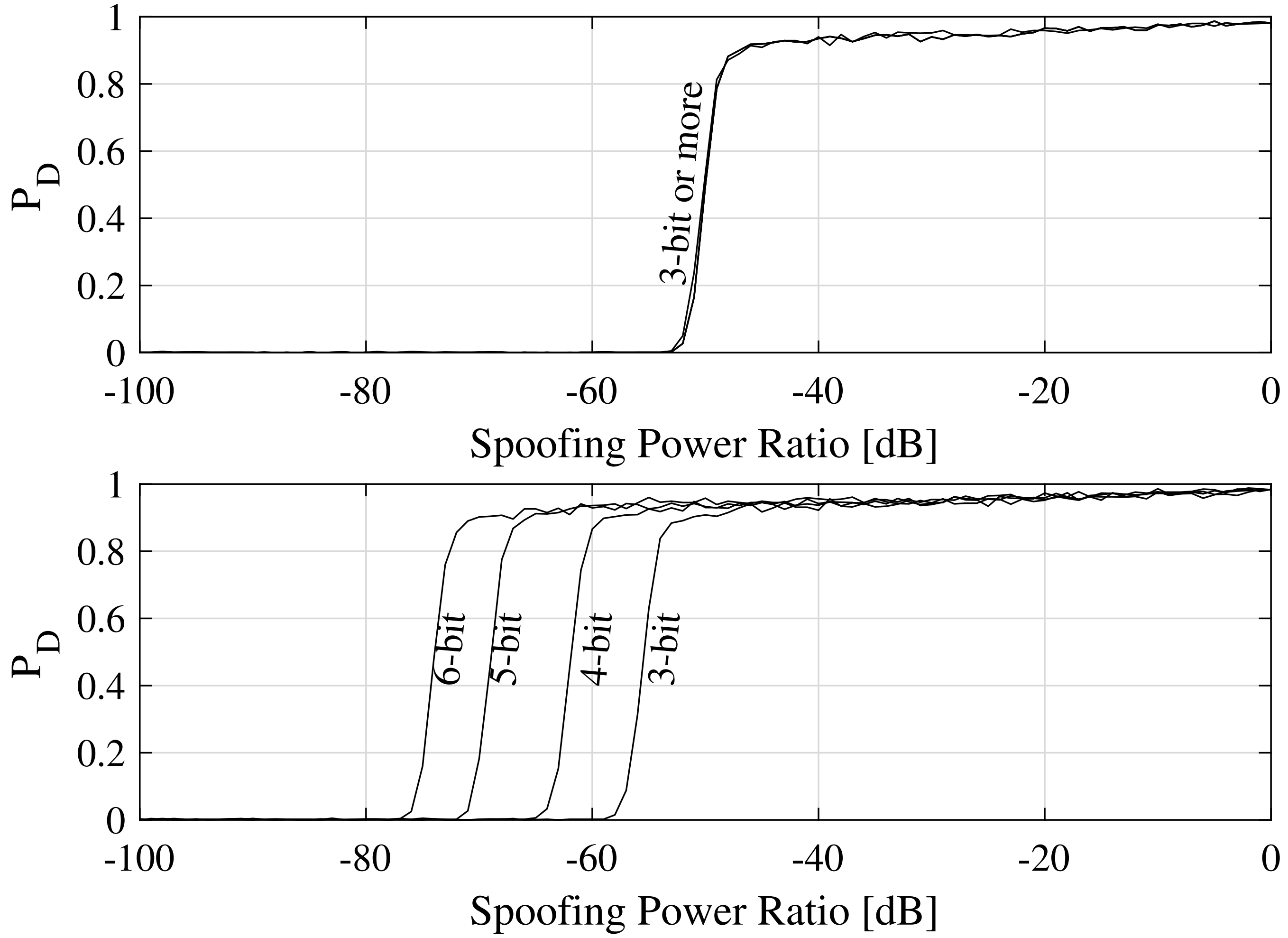


Fig. 68: Simulation results of the GLRT detector with different ADC bit depth, for a 0 dBW (top) and 40 dBW (bottom) transmitter located at 10 km away from a listening beacon.

ABSTRACT

Intersection movement assist (IMA) is a connected vehicle (CV) application to improve vehicle safety. GPS spoofing attack is one major threat to the IMA application since inaccurate localization results may generate fake warnings that increase rear-end crashes, or cancel real warnings that may lead to angle or swipe crashes. In this work, we first develop a GPS spoofing attack model to trigger the IMA warning of entry vehicles at a roundabout driving scenario. The attack model can generate realistic trajectories while achieving the attack goal. To defend against such attacks, we further design a one-class classifier to distinguish the normal vehicle trajectories from the trajectories under attack. The proposed model is validated with a real-world data set collected from Ann Arbor, Michigan. Results show that although the attack model triggers the IMA warning in a short time (i.e., in a few seconds), the detection model can still identify the abnormal trajectories before the attack succeeds with low false positive and false negative rates.

INTRODUCTION

Connected vehicle (CV) technology has great potential to benefit the transportation system in terms of improving system efficiency, sustainability, and safety. Vehicle-to-vehicle (V2V) communication enables the CVs to send and receive real-time information from other nearby vehicles, for example, Basic Safety Messages (BSMs) to avoid collisions. BSMs play an important role in multiple CV applications, such as intersection movement assist (IMA), a widely implemented CV application to improve vehicle safety [341]. The IMA system can be applied when vehicles pass through unsignalized

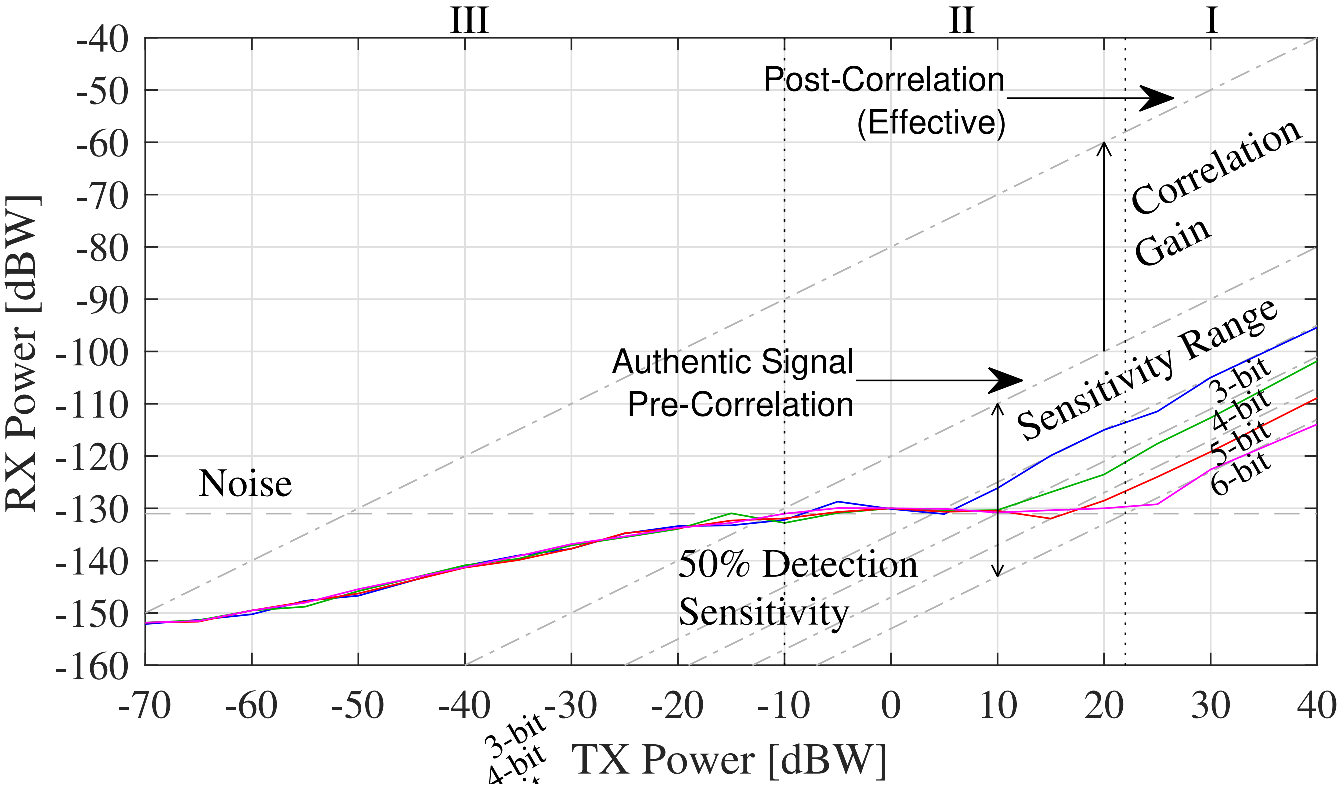


Fig. 69: Simulation results of the GLRT detector under different transmitter power, showing the 50% detection sensitivity curve with different levels of quantization. The boundary between Regions I and II varies with depth and is shown for 6-bit quantization.

intersections. It receives other approaching vehicles' information such as location and speed to determine whether it is safe to enter the intersection. If a potential collision is detected, a warning message will be generated and sent to the driver (e.g., through an in-vehicle display or an audio warning). Among all information that is shared through V2V communication, vehicle position is critical in deciding whether to generate warnings to drivers. To obtain a vehicle's real-time position, a GPS receiver is commonly used for vehicle localization and navigation [342], [343]. Commercial-grade GPS receivers can get vehicle position within a meter accuracy [342] while AV-grade GPS receiver has centimeter-level positioning accuracy [344]. It is important to guarantee that the vehicle's localization module is accurate and reliable.

Existing studies show that GPS receivers are vulnerable to multiple cyber attacks. One major threat is the spoofing attack, which has been proved feasible both theoretically [345] and practically on various systems [346], including in autonomous vehicles [344]. To defend against GPS spoofing attacks, multiple detection methods have been proposed, including filter-based methods and observer-based methods [347] [348]. However, the proposed methods either rely on other onboard sensors or V2V information from surrounding vehicles, which may not be available in the CV environment with a low penetration rate. Our previous work [349] proposed a GPS spoofing attack detection method, which combines learning from demonstration and a decision tree classifier. The decision tree classifier needs to be trained using both ground truth trajectory and known attack trajectory. As a result, the proposed detection framework can be only applied to detect known attacks. However, in reality, new attacks are usually unknown to the detector, where the attack trajectories can not be obtained for training the classifier.

In this paper, we first propose a GPS spoofing attack model which aims to trigger the IMA warning of entry vehicles in a roundabout scenario. We further design a one-class classifier to distinguish the normal trajectories from the trajectories under attack, where only the normal trajectories are needed for training. Our work can be briefly summarized as follows. The attack model is formulated as an optimization problem, with specifically designed features to trigger the IMA warning while generating as normal driving behaviors as possible. The detection framework includes a feature extractor and a one-class neural network classifier. In the case study, a real-world data set, which is collected from a two-lane roundabout in Ann Arbor, Michigan [350] is applied to test both the CV threat model and the detection model. Results show that the proposed threat model can trigger the IMA warning in a short time (less than 1.7s) which poses great challenges to the detection. The online detection results denote that the proposed detection framework can differentiate the normal and abnormal trajectories before the attack succeeds time, with both low false positive rates and low false negative rates.

The main contributions of the paper are listed as follows:

1. We formulate the GPS spoofing attack as an optimization model, with the goal to trigger an IMA warning as well as generating smooth and realistic trajectories. The proposed threat model takes vehicle's initial state and road geometry into consideration and can be applied to different scenarios, not limited to the roundabout.

2. A generic detection framework is proposed. The detection framework combines a feature extractor with a one-class classifier. The feature extractor can be adjusted according to different driving scenarios. Besides, the proposed framework only requires normal trajectories for training, which enables it to detect unknown attacks.

The remainder of the paper is arranged as follows. Section II reviews related literature on GPS spoofing attacks and related detection methods. Section III introduces the threat model, including the problem statement, objective function, and implementation framework. Section IV presents the detection methodology. Numerical experiments from the roundabout scenario are introduced in section V. Section VI concludes the work and lays out future research directions.

LITERATURE REVIEW

In this section, we reviewed literature related to GPS spoofing attacks and related detection models.

GPS Spoofing Attack

GPS spoofing attack has been a long-existing problem. The attack broadcasts incorrect but valid GPS signals to mislead GPS receivers [351]. By providing falsified information, GPS spoofing attacks can deviate vehicles to random positions [346] or guide the vehicles to the wrong destinations. Zeng et al. [67] proposed a stealthy attack against the road navigation system. GPS locations were spoofed slightly to trigger the turn-by-turn navigation and guided the vehicle to the wrong destination without recognizing the attack. To prove the feasibility, the proposed GPS spoofing model was tested on real vehicles. Narain et al. [352] evaluated the INS-aided GPS system and developed algorithms to deviate the vehicle to alternative locations without being detected. The result showed that the proposed algorithm could deviate vehicles as far as 30km from the origin without raising alarms. Multi-Sensor Fusion (MSF) is usually considered one approach to defending GPS spoofing attacks. An MSF system combines inputs from multiple sensors for vehicle localization. It is highly unlikely that all sensors can be compromised at the same time. For example, Liu et al. [353] proposed an Extended Kalman filter (EKF) based algorithm to fuse the measurements from multiple sensors. The proposed algorithm performed well under GPS spoofing attacks where the GPS signal was deviated by a fixed bias. However, Shen at al. [344] proposed a GPS spoofing attack algorithm that penetrated the MSF based localization system with GPS, IMU and Lidar. The proposed algorithm only spoofed GPS to cause large deviations in the MSF output. It could deviate the vehicle from the original lane, or cross the road boundary, which may lead to collisions with other vehicles.

GPS Spoofing Attack Detection

To defend against GPS spoofing attacks, anomaly detection methods have been proposed. The detection methods can be divided into filter-based methods and observer-based methods [347]. Filter-based methods consider uncertainties and measurement noises and apply filters such as Kalman Filter to detect attacks on sensors. Van et al. [354] proposed an anomaly detection approach that combines convolutional neural network (CNN) and Kalman filtering with χ^2 detector. CNN was used for detecting anomalies in time-series sensor data and KF-based χ^2 detector is applied to detect the abnormal data which are undetected by CNN. Ju et al. [355] proposed a simple distributed Kalman filter based on neighboring vehicle measurement exchange. A Generalized likelihood ratio (GLR) detector was proposed to detect position sensor attacks based on the Kalman filter's result.

Compared with filter-based methods, observer-Based methods are usually applied to deterministic vehicle models. Wang et al. [348] proposed an observer-based method. An adaptive extended Kalman filter was applied to smooth vehicle sensor data based on a nonlinear car-following model. One Class Support Vector Machine (OCSVM) is applied to detect anomalies on sensors. He et al. [356] proposed an observer-based detection framework. A detector was developed according to the potentially compromised sensor measurements and the observer's estimation. Measurement data was discarded if it was larger than a threshold.

The existing methods either need input from multiple sensors [353], input from known attacks [349], or only detect the anomaly in the longitudinal vehicle dynamics [354]. Besides, surrounding vehicle states such as leading vehicles or information from other vehicles in the platoon are needed [356]. They may not be applicable to our case because 1) in the CV environment, there may not exist other onboard sensors to perform multi-sensor fusion or cross-validate the results from the GPS, especially if a vehicle is equipped with aftermearket safety devices (ASDs). 2) in the roundabout scenario, vehicles have lateral movement due to road curvature, which significantly increases the detection difficulty.

THREAT MODEL

In this section, the threat model towards the IMA application on the CV is presented. IMA is an important CV safety application [341] [357]. When approaching an intersection, the IMA system first receives information (i.e., BSMs) from other vehicles. According to the received data, the IMA system determines whether it is unsafe to enter an unsignalized intersection due to potential collision with other vehicles and sends warnings to drivers. Drivers receiving collision warning information from the IMA system should perform actions to avoid crashes at the intersection. In this work, it is assumed that the IMA warnings are triggered only based on received BSMs from other vehicles at the intersection. We propose a threat model towards the IMA system at the roundabout scenario, which is a common type of unsignalized intersection.

Problem Statement

The proposed CV threat model generates falsified BSMs to trigger the IMA warning of CV at the entry of the roundabout. Figure 70 demonstrates the attack concept. The figure contains two vehicles, the vehicle under attack and the victim vehicle and both vehicles are CVs. The attack vehicle is the vehicle located in the inner lane of the roundabout. The victim vehicle is the vehicle located at the entry of the roundabout. The blue rectangles denote the real vehicle trajectory in the inner lane of the roundabout. The red rectangles denote the falsified BSM trajectory when the vehicle is under attack, changing lanes from the inner lane to the outer lane. The yellow rectangles denote the victim vehicle trajectory. A conflict point is defined as the intersection between the center line of the outer lane of the roundabout and the entry path. Note that lane changing is forbidden within the multi-lane roundabout. When the vehicle is not under attack, the BSM sent from the CV in the roundabout should be consistent with the real trajectory (the blue rectangles). There is no conflict between the real CV trajectory and the victim vehicle trajectory. The IMA warning will not be triggered for the entry vehicle. When the vehicle is under attack, the CV in the roundabout sends out falsified BSMs (the red rectangles) to trigger the IMA warning of the victim vehicle, without really controlling vehicle movement. The values within the rectangles denote the timestamps. The attack starts at time t_0 and the attack successfully triggers the IMA warning of the victim vehicle at time t_2.

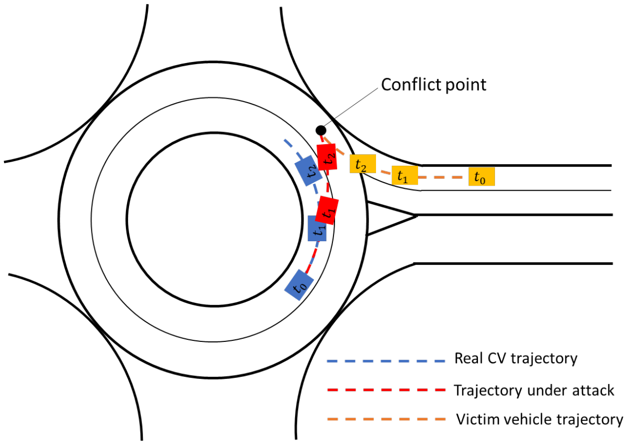


Fig. 70: Threat model on intersection movement assist system

To generate the falsified trajectory (the red rectangles), an optimization problem is formulated, as shown in Equation 65. The objective function is presented as $\theta^T f(\mathbf{s}, \mathbf{u})$. θ is the weight vector and $f(\mathbf{s}, \mathbf{u})$ is a function mapping a trajectory to feature vectors. **s** is the variable of the optimization model. $\mathbf{s} = (s_1, s_2, ..., s_N)$ denotes the set of trajectory points, where s_i is the trajectory point at time step i. Each trajectory point s_i consists of $(x_i, y_i, v_i, a_i, \psi_i)$, where x_i,y_i denotes vehicle's longitudinal and lateral coordinate at time step i. v_i and a_i represent vehicle speed and acceleration at time step i. ψ_i is the vehicle's heading angle. N is the planning horizon for the attack trajectory, which is determined by the estimated arrival time for the victim vehicle to reach the conflict point. **u** denotes the vehicle's initial state and road geometry. The vehicle's initial state includes its initial position and status (speed, heading, acceleration). Road geometry includes the radius of the inner and outer lanes of the roundabout and the coordinate of the conflict point. The feature selection for the objective function and the constraints are introduced in the following section.

$$\begin{aligned} \min_{\mathbf{s}} \quad & \theta^T f(\mathbf{s}, \mathbf{u}) \\ \text{s.t.} \quad & \text{vehicle dynamic constraints} \end{aligned} \tag{65}$$

Objective Function

3) Feature Vectors: The objective function contains two parts: 1) trigger the IMA warning of the victim vehicle. 2) generate a trajectory close to the normal driving behavior considering smoothness and comfort. A realistic attack trajectory will increase the difficulty in detection. To achieve the attack goal, five features are selected and elaborated as follows:

(1) Acceleration: $f_1 = \frac{1}{N}\sum_i a_i^2$. f_1 sums up the a_i^2 for the entire trajectory. Uncomfortable driving behavior such as large accelerations are penalized by minimizing f_1.

(2) Heading rate: $f_2 = \frac{1}{N}\sum_i(\dot{\psi}_i)^2$. ψ_i denotes the heading angle change rate at time step i. f_2 minimizes the difference of heading rate for two consecutive time steps.

(3) Curvature: $f_3 = \frac{1}{N}\sum_i(\sqrt{(x_i - x^c)^2 + (y_i - y^c)^2} - r^c)^2$. x^c and y^c denote the coordinate of the center of the roundabout. r^c denotes the radius of the roundabout. f_3 calculates the difference between the vehicle's distance to the center of the roundabout and the roundabout radius at time step i. f_3 guarantees the vehicle stays in the roundabout.

(4) Lateral terminal position: $f_4 = (x_N - x^{con})^2$ x_N denotes the vehicle's lateral coordinate at the end of the planning horizon. x^{con} represents the conflict point's lateral coordinate.

(5) Longitudinal terminal position: $f_5 = (y_N - y^{con})^2$ y_N is the vehicle's longitudinal coordinate at the end of the planning horizon. y^{con} is the conflict point's longitudinal coordinate. f_4 and f_5 push the vehicle to reach the conflict point at the end of the planning horizon and trigger the IMA warning of the victim vehicle.

4) Vehicle Dynamic Constraints: In this section, vehicle dynamic constraints are introduced. Constraint 66-69 denotes the evolution of vehicle state, including vehicle position, speed, acceleration, and heading angle. Equation 70-72 limits vehicle kinematic parameters within boundaries. Equation 70 bounds the vehicle's maximum acceleration and deceleration to be less than $8m/s^2$. Equation 71 limits vehicle's heading rate within range $(-\frac{\pi}{3}, \frac{\pi}{3})$. Equation 72 limits the vehicle's maximum speed.

$$x(i+1) = x(i) + v(i)cos(\psi(i))\tau \tag{66}$$

$$y(i+1) = y(i) + v(i)sin(\psi(i))\tau \tag{67}$$

$$\dot{\psi}(i) = \frac{(\psi(i+1) - \psi(i))}{\tau} \tag{68}$$

$$v(i+1) = v(i) + a(i)\tau \tag{69}$$

$$-8 \leq a(i) \leq 8 \tag{70}$$

$$-\frac{\pi}{3} \leq \dot{\psi}(i) \leq \frac{\pi}{3} \tag{71}$$

$$v(i) \leq 16.7 \tag{72}$$

Implementation Framework

The implementation framework of the attack model is described in this section. Details of the attack trajectory generation procedure is described in the following steps and shown in Figure 71.

Step 1: Collect vehicle state. The vehicle under attack (roundabout vehicle) collects its own state and the entry vehicle state, including vehicle speed and position.

Step 2: Calculate estimated arrival times to the conflict point for the roundabout vehicle and the entry vehicle, denote as t_r^{est} and t_e^{est}. D is vehicle's distance to conflict point. v denotes vehicle's current speed. The estimated arrival time is calculated using Equation 73, assuming the vehicle keeps current speed to reach the conflict point.

$$t^{est} = \frac{D}{v} \tag{73}$$

Step 3: Determine the attack start time. The attack starts when $|t_r^{est} - t_e^{est}|$ is less than 4s. If the criterion to start attack is satisfied, go to **Step 4**. Otherwise, go to **Step 1**. The threshold of launching attack is a hyper-parameter and needs to calibrated based on real-world data.

Step 4: Update entry vehicle state.

Step 5: Generate attack trajectory. The attack trajectory is generated according to Equation 65, assuming victim vehicle (entry vehicle) keeps a constant speed. The length of the attack trajectory (i.e., planning horizon) is set to be the same as the estimated arrival time of the roundabout vehicle to the conflict point, calculated in Equation 73.

Step 6: Determine whether the attack success criterion is satisfied. If the attack success criterion is satisfied, the attack ends. Otherwise, go to **Step 7**. Equation 74 and 75 denote the attack success criterion. D_{atk} is the attack vehicle distance to the conflict point. v_{atk} is the attack vehicle speed. r_{atk} is the distance between the attack vehicle trajectory point and the center of the roundabout. r_l denotes the lane boundary radius between the inner and outer lanes. The attack succeeds when the post encroachment time (PET) to the conflict point between the attack trajectory and the victim vehicle is less than T_g and the attack trajectory is deviated from the inner lane and crosses the road boundary. In this work, T_g is equal to $2s$. Both attack success criteria guarantee the two vehicles have a potential collision at the conflict point.

$$\frac{D_{atk}}{v_{atk}} \leq T_g \tag{74}$$

$$r_{atk} \geq r_l \tag{75}$$

Step 7: Determine when to stop generating attack trajectory. When the entry vehicle has reached the conflict point and the attack success criterion is not satisfied, continue attacking the vehicle will no longer trigger the IMA warning of the entry vehicle. Therefore, the attack should end. Otherwise, go to **Step 4**.

In order to minimize the prediction error, a rolling horizon framework is applied to update the entry vehicle information (speed and position) and calculate the planning horizon once the attack starts (**Step 4-Step 7**). The attack trajectory is generated for the whole planning horizon, but only the first 0.4s will be used.

DETECTION METHODOLOGY

In this section, we introduce a one-class classifier that is designed to detection the GPS spoofing attack toward the IMA application.

Detection Framework

Figure 72 demonstrates the detection framework. It consists of two parts, an offline training step and an online detection step. First, a training data set that includes historical normal trajectories is collected. A feature extractor is applied that maps trajectories to feature vectors, which represent different aspects of driving behaviors. A one-class neural network classifier

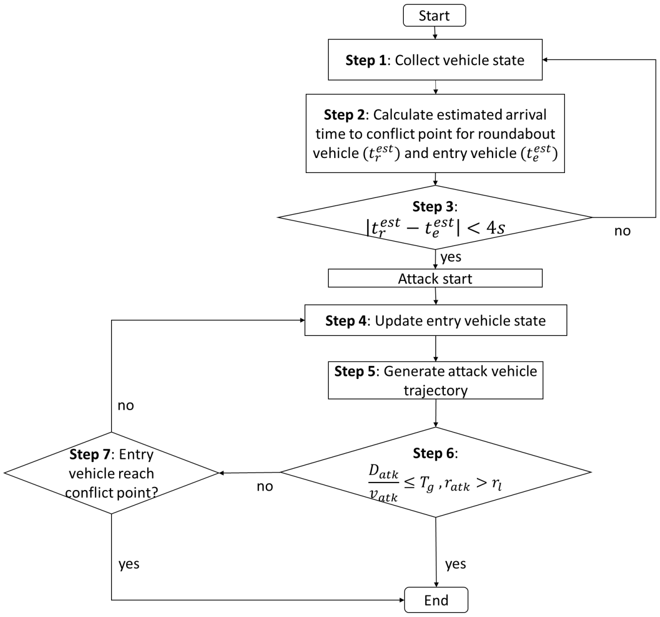

Start
Step 1: Collect vehicle state
Step 2: Calculate estimated arrival time to conflict point for roundabout vehicle (t_r^{est}) and entry vehicle (t_e^{est})
Step 3: $|t_r^{est} - t_e^{est}| < 4s$
no
yes
Attack start
Step 4: Update entry vehicle state
Step 5: Generate attack vehicle trajectory
no
Step 7: Entry vehicle reach conflict point?
no
Step 6: $\frac{D_{atk}}{v_{atk}} \leq T_g, r_{atk} > r_l$
yes
yes
End

Fig. 71: Threat model implementation framework

is trained with extracted features from normal trajectories. The trained classifier is then applied to the online detection, as shown at the bottom of Figure 72. Given the observed trajectory, the same feature extractor is applied to extract driving related features. The extracted features are sent to the trained anomaly classifier, to determine if the vehicle is under attack or not.

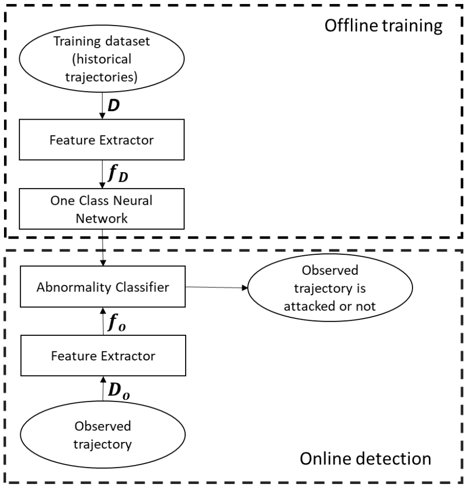


Fig. 72: Anomaly detection framework

One class classification

Figure 73 demonstrates the structure of the one-class classifier, which contains a feature extractor and a classifier network. The feature extractor is predefined and maps the input trajectory into a feature vector. Pseudo-positive data are generated from a Gaussian distribution $N(\overline{\mu}, \sigma^2)$. σ and $\overline{\mu}$ are parameters of the Gaussian distribution. Denote N is the number of input data and D is the dimension of the feature vector. The generated pseudo-positive data has the same dimension as the feature data set. The generated data are then combined with the extracted feature data set and fed into a classifier, following the method shown in [358]. The classification network is composed of three fully connected layers, followed by a sigmoid function. The output of the classifier is 0 or 1. 0 denotes that the data sample belongs to normal and 1 denotes that the data sample is abnormal. The binary cross entropy loss is applied as the loss function to train the classification network.

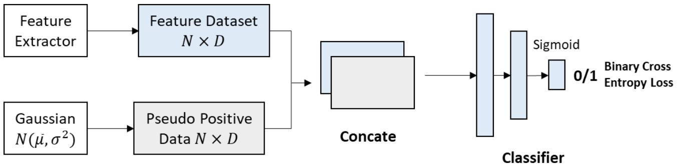

Feature Extractor
Feature Dataset $N \times D$
Gaussian $N(\mu, \sigma^2)$
Pseudo Positive Data $N \times D$
Concate
Sigmoid
0/1
Binary Cross Entropy Loss
Classifier

Fig. 73: One Class Classifier Framework

The classification network is optimized using the SGD optimizer, with the learning rate equals to 10^{-3} and the batch size equals to 64. μ equals 0 and σ equals 3 for the Gaussian distribution to generate pseudo-positive data.

Feature extractor

In the proposed anomaly detection model, ten features are designed to describe normal driving behavior, including both longitudinal and lateral behaviors. The designed features are elaborated as follows:

(1) Average lateral acceleration: $f_1 = \frac{1}{N}\sum_i^N |a_i \sin\psi_i|$. N is the trajectory length. ψ_i is the vehicle heading at time step i. f_1 calculates the average of lateral acceleration at each time step.

(2) Maximum lateral acceleration: $f_2 = \max_{i=1,...N} |a_i \sin\psi_i|$. f_2 is the max value of the lateral acceleration for the entire trajectory. f_1 and f_2 measure the smoothness of the lateral driving behavior.

(3) Average lateral speed: $f_3 = \frac{1}{N}\sum_i^N |v_i \sin\psi_i|$. f_3 calculates the average lateral speed.

(4) Maximum lateral speed: $f_4 = \max_{i=1,...N} |a_i \sin\psi_i|$. f_4 is the maximum lateral speed. f_3 and f_4 denotes the vehicle's lateral driving behavior.

(5) Average longitudinal acceleration: $f_5 = \frac{1}{N}\sum_i^N |a_i \cos\psi_i|$. f_5 calculates the average longitudinal acceleration.

(6) Maximum longitudinal acceleration: $f_6 = \max_{i=1,...N} |a_i \sin\psi_i|$. f_6 calculates the maximum longitudinal acceleration. f_5 and f_6 represent the smoothness of the longitudinal driving behavior.

(7) Average longitudinal speed: $f_7 = \frac{1}{N}\sum_i^N |v_i \cos\psi_i|$. f_7 is the average longitudinal speed.

(8) Maximum longitudinal speed: $f_8 = \max_{i=1,...N} |a_i \cos\psi_i|$. f_7 and f_8 denote the driver's longitudinal driving efficiency at the roundabout.

(9) Maximum heading rate: $f_9 = \max_{i=1,...N} \left|\dot{\psi}_i\right|$. $\dot{\psi}_i$ denotes vehicle heading change rate at time step i.

(10) Average heading rate: $f_{10} = \frac{1}{N}\sum_i^N \left|\dot{\psi}_i\right|$. f_9 and f_{10} demonstrates vehicle's driving smoothness at the roundabout.

Note that for different driving scenarios, the features may be designed differently. A Greedy Algorithm is applied to select critical features from the designed feature list.

Algorithm denotes the greedy algorithm that selects the critical features used for the one-class classification neural network. S denotes the selected feature set. S is initiated as an empty set at the beginning of the algorithm. NS denotes the feature set which is not selected. A^* denotes the highest accuracy, A^* is initiated with 0. While NS is not an empty set, the one class classifier is trained with $S \cup f_i$, where $f_i \in NS$. Testing accuracy is calculated and denoted as A_i. The algorithm loops through all features in NS and selects the feature that maximizes testing accuracy denoted as f^* and the relative testing accuracy is denoted as a^*. If $a^* > A^*$, then S will be updated by adding f^* into it, and f^* will be extracted from NS and A^* will be updated as a^*. If no improvement is made by adding feature $f_i \in NS$, the iteration stops and returns with S, which is the feature set that achieves the highest testing accuracy. f_1, f_2, f_3, f_5, f_9 are selected and used as feature extractor. The selected features can describe both longitudinal and lateral driving behaviors.

Algorithm 1 Greedy Algorithm

```
$S \leftarrow \emptyset$
$NS \leftarrow \{f_1, f_2, ...f_{10}\}$
$A^* = 0$
while $NS \neq \emptyset$ do
    for $f_i \in NS$ do
        Calculate the One class classifier accuracy using features $S \cup f_i$, the accuracy on testing set is denoted as $A(f_i)$
    end for
    $f^* = argmax_{f_i} A(f_i)$
    $a^* = A(f^*)$
    if $a^* > A^*$ then
        $S = S \cup f^*$
        $A^* = a^*$
        $NS = NS \setminus f^*$
    else
        Break
    end if
end while
return $S$
```

EXPERIMENTS

To validate the proposed anomaly detection framework, a roundabout data set collected at Ann Arbor, Michigan is applied. The data set is collected at a two-lane roundabout. The roundabout is equipped with infrastructure sensors such as radars and cameras are installed at the four corners of the roundabout. Vehicle trajectories approaching and within the roundabout are extracted from the video data with a time step equal to 0.4s [350].

This roundabout is of high-interest because of its high crash rates. 69 crashes happened at the intersection in year 2021. Among them, 66 crashes happened between the vehicle travelling inside the roundabout and the vehicle at the entry [359]. One possible solution to reduce the crash counts is to apply the IMA. However, if the IMA application is under cyber attack, generating fake warnings and/or canceling true warnings may even aggravate the crash risks.

In this section, we first show the results of the attack model and then evaluate the anomaly detection framework with the threat model.

Attack model results

In the experiment, qualified vehicle trajectory pairs in the roundabout data set are extracted and used to generate attack trajectories. Each trajectory pair consists of the vehicle traveling within the roundabout and an entry vehicle. A qualified vehicle pair must have a similar arrival time to the defined conflict point so that the driver of the entry vehicle would actually observe a real approaching vehicle in the roundabout. In this way, when the IMA warning is triggered, the driver of the entry vehicle may take real actions to avoid the (fake) conflict. Based on this criteria, a total number of 927 vehicle pairs are selected.

Using the attack model illustrated in section III, 744 attack trajectories are generated. Figure 74 shows an example of the attack trajectory. The red line represents the original vehicle trajectory. The green line represents the vehicle trajectory generated by the attack model. The black line denotes the victim vehicle trajectory. In this case, the average vehicle speed at the roundabout is around $7m/s$, which is consistent with the speed limit in the round about (15mph). At the end of the green vehicle trajectory, the IMA warning is triggered. The result shows that the proposed algorithm can generate falsified lane changing trajectory that follows roundabout's geometry to trigger the IMA warning of the entry vehicle.

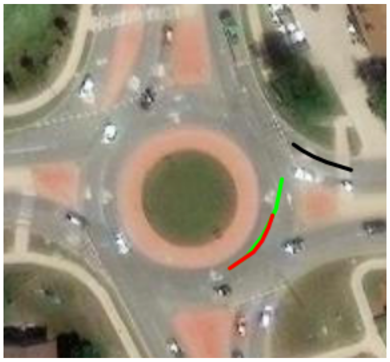


Fig. 74: Attack Trajectory at the Roundabout

The overall attack success rate is 77.970% with the average attack success time of 2.096s. The attack success time is calculated as the difference between the attack start time and the time when the IMA warning of the victim vehicle is triggered. Given that the frequency of the trajectory data is 2.5HZ, the average attack succeeds at around the fifth time step.

One explanation for the attack failure is the vehicle speed variation before entering the roundabout. Figure 75 shows an example of the speed profile of an entry vehicle (victim vehicle). The proposed model fails to generate an attack trajectory to trigger the IMA warning in this case. The entry vehicle accelerates from $2m/s$ to $7m/s$ in two seconds and the estimated arrival time changes from 6.23s to 0.33s. Even though the estimated arrival time is updated every 0.4s, the large variation makes it impossible for the attack trajectory to reach the conflict point in time, without violating vehicle dynamic constraints.

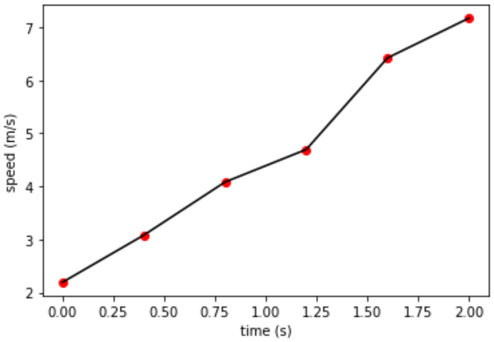


Fig. 75: Attack Failure example at Roundabout

Detection framework evaluation

To evaluate the detection framework, 2564 ground truth trajectories from the data set are extracted. 490 attack trajectories are generated with the proposed attack model. 40% of the ground truth are used to train the one-class classifier and the rest 60% are used for testing. All of the attack trajectories are used for testing. Both offline and online detection are conducted and the results are presented as follows.

In the offline mode, the detection is not performed until the full trajectory is observed. 99.59% (488/490) of the attack trajectory and 99.48% (1531/1539) of the ground truth trajectory are identified correctly. A false positive means that a ground truth trajectory is classified as an abnormal trajectory while a false negative means that an abnormal trajectory is identified as a normal trajectory. The false positive rate and false negative rate for the offline detection is 0.52% and 0.40% respectively. Figure 76 shows a false negative case in which the attack succeeds within only one time step. The short attack success time leads to little information can be used for detection. Therefore, the trajectory is not identified correctly.

The online detection is more important in real-world implementations. The detection starts after sufficient number of trajectory points (e.g., 3 data points) and is conducted every time step until the trajectory is identified as abnormal or the attack succeeds. The trajectory will be identified as abnormal if it is classified as abnormal in two consecutive time steps. Therefore, trajectories

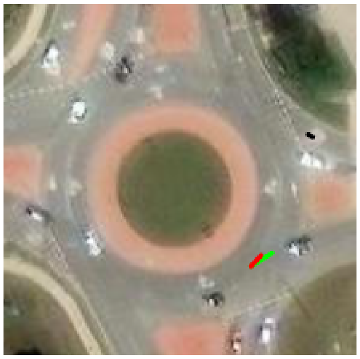


Fig. 76: Misclassification example (FN case)

TABLE XIII: Performance of online detection

FP	**FN**	**Mean attack success time (s)**	**Mean detection time (s)**	**Mean time to attack success (s)**
14/1539 (0.91%)	0/314 (0%)	2.096	1.600	0.497

used for online detection should be at least 5 time steps long. 314 attack trajectories and 1539 ground truth trajectories are used to test the online detection. The online detection performance is shown in Table XIII. The mean detection time is the elapsed time when the trajectory is identified as abnormal. The time to attack succeed is the difference between the attack success time and the detection time. Results show that the anomaly classifier can identify attack trajectories 0.49s before attack succeed in average, with a standard deviation equals to 0.22s.

Figure 77 shows a false positive case in the online detection. The attack trajectory speed profile shows that the vehicle's speed fluctuates between $1m/s$ to $7m/s$ in 1.2s. The average speed of the vehicle is $5.8m/s$. The large fluctuation under low travel speed are rare in the roundabout data set. As a result, the trajectory is classified as abnormal. A possible reason is due to the error in the trajectory processing. Even this trajectory is not under attack, its erroneous behavior indicates that it is not a normal trajectory and further attention is needed to identify the root cause.

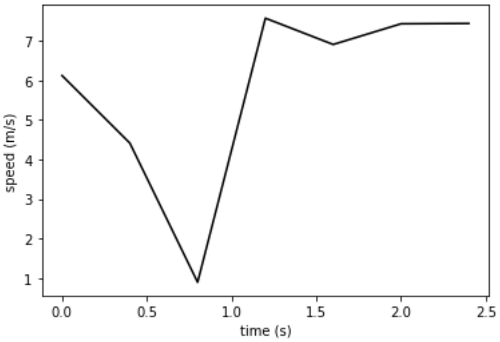


Fig. 77: Misclassification example (FP case)

CONCLUSIONS AND DISCUSSIONS

In this paper, a GPS spoofing attack model towards the CV IMA application and an anomaly detection framework to detect such attacks using one class classification is introduced. An optimization model is formulated and served as the threat model which aims to trigger the IMA warning of the entry vehicle at the roundabout. Both models are evaluated with a real world data set and the results show that the threat model can generate falsified trajectory and trigger the IMA warning within a short time, which is very aggressive and raises challenges to the detection model. However, the detection result shows satisfactory performance that most of the abnormal trajectories can be identified correctly and in time.

Comparing with previous work on GPS spoofing attack, the proposed attack model in this paper is much more aggressive. For example, the average attack success time using algorithm proposed by [344] is 28.7s. In our work, the attack success time is only around 1.7s, which leaves little time for the detection model. In addition, the proposed detection model is more generic. Comparing with our previous work [349] which uses both ground truth trajectories and attack trajectories in the training process, the proposed detection framework based on one class classification only need ground truth trajectories for training, which makes it applicable to detect unknown attacks. Besides, the proposed anomaly detection model can be applied to multiple scenarios including IMA warning at the roundabout and unsignalized intersection, as well as Red Light Violation Warning (RLVW) at signalized intersections, since the proposed model focus on learning the normal driving behaviors. As long as the normal driving behavior is affected, the proposed method can be applied to detect the anomaly.

5e GPS Spoofing Attack Detection on Intersection Movement Assist using One-Class Classification

ABSTRACT

Modern Positioning, Navigation, and Timing (PNT) systems heavily rely on Global Navigation Satellite Systems (GNSS). Meanwhile, GNSS-based PNT systems are increasingly becoming susceptible to unintentional and deliberate Radio Frequency (RF) interference. In particular, as technology keeps advancing and hardware is becoming so inexpensive, it takes a modest effort to disrupt the normal operation of almost any PNT systems, thus posing an extreme threat to autonomous transportation systems that rely on precise PNT. As communication capabilities are expanding, a group of vehicles can easily share data when they operate in close vicinity. This gives opportunity to position and navigate the vehicles based on a jointly computed navigation solution, which is usually called collaborative navigation, resulting in a potentially more accurate and reliable operation. In this study, the feasibility and performance potential of collaborative navigation on the detection and mitigation of GNSS-based PNT system operational anomalies are evaluated on some real data and simulated anomaly scenarios. By incorporating an outlier detection method based on least squares adjustment, the collaborative navigation has shown to be able to maintain the differences to the reference solution to within m, m, and m for the biased case, noisy case, and anchor case, respectively, for all test vehicles.

INTRODUCTION

Modern Positioning, Navigation, and Timing (PNT) systems heavily rely on Global Navigation Satellite Systems (GNSS). Meanwhile, GNSS-based PNT systems are increasingly becoming susceptible to unintentional and deliberate Radio Frequency (RF) interference. In particular, as technology keeps advancing and hardware is becoming so inexpensive, it takes a modest effort to disrupt the normal operation of almost any PNT systems, thus posing an extreme threat to autonomous transportation systems that rely on precise PNT.

Finding protection against any interference to PNT systems is happening on multiple levels. Modernization of Global Positioning System (GPS) has introduced new signals to provide significant capabilities to increase protection at signal and receiver level. In parallel, the proliferation of GNSS systems in the past decade has substantially increased the signal availability, so PNT systems using all the potentially available signals can exploit the benefits of redundancy. Nevertheless, there are limits on what can be done against RF interference in PNT systems, and therefore, using totally independent sensor technologies is mandatory to detect malfunctioning and potentially offering mitigation to some extent.

As communication capabilities are expanding, a group of vehicles can easily share data when they operate in close vicinity. This gives opportunity to position and navigate the vehicles based on a jointly computed navigation solution, resulting in a potentially more accurate and reliable operation. This method, usually called collaborative navigation (or cooperative navigation), is considered here for detecting any malfunctioning of a GNSS-based PNT system, such as hardware problem, jamming or spoofing.

RF interference and mitigation

RF interference has become a serious threat to the GNSS navigation systems. Even though the GNSS signals have been designed to withstand a certain level of interference through Direct Sequence Spread Spectrum (DSSS) technique, they are so weak at the receiving end on Earth that most modern electronic equipment interferes with them at close range [360]. The open GNSS signals to civil users are especially vulnerable due to their open structure and lacking encryption and authentication [360], [361]. The jamming, either unintentional or deliberate, will block the genuine GNSS signal from being tracked. The spoofing fools the receivers with counterfeit signal to produce precise but erroneous solution [360].

The effects of simple spoofing attacks on GNSS receivers integrated in Android smartphones are investigated in [362]. A portable spoofer was developed with a Software Defined Radio (SDR) and a low cost front-end. The spoofer was placed within m from the test smartphones under open sky. It broadcast spoofing signals over GPS L1 band as if received one day ago at a different location that was km away. The carrier-to-noise density ratio (C/N_0), Automatic Gain Control (AGC), and time of signal transmission and reception from the test smartphones were analyzed. The results show that during the spoofing period, tracking of the real signals from low elevation satellites is lost; the fake signal is not acquired but the real signals of satellites with the same satellite vehicle identification number as the fake satellite is lost too. The position outputs deviate a few meters during spoofing that is blamed to loss of lock to some GPS satellites. Actually, the test smartphone uses a multi-constellation GNSS chip that has been claimed to be able to reach cm accuracy in open environment [363].

Varying methods to protect GNSS receivers from jamming and interference are summarized in [361] into four main categories: inertial aiding, spatial filtering, time-frequency filtering, and vector tracking. Deep integration of Inertial Navigation System (INS) and GNSS will improve jamming-to-signal (J/S) ratio, system accuracy, and high dynamic performance. Spatial filtering uses antenna arrays to point the receiver antenna beam towards the GNSS satellites and away from jammers. Time-frequency filtering methods are based on GNSS signal conditioning and filtering. Vector tracking achieves enhanced tracking robustness under degraded conditions by utilizing the fact that signal channels are coupled through the shared receiver states of position, velocity, and time. Inertial aiding and vector tracking improve the receiver robustness by lowering the minimum required C/N_0 level for receiver acquisition and tracking; whereas, the incident interfering signals are suppressed before entering the receiver in the spatial and time-frequency filtering approaches [361].

Collaborative navigation

Collaborative Navigation entails a concept of a group of platforms, referred to as network nodes, navigating collectively (as a network) and supporting each other's positioning solution to obtain higher accuracy and availability for all platforms. Early works have been focused on integrating the inter-nodal range/bearing measurements or locally generated maps [364]–[366]. In fact, a similar concept, community relative navigation, was studied before GPS was established, and can be traced back at least to the 1970s.

Recent technology advancements have made the inter-nodal measurements, especially the range measurements, more accurate and affordable than before. Inter-nodal range can be measured with a radio signal in a wireless sensor network through RSS (Received Signal Strength), TOA (Time of Arrival), TDOA (Time Difference of Arrival), and AOA (Angle of Arrival) techniques, or with optical sensors through computer vision techniques [366]. Among them, Ultra-Wide Band (UWB) ranging technology, with a broad bandwidth available for time transfer and centimeter level ranging capability, is of particular interest to positioning and navigation applications [367]. Moreover, the UWB transceivers mounted on the nodes form an ad-hoc network that can provide a datalink among the nodes without any additional infrastructure and aid the INS/GNSS solution by giving accurate inter-nodal range measurements [368].

The benefits of cooperative navigation have been demonstrated through simulation in [366]. The result shows that, for repeated 60-second GPS gaps, separated by 10-second signal availability, cooperative navigation can maintain the accuracy at m level for nodes only equipped with consumer grade Inertial Measurement Unit (IMU). In the simulation, a decentralized Extended Kalman Filter (EKF) was used to integrate all the measurements. Similar research is presented for cooperative navigation concept verification prototype in [368], in which all nodes were equipped with GPS, Micro-Electro-Mechanical Systems (MEMS) inertial sensors, barometric altimeter, and UWB transceiver. The results show that with UWB aiding, a horizontal position accuracy of $\pm\mathrm{m}$, horizontal velocity accuracy of $\pm\mathrm{m/s}$, and orientation accuracy of $\pm^\circ$ for roll and pitch and $\pm^\circ$ for heading were obtained for the mobile node during several second GPS outages [368].

An outlier detection algorithm for collaborative navigation is presented in [369]. The algorithm models the GNSS measurements and inter vehicle range measurements into common and specific parts and excludes the faulty measurements with a greedy search strategy. The test results show the algorithm has a better detection of GNSS faults than tradition Receiver Autonomous Integrity Monitoring (RAIM) and good sensitivity to the faulty UWB range measurements.

Contributions

This study investigates the detection and mitigation of GNSS-based PNT system operational anomalies on individual vehicles in the context of collaborative navigation. The contributions can be summarized as follows.

1) Establishment of a framework for collaborative navigation to integrate individual vehicle's GNSS or INS solutions, inter-vehicle ranges, and ranges to infrastructures.
2) Demonstration of the effectiveness of collaborative navigation in detecting and mitigating PNT system operational anomalies on individual vehicles in post processing mode.
3) Utilization of a least squares adjustment based method for outlier detection in collaborative navigation.
4) Analysis of the theoretic limitation of the "anchor" concept that relies on the navigation solution of an anchor vehicle and ranges to other vehicles.

METHODOLOGY

This study adopts centralized EKF to integrate GNSS or INS solutions of individual vehicles and range measurements among vehicles and from a vehicle to beacons along the road. A least squares adjustment based method is used to detect outliers in GNSS or INS solutions of individual vehicles before the EKF measurement update.

Collaborative navigation based on range measurements

The range measurement is a function of the positions of the vehicles involved. After linearization, it has the following form.

$$r_{ij,t} - r^{0}_{ij,t} = [\vec{J}, -\vec{J}][\vec{x_i}, \vec{x_j}]^T + e_{ij,t}, \tag{76}$$

where $r_{ij,t}$ is the range between vehicle i and j at epoch t, $r^{0}_{ij,t}$ is the computed range from the approximate coordinates, $\vec{J}$ is the Jacobian coefficient vector, $[\vec{x_i}, \vec{x_j}]$ are (unknown) correction vector to the approximate coordinates of the ith and jth platforms, and $e_{ij,t}$ is the error term.

The collaborative navigation can be implemented in a central Extended Kalman Filter (EKF), in which the communication and range measurements between the vehicles are assumed. The centralized architecture allows near-optimal behaviors in well-understood environments. The state model can be described as follows.

$$\vec{x_k} = \Phi_{k,k-1}\vec{x_{k-1}} + G_k\vec{w_k}, \quad \vec{w_k} \sim \mathcal{N}(\vec{0}, Q_k), \tag{77}$$

where $\vec{x_k}$ is the state vector, $\Phi_{k,k-1}$ is the state transition matrix, $\vec{w_k}$ is a Gaussian, zero-mean, white noise vector with a covariance matrix of Q_k, k and $k-1$ are time instants.

The linearized observation model is as follows.

$$\vec{y_k} = H_k\vec{x_k} + \vec{v_k}, \quad \vec{v_k} \sim \mathcal{N}(\vec{0}, R_k), \tag{78}$$

where the observation vector $\vec{y_k}$ may include the GNSS or INS solutions of individual vehicles as well as Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) ranges, R_k is the variance of the observation error vector $\vec{v_k}$.

Following [370], the EKF solution can be expressed in the following equations.

$$\hat{\vec{x}}_k^- = \Phi_{k,k-1}\hat{\vec{x}}_{k-1}, \tag{79}$$
$$P_k^- = \Phi_{k,k-1}P_{k-1}\Phi_{k,k-1}^T + G_kQ_kG_k^T, \tag{80}$$
$$\vec{\gamma_k} = \vec{y_k} - H_k\hat{\vec{x}}_k^-, \tag{81}$$
$$K_k = P_k^-H_k^T(H_kP_k^-H_k^T + R_k)^{-1}, \tag{82}$$
$$\hat{\vec{x}}_k = \hat{\vec{x}}_k^- + K_k\vec{\gamma_k}, \tag{83}$$
$$P_k = (I_n - K_kH_k)P_k^-(I_n - K_kH_k)^T + K_kR_kK_k^T, \tag{84}$$

where $\hat{\vec{x}}_k^-$ is the a priori expected value of the state vector, P_k^- is the error covariance of the expected state, $\vec{\gamma_k}$ is the innovation vector, K_k is the Kalman Gain matrix, $\hat{\vec{x}}_k$ is the a posteriori estimate of the state after the measurement update, and P_k is the a posteriori covariance of the state vector.

Least squares adjustment based detection method

Generally, the innovation vector as in equation (81) in EKF is exploited to detect the measurement faults, as the normalized-innovation-squared test statistic employed in [369], [371], [372]. Meanwhile, it is found in the study that the innovation based method has higher false alarm rate for the same detection threshold. Hence, least squares adjustment is adopted to detect any anomalies in individual vehicle's GNSS or INS solutions in collaborative navigation.

There is a rank deficiency problem in least squares adjustment based on range measurements only. As an example, to localize a quadrilateral from lengths among its corners on a plane, the rank deficiency is three. There are some methods to solve the rank deficiency problem. A straightforward one would be reducing the number of unknown parameters, that is, treating the GNSS or INS position of some vehicles as known and removing the corresponding corrections from the parameter list. This method will be used in the anchor case later for cross validation. The issue with the method is there will be no chance to detect any errors in those GNSS or INS positions in collaborative navigation.

Among some other methods, Minimum Norm Least Squares Solution (MINOLESS) is selected in this study because the solution norm is minimized among all possible (biased) solutions. The adjustment method can be described with the following equations [373].

$$\vec{y} = \underset{n\times m}{A}\xi + \vec{e}, \vec{e} \sim (\vec{0}, \sigma_0^2 P^{-1}), rk(A) < \{m, n\}, \tag{85}$$

$$\underset{n\times m}{E}\vec{\xi} = \vec{0}, \text{with } AE^T = 0, R(A^T) \overset{\perp}{\oplus} R(E^T) = R^m, \tag{86}$$

$$N = A^T P A, \tag{87}$$

$$\vec{c} = A^T P \vec{y}, \tag{88}$$

$$\hat{\vec{\xi}} = (N + E^T E)^{-1}\vec{c}, \tag{89}$$

$$D\{\hat{\vec{\xi}}\} = {\sigma_0}^2 (N + E^T E)^{-1} N (N + E^T E)^{-1}, \tag{90}$$

where the rank of the coefficient matrix A is less than its dimension (n, m). With the introduction of "inner datum constraints" in equation (86), the columns of E^T form a basis for the nullspace of A, and the orthogonal direct sum of rangespaces of A^T and E^T is in the m-dimensional real space R^m. Equation (89) and (90) represent the estimate of parameters and its covariance, respectively.

Locations derived from the least squares adjustment are then compared to GNSS or INS solution of individual vehicles against a threshold to detect any anomalies. The detected faulty GNSS or INS solution can then be downweighted or simply excluded from participating in the measurement update of EKF. Without the effects of faulty GNSS or INS solutions of individual vehicles, the collaborative navigation will be more accurate and reliable.

TEST DATASET

A real dataset was collected at a parking lot of The Ohio State University and included four vehicles. The anchor vehicle A was equipped with GNSS receivers, high-end and MEMS IMU sensors, cameras, laser scanners, as well as UWB transmitters to measure the ranges to other vehicles and UWB beacons along the road. Three other vehicles, B, C, and D, were equipped with GNSS receivers and multiple UWB transmitters. The GNSS receivers were operating in standard point positioning mode. Two networks of UWB devices independently measured the V2V ranges using different channels. UWB transmitters on the right side of the four vehicles formed Network 1 (NET1), whereas the left side sensors formed Network 2 (NET2). Ten UWB beacons were deployed along the road to enable the range measurements from vehicle A to get the V2I range measurements. The UWB beacons were regarded as the infrastructure of the road and their locations had been accurately surveyed. The vehicles were driven in formation in the test with most of time A and B being side by side and in the front, followed by C and D. Fig. 78 and Fig. 79 show the scene of data collection and a trajectory of the anchor vehicle, respectively. More information about the data collection campaign can be found in [374].

A period of 42.6 seconds (214 epochs) of data were selected for this study. As the UWB success rate was below 50%, V2V ranges were simulated from GNSS solutions with a standard deviation of m. For the same reason, V2I ranges were also simulated at the same level of accuracy from the known locations of UWB beacons and vehicle Aâ€™s GNSS solutions.

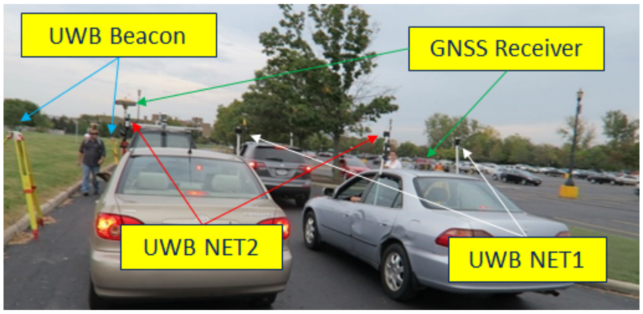


Fig. 78: The scene of data collection.

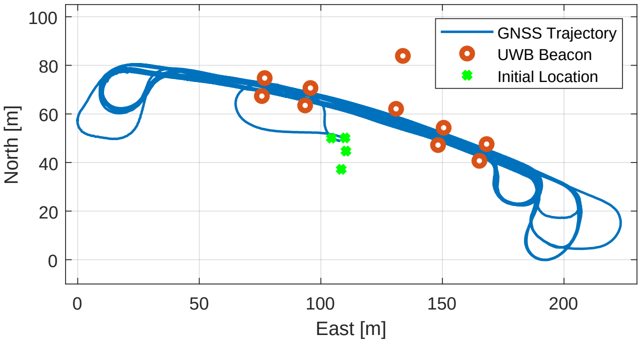


Fig. 79: GNSS trajectory of vehicle A, UWB beacon locations, and the initial locations of the four vehicles.

TEST RESULTS

Centralized integration architecture is adopted for this study by assuming the GNSS solution of four vehicles as well as the V2V and V2I range measurements are available to a central EKF without any communication delay. GNSS solutions of four vehicles, V2I ranges from A, and V2V ranges among four vehicles are the measurements for the EKF. The state vector includes two dimensional positions, velocities, and accelerations of these four vehicles.

The least squares adjustment based detection method as described above is used for the fault detection of GNSS solutions of individual vehicles.

Collaborative navigation

The collaborative navigation results are depicted in Fig. 80 and 81 and are used as reference for the test scenarios discussed below. Since the GNSS solutions are from standard point positioning, the differences between collaborative navigation and GNSS solutions are at sub-meter level as shown in table XIV.

TABLE XIV: Statistics of norms of differences between collaborative navigation and GNSS solution [m]

Vehicle	Min	Max	Mean	Std.
A	0.14	0.74	0.58	0.07
B	0.16	0.72	0.58	0.06
C	0.17	0.70	0.58	0.06
D	0.14	0.71	0.58	0.07

Constant bias to one vehicleâ€™s GNSS solution

As for a spoofing attack case like in [362], a constant bias of 1.0 m is added to each component of vehicle Dâ€™s GNSS solution starting from the 57th epoch for 100 epochs, and then it is used to derive the collaborative navigation with the other vehiclesâ€™ GNSS solution as well as V2I and V2V ranges. As seen from Fig. 82 and table XV, using the information

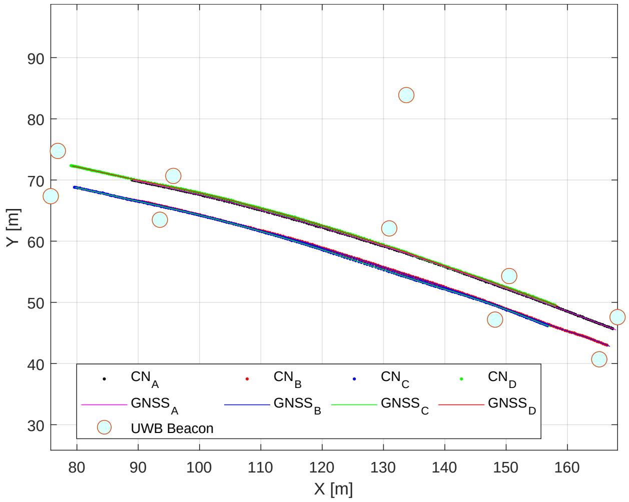


Fig. 80: Trajectories of collaborative navigation solution.

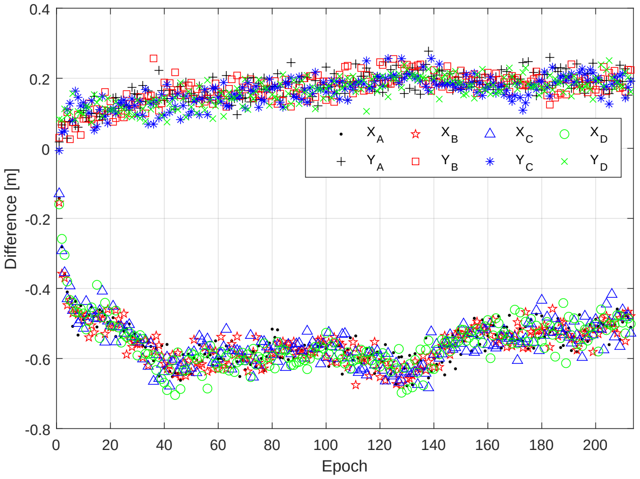


Fig. 81: Differences between collaborative navigation and GNSS solution.

of other vehicles and the range measurements, the position error of vehicle D has been restricted to less than m. After the bias-added periods, the solution gradually converged back to the reference solution.

TABLE XV: Statistics of norms of differences between collaborative navigation for biased case and the reference solution [m]

Vehicle	Min	Max	Mean	Std.
A	0.00	0.09	0.03	0.03
B	0.00	0.07	0.02	0.02
C	0.00	0.08	0.02	0.02
D	0.00	0.11	0.03	0.03

Obviously, any added biases larger than m in GNSS solutions will be detected with precise inter-vehicle range measurements.

Noisy GNSS solution of one vehicle

To investigate the situation that the GNSS receiver produces noisy positioning results due to interference, random noise with zero mean and variance of m is added to each component of the GNSS solution of vehicle D. The collaborative navigation solution for this case is presented in Fig. 83 and table XVI. The collaborative navigation clearly helps decrease the position

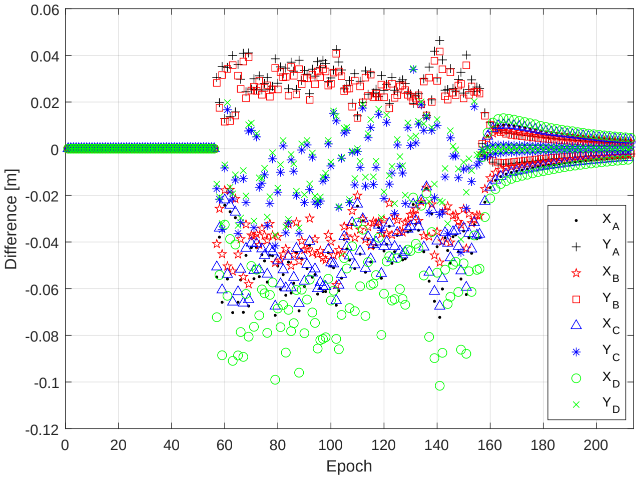


Fig. 82: Differences between collaborative navigation for the biased case and reference solution.

error for vehicle D with the maximum 2D distance to the reference solution being below m. Comparing to the result of the constant bias case above, the increased noise level creates challenges for detecting and mitigating GNSS anomalies.

In the meantime, if decreasing the weight of Dâ€™s GNSS solution for the whole duration of the affected periods before deriving the collaborative navigation, the offsets of Dâ€™s solution can be controlled around m, as shown in Fig. 84 and table XVII. Obviously, techniques in the RF domain, such as C/N_0 monitoring and received power monitoring as in [360], should be combined to detect and mitigate the anomalies for this case.

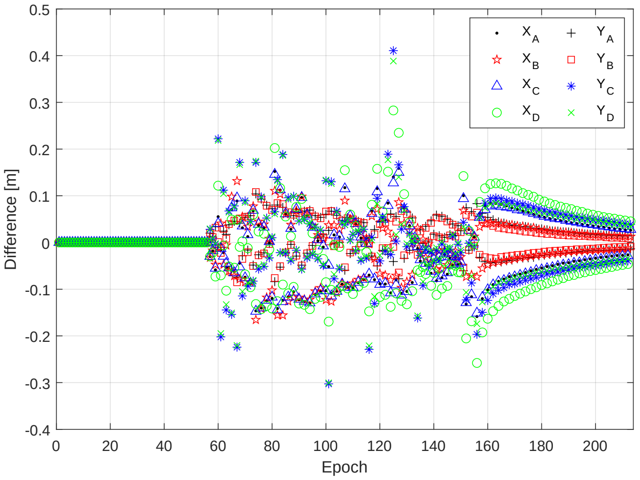


Fig. 83: Differences between collaborative navigation for the noisy case and reference solution.

TABLE XVI: Statistics of norms of differences between collaborative navigation for noisy case and the reference solution [m]

Vehicle	Min	Max	Mean	Std.
A	0.00	0.18	0.05	0.05
B	0.00	0.20	0.04	0.04
C	0.00	0.43	0.07	0.07
D	0.00	0.48	0.08	0.07

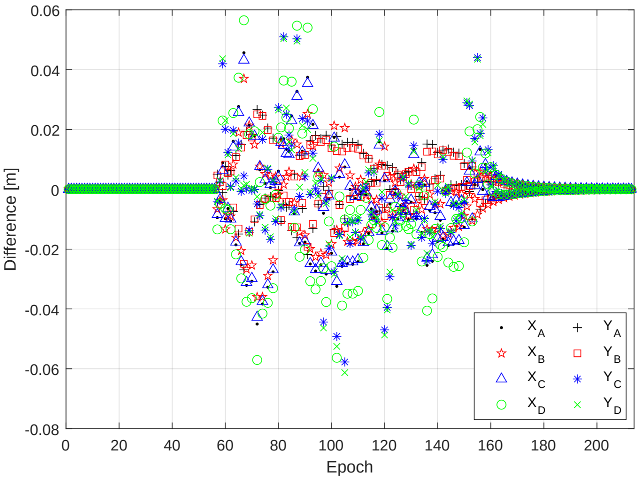


Fig. 84: Differences between solution of always-downweighting for noisy case and the reference solution.

GNSS outages for three vehicles

In the presence of strong interference, most of the GNSS receivers may lose tracking of the satellite signals and produce no positioning results. There are some high-end, interference resistant GNSS receivers, however, that may still be able to position. This case is also investigated by assuming the GNSS solution is only available on vehicle A and thus trying to derive the navigation solution for the other vehicles with the V2V range measurements. The results are plotted in Fig. 85 and tabulated in table XVIII, in which it can be found that the differences to the reference solution are less than m for B and m for C and D. It demonstrates that the collaborative navigation can generate the navigation solution for the other three vehicles during the outage periods, which is rather unreal in single platform navigation without aiding from other sensors, such as IMUs and odometers/speedometers. However, the solution differences seem to be larger than expected at first glance, considering the V2V ranges are simulated from the GNSS solutions.

For cross validation, a least squares adjustment has also been used to estimate the positions of the vehicles. In the computation, the coordinate corrections of vehicle A are excluded from the parameter list and the number of unknowns is reduced to six. Meanwhile, as explained before, the rank of the coefficient matrix A in equation (85) is still rank deficient, which makes the estimation an ill-posed problem. Accordingly, it is not a surprise to find from Fig. 86 that the maximum differences of a single coordinate component are close to m.

The difficulty in location estimation for the anchor case can also be illustrated geometrically as in Fig. 87. Consider rotating the quadrilateral $ABCD$ about its corner A to two different locations. The location of A has not changed, neither have the lengths among all four corners, but the locations of the other corners are totally different. Therefore, for this anchor case,

TABLE XVII: Statistics of norms of differences between always-downweighting solution for noisy case and the reference solution [m]

Vehicle	Min	Max	Mean	Std.
A	0.00	0.05	0.01	0.01
B	0.00	0.04	0.01	0.01
C	0.00	0.06	0.01	0.01
D	0.00	0.08	0.01	0.02

TABLE XVIII: Statistics of norms of differences between collaborative navigation for anchor case and the reference solution [m]

Vehicle	Min	Max	Mean	Std.
A	0.10	0.39	0.30	0.04
B	0.13	1.12	0.89	0.17
C	0.13	2.59	2.00	0.52
D	0.11	2.40	1.89	0.45

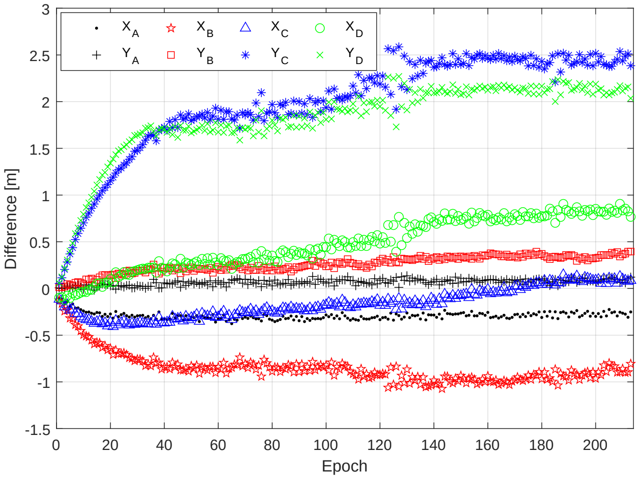


Fig. 85: Differences between collaborative navigation for anchor case and reference solution.

knowing the location of the anchor vehicle and the V2V ranges is not adequate to accurately determine the location of all vehicles.

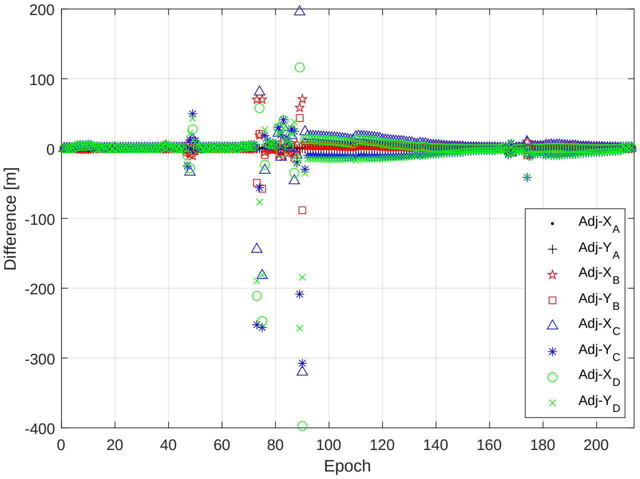


Fig. 86: Differences between adjustment for anchor case and reference solution.

CONCLUSION

The feasibility and performance potential of collaborative navigation on the detection and mitigation of GNSS-based PNT system operational anomalies are evaluated on some real data and simulated anomaly scenarios in this study. The collaborative navigation is based on the GNSS solution of four vehicles as well as ranges among vehicles and from one vehicle to beacons along the road. A least squares adjustment based method is used to detect outliers in GNSS solutions before the EKF measurement update.

The effectiveness of collaborative navigation in detecting and mitigating PNT system operational anomalies on individual vehicles are demonstrated in three simulated cases.

For a simulated spoofing attack in which one vehicle's GNSS solutions are deviated m constantly in all components, collaborative navigation can keep the solution differences to the reference solution from clean data to below m for all affected epochs.

For a simulated interference case with increased noise level, by adding white noises with a variance of m to each component of one vehicleâ€™s GNSS solution, the vehicleâ€™s total position difference to the reference solution can be contained

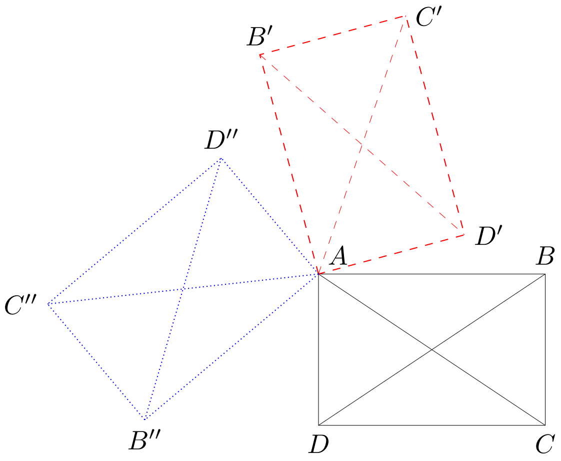


Fig. 87: Rotating a quadrilateral around one corner A wonâ€™t change the location of A and lengths to other corners.

to below m in collaborative navigation. If combining interference detection techniques from RF domain, the navigation performance could be further improved.

In the anchor case, it is simulated that all lower-end receivers lose positioning capabilities due to strong interference and the collaborative navigation has to rely on the GNSS solution of one anchor vehicle and V2V ranges. The collaborative navigation can generate the navigation solution within m differences to the reference solution for the other three vehicles during the outage periods, which is rather unreal in single platform navigation without aiding from other sensors. However, due to the inherent defects of rank deficiency, the solution cannot be accurately derived in this case.

REFERENCES

[1] John A. Volpe National Transportation Systems Center, "Vulnerability assessment of the transportation infrastructure relying on the Global Positioning System," 2001.

[2] Psiaki, M. L. and Humphreys, T. E., *Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications*, Vol. 1, chap. Civilian GNSS Spoofing, Detection, and Recovery, Wiley-IEEE, 2020, pp. 655–680.

[3] Mit, R., Zangvil, Y., and Katalan, D., "Analyzing Tesla's Level 2 Autonomous Driving System Under Different GNSS Spoofing Scenarios and Implementing Connected Services for Authentication and Reliability of GNSS Data," *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 621–646.

[4] Psiaki, M. L. and Humphreys, T. E., "GNSS Spoofing and Detection," *Proceedings of the IEEE*, Vol. 104, No. 6, 2016, pp. 1258–1270.

[5] Wesson, K. D., Gross, J. N., Humphreys, T. E., and Evans, B. L., "GNSS Signal Authentication Via Power and Distortion Monitoring," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 2, April 2018, pp. 739–754.

[6] Gross, J. N., Kilic, C., and Humphreys, T. E., "Maximum-likelihood power-distortion monitoring for GNSS-signal authentication," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 55, No. 1, 2018, pp. 469–475.

[7] Humphreys, T. E., "Interference," *Springer Handbook of Global Navigation Satellite Systems*, Springer International Publishing, 2017, pp. 469–503.

[8] Montgomergy, P. Y., Humphreys, T. E., and Ledvina, B. M., "Receiver-Autonomous Spoofing Detection: Experimental Results of a Multi-antenna Receiver Defense Against a Portable Civil GPS Spoofer," *Proceedings of the ION International Technical Meeting*, Anaheim, CA, Jan. 2009.

[9] Psiaki, M. L., O'Hanlon, B. W., Powell, S. P., Bhatti, J. A., Wesson, K. D., Humphreys, T. E., and Schofield, A., "GNSS Spoofing Detection using Two-Antenna Differential Carrier Phase," *Proceedings of the ION GNSS+ Meeting*, Institute of Navigation, Tampa, FL, 2014.

[10] Reid, T. G., Houts, S. E., Cammarata, R., Mills, G., Agarwal, S., Vora, A., and Pandey, G., "Localization Requirements for Autonomous Vehicles," *SAE International Journal of Connected and Automated Vehicles*, Vol. 2, No. 12-02-03-0012, 2019, pp. 173–190.

[11] Ye, H., Chen, Y., and Liu, M., "Tightly coupled 3d lidar inertial odometry and mapping," *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 3144–3150.

[12] Chiang, K.-W., Tsai, G.-J., Li, Y.-H., Li, Y., and El-Sheimy, N., "Navigation Engine Design for Automated Driving Using INS/GNSS/3D LiDAR-SLAM and Integrity Assessment," *Remote Sensing*, Vol. 12, No. 10, 2020, pp. 1564.

[13] Narula, L., Iannucci, P. A., and Humphreys, T. E., "All-weather sub-50-cm radar-inertial positioning," *Field Robotics*, Vol. 2, 2022, pp. 525–556.

[14] Yoder, J. E. and Humphreys, T. E., "Low-Cost Inertial Aiding for Deep-Urban Tightly-Coupled Multi-Antenna Precise GNSS," *Navigation, Journal of the Institute of Navigation*, 2022, To be published.

[15] Teunissen, P. J., *Springer Handbook of Global Navigation Satellite Systems*, chap. Carrier Phase Integer Ambiguity Resolution, Springer, 2017, pp. 661–685.

[16] Humphreys, T. E., Murrian, M. J., and Narula, L., "Deep-Urban Unaided Precise Global Navigation Satellite System Vehicle Positioning," *IEEE Intelligent Transportation Systems Magazine*, Vol. 12, No. 3, 2020, pp. 109–122.

[17] Khanafseh, S., Roshan, N., Langel, S., Cheng-Chan, F., Joerger, M., and Pervan, B., "GPS Spoofing Detection Using RAIM with INS Coupling," *Proceedings of the IEEE/ION PLANS Meeting*, May 2014.

[18] Tanıl, Ç., Khanafseh, S., and Pervan, B., "Detecting Global Navigation Satellite System Spoofing Using Inertial Sensing of Aircraft Disturbance," *Journal of Guidance, Control, and Dynamics*, 2017.

[19] Tanil, C., Khanafseh, S., Joerger, M., and Pervan, B., "An INS Monitor to Detect GNSS Spoofers Capable of Tracking Vehicle Position," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 1, Feb. 2018, pp. 131–143.

[20] Tanil, C., Jimenez, P. M., Raveloharison, M., Kujur, B., Khanafseh, S., and Pervan, B., "Experimental validation of INS monitor against GNSS spoofing," *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, 2018, pp. 2923–2937.

[21] Tanil, C., Khanafseh, S., Joerger, M., and Pervan, B., "Sequential integrity monitoring for Kalman filter innovations-based detectors," *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, 2018, pp. 2440–2455.

[22] Kujur, B., Khanafseh, S., and Pervan, B., "A Solution Separation Monitor using INS for Detecting GNSS Spoofing," *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 3210–3226.

[23] Liu, Y., Li, S., Fu, Q., Liu, Z., and Zhou, Q., "Analysis of Kalman filter innovation-based GNSS spoofing detection method for INS/GNSS integrated navigation system," *IEEE Sensors Journal*, Vol. 19, No. 13, 2019, pp. 5167–5178.

[24] Curran, J. T. and Broumendan, A., "On the use of low-cost IMUs for GNSS spoofing detection in vehicular applications," *Proc. ITSNT*, 2017, pp. 1–8.

[25] Psiaki, M. L. and Mohiuddin, S., "Relative navigation of high-altitude spacecraft using dual-frequency civilian CDGPS," *Proceedings of the ION GNSS Meeting*, 2005, pp. 1191–1207.

[26] Psiaki, M. and Mohiuddin, S., "Global Positioning System Integer Ambiguity Resolution Using Factorized Least-Squares Techniques," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, March-April 2007, pp. 346–356.

[27] Mohiuddin, S. and Psiaki, M. L., "High-Altitude Satellite Relative Navigation Using Carrier-Phase Differential Global Positioning System Techniques," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, Sept.-Oct. 2007, pp. 1628–1639.

[28] Narula, L., LaChapelle, D. M., Murrian, M. J., Wooten, J. M., Humphreys, T. E., Lacambre, J.-B., de Toldi, E., and Morvant, G., "TEX-CUP: The University of Texas Challenge for Urban Positioning," *Proceedings of the IEEE/ION PLANSx Meeting*, 2020.

[29] Humphreys, T. E., Ledvina, B. M., Psiaki, M. L., and Kintner, Jr., P. M., "GNSS Receiver Implementation on a DSP: Status, Challenges, and Prospects," *Proceedings of the ION GNSS Meeting*, Institute of Navigation, Fort Worth, TX, 2006, pp. 2370–2382.

[30] Lightsey, E. G., Humphreys, T. E., Bhatti, J. A., Joplin, A. J., O'Hanlon, B. W., and Powell, S. P., "Demonstration of a Space Capable Miniature Dual Frequency GNSS Receiver," *Navigation*, Vol. 61, No. 1, Mar. 2014, pp. 53–64.

[31] Humphreys, T. E., Bhatti, J., Pany, T., Ledvina, B., and O'Hanlon, B., "Exploiting multicore technology in software-defined GNSS receivers," *Proceedings of the ION GNSS Meeting*, Institute of Navigation, Savannah, GA, 2009, pp. 326–338.

[32] Yoder, J. E., Iannucci, P. A., Narula, L., and Humphreys, T. E., "Multi-Antenna Vision-and-Inertial-Aided CDGNSS for Micro Aerial Vehicle Pose Estimation," *Proceedings of the ION GNSS+ Meeting*, Online, 2020, pp. 2281–2298.

[33] Clements, Z., Iannucci, P. A., Humphreys, T. E., and Pany, T., "Optimized Bit-Packing for Bit-Wise Software-Defined GNSS Radio," *Proceedings of the ION GNSS+ Meeting*, St. Louis, MO, 2021.

[34] Kerns, A. J., Shepard, D. P., Bhatti, J. A., and Humphreys, T. E., "Unmanned Aircraft Capture and Control Via GPS Spoofing," *Journal of Field Robotics*, Vol. 31, No. 4, 2014, pp. 617–636.

[35] Shepard, D. P., Bhatti, J. A., and Humphreys, T. E., "Drone Hack: Spoofing Attack Demonstration on a Civilian Unmanned Aerial Vehicle," *GPS World*, Aug. 2012.

[36] Bhatti, J. and Humphreys, T. E., "Hostile control of ships via false GPS signals: Demonstration and detection," *Navigation*, Vol. 64, No. 1, 2017, pp. 51–66.

[37] Humphreys, T. E., Shepard, D. P., Bhatti, J. A., and Wesson, K. D., "A Testbed for Developing and Evaluating GNSS Signal Authentication Techniques," *Proceedings of the International Symposium on Certification of GNSS Systems and Services (CERGAL)*, Dresden, Germany, July 2014.

[38] Humphreys, T. E., Bhatti, J. A., Shepard, D. P., and Wesson, K. D., "The Texas Spoofing Test Battery: Toward a Standard for Evaluating GNSS Signal Authentication Techniques," *Proceedings of the ION GNSS Meeting*, 2012.

[39] Laboratory, T. R., "Texas Spoofing Test Battery (TEXBAT)," July 2017, http://radionavlab.ae.utexas.edu/texbat.

[40] SAE On-Road Automated Vehicle Standards Committee and others, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," *SAE International: Warrendale, PA, USA*, 2021.

[41] Armen Hareyan, "Baidu To Operate 3,000 Driverless Apollo Go Robotaxis in 30 Cities in 3 Years," https://www.torquenews.com/1/baidu-operate-3000-driverless-apollo-go-robotaxies-30-cities-3-years.

[42] "Waymo has launched its commercial self-driving service in Phoenix - and it's called 'Waymo One'," https://www.businessinsider.com/waymo-one-driverless-car-service-launches-in-phoenix-arizona-2018-12.

[43] Tanner Brown, "Baidu Has Been Working on Autonomous Vehicles. It Got the Green Light for Commercial Self-Driving Bus Service in China," https://www.barrons.com/articles/baidu-gets-green-light-for-commercial-self-driving-bus-service-in-china-51618390800.

[44] Ioanna Lykiardopoulou, "Britain's first self-driving shuttle bus hits the streets, but scares passengers away," https://thenextweb.com/news/uk-first-self-driving-shuttle-bus-scares-passengers-away.

[45] "UPS joins race for future of delivery services by investing in self-driving trucks," https://abcnews.go.com/Business/ups-joins-race-future-delivery-services-investing-driving/story?id=65014414.

[46] Jerry Hirsch, "Aurora Expands Autonomous Trucking Tests in Texas," https://www.ttnews.com/articles/aurora-expands-autonomous-trucking-tests-texas.

[47] Kirsten Korosec, "Waymo's driverless taxi service can now be accessed on Google Maps," https://techcrunch.com/2021/06/03/waymos-driverless-taxi-service-can-now-be-accessed-on-google-maps/.

[48] Kim Lyons, "Cruise gets permit from California to provide passenger test rides in driverless vehicles," https://www.theverge.com/2021/6/5/22520227/cruise-permit-california-driverless-autonomous-vehicles.

[49] Kim Lyons, "Chinese startup Pony.ai gets approval to test driverless vehicles in California," https://www.theverge.com/2021/5/22/22449084/chinese-startup-pony-ai-autonomous-vehicles-california.

[50] Levinson, J., Montemerlo, M., and Thrun, S., "Map-Based Precision Vehicle Localization in Urban Environments." *Robotics: Science and Systems*, Vol. 4, Citeseer, 2007, p. 1.

[51] "Report On Road User Needs And Requirements," Tech. rep., European GNSS Agency, 2019.

[52] Wan, G., Yang, X., Cai, R., Li, H., Zhou, Y., Wang, H., and Song, S., "Robust and Precise Vehicle Localization based on Multi-Sensor Fusion in Diverse City Scenes," *ICRA*, IEEE, 2018, pp. 4670–4677.

[53] Gao, Y., Liu, S., Atia, M., and Noureldin, A., "INS/GPS/LiDAR Integrated Navigation System for Urban and Indoor Environments Using Hybrid Scan Matching Algorithm," *Sensors*, Vol. 15, No. 9, 2015.

[54] Soloviev, A., "Tight Coupling of GPS, Laser Scanner, and Inertial Measurements for Navigation in Urban Environments," *IEEE/ION Position, Location and Navigation Symposium*, IEEE, 2008.

[55] "Self-Driving Fundamentals: Featuring Apollo," https://www.udacity.com/course/self-driving-car-fundamentals-featuring-apollo--ud0419.

[56] "Self-Driving Car Engineer Nanodegree," https://www.udacity.com/course/self-driving-car-engineer-nanodegree--nd013.

[57] "State Estimation and Localization for Self-Driving Cars," https://www.coursera.org/learn/state-estimation-localization-self-driving-cars.

[58] Shen, J., Won, J. Y., Chen, Z., and Chen, Q. A., "Drift with Devil: Security of Multi-Sensor Fusion based Localization in High-Level Autonomous Driving under GPS Spoofing," *USENIX Security*, 2020.

[59] "Tesla Model S and Model 3 Vulnerable to GNSS Spoofing Attacks," https://www.gpsworld.com/tesla-model-s-and-model-3-vulnerable-to-gnss-spoofing-attacks/.

[60] Humphreys, T. E. and Psiaki, M. L., "GNSS Spoofing and Detection," Vol. 104, 2016, pp. 1258–1270.

[61] Federal Highway Administration, "Roadway Departure Safety," https://safety.fhwa.dot.gov/roadway_dept/.

[62] Baidu, "Baidu Apollo," https://github.com/ApolloAuto/apollo.

[63] Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monrroy, A., Ando, T., Fujii, Y., and Azumi, T., "Autoware On Board: Enabling Autonomous Vehicles with Embedded Systems," *ICCPS'18*, IEEE Press, 2018, pp. 287–296.

[64] NovAtel, "An Introduction to GNSS: Chapter 4 - GNSS Error Sources," https://novatel.com/an-introduction-to-gnss/chapter-4-gnsserror-sources.

[65] Biber, P. and Straßer, W., "The Normal Distributions Transform: A New Approach to Laser Scan Matching," *IROS*, IEEE, 2003.

[66] Levinson, J. and Thrun, S., "Robust vehicle localization in urban environments using probabilistic maps," *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, 2010, pp. 4372–4378.

[67] Zeng, K. C., Liu, S., Shu, Y., Wang, D., Li, H., Dou, Y., Wang, G., and Yang, Y., "All Your GPS Are Belong To Us: Towards Stealthy Manipulation of Road Navigation Systems," *USENIX Security*, 2018.

[68] Narain, S., Ranganathan, A., and Noubir, G., "Security of GPS/INS based On-Road Location Tracking Systems," *IEEE Symposium on Security and Privacy (SP)*, 2019.

[69] C4ADS, "Above Us Only Stars - Exposing GPS Spoofing in Russia and Syria," https://www.c4reports.org/aboveusonlystars.

[70] Humphreys, T. E., Ledvina, B. M., Psiaki, M. L., O'Hanlon, B. W., and Kintner, P. M., "Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer," *ION GNSS'08*, 2008.

[71] Tippenhauer, N. O., Pöpper, C., Rasmussen, K. B., and Capkun, S., "On the Requirements for Successful GPS Spoofing Attacks," *CCS*, 2011.

[72] Davidson, D., Wu, H., Jellinek, R., Singh, V., and Ristenpart, T., "Controlling UAVs with Sensor Input Spoofing Attacks," *WOOT*, 2016.

[73] Lee, S., Cho, Y., and Min, B.-C., "Attack-Aware Multi-Sensor Integration Algorithm for Autonomous Vehicle Navigation Systems," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2017.

[74] Cardenas, A., "Cyber-Physical Systems Security Knowledge Area," *The Cyber Security Body Of Knowledge (cybok)*, 2019.

[75] Paden, B., Čáp, M., Yong, S. Z., Yershov, D., and Frazzoli, E., "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," *IEEE Transactions on intelligent vehicles*, Vol. 1, No. 1, 2016, pp. 33–55.

[76] Quinonez, R., Giraldo, J., Salazar, L., Bauman, E., Cardenas, A., and Lin, Z., "SAVIOR: Securing Autonomous Vehicles with Robust Physical Invariants," *USENIX Security*, 2020.

[77] Choi, H., Lee, W.-C., Aafer, Y., Fei, F., Tu, Z., Zhang, X., Xu, D., and Deng, X., "Detecting Attacks Against Robotic Vehicles: A Control Invariant Approach," *CCS*, 2018.

[78] Kong, J., Pfeiffer, M., Schildbach, G., and Borrelli, F., "Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design," *IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2015.

[79] Polack, P., Altché, F., d'Andréa Novel, B., and de La Fortelle, A., "The Kinematic Bicycle Model: a Consistent Model for Planning Feasible Trajectories for Autonomous Vehicles?" *IEEE intelligent vehicles symposium (IV)*, IEEE, 2017.

[80] Hillel, A. B., Lerner, R., Levi, D., and Raz, G., "Recent progress in road and lane detection: a survey," *Machine vision and applications*, Vol. 25, No. 3, 2014, pp. 727–745.

[81] Pan, X., Shi, J., Luo, P., Wang, X., and Tang, X., "Spatial as deep: Spatial cnn for traffic scene understanding," *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, 2018.

[82] comma.ai, "openpilot," https://github.com/commaai/openpilot.

[83] Tesla, "Autopilot," https://www.tesla.com/autopilot.

[84] Kang, J. M., Yoon, T. S., Kim, E., and Park, J. B., "Lane-Level Map-Matching Method for Vehicle Localization Using GPS and Camera on a High-Definition Map," *Sensors*, Vol. 20, No. 8, 2020, pp. 2166.

[85] Evlampev, A., Shapovalov, I., and Gafurov, S., "Map relative localization based on road lane matching with Iterative Closest Point algorithm," *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition*, 2020, pp. 232–236.

[86] "See What Tesla Autopilot Sees At Night In Rain: Video," https://insideevs.com/news/348362/video-what-tesla-autopilot-sees-night-rain/.

[87] Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., and Van Gool, L., "Towards End-to-End Lane Detection: an Instance Segmentation Approach," *2018 IEEE intelligent vehicles symposium (IV)*, IEEE, 2018, pp. 286–291.

[88] "The Autonomous Truck Revolution Is Right Around The Corner," https://www.forbes.com/sites/stevebanker/2021/05/11/the-autonomous-truck-revolution-is-right-around-the-corner/?sh=3d0022222c96.

[89] "Walmart First to Deliver Driverless Middle Mile," https://multichannelmerchant.com/operations/walmart-first-to-deliver-driverless-middle-mile/.

[90] Choi, H., Kate, S., Aafer, Y., Zhang, X., and Xu, D., "Software-based Realtime Recovery from Sensor Attacks on Robotic Vehicles," *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, 2020, pp. 349–364.

[91] Zhang, L., Chen, X., Kong, F., and Cardenas, A. A., "Real-Time Attack-Recovery for Cyber-Physical Systems Using Linear Approximations," *2020 IEEE Real-Time Systems Symposium (RTSS)*, IEEE, 2020, pp. 205–217.

[92] Sato, T., Shen, J., Wang, N., Jia, Y., Lin, X., and Chen, Q. A., "Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack," *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3309–3326.

[93] National Association of City Transportation Officials (NACTO), "Vehicle Stopping Distance and Time," https://nacto.org/docs/usdg/vehicle_stopping_distance_and_time_upenn.pdf.

[94] Aptiv, Audi, Baidu, BMW, Continental, Daimler, Fiat Chrysler Automobiles, HERE, Infineon, Intel and Volkswagen, "Safety First for Automated Driving," https://www.daimler.com/documents/innovation/other/safety-first-for-automated-driving.pdf, 2019.

[95] Lyft, "Semantic Maps for Autonomous Vehicles," https://medium.com/lyftself-driving/semantic-maps-for-autonomous-vehicles-470830ee28b6.

[96] Urbina, D. I., Giraldo, J. A., Cardenas, A. A., Tippenhauer, N. O., Valente, J., Faisal, M., Ruths, J., Candell, R., and Sandberg, H., "Limiting the Impact of Stealthy Attacks on Industrial Control Systems," *CCS*, 2016.

[97] Police Radar Information Center, "Vehicle Acceleration and Braking Parameters," https://copradar.com/chapts/references/acceleration.html.

[98] Thrun, S., Burgard, W., and Fox, D., *Probabilistic robotics*, MIT press, 2005.

[99] Friedland, B., *Control System Design: An Introduction to State-Space Methods*, Courier Corporation, 2012.

[100] Jeong, J., Cho, Y., Shin, Y.-S., Roh, H., and Kim, A., "Complex Urban Dataset with Multi-Level Sensors from Highly Diverse Urban Environments," *IJRR*, Vol. 38, No. 6, 2019, pp. 642–657.

[101] Rajamani, R., *Vehicle Dynamics and Control*, Springer Science & Business Media, 2011.

[102] Stein, W. J. and Neuman, T. R., "Mitigation Strategies for Design Exceptions," Tech. rep., United States. Federal Highway Administration. Office of Safety, 2007.

[103] Baidu, "Apollo Planning Module," https://github.com/ApolloAuto/apollo/tree/master/modules/planning.

[104] comma.ai, "Announcing the EON Dashcam DevKit," https://blog.comma.ai/announcing-the-eon-dashcam-devkit/.

[105] LG, "LGSVL Simulator: An Autonomous Vehicle Simulator," https://github.com/lgsvl/simulator.

[106] Lin, G.-T., Santoso, P. S., Lin, C.-T., Tsai, C.-C., and Guo, J.-I., "Stop Line Detection and Distance Measurement for Road Intersection based on Deep Learning Neural Network," *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2017, pp. 692–695.

[107] Bailo, O., Lee, S., Rameau, F., Yoon, J. S., and Kweon, I. S., "Robust Road Marking Detection and Recognition Using Density-Based Grouping and Machine Learning Techniques," *2017 IEEE winter conference on applications of computer vision (WACV)*, IEEE, 2017, pp. 760–768.

[108] Feng, Z., Guan, N., Lv, M., Liu, W., Deng, Q., Liu, X., and Yi, W., "An Efficient UAV Hijacking Detection Method Using Onboard Inertial Measurement Unit," *ACM Transactions on Embedded Computing Systems (TECS)*, Vol. 17, No. 6, 2018, pp. 1–19.

[109] Feng, Z., Guan, N., Lv, M., Liu, W., Deng, Q., Liu, X., and Yi, W., "Efficient Drone Hijacking Detection using Onboard Motion Sensors," *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, IEEE, 2017, pp. 1414–1419.

[110] Aguilar, W. G., Salcedo, V. S., Sandoval, D. S., and Cobeña, B., "Developing of a Video-Based Model for UAV Autonomous Navigation," *Latin American Workshop on Computational Neuroscience*, Springer, 2017, pp. 94–105.

[111] Khanafseh, S., Roshan, N., Langel, S., Chan, F.-C., Joerger, M., and Pervan, B., "GPS spoofing detection using RAIM with INS coupling," *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*, IEEE, 2014, pp. 1232–1239.

[112] Lee, B.-H., Song, J.-H., Im, J.-H., Im, S.-H., Heo, M.-B., and Jee, G.-I., "GPS/DR Error Estimation for Autonomous Vehicle Localization," *Sensors*, Vol. 15, No. 8, 2015, pp. 20779–20798.

[113] Petit, J., Stottelaar, B., Feiri, M., and Kargl, F., "Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and Lidar," *Black Hat Europe*, Vol. 11, 2015, pp. 2015.

[114] Yan, C., Xu, W., and Liu, J., "Can You Trust Autonomous Vehicles: Contactless Attacks Against Sensors of Self-Driving Vehicle," *DEF CON*, Vol. 24, 2016.

[115] Nassi, B., Nassi, D., Ben-Netanel, R., Mirsky, Y., Drokin, O., and Elovici, Y., "Phantom of the ADAS: Phantom Attacks on Driver-Assistance Systems," *IACR Cryptol. ePrint Arch.*, Vol. 2020, 2020, pp. 85.

[116] Sayles, A., Hooda, A., Gupta, M., Chatterjee, R., and Fernandes, E., "Invisible Perturbations: Physical Adversarial Examples Exploiting the Rolling Shutter Effect," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14666–14675.

[117] Köhler, S., Lovisotto, G., Birnbach, S., Baker, R., and Martinovic, I., "They See Me Rollin': Inherent Vulnerability of the Rolling Shutter in CMOS Image Sensors," *arXiv preprint arXiv:2101.10011*, 2021.

[118] Jing, P., Tang, Q., Du, Y., Xue, L., Luo, X., Wang, T., Nie, S., and Wu, S., "Too Good to Be Safe: Tricking Lane Detection in Autonomous Driving with Crafted Perturbations," *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[119] Cao, Y., Xiao, C., Cyr, B., Zhou, Y., Park, W., Rampazzi, S., Chen, Q. A., Fu, K., and Mao, Z. M., "Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving," *CCS*, 2019.

[120] Son, Y., Shin, H., Kim, D., Park, Y., Noh, J., Choi, K., Choi, J., and Kim, Y., "Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors," *USENIX Security*, 2015.

[121] Giraldo, J., Sarkar, E., Cardenas, A. A., Maniatakos, M., and Kantarcioglu, M., "Security and Privacy in Cyber-Physical Systems: A Survey of Surveys," *IEEE Design & Test*, Vol. 34, No. 4, 2017, pp. 7–17.

[122] Schnabel, R., Wahl, R., and Klein, R., "Efficient RANSAC for Point-Cloud Shape Detection," *Computer graphics forum*, Vol. 26, Wiley Online Library, 2007, pp. 214–226.

[123] Cohen, J., *Statistical Power Analysis for the Behavioral Sciences*, Routledge, 2013.

[124] Merten, H., "The Three-Dimensional Normal-Distributions Transform," *threshold*, Vol. 10, 2008, pp. 3.

[125] "SAVIOR codebase," https://github.com/Cyphysecurity/SAVIOR.

[126] MathWorks, "System Identification Toolbox," https://www.mathworks.com/products/sysid.html.

[127] Schafer, H., Santana, E., Haden, A., and Biasini, R., "A Commute in Data: The comma2k19 Dataset," *arXiv:1812.05752*, 2018.

[128] "Lane Keeping Assist System Using Model Predictive Control," https://www.mathworks.com/help/mpc/ug/lane-keeping-assist-system-using-model-predictive-control.html, 2020.

[129] Lee, J.-W. and Litkouhi, B., "A Unified Framework of the Automated Lane Centering/Changing Control for Motion Smoothness Adaptation," *International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 282–287.

[130] "Super Cruise - Hands Free Driving — Cadillac Ownership," https://www.cadillac.com/world-of-cadillac/innovation/super-cruise.

[131] "Tesla Autopilot," https://www.tesla.com/autopilot.

[132] "2020 Accord Hybrid Owner's Manual," http://techinfo.honda.com/rjanisis/pubs/OM/AH/ATWA2020OM/enu/ATWA2020OM.PDF.

[133] "Toyota 2020 RAV4 Owner's Manual," https://www.toyota.com/t3Portal/document/om-s/OM0R024U/xhtml/OM0R024U.html.

[134] Wang, Z., Ren, W., and Qiu, Q., "LaneNet: Real-Time Lane Detection Networks for Autonomous Driving," *arXiv:1807.01726*, 2018.

[135] Ko, Y., Jun, J., Ko, D., and Jeon, M., "Key Points Estimation and Point Instance Segmentation Approach for Lane Detection," *arXiv:2002.06604*, 2020.

[136] Li, J., Mei, X., Prokhorov, D., and Tao, D., "Deep Neural Network for Structural Prediction and Lane Detection in Traffic Scene," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 28, No. 3, 2016, pp. 690–703.

[137] Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., and Wang, Q., "Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks," *IEEE Transactions on Vehicular Technology*, 2019.

[138] Smuda, P., Schweiger, R., Neumann, H., and Ritter, W., "Multiple Cue Data Fusion With Particle Filters for Road Course Detection in Vision Systems," *IEEE Intelligent Vehicles Symposium (IV)*, 2006.

[139] Gackstatter, C., Heinemann, P., Thomas, S., and Klinker, G., "Stable Road Lane Model Based on Clothoids," *Advanced Microsystems for Automotive Applications*, Springer, 2010, pp. 133–143.

[140] Yenikaya, S., Yenikaya, G., and Düven, E., "Keeping the Vehicle on the Road - A Survey on On-Road Lane Detection Systems," *ACM Computing Surveys (CSUR)*, Vol. 46, No. 1, 2013, pp. 1–43.

[141] Becker, C., Yount, L. J., Rozen-Levy, S., and Brewer, J. D., "Functional Safety Assessment of an Automated Lane Centering System," *National Highway Traffic Safety Administration*, 2018.

[142] Dorf, R. C. and Bishop, R. H., *Modern Control Systems*, Pearson, 2011.

[143] Richalet, J. and Rault, A. and Testud, J. L. and Papon, J., "Model Predictive Heuristic Control," *Automatica*, Vol. 14, 1978, pp. 413–428.

[144] “Tinkla: Tinkering with Tesla,” https://tinkla.us/.

[145] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R., “Intriguing Properties of Neural Networks,” *International Conference on Learning Representation (ICLR)*, 2014.

[146] Goodfellow, I. J., Shlens, J., and Szegedy, C., “Explaining and Harnessing Adversarial Examples,” *arXiv:1412.6572*, 2014.

[147] Kurakin, A., Goodfellow, I., and Bengio, S., “Adversarial Examples in the Physical World,” *arXiv:1607.02533*, 2016.

[148] Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K., “Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition,” *ACM SIGSAC Conference on Computer and Communications Security (ACM CCS)*, 2016, pp. 1528–1540.

[149] Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K., “Synthesizing Robust Adversarial Examples,” *International Conference on Machine Learning (ICML)*, 2018.

[150] Brown, T., Mane, D., Roy, A., Abadi, M., and Gilmer, J., “Adversarial Patch,” *arXiv:1712.09665*, 2017.

[151] Chen, S.-T., Cornelius, C., Martin, J., and Chau, D. H. P., “Shapeshifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector,” *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2018, pp. 52–68.

[152] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramer, F., Prakash, A., Kohno, T., and Song, D., “Physical Adversarial Examples for Object Detectors,” *WOOT*, 2018.

[153] Zhong, Z., Xu, W., Jia, Y., and Wei, T., “Perception Deception: Physical Adversarial Attack Challenges and Tactics for DNN-Based Object Detection,” *Black Hat Europe*, 2018.

[154] Zhao, Y., Zhu, H., Liang, R., Shen, Q., Zhang, S., and Chen, K., “Seeing isn’t Believing: Practical Adversarial Attack Against Object Detectors,” *ACM SIGSAC Conference on Computer and Communications Security (ACM CCS)*, 2019, p. 1989–2004.

[155] Pei, K., Cao, Y., Yang, J., and Jana, S., “Deepxplore: Automated Whitebox Testing of Deep Learning Systems,” *Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[156] Tian, Y., Pei, K., Jana, S., and Ray, B., “Deeptest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars,” *International Conference on Software Engineering*, 2018, pp. 303–314.

[157] Chernikova, A., Oprea, A., Nita-Rotaru, C., and Kim, B., “Are Self-Driving Cars Secure? Evasion Attacks Against Deep Neural Networks for Steering Angle Prediction,” *IEEE Security and Privacy Workshops (SPW)*, 2019, pp. 132–137.

[158] Zhou, H., Li, W., Zhu, Y., Zhang, Y., Yu, B., Zhang, L., and Liu, C., “Deepbillboard: Systematic Physical-World Testing of Autonomous Driving Systems,” *International Conference on Software Engineering*, 2020.

[159] “Tesla Autopilot Support,” https://www.tesla.com/support/autopilot.

[160] SAE On-Road Automated Vehicle Standards Committee and others, “Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,” *SAE International: Warrendale, PA, USA*, 2018.

[161] Zhao, D., Guo, Y., and Jia, Y. J., “Trafficnet: An Open Naturalistic Driving Scenario Library,” *IEEE International Conference on Intelligent Transportation Systems*, 2017, pp. 1–8.

[162] Boora, A., Ghosh, I., and Chandra, S., “Identification of Free Flowing Vehicles on Two Lane Intercity Highways under Heterogeneous Traffic condition,” *Transportation Research Procedia*, Vol. 21, 2017, pp. 130–140.

[163] “California Vehicle Code 21663,” https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=VEH&sectionNum=21663.

[164] “Does your car have automated emergency braking? It’s a big fail for pedestrians,” https://www.zdnet.com/article/does-your-car-have-automated-emergency-braking-its-a-big-fail-for-pedestrians/.

[165] “Experimental Security Research of Tesla Autopilot,” https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf, 2019.

[166] Rajamani, R., *Vehicle Dynamics and Control*, Springer Science & Business Media, 2011.

[167] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D., “Robust Physical-World Attacks on Deep Learning Visual Classification,” *CVPR*, 2018.

[168] “California Penal Code 594,” https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=PEN&sectionNum=594, 1872.

[169] Li, S., Neupane, A., Paul, S., Song, C., Krishnamurthy, S. V., Roy-Chowdhury, A. K., and Swami, A., “Stealthy Adversarial Perturbations Against Real-Time Video Classification Systems,” *Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[170] Carlini, N. and Wagner, D., “Audio Adversarial Examples: Targeted Attacks on Speech-to-Text,” *2018 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2018, pp. 1–7.

[171] Ebrahimi, J., Lowd, D., and Dou, D., “On Adversarial Examples for Character-Level Neural Machine Translation,” *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, Association for Computational Linguistics, 2018.

[172] “Adhesive Patch can Seal Potholes and Cracks on the Road,” https://www.startupselfie.net/2019/05/07/american-road-patch-seals-potholes-road-cracks/, 2019.

[173] “GM Cadillac CT6 Owner’s Manual,” https://www.cadillac.com/content/dam/cadillac/na/us/english/index/ownership/technology/supercruise/pdfs/2020-cad-ct6-owners-manual.pdf, 2019.

[174] “Volvo XC90 Owner’s Manual,” https://volvornt.harte-hanks.com/manuals/2020/XC90_OwnersManual_MY20_en-US_TP29159[1].pdf, 2020.

[175] “KIA Seltos Owner’s Manual,” https://www.kia.ca/content/ownership/ownersmanual/21seltos.pdf, 2020.

[176] “Ford Escape Owner’s Manual,” https://www.fordservicecontent.com/Ford_Content/Catalog/owner_information/2020-Ford-Escape-Gas-HEV-PHEV-Owners-Manual-version-3_om_EN_08_2020.pdf, 2020.

[177] “Nissan Rogue Sports Owner’s Manual,” https://www.nissan-cdn.net/content/dam/Nissan/pr/Owners-manuals/rogue-sport/2019-RogueSport-owner-manual.pdf, 2019.

[178] “Hyundai Sonata Owner’s Manual,” https://owners.hyundaiusa.com/content/dam/hyundai/us/myhyundai/glovebox-manual/2020/sonata/2020SonataOwner’ sManual.pdf, 2020.

[179] Carlini, N. and Wagner, D., “Towards Evaluating the Robustness of Neural Networks,” *IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 39–57.

[180] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A., “Towards Deep Learning Models Resistant to Adversarial Attacks,” *International Conference on Learning Representation (ICLR)*, 2018.

[181] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V., “CARLA: An Open Urban Driving Simulator,” *Annual Conference on Robot Learning*, 2017.

[182] Hartley, R. and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd ed., 2003.

[183] Tanaka, S., Yamada, K., Ito, T., and Ohkawa, T., “Vehicle Detection Based on Perspective Transformation Using Rear-View Camera,” *Hindawi Publishing Corporation International Journal of Vehicular Technology*, Vol. 9, 03 2011.

[184] Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., and Weaver, J., “Google Street View: Capturing the World at Street Level,” *Computer*, Vol. 43, No. 6, 2010, pp. 32–38.

[185] "Introduction to Self-Driving Cars," https://www.coursera.org/learn/intro-self-driving-cars.

[186] Watzenig, D. and Horn, M., *Automated Driving: Safer and More Efficient Future Driving*, Springer, 2016.

[187] Kingma, D. P. and Ba, J., "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representation (ICLR)*, 2015.

[188] Hamilton, E., "JPEG File Interchange Format," 2004.

[189] "Is a $1000 Aftermarket Add-On as Capable as Tesla's Autopilot and Cadillac's Super Cruise?" https://www.caranddriver.com/features/a30341053/self-driving-technology-comparison/, 2020.

[190] "We Hit the Road with Comma.ai's Assisted-Driving Tech at CES 2020," https://www.cnet.com/roadshow/news/comma-ai-assisted-driving-george-hotz-ces-2020/, 2020.

[191] "Hands-on with Comma.ai's Add-on Level 2 Autonomous Tech," https://www.cnet.com/roadshow/news/hands-on-with-comma-ais-add-on-level-2-autonomous-tech/, 2018.

[192] of California Department of Motor Vehicles, S., *California Commercial Driver Handbook: Section 2 – Driving Safely*, 2019, Available at https://www.dmv.ca.gov/portal/dmv/detail/pubs/cdl_htm/sec2.

[193] Ling, X., Ji, S., Zou, J., Wang, J., Wu, C., Li, B., and Wang, T., "Deepsec: A Uniform Platform for Security Analysis of Deep Learning Model," *IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 673–690.

[194] Athalye, A., Carlini, N., and Wagner, D., "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples," *International Conference on Machine Learning (ICML)*, 2018.

[195] "Manual on Uniform Traffic Control Devices Part 3 Markings," https://mutcd.fhwa.dot.gov/pdfs/millennium/06.14.01/3ndi.pdf, 2020.

[196] "American Road Patch - Deployment Demonstration Video from 4:54 to 5:04," https://youtu.be/Vr_Dxg1LdxU?t=294, 2019.

[197] "Dirty Road Patch Attack Project Website," https://sites.google.com/view/cav-sec/drp-attack.

[198] "Toyota Safety Sense Pre-Collision System (PCS) Settings and Controls," https://youtu.be/IY4g_zG1Qj0, 2017.

[199] "Honda's Collision Mitigation Braking System CMBS," https://youtu.be/NJcy5ySOrM4, 2013.

[200] "IIHS Issues First Crash Avoidance Ratings Under New Test Program," https://www.iihs.org/news/detail/iihs-issues-first-crash-avoidance-ratings-under-new-test-program, 2013.

[201] "Collision Avoidance Strikeable Targets for AEB," http://www.pedstrikeabletargets.com/, 2020.

[202] "California Vehicle Code 23113," https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=VEH&sectionNum=23113, 2000.

[203] "California Vehicle Code 23112," https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=VEH&sectionNum=23112, 2000.

[204] "Idaho Statutes Title 49, Motor Vehicles, Chapter 6, Rules of the Road," https://legislature.idaho.gov/statutesrules/idstat/title49/t49ch6/sect49-613/, 2015.

[205] "Toyota 2019 Camry Owner's Manual," https://www.toyota.com/t3Portal/document/om-s/OM06142U/pdf/OM06142U.pdf, 2019.

[206] Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., and Zhu, J., "Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[207] Xie, C., Wu, Y., Maaten, L. v. d., Yuille, A. L., and He, K., "Feature Denoising for Improving Adversarial Robustness," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[208] Xu, W., Evans, D., and Qi, Y., "Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks," *arXiv:1704.01155*, 2017.

[209] Guo, C., Rana, M., Cisse, M., and Van Der Maaten, L., "Countering Adversarial Images using Input Transformations," *International Conference on Learning Representation (ICLR)*, 2018.

[210] Raghunathan, A., Steinhardt, J., and Liang, P., "Certified Defenses Against Adversarial Examples," 2018.

[211] Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S., "Certified Robustness to Adversarial Examples with Differential Privacy," *IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 656–672.

[212] Cohen, J., Rosenfeld, E., and Kolter, Z., "Certified Adversarial Robustness via Randomized Smoothing," *International Conference on Machine Learning (ICML)*, 2019, pp. 1310–1320.

[213] Dziugaite, G. K., Ghahramani, Z., and Roy, D. M., "A Study of the Effect of JPG Compression on Adversarial Images," *arXiv:1608.00853*, 2016.

[214] Zhang, Y. and Liang, P., "Defending Against Whitebox Adversarial Attacks via Randomized Discretization," *International Conference on Artificial Intelligence and Statistics*, Vol. 89, 2019, pp. 684–693.

[215] Meng, D. and Chen, H., "Magnet: a Two-pronged Defense Against Adversarial Examples," *ACM SIGSAC Conference on Computer and Communications Security (ACM CCS)*, 2017, pp. 135–147.

[216] "HD Maps: New Age Maps Powering Autonomous Vehicles," https://www.geospatialworld.net/article/hd-maps-autonomous-vehicles/.

[217] Bai, M., Mattyus, G., Homayounfar, N., Wang, S., Lakshmikanth, S. K., and Urtasun, R., "Deep Multi-Sensor Lane Detection," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3102–3109.

[218] "Waymo Has Launched its Commercial Self-Driving Service in Phoenix — and it's Called 'Waymo One'," https://www.businessinsider.com/waymo-one-driverless-car-service-launches-in-phoenix-arizona-2018-12, 2018.

[219] "Velodyne Just Cut the Price of Its Most Popular Lidar Sensor in Half," https://www.thedrive.com/tech/17297/velodyne-just-cut-the-price-of-its-most-popular-lidar-sensor-in-half, 2018.

[220] "'Anyone Relying on Lidar is Doomed,' Elon Musk Says," https://techcrunch.com/2019/04/22/anyone-relying-on-lidar-is-doomed-elon-musk-says/, 2019.

[221] "Tesla Admits its Approach to Self-Driving is Harder But Might be Only Way to Scale," https://electrek.co/2020/06/18/tesla-approach-self-driving-harder-only-way-to-scale/, 2020.

[222] "Building Maps for a Self-Driving Car," https://link.medium.com/Bo5pCOov95, 2016.

[223] "Baidu Apollo HD Map," http://ggim.un.org/unwgic/presentations/2.2_Ma_Changjie.pdf, 2018.

[224] Shin, H., Kim, D., Kwon, Y., and Kim, Y., "Illusion and Dazzle: Adversarial Optical Channel Exploits Against Lidars for Automotive Applications," *International Conference on Cryptographic Hardware and Embedded Systems*, Springer, 2017, pp. 445–467.

[225] Tu, Y., Lin, Z., Lee, I., and Hei, X., "Injected and Delivered: Fabricating Implicit Control over Actuation Systems by Spoofing Inertial Sensors," *USENIX Security*, 2018.

[226] Trippel, T., Weisse, O., Xu, W., Honeyman, P., and Fu, K., "WALNUT: Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks," *EuroS&P*, IEEE, 2017, pp. 3–18.

[227] "Model Hacking ADAS to Pave Safer Roads for Autonomous Vehicles," https://www.mcafee.com/blogs/other-blogs/mcafee-labs/model-hacking-adas-to-pave-safer-roads-for-autonomous-vehicles/, 2020.

[228] Jia, Y., Lu, Y., Shen, J., Chen, Q. A., Chen, H., Zhong, Z., and Wei, T., "Fooling Detection Alone is Not Enough: Adversarial Attack Against Multiple Object Tracking," *International Conference on Learning Representations (ICLR)*, 2019.

[229] Shen, J., Won, J. Y., Chen, Z., and Chen, Q. A., "Drift with Devil: Security of Multi-Sensor Fusion based Localization in High-Level Autonomous Driving under GPS Spoofing," *USENIX Security Symposium*, 2020.

[230] Tang, K., Shen, J., and Chen, Q. A., “Fooling Perception via Location: A Case of Region-of-Interest Attacks on Traffic Light Detection in Autonomous Driving,” *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2021.

[231] Li, J., Schmidt, F., and Kolter, Z., “Adversarial Camera Stickers: A Physical Camera-Based Attack on Deep Learning Systems,” *International Conference on Machine Learning*, 2019, pp. 3896–3904.

[232] “TuSimple Lane Detection Challenge,” https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection, 2017.

[233] Sato, T., Shen, J., Wang, N., Jia, Y., Lin, X., and Chen, Q. A., “Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack,” *29th USENIX Security*, 2021.

[234] Jing, P., Tang, Q., Du, Y., Xue, L., Luo, X., Wang, T., Nie, S., and Wu, S., “Too Good to Be Safe: Tricking Lane Detection in Autonomous Driving with Crafted Perturbations,” *30th USENIX Security Symposium*, 2021.

[235] Tabelini, L., Berriel, R., ao, T. M. P., Badue, C., Souza, A. F. D., and Oliveira-Santos, T., “Keep Your Eyes on the Lane: Real-time Attention-guided Lane Detection,” *CVPR*, 2021.

[236] Liu, L., Chen, X., Zhu, S., and Tan, P., “CondLaneNet: A Top-To-Down Lane Detection Framework Based on Conditional Convolution,” *ICCV*, 2021.

[237] Hsu, Y.-C., Xu, Z., Kira, Z., and Huang, J., “Learning to Cluster for Proposal-Free Instance Segmentation,” *IJCNN*, 2018.

[238] Zheng, T., Fang, H., Zhang, Y., Tang, W., Yang, Z., Liu, H., and Cai, D., “RESA: Recurrent Feature-Shift Aggregator for Lane Detection,” 2020.

[239] Hou, Y., Ma, Z., Liu, C., and Loy, C. C., “Learning Lightweight Lane Detection CNNs by Self Attention Distillation,” *CVPR*, 2019.

[240] Zheng, T., Fang, H., Zhang, Y., Tang, W., Yang, Z., Liu, H., and Cai, D., “RESA: Recurrent Feature-Shift Aggregator for Lane Detection,” *AAAI*, 2021.

[241] Qin, Zequn and Wang, Huanyu and Li, Xi, “Ultra Fast Structure-Aware Deep Lane Detection,” *ECCV*, 2020.

[242] Yoo, S., Lee, H. S., Myeong, H., Yun, S., Park, H., Cho, J., and Kim, D. H., “End-to-End Lane Marker Detection via Row-Wise Classification,” *CVPR Workshop*, 2020.

[243] Hou, Y., Ma, Z., Liu, C., Hui, T.-W., and Loy, C. C., “Inter-Region Affinity Distillation for Road Marking Segmentation,” *CVPR*, 2020.

[244] Tabelini, L., Berriel, R., Paixao, T. M., Badue, C., De Souza, A. F., and Oliveira-Santos, T., “Polylanenet: Lane Estimation via Deep Polynomial Regression,” *ICPR*, 2021.

[245] Philion, J., “FastDraw: Addressing the Long Tail of Lane Detection by Adapting a Sequential Prediction Network,” *CVPR*, 2019.

[246] Li, X., Li, J., Hu, X., and Yang, J., “Line-CNN: End-to-End Traffic Line Detection with Line Proposal Unit,” *IEEE T-ITS*, 2019.

[247] Qu, Z., Jin, H., Zhou, Y., Yang, Z., and Zhang, W., “Focus on Local: Detecting Lane Marker From Bottom Up via Key Point,” *CVPR*, 2021.

[248] Ren, S., He, K., Girshick, R., and Sun, J., “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, 2015, pp. 91–99.

[249] Ilyas, A., Engstrom, L., Athalye, A., and Lin, J., “Black-Box Adversarial Attacks with Limited Queries and Information,” *ICML*, 2018.

[250] Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B., “Algorithms for Hyper-Parameter Optimization,” *NeurIPS*, 2011.

[251] “Hyperopt,” https://github.com/hyperopt/hyperopt.

[252] Sato, T. and Chen, Q. A., “On Robustness of Lane Detection Models to Physical-World Adversarial Attacks in Autonomous Driving,” *arXiv preprint arXiv:2107.02488*, 2021.

[253] Li, Y., Li, L., Wang, L., Zhang, T., and Gong, B., “Nattack: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks,” *ICML*, 2019.

[254] Jwa, S., Ozguner, Ü., and Tang, Z., “Information-theoretic data registration for UAV-based sensing,” *IEEE Transactions on intelligent transportation systems*, Vol. 9, No. 1, 2008, pp. 5–15.

[255] Adamey, E. and Ozguner, U., “Cooperative multitarget tracking and surveillance with mobile sensing agents: A decentralized approach,” *2011 14th International IEEE conference on intelligent transportation systems (ITSC)*, IEEE, 2011, pp. 1916–1922.

[256] Adamey, E., Ozbilgin, G., and Ozguner, U., “Collaborative vehicle tracking in mixed-traffic environments: Scaled-down tests using simville,” Tech. rep., SAE Technical Paper, 2015.

[257] Borges, P. V., Tews, A., and Haddon, D., “Pedestrian detection in industrial environments: Seeing around corners,” *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 4231–4232.

[258] Gelbal, S. Y., Arslan, S., Wang, H., Aksun-Guvenc, B., and Guvenc, L., “Elastic band based pedestrian collision avoidance using V2X communication,” *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 270–276.

[259] Flores, C., Merdrignac, P., de Charette, R., Navas, F., Milanés, V., and Nashashibi, F., “A cooperative car-following/emergency braking system with prediction-based pedestrian avoidance capabilities,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 5, 2018, pp. 1837–1846.

[260] Sugimoto, C., Nakamura, Y., and Hashimoto, T., “Prototype of pedestrian-to-vehicle communication system for the prevention of pedestrian accidents using both 3G wireless and WLAN communication,” *2008 3rd International Symposium on Wireless Pervasive Computing*, IEEE, 2008, pp. 764–767.

[261] ISO, “ISO 22737:2021,” June 2022.

[262] Miura, S., Hsu, L.-T., Chen, F., and Kamijo, S., “GPS error correction with pseudorange evaluation using three-dimensional maps,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 6, 2015, pp. 3104–3115.

[263] Cincinnati, C. o., “Vehicle GPS DATA: Department of Public Services: Open Data: Socrata,” Nov 2021.

[264] Adamey, E., Kurt, A., and Ozgüner, U., “Cooperative traffic mapping using onboard sensing and V2V communication in mixed-traffic environments,” *Second International Symposium on Future Active Safety Technology, FAST-zero*, 2013, pp. 1–6.

[265] Acarman, T., Pan, Y., and Ozguner, U., “A control authority transition system for collision avoidance,” *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, IEEE, 2001, pp. 466–471.

[266] CARLA, “Carla,” May 2022.

[267] Cui, Y. and Ge, S. S., “Autonomous vehicle positioning with GPS in urban canyon environments,” *IEEE transactions on robotics and automation*, Vol. 19, No. 1, 2003, pp. 15–25.

[268] Maier, D. and Kleiner, A., “Improved GPS sensor model for mobile robots in urban terrain,” *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 4385–4390.

[269] Enge, P., “Local area augmentation of GPS for the precision approach of aircraft,” *Proceedings of the IEEE*, Vol. 87, No. 1, 1999, pp. 111–132.

[270] Green, G. N. and Humphreys, T., “Position-Domain Integrity Analysis for Generalized Integer Aperture Bootstrapping,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 55, No. 2, 2018, pp. 734–746.

[271] Torens, C., Volkert, A., Becker, D., Gerbeth, D., Schalk, L., Garcia Crespillo, O., Zhu, C., Stelkens-Kobsch, T., Gehrke, T., Metz, I. C., et al., “HorizonUAM: Safety and Security Considerations for Urban Air Mobility,” *AIAA Aviation 2021 Forum*, 2021, p. 3199.

[272] Clements, Z., Yoder, J. E., and Humphreys, T. E., “Carrier-phase and IMU based GNSS Spoofing Detection for Ground Vehicles,” *Proceedings of the ION International Technical Meeting*, Long Beach, CA, 2022.

[273] Rizos, C., Grejner-Brzezinska, D., Toth, C., Dempster, A., Li, Y., Politi, N., Barnes, J., and Sun, H., "A hybrid system for navigation in GPS-challenged environments: Case study," Vol. 4, 08 2009.

[274] Grejner-Brzezinska, D. A., Toth, C. K., Sun, H., Wang, X., and Rizos, C., "A Robust Solution to High-Accuracy Geolocation: Quadruple Integration of GPS, IMU, Pseudolite, and Terrestrial Laser Scanning," *IEEE Transactions on Instrumentation and Measurement*, Vol. 60, No. 11, 2011, pp. 3694–3708.

[275] Jiang, W., Li, Y., and Rizos, C., "Improved decentralized multi-sensor navigation system for airborne applications," *GPS Solutions*, Vol. 22, No. 78, 2018, https://doi.org/10.1007/s10291-018-0743-9.

[276] Jiang, W., Li, Y., Rizos, C., and Barnes, J., "Flight Evaluation of a Locata-augmented Multisensor Navigation System," *Journal of Applied Geodesy*, Vol. 7, 11 2013.

[277] Jiang, W., Li, Y., and Rizos, C., "Optimal Data Fusion Algorithm for Navigation Using Triple Integration of PPP-GNSS, INS, and Terrestrial Ranging System," *IEEE Sensors Journal*, Vol. 15, No. 10, 2015, pp. 5634–5644.

[278] Mohanty, A., Wu, A., Bhamidipati, S., and Gao, G., "Precise Relative Positioning via Tight-Coupling of GPS Carrier Phase and Multiple UWBs," *IEEE Robotics and Automation Letters*, 2022, pp. 1–1.

[279] Maaref, M., Khalife, J., and Kassas, Z. M., "Aerial Vehicle Protection Level Reduction by Fusing GNSS and Terrestrial Signals of Opportunity," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 9, 2021, pp. 5976–5993.

[280] Jia, M., Lee, H., Khalife, J., Kassas, Z. M., and Seo, J., "Ground Vehicle Navigation Integrity Monitoring for Multi-Constellation GNSS Fused with Cellular Signals of Opportunity," *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 3978–3983.

[281] Brown, R. G. and Hwang, P. Y., *Introduction to Random Signals and Applied Kalman Filtering*, Wiley, 2012.

[282] Humphreys, T. E., Kor, R. X. T., and Iannucci, P. A., "Open-World Virtual Reality Headset Tracking," *Proceedings of the ION GNSS+ Meeting*, Online, 2020.

[283] Rizos, C., Roberts, G., Barnes, J., and Gambale, N., "Experimental results of Locata: A high accuracy indoor positioning system," *2010 International Conference on Indoor Positioning and Indoor Navigation*, IEEE, 2010, pp. 1–7.

[284] Meiyappan, S., Raghupathy, A., and Pattabiraman, G., "Positioning in GPS challenged locations-The NextNav terrestrial positioning constellation," *Proc. ION GNSS+ 2013*, 2013.

[285] Rizos, C., Grejner-Brzezinska, D. A., Toth, C. K., Dempster, A. G., Li, Y., Politi, N., Barnes, J., and Sun, H., "A hybrid system for navigation in GPS-challenged environments: Case study," *Proceedings, ION GNSS, Savannah, Georgia, Sept*, 2008, pp. 16–19.

[286] Rizos, C. and Yang, L., "Background and Recent Advances in the Locata Terrestrial Positioning and Timing Technology," *Sensors*, Vol. 19, No. 8, 2019, pp. 1821.

[287] Barnes, J., Rizos, C., Kanli, M., Small, D., Voigt, G., Gambale, N., Lamance, J., Nunan, T., and Reid, C., "Indoor industrial machine guidance using Locata: A pilot study at BlueScope Steel," *60th Annual Meeting of the US Inst. Of Navigation*, 2004, pp. 533–540.

[288] Khan, F. A., Rizos, C., and Dempster, A. G., "Novel time-sharing scheme for virtual elimination of Locata-WiFi interference effects," *Int. Symp. on GPS/GNSS*, 2008, pp. 526–530.

[289] Khan, F. A., Rizos, C., and Dempster, A. G., "Locata performance evaluation in the presence of wide-and narrow-band interference," *Journal of Navigation*, Vol. 63, No. 3, 2010, pp. 527.

[290] Scott, L., "Anti-spoofing and authenticated signal architectures for civil navigation systems," *Proceedings of the ION GNSS Meeting*, 2003, pp. 1542–1552.

[291] Anderson, J. M., Carroll, K. L., DeVilbiss, N. P., Gillis, J. T., Hinks, J. C., O'Hanlon, B. W., Rushanan, J. J., Scott, L., and Yazdi, R. A., "Chips-message robust authentication (Chimera) for GPS civilian signals," *ION GNSS*, 2017, pp. 2388–2416.

[292] Margaria, D., Motella, B., Anghileri, M., Floch, J.-J., Fernandez-Hernandez, I., and Paonni, M., "Signal structure-based authentication for civil GNSSs: Recent solutions and perspectives," *IEEE Signal Processing Magazine*, Vol. 34, No. 5, 2017, pp. 27–37.

[293] Humphreys, T. E., Ledvina, B. M., Psiaki, M. L., O'Hanlon, B. W., and Kintner, Jr., P. M., "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," *Proceedings of the ION GNSS Meeting*, Institute of Navigation, Savannah, GA, 2008.

[294] Psiaki, M. L., O'Hanlon, B. W., Powell, S. P., Bhatti, J. A., Humphreys, T. E., and Schofield, A., "GNSS lies, GNSS truth: Spoofing Detection with Two-Antenna Differential Carrier Phase," *GPS World*, Vol. 25, No. 11, Feb. 2014, pp. 36–44.

[295] C4ADS, "Above us only stars: Exposing GPS spoofing in Russia and Syria," April 2019, https://c4ads.org/reports.

[296] Murrian, M. J., Narula, L., and Humphreys, T. E., "Characterizing Terrestrial GNSS Interference from Low Earth Orbit," *Proceedings of the ION GNSS+ Meeting*, Institute of Navigation, Oct. 2019.

[297] Harris, M., "Ghost ships, crop circles, and soft gold: A GPS mystery in Shanghai," *MIT Technology Review*, 11 2019.

[298] Bergman, B., "AIS Ship Tracking Data Shows False Vessel Tracks Circling Above Point Reyes, Near San Francisco," 05 2020.

[299] Humphreys, T. E., "Lost in Space: How Secure Is the Future of Mobile Positioning?" 02 2016.

[300] Yang, C., Soloviev, A., Veth, M., and Qiu, D., "Opportunistic Use of Metropolitan RF Beacon Signals for Urban and Indoor Positioning," *Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016), Portland, Oregon*, 2016, pp. 394–403.

[301] Meurer, M., Konovaltsev, A., Cuntz, M., and Hättich, C., "Robust joint multi-antenna spoofing detection and attitude estimation using direction assisted multiple hypotheses RAIM," *Proceedings of the 25th Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2012)*, ION, 2012.

[302] Borio, D., "PANOVA tests and their application to GNSS spoofing detection," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, No. 1, Jan. 2013, pp. 381–394.

[303] He, L., Li, H., and Lu, M., "Dual-antenna GNSS spoofing detection method based on Doppler frequency difference of arrival," *GPS Solutions*, Vol. 23, No. 3, 2019, pp. 78.

[304] Psiaki, M., Powell, S. P., and O'Hanlon, B. W., "GNSS Spoofing Detection Using High-Frequency Antenna Motion and Carrier-Phase Data," *Proceedings of the ION GNSS+ Meeting*, 2013, pp. 2949–2991.

[305] Broumandan, A., Jafarnia-Jahromi, A., Dehghanian, V., Nielsen, J., and Lachapelle, G., "GNSS Spoofing Detection in Handheld Receivers based on Signal Spatial Correlation," *Proceedings of the IEEE/ION PLANS Meeting*, Institute of Navigation, Myrtle Beach, SC, April 2012.

[306] McMilin, E., De Lorenzo, D. S., Walter, T., Lee, T. H., and Enge, P., "Single antenna GPS spoof detection that is simple, static, instantaneous and backwards compatible for aerial applications," *Proceedings of the 27th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2014), Tampa, FL*, Citeseer, 2014, pp. 2233–2242.

[307] Poncelet, J.-P. and Akos, D. M., "A low-cost monitoring station for detection & localization of interference in GPS L1 band," *2012 6th ESA Workshop on Satellite Navigation Technologies (Navitec 2012) & European Workshop on GNSS Signals and Signal Processing*, IEEE, 2012, pp. 1–6.

[308] Lee, Y. C. and O'Laughlin, D. G., "Performance Analysis of a Tightly Coupled GPS/Inertial System for Two Integrity Monitoring Methods 1," *Navigation*, Vol. 47, No. 3, 2000, pp. 175–189.

[309] Wesson, K. D., Shepard, D. P., Bhatti, J. A., and Humphreys, T. E., “An Evaluation of the Vestigial Signal Defense for Civil GPS Anti-Spoofing,” *Proceedings of the ION GNSS Meeting*, Portland, OR, 2011.

[310] Wullems, C., Pozzobon, O., and Kubik, K., “Signal Authentication and Integrity Schemes for Next Generation Global Navigation Satellite Systems,” *Proc. European Navigation Conference GNSS*, Munich, July 2005.

[311] Kerns, A. J., Wesson, K. D., and Humphreys, T. E., “A Blueprint for Civil GPS Navigation Message Authentication,” *Proceedings of the IEEE/ION PLANS Meeting*, May 2014.

[312] Curran, J. T. and Paonni, M., “Securing GNSS: An end-to-end feasibility study for the Galileo open service,” *International Technical Meeting of the Satellite Division of The Institute of Navigation, ION GNSS*, 2014, pp. 1–15.

[313] Fernández-Hernández, I., Rijmen, V., Seco-Granados, G., Simon, J., Rodríguez, I., and Calle, J. D., “A Navigation Message Authentication Proposal for the Galileo Open Service,” *Navigation*, Vol. 63, No. 1, 2016, pp. 85–102.

[314] Chino, K., Manandhar, D., and Shibasaki, R., “Authentication technology using QZSS,” *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014*, IEEE, 2014, pp. 367–372.

[315] Lo, S. C. and Enge, P. K., “Authenticating aviation augmentation system broadcasts,” *IEEE/ION Position, Location and Navigation Symposium*, IEEE, 2010, pp. 708–717.

[316] Neish, A., Walter, T., and Enge, P., “Quantum-resistant authentication algorithms for satellite-based augmentation systems,” *Navigation*, Vol. 66, No. 1, 2019, pp. 199–209.

[317] Neish, A., Walter, T., and Powell, J. D., “Design and analysis of a public key infrastructure for SBAS data authentication,” *Navigation*, Vol. 66, No. 4, 2019, pp. 831–844.

[318] Gutierrez, P., “Galileo to Transmit Open Service Authentication,” *Inside GNSS*, 2020.

[319] Dovis, F., Luciano, M., Motella, B., and Falletti, E., *GNSS interference threats and countermeasures*, chap. Classification of Interfering Sources and Analysis of the Effects on GNSS Receivers, Artech House, 2015, pp. 31–66.

[320] Perrig, A., Canetti, R., Tygar, J., and Song, D., “The TESLA broadcast authentication protocol,” *RSA CryptoBytes*, Vol. 5, No. 2, 2002, pp. 2–13.

[321] NIST, “Recommendation for Key Management—Part I: General (Revised),” SP 800-57, National Institute of Standards and Technology, July 2012.

[322] Dang, Q., “Recommendation for Applications Using Approved Hash Algorithms (Revised),” SP 800-107, National Institute of Standards and Technology, Aug. 2007.

[323] Caparra, G., Sturaro, S., Laurenti, N., and Wullems, C., “Evaluating the security of one-way key chains in TESLA-based GNSS Navigation Message Authentication schemes,” *2016 International Conference on Localization and GNSS (ICL-GNSS)*, IEEE, 2016, pp. 1–6.

[324] Meiyappan, S., Raghupathy, A., and Pattabiraman, G., *Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications*, Vol. 2, chap. Position, Navigation and Timing with Dedicated Metropolitan Beacon Systems, Wiley-IEEE, 2020, pp. 1225–1241.

[325] Kor, R. X., Iannucci, P. A., Narula, L., and Humphreys, T. E., “A Proposal for Securing Terrestrial Radio-Navigation Systems,” *Proceedings of the ION GNSS+ Meeting*, Online, 2020.

[326] Hegarty, C., “Analytical model for GNSS receiver implementation losses,” *Navigation, Journal of the Institute of Navigation*, Vol. 58, No. 1, 2011, pp. 29.

[327] Dempster, A. G. and Cetin, E., “Interference Localization for Satellite Navigation Systems,” *Proceedings of the IEEE*, Vol. 104, No. 6, June 2016, pp. 1318–1326.

[328] Akos, D. M., “Who’s Afraid of the Spoofer? GPS/GNSS Spoofing Detection via Automatic Gain Control (AGC),” *Navigation, Journal of the Institute of Navigation*, Vol. 59, No. 4, 2012, pp. 281–290.

[329] Egea-Roca, D., Seco-Granados, G., and López-Salcedo, J. A., “Comprehensive overview of quickest detection theory and its application to GNSS threat detection,” *Gyroscopy and Navigation*, Vol. 8, No. 1, 2017, pp. 1–14.

[330] Broumandan, A., Kennedy, S., and Schleppe, J., “Demonstration of a Multi-Layer Spoofing Detection Implemented in a High Precision GNSS Receiver,” *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, IEEE, 2020, pp. 538–547.

[331] Cavaleri, A., Pini, M., Presti, L. L., Fantino, M., Boella, M., and Ugazio, S., “Signal quality monitoring applied to spoofing detection,” *Proceedings of the ION GNSS Meeting*, 2011.

[332] Hegarty, C., Odeh, A., Shallberg, K., Wesson, K., Walter, T., and Alexander, K., “Spoofing detection for airborne GNSS equipment,” *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, 2018, pp. 1350–1368.

[333] Kor, R. X. T., *A Comprehensive Proposal for Securing Radionavigation Systems*, Master’s thesis, The University of Texas at Austin, 2021.

[334] Narula, L. and Humphreys, T. E., “Requirements for Secure Clock Synchronization,” *IEEE Journal of Selected Topics in Signal Processing*, Vol. 12, No. 4, Aug. 2018, pp. 749–762.

[335] Humphreys, T. E., “Detection Strategy for Cryptographic GNSS Anti-Spoofing,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, No. 2, 2013, pp. 1073–1090.

[336] Rocken, C. and Meertens, C., “Monitoring selective availability dither frequencies and their effect on GPS data,” *Bulletin géodésique*, Vol. 65, No. 3, 1991, pp. 162–169.

[337] Van Trees, H. L., *Detection, Estimation, and Modulation Theory*, Wiley, 2001.

[338] Gross, J. and Humphreys, T. E., “GNSS Spoofing, Jamming, and Multipath Interference Classification using a Maximum-Likelihood Multi-Tap Multipath Estimator,” *Proceedings of the ION International Technical Meeting*, Jan. 2017.

[339] Blanco-Delgado, N. and Nunes, F. D., “Multipath estimation in multicorrelator GNSS receivers using the maximum likelihood principle,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 48, No. 4, 2012, pp. 3222–3233.

[340] Rappaport, T. S. et al., *Wireless communications: principles and practice*, Vol. 2, prentice hall PTR New Jersey, 1996.

[341] Seo, H.-S., Noh, D.-G., Lee, C.-J., and Lee, S.-S., “Design and implementation of intersection movement assistant applications using V2V communications,” *2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2013, pp. 49–50.

[342] Campbell, S., O’Mahony, N., Krpalcova, L., Riordan, D., Walsh, J., Murphy, A., and Ryan, C., “Sensor Technology in Autonomous Vehicles : A review,” *2018 29th Irish Signals and Systems Conference (ISSC)*, 2018, pp. 1–4.

[343] Tian, Z., Cai, Y., Huang, S., Hu, F., Li, Y., and Cen, M., “Vehicle tracking system for intelligent and connected vehicle based on radar and V2V fusion,” *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 6598–6603.

[344] Shen, J., Won, J. Y., Chen, Z., and Chen, Q. A., “Drift with Devil: Security of {Multi-Sensor} Fusion based Localization in {High-Level} Autonomous Driving under {GPS} Spoofing,” *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 931–948.

[345] Tippenhauer, N. O., Pöpper, C., Rasmussen, K. B., and Capkun, S., “On the requirements for successful GPS spoofing attacks,” *Proceedings of the 18th ACM conference on Computer and communications security*, ACM, 2011, pp. 75–86.

[346] Humphreys, T. E., Ledvina, B. M., Psiaki, M. L., O'Hanlon, B. W., Kintner, P. M., et al., "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, 2008, pp. 2314–2325.

[347] Ju, Z., Zhang, H., Li, X., Chen, X., Han, J., and Yang, M., "A Survey on Attack Detection and Resilience for Connected and Automated Vehicles: From Vehicle Dynamics and Control Perspective," *IEEE Transactions on Intelligent Vehicles*, Vol. 7, No. 4, 2022, pp. 815–837.

[348] Wang, Y., Masoud, N., and Khojandi, A., "Real-time sensor anomaly detection and recovery in connected automated vehicle sensors," *IEEE transactions on intelligent transportation systems*, Vol. 22, No. 3, 2020, pp. 1411–1421.

[349] Yang, Z., *An Infrastructure-based Cooperative Driving Framework for Connected and Automated Vehicles*, Ph.D. thesis, 2022.

[350] Zhang, R., Zou, Z., Shen, S., and Liu, H. X., "Design, Implementation, and Evaluation of a Roadside Cooperative Perception System," *Transportation Research Record*, 2022, pp. 03611981221092402.

[351] Parkinson, S., Ward, P., Wilson, K., and Miller, J., "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE transactions on intelligent transportation systems*, Vol. 18, No. 11, 2017, pp. 2898–2915.

[352] Narain, S., Ranganathan, A., and Noubir, G., "Security of GPS/INS based on-road location tracking systems," *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2019, pp. 587–601.

[353] Liu, Q., Mo, Y., Mo, X., Lv, C., Mihankhah, E., and Wang, D., "Secure pose estimation for autonomous vehicles under cyber attacks," *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 1583–1588.

[354] van Wyk, F., Wang, Y., Khojandi, A., and Masoud, N., "Real-Time Sensor Anomaly Detection and Identification in Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21, No. 3, 2020, pp. 1264–1276.

[355] Ju, Z., Zhang, H., and Tan, Y., "Distributed Deception Attack Detection in Platoon-Based Connected Vehicle Systems," *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 5, 2020, pp. 4609–4620.

[356] He, X., Hashemi, E., and Johansson, K. H., "Distributed control under compromised measurements: Resilient estimation, attack detection, and vehicle platooning," *Automatica*, Vol. 134, 2021, pp. 109953.

[357] Maile, M., Chen, Q., Brown, G., and Delgrossi, L., "Intersection Collision Avoidance: From Driver Alerts to Vehicle Control," *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015, pp. 1–5.

[358] Oza, P. and Patel, V. M., "One-class convolutional neural network," *IEEE Signal Processing Letters*, Vol. 26, No. 2, 2018, pp. 277–281.

[359] of Highway Safety Planning, M. O., "Michigan Traffic Crash Facts," .

[360] Humphreys, T., "Interference," *Springer Handbook of Global Navigation Satellite Systems*, edited by P. Teunissen and O. Montenbruch, chap. 16, Springer, 2017, pp. 469–504.

[361] Gao, G., Sgammini, M., Lu, M., and Kubo, N., "Protecting GNSS receivers from jamming and interference," *Proceedings of the IEEE*, Vol. 104, No. 6, 2016, pp. 1327–1338.

[362] Rustamov, A., Gogoi, N., Minetto, A., and Dovis, F., "Assessment of the vulnerability to spoofing attacks of GNSS receivers integrated in consumer devices," *2020 International Conference on Localization and GNSS (ICL-GNSS)*, 2020.

[363] Moore, S., "Super-accurate GPS coming to smartphones in 2018," *IEEE Spectrum*, Vol. 54, No. 11, 2017, pp. 10–11.

[364] Roumeliotis, S. and Beke, G., "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, 2002, pp. 781–795.

[365] Bryson, M. and Sukkarieh, S., "Architectures for Cooperative Airborne Simultaneous Localisation and Mapping," *Journal of Intelligent and Robotic Systems*, Vol. 55, No. 4-5, 2009, pp. 267–297.

[366] Grejner-Brzezinska, D., Toth, C., Li, L., Park, J., Wang, X., Sun, H., Gupta, I., Huggins, K., and Zheng, Y., "Positioning in GPS-challenged environments: dynamic sensor network with distributed GPS aperture and internodal ranging signals," *Proceedings of the 22nd International Technical Meeting of The Institute of Navigation*, 2009, pp. 111–123.

[367] MacGougan, G., O'Keefe, K., and Klukas, R., "Ultra-Wideband Ranging Precision and Accuracy," *Measurement Science and Technology*, Vol. 20, No. 9, 2009, pp. 095105.

[368] Vydhyanathan, A., Luinge, H., Tanigawa, M., Dijkstra, F., Braasch, M., and de Haag, M. U., "Augmenting Low-Cost GPS/INS with Ultra-Wideband Transceivers for Multi-Platform Relative Navigation," *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation*, 2009, pp. 547–554.

[369] Xiong, J., Cheong, J., Xiong, Z., Dempster, A., Tian, S., and Wang, R., "Integrity for multi-sensor cooperative positioning," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 2, 2021, pp. 792–807.

[370] Jekeli, C., *Inertial Navigation Systems with Geodetic Applications*, Walter de Gruyter, Berlin, New York, 2001.

[371] Maaref, M., Khalife, J., and Kassas, Z., "Integrity monitoring of LTE signals of opportunity-based navigation for autonomous ground vehicles," *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation*, 2018, pp. 2456–2466.

[372] Zhu, N., Betaille, D., Marais, J., and Berbineau, M., "GNSS Integrity Monitoring Schemes for Terrestrial Applications in Harsh Signal Environments," *IEEE Intelligent Transportation Systems Magazine*, Vol. 12, No. 3, 2020, pp. 81–91.

[373] Schaffrin, B. and Snow, K., "Advanced Adjustment Notes," 2017, visited on 2022-06-15.

[374] Retscher, G., Kealy, A., Gabela, J., Li, Y., Goel, S., Toth, C., Masiero, A., Blaszczak-Bak, W., Gikas, V., Perakis, H., Koppanyi, Z., and Grejner-Brzezinska, D., "A benchmarking measurement campaign in GNSS-denied/challenged indoor/outdoor and transitional environments," *Journal of Applied Geodesy*, Vol. 14, No. 2, 2020, pp. 215–229.