sensors

MDPI

Article

A GPS Spoofing Generator Using an Open Sourced Vector Tracking-Based Receiver

Qian Meng [1], Li-Ta Hsu [1,*], Bing Xu [1], Xiapu Luo [2] and Ahmed El-Mowafy [3]

1 Interdisciplinary Division of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong; qian2019.meng@polyu.edu.hk (Q.M.); pbing.xu@polyu.edu.hk (B.X.)
2 Department of Computing, The Hong Kong Polytechnic University, Hong Kong; daniel.xiapu.luo@polyu.edu.hk
3 School of Earth and Planetary Sciences, Curtin University, 6102 Perth, Australia; a.el-mowafy@curtin.edu.au
* Correspondence: lt.hsu@polyu.edu.hk

Received: 6 August 2019; Accepted: 12 September 2019; Published: 16 September 2019

Abstract: Spoofing can seriously threaten the use of the Global Positioning System (GPS) in critical applications such as positioning and navigation of autonomous vehicles. Research into spoofing generation will contribute to assessment of the threat of possible spoofing attacks and help in the development of anti-spoofing methods. However, the recent commercial off-the-shelf (COTS) spoofing generators are expensive and the technology implementation is complicated. To address the above problem and promote the GPS safety-critical applications, a spoofing generator using a vector tracking-based software-defined receiver is proposed in this contribution. The spoofing generator aims to modify the raw signals by cancelling the actual signal component and adding the spoofing signal component. The connections between the spreading code and carrier, and the states of the victim receiver are established through vector tracking. The actual signal can be predicted effectively, and the spoofing signal will be generated with the spoofing trajectory at the same time. The experimental test results show that the spoofing attack signal can effectively mislead the victim receiver to the designed trajectory. Neither the tracking channels nor the positioning observations have abnormal changes during this processing period. The recent anti-spoofing methods cannot detect this internal spoofing easily. The proposed spoofing generator can cover all open-sky satellites with a high quality of concealment. With the superiority of programmability and diversity, it is believed that the proposed method based on an open source software-defined receiver has a great value for anti-spoofing research of different GNSS signals.

Keywords: spoofing generator; GPS; vector tracking; autonomous vehicles

1. Introduction

Autonomous vehicles require an extremely accurate, robust, and reliable navigation system [1,2]. Global Navigation Satellite Systems (GNSSs), such as Global Positioning System (GPS) receivers are heavily relied upon in current autonomous vehicular navigation solutions. However, it is well-known that GPS is vulnerable to interference, such as multipath, jamming, and spoofing [3,4]. The impacts of multipath and jamming can result in a positioning error of several tens of meters or even cause the malfunction of GPS receivers [5,6]. Different from multipath and jamming, spoofing signals are intentionally designed to mislead GPS receivers to fake navigation solutions by generating fabricated synchronized navigation signals [7]. Spoofing seriously limits the use of GPS in applications related to life safety such as autonomous vehicles [8]. Although most GPS receivers have a function to detect and exclude faults, such as receiver autonomous integrity monitoring (RAIM), the need for redundant observations to perform a consistency check still limits its capability in performing anti-spoofing [9,10].

A recent research test on a commercial autopilot system revealed that when facing a spoofing attack implemented by commercially available hardware and software, the vehicle was vulnerable and was spoofed off its intended route easily [11]. This test proved beyond doubt the crucial dependence on GPS for any level 2+ autonomous navigation and the high threat spoofing poses to drivers and passengers utilizing this system.

To generate the spoofing signal, the methods can be broadly divided into meaconing, simulator-based spoofing, and receiver-based spoofing [12]. In meaconing, the GPS signals are recorded and simply replayed after a set delay. This basic meaconing technique, while capable of spoofing encrypted signals, cannot generate an arbitrary trajectory. In simulator-based spoofing, a GPS simulator is used to replicate the signals as they would appear at a chosen location, misleading the receiver to produce an incorrect position, velocity and time (PVT) solution. However, besides the high cost of a commercial signal generator, the software and hardware are not easy to be updated with the development of new signals, channel structures, and navigation message coding rules. In receiver-based spoofing, the receiver processes the actual signals to extract the accurate position and ephemeris. Then the spoofing signals can be generated with the code phase and Doppler shift matching the victim ones at the spoofing position. An advanced receiver-based spoofing technique, which is referred to as nulling, tries to transmit two signals to the victim receiver. One is the spoofing attack signal and the other is the negative of the actual signal. For the signal received by the victim receiver, the actual signal component is cancelled out and only the spoofing component is left. The threat of this spoofing attack is enormous. However, the nulling attack is extremely difficult to be implemented due to exact carrier phase alignment and amplitude matching [13]. In recent research, a way to convert a software-defined receiver (SDR) into a GPS software transceiver was proposed to reuse the sophisticated and optimized infrastructure of the software receiver for the signal generator [14]. This approach makes it possible to realize receiver-based spoofing. The key element in this approach is the usage of software receiver vector-tracking architecture to create the desired line-of-sight (LOS) parameters for updating the numerically controlled oscillator (NCO) and therefore the code and carrier replica generation [15].

Protecting GPS from spoofing is critical to autonomous vehicle navigation and understanding the spoofing mode is the first step to realizing spoofing detection. Spoofing attacks can be divided into two scenarios, an overlapped scenario and a non-overlapped scenario, according to whether the actual signal exists [16]. In a traditional overlapped spoofing scenario, the victim receiver will receive the actual signal and the spoofing signal synchronously. The correlation peak in the tracking channel is overlapped by the spoofing signal and the actual signal. To oppress the actual signal, it is necessary to modify some parameters in the spoofing signal, such as the amplitude and code delay. This kind of spoofing attack is complex and easy to be detected by signal features. Instead, in a non-overlapped scenario, the actual signal is blocked directly, and the victim receiver will receive and process the spoofing signal only. Whether based on communication technology or aided by the urban environment, this scenario is not hard to be implemented. With the recent development of communication technology, the GPS-denied technology can effectively block the actual circumstance. The actual signal will be classified as noise and the spoofing signal will take its place. Particularly, the non-overlapped scenario provides a chance to implement a nulling attack. Compared to GPS-denied technology, the complexity of the urban environment additionally provides many chances to create non-overlapped scenarios in a more natural way. Tall buildings, multi-decked roads, interchanges, and tunnels provide boundaries to block the actual signal. The 3D mapping aided (3DMA) technology can generate both multipath and non-line-of-sight (NLOS) signal interference to facilitate this kind of spoofing [17,18].

Extending from the above spoofing attack on autonomous vehicles, hacker cyberattacks are hazardous and should not be neglected [19], where the non-overlapped scenario still can be created even after the raw signal has been collected by the antenna. The developing hacker cyberattacks make it so that infiltrating the electronic control units and implanting the spoofing signal component are no longer a plot in science fiction or Hollywood movies. The actual signal component will be cancelled

and modified to a spoofing signal directly before baseband processing. This internal spoofing solution is more hazardous and concealed compared to external spoofing attacks. The recent anti-spoofing technologies are less able to overwhelm it.

Many methods have been proposed for spoofing detection, for example, the cryptographic signal method [20–22], the multi-sensor aided method [23–25], the antenna aided method [26–28], and the signal features method [29–31]. All these spoofing detection methods show limitations to detection of the non-overlapped spoofing attack, where it can be easily concealed as it does not need to change the signal power or C/N_0 to suppress the actual signal. The implementation of cryptographic methods is not feasible for civil GPS signals at present. The multi-sensor method is based on the performance of information fusion and the support of various hardware. The aiding sensors also have their limitations in application scenarios, for instance, the vision system cannot work at night. The multi-sensor aided method is not able to work under only receiver available circumstances. The antenna array method is based on more than one antenna and its implementation technology is complicated. In the signal features method, the features of the spoofing signal are quite similar to those of the actual signal and there is no sudden change in the transition process; but still, the signal feature method has not proven to work well. In addition, some crossing methods were proposed to detect spoofing, for instance machine learning [32], maximum likelihood estimation [33], and cooperation of multiple detections [34]. However, these methods are still dependent on prior information or actual signal features [35].

Furthermore, for a spoofing generator under a non-overlapped scenario, although the actual signal is no longer considered, it is still a key question to connect the actual signal seamlessly at the transition moment. It is easy to detect if out-of-lock happens or if the signal features are different from those of the previous actual signals. On the other hand, creating a vivid spoofing signal almost the same as the actual signal is much more harmful to autonomous vehicles and thus is more helpful to spoofing detection research. In this paper, a GPS spoofing generator based on actual raw signal is proposed. The suggested generator is implemented using the open sourced vector tracking on the SDR platform [36]. Code phase and carrier frequency are generated using a vector delay frequency lock loop (VDFLL) architecture. The proposed spoofing method is suitable for nulling an attack under a non-overlapped scenario. The functional implementation is shown in Figure 1. Firstly, the generator will track the actual signal synchronously to extract the ephemeris of visible satellites, their signal amplitude, and other parameters. Then, the generator will predict the actual signal in the next epoch and generate the cancellation component. At the same time, the spoofing trajectory will be converted to the corresponding spreading code frequency and carrier frequency to generate the spoofing signal component. Finally, the cancellation signal component and spoofing signal component will be combined as the attack signal. The proposed spoofing attack can be launched via a GPS-denied strategy or by using a 3DMA multipath interference approach. In the development of future cyberattacks, the hacker will be able to plant the attack signal into the raw signal. The contributions of this method include two 'consistency' and one 'expansibility' criteria. The first consistency criterion is that the spoofing signal is generated by modifying the actual signal. The signal power, code phase, and carrier phase are extracted from the actual tracking outputs. The signal features keep consistency with the actual signal. The second consistency criterion is that the proposed method is based on a vector tracking receiver. It can take advantage of the relationship between loop information and receiver states to attack visible satellites to preserve observation consistency. The 'expansibility' criterion refers to the detailed implementation based on an open sourced receiver being given. In general, the method is easy to implement and extend to different kinds of satellite navigation systems and signal structures.

The rest of the paper is organized as follows: The design of vector tracking is introduced in Section 2. After that, details about the actual signal prediction and spoofing signal generation based on actual raw signal are given in Section 3. Next, in Section 4, the experimental test evaluating the performance of the proposed spoofing method and the hidden characteristic of the proposed method is analyzed. Finally, Section 5 draws the conclusion.

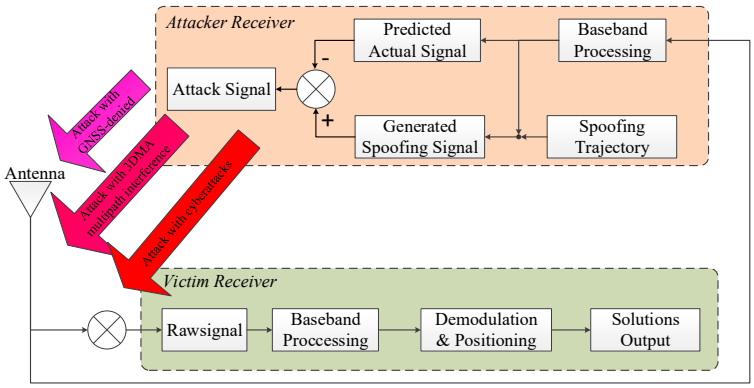

Figure 1. Functional diagram of internal spoofing generator. GNSS is global navigation satellite system; 3DMA is 3D mapping aided.

2. Spoofing Attack Using Vector Tracking

Vector-tracking is an advanced signal tracking technology, different from the traditional signal tracking, in which all tracking channels are independent to each other and no information exchange is performed between signal tracking. The channels in a vector-tracking receiver are coupled together through the navigation processor. The vector-tracking shows superiority in performance under harsh environments, e.g., increased capabilities against weak signal or high dynamic conditions. In recent years, with the increasing development of intelligent transportation systems and location-based services in urban canyon areas, vector-tracking shows more potential superiorities. For example, vector-tracking is applied to multipath or NLOS reception mitigation in the signal processing stage [37,38]. The fundamental principle behind vector-tracking is the relationship between the code or carrier phase and the receiver states of position, velocity, and time. It gives a feasible opportunity to generate spoofing signals with the given receiver trajectory, as suggested in [14].

In this paper, we use vector-tracking architecture to implement the spoofing attack. From the aspect of demodulating the actual signals, the vector-tracking SDR can track the actual code and carrier much more accurate and robust in urban environments. From the aspect of modulating the spoofing signal, the vector-tracking has the function of converting the predicted receiver position and velocity to the corresponding code frequency and carrier frequency. The detailed implementation architecture is shown in Figure 2. It includes three blocks: tracking channel, actual signal prediction, and spoofing signal generation. All these three blocks are connected with an extended Kalman filter (EKF).

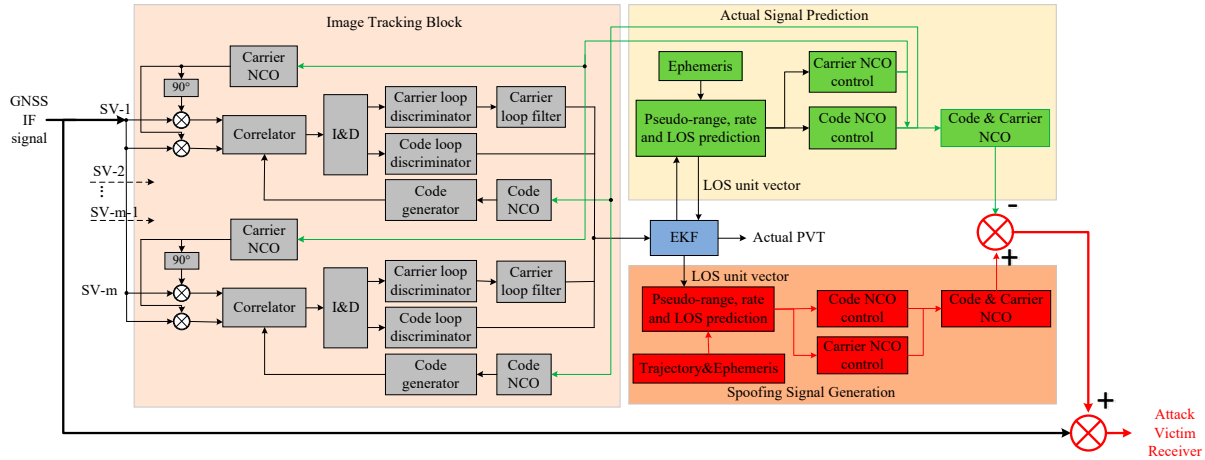

Figure 2. Implementation architecture of the proposed spoofing generator based on vector tracking. 'SV-m' represents the m-th satellite. 'I & D' means the In-phase and quadrature tracking branches.

The EKF estimates the actual PVT based on its system propagation and the measurements. After obtaining the navigation solution, the pseudo-range and its rate and the line-of-sight (LOS) vector between the receiver and the satellites are predicted. To do this, the satellite ephemeris data must be known a priori, which means the attacker should process the actual signal and decode the ephemeris data first. The state vector of the EKF is:

$$\mathbf{X} = \left[\Delta p_x, \Delta p_y, \Delta p_z, \Delta v_x, \Delta v_y, \Delta v_z, \Delta b, \Delta d\right]^T \tag{1}$$

where $\left[\Delta p_x, \Delta p_y, \Delta p_z\right]$ and $\left[\Delta v_x, \Delta v_y, \Delta v_z\right]$ are the three-dimensional receiver position and velocity error vectors in an earth-centered and earth-fixed (ECEF) frame; Δb and Δd are the receiver clock bias and drift in the units of m and m/s, respectively. The system propagation at epoch k is:

$$\hat{\mathbf{X}}_k^- = \mathbf{\Phi}_{k-1}\hat{\mathbf{X}}_{k-1}^+ \tag{2}$$

where

$$\mathbf{\Phi}_{k-1} = \begin{bmatrix} \mathbf{I}_{3\times3} & \tau\mathbf{I}_{3\times3} & 0_{3\times2} \\ 0_{3\times3} & \mathbf{I}_{3\times3} & 0_{3\times2} \\ 0_{2\times3} & 0_{2\times3} & \mathbf{K} \end{bmatrix}_{8\times8} \tag{3}$$

$$\mathbf{K} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}. \tag{4}$$

In Equation (2), τ is the update interval of the EKF. The superscripts "–" and "+" denote the system state before and after measurement update, respectively. The symbol " ˆ " represents the EKF estimates. $\mathbf{I}_{m\times n}$ represents the identity matrix of $(m \times n)$.

The measurement vector can be expressed as

$$\mathbf{Z} = \left[\Delta\rho^j, \Delta\dot{\rho}^j\right] \tag{5}$$

where $\Delta\rho^j$ and $\Delta\dot{\rho}^j$ are the pseudo-range error and pseudo-range rate error of satellite j, respectively. The detailed calculation method will be given in the following section.

The relationship between the state vector and the measurement vector at epoch k is linearized by a first-order Taylor's expression as follows:

$$\mathbf{Z}_k = \mathbf{H}_k \cdot \mathbf{X}_k \tag{6}$$

where $\mathbf{H}$ is the measurement matrix, calculated as

$$\mathbf{H} = \begin{bmatrix} -\mathbf{l}_x^1 & -\mathbf{l}_y^1 & -\mathbf{l}_z^1 & 0 & 0 & 0 & 1 & 0 \\ -\mathbf{l}_x^2 & -\mathbf{l}_y^2 & -\mathbf{l}_z^2 & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{l}_x^m & -\mathbf{l}_y^m & -\mathbf{l}_z^m & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\mathbf{l}_x^1 & -\mathbf{l}_y^1 & -\mathbf{l}_z^1 & 0 & 1 \\ 0 & 0 & 0 & -\mathbf{l}_x^2 & -\mathbf{l}_y^2 & -\mathbf{l}_z^2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -\mathbf{l}_x^m & -\mathbf{l}_y^m & -\mathbf{l}_z^m & 0 & 1 \end{bmatrix} \tag{7}$$

where m is the number of satellites involving positioning; the subscript of the LOS unit vector denotes its x, y, and z components, and the superscript denotes the satellite.

The process noise comes from two sources, the receiver dynamics and clock noise, as follows:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{dyn} & 0_{6\times2} \\ 0_{2\times6} & \mathbf{Q}_{clk} \end{bmatrix}. \tag{8}$$

The values of $\mathbf{Q}_{dyn}$ and $\mathbf{Q}_{clk}$ can be set empirically according to the expected receiver motion state and the oscillator used. Alternatively, they can be calculated as

$$\mathbf{Q}_{dyn} = \begin{bmatrix} \tau^3/3\cdot\mathbf{I}_{3\times3} & \tau^2/2\cdot\mathbf{I}_{3\times3} \\ \tau^2/2\cdot\mathbf{I}_{3\times3} & \tau\cdot\mathbf{I}_{3\times3} \end{bmatrix}\cdot S_v \tag{9}$$

$$\mathbf{Q}_{clk} = \begin{bmatrix} S_f\cdot\tau + S_g\tau^3/3 & S_g\tau^2/2 \\ S_g\tau^2/2 & S_g\cdot\tau \end{bmatrix} \tag{10}$$

where S_v is the receiver velocity noise power spectral density (PSD); S_f and S_g are the PSD of receiver clock phase and frequency, respectively. The value of S_v should be set according to the expected level of dynamics. Settings of S_f and S_g are usually based on the rule of thumb values of the type of oscillator used, or calculated using the following formulas:

$$S_f = c^2\cdot\frac{h_0}{2} \tag{11}$$

$$S_g = c^2\cdot2\pi^2\cdot h_{-2} \tag{12}$$

where h_0 and h_{-2} are the coefficients of white frequency modulation noise and flicker frequency modulation noise of the oscillator used, respectively.

The measurement noise covariance matrix is calculated adaptively using the innovation-based adaptive estimation technique. The measurement innovation at epoch k in this paper is

$$\mathbf{V}_k = \mathbf{Z}_k - \mathbf{Z}_k^- \tag{13}$$

$$\mathbf{Z}_k^- = \mathbf{H}_k\hat{\mathbf{X}}_k^-. \tag{14}$$

The diagonal element of the measurement covariance matrix is the variance of the measurement innovation. The off-diagonal terms are assumed to be zero due to the weak correlation between channels.

3. Actual Signal Prediction and Spoofing Signal Generation

The implementation details of the EKF used in this GPS signal generator are described above. This section will take the advantage of vector tracking to control the local code and carrier generation in two different scenarios: actual signal prediction and spoofing signal generation. Then, the final attacking signal is given after that.

In actual signal prediction, the code NCO control algorithm is implemented using the estimated navigation solution as:

$$\widetilde{f}^j_{code,k+1} = f_{CA}\left[1 - \frac{\widetilde{\rho}^j_{k+1} - \hat{\rho}^j_k}{c\tau}\right] \tag{15}$$

where $\widetilde{\rho}^j_{k+1}$ and $\hat{\rho}^j_k$ are the predicted pseudo-range at epoch $k+1$ and the estimated pseudo-range at epoch k; f_{CA} is the code chipping rate (e.g., 1.023 MHz for GPS L1 C/A); c denotes the speed of light. The predicted pseudo-range is calculated using

$$\widetilde{\rho}^j_{k+1} = \|\widetilde{\mathbf{r}}_{u,k+1} - \mathbf{r}^j_{k+1}\| + \delta\hat{\rho}^j_{sv,c} + \delta\hat{\rho}^j_I + \delta\hat{\rho}^j_T - \hat{b}_{clk}. \tag{16}$$

It consists of two parts: the first part is the predicted range between satellite and receiver, where $\mathbf{r}^j_{k+1}$ is the satellite position at epoch $k+1$, which is calculated based on the broadcast ephemeris. $\widetilde{\mathbf{r}}_{u,k+1}$ is the predicted receiver position, which can be calculated based on the system propagation according to Equation (2). The second part is the pseudo-range errors, including the satellite clock error $\delta\hat{\rho}^j_{sv,c}$, ionospheric delay $\delta\hat{\rho}^j_I$, tropospheric delay $\delta\hat{\rho}^j_T$, and the estimated receiver clock bias $\hat{b}_{clk}$, respectively. The receiver clock is also obtained from the propagated EKF state vector.

$f^j_{code,k+1}$ is then fed back to the code NCO in each channel to generate local code replicas to keep tracking the actual signal.

The carrier NCO control algorithm is implemented using the predicted pseudo-range rate at epoch $k+1$ as follows:

$$\widetilde{f}^j_{doppler,k+1} = -\widetilde{\dot{\rho}}^j_{k+1}\frac{f_{L1}}{c} \tag{17}$$

where f_{L1} is the carrier frequency (1575.42 MHz for GPS L1). The predicted pseudo-range rate is calculated using

$$\widetilde{\dot{\rho}}^j_{k+1} = \left(\mathbf{v}^j_{sv,k+1} - \widetilde{\mathbf{v}}_{u,k+1}\right)\mathbf{l}^j + \hat{d}_{u,clk} - d^j_{sv,clk} \tag{18}$$

where $\widetilde{\mathbf{v}}_{u,k+1}$ and $\mathbf{v}^j_{sv,k+1}$ are the velocity vectors of the receiver and satellite j, respectively, at epoch $k+1$; $\mathbf{l}^j$ is the LOS unit vector from the receiver to satellite j; $\hat{d}_{u,clk}$ and $d^j_{sv,clk}$ are the estimated receiver clock drift and the j^{th} satellite clock drift, respectively, both in m/s.

Then, the measurement vector of EKF at epoch $k+1$ can be obtained from

$$\Delta\rho^j = \Delta\tau^j \cdot \frac{c}{f_{CA}} \tag{19}$$

$$\Delta\dot{\rho}^j_{k+1} = f^j_{doppler}\frac{c}{f_{L1}} - \left(\mathbf{v}^j_{sv,k+1} - \widetilde{\mathbf{v}}_{u,k+1}\right)\mathbf{l}^j - \hat{d}_{u,clk} + d^j_{sv,clk} \tag{20}$$

where $\Delta\tau^j$ is the code discriminator output in chips, $f^j_{Doppler}$ is the Doppler shift frequency in Hz.

The mechanism of spoofing code generation is similar to that of actual code prediction. The main difference is that the 'receiver position' and 'receiver velocity' are replaced by the spoofing trajectory. The spoofing pseudo-range and pseudo-range rates are calculated as:

$$\widetilde{\rho}^j_{spoof,k+1} = \|\mathbf{r}_{trj,k+1} - \mathbf{r}^j_{k+1}\| + \delta\hat{\rho}^j_{sv,c} + \delta\hat{\rho}^j_I + \delta\hat{\rho}^j_T - \hat{b}_{clk} \tag{21}$$

$$\widetilde{\dot{\rho}}^j_{spoof,k+1} = \left(\mathbf{v}^j_{sv,k+1} - \mathbf{v}_{trj,k+1}\right)\mathbf{l}^j + \hat{d}_{u,clk} - d^j_{sv,clk} \tag{22}$$

where $\mathbf{r}_{trj,k+1}$ and $\mathbf{v}_{trj,k+1}$ are the spoofing receiver position and velocity extracted from the spoofing trajectory. The details can be found in [14], which includes a 4th degree spline interpolation and a second extrapolation.

Attack Signal Generation

To generate a whole GPS signal, besides the code and carrier, the amplitude and navigation data are also essential. In the actual signal prediction, the navigation data is obtained from the prompt branch as

$$\hat{D}^j_{nav,actual} = r_{IF} .* C^j_{prompt} .* Carr^j_{\cos} \tag{23}$$

where r_{IF} is the raw signal, C^j_{prompt} and $Carr^j_{\cos}$ are the code and carrier in the prompt branch of the satellite j channel. Using $\hat{D}^j_{nav,actual}$ to generate the actual signal is better as it includes the Doppler residual between two successive epochs.

In spoofing signal generation, as we do not to consider the Doppler residual, the navigation data is calculated as

$$\hat{D}^{j}_{nav,spoof} = \begin{cases} 1, & if\ I_p > 0 \\ -1, & if\ I_p < 0 \end{cases} \quad where\ I_p = \sum_{i=1}^{N_{sample}} \left(r_{IF}. * C^{j}_{prompt}. * Carr^{j}_{\cos} \right) \tag{24}$$

where N_{sample} represents the number of samples in one tracking epoch.

Regarding the signal amplitude, a simple method to estimate it, as mentioned in [39], is

$$\hat{A}^{j} = \frac{\sum_{1}^{N_{sample}} \left(r_{IF}. * C^{j}_{prompt}. * \hat{D}^{j}_{nav,actual}. * Carr^{j}_{\cos} \right)}{\sum_{1}^{N_{sample}} \left(C^{j}_{prompt}. * \hat{D}^{j}_{nav,actual}. * Carr^{j}_{\cos} \right)^2}. \tag{25}$$

Finally, the attack signal is combined with the predicted actual signal component to generate the spoof signal component as

$$r_{attack} = r_{spoof} - r_{actual}. \tag{26}$$

4. Experimental Test and Analysis

Experimental tests were conducted to evaluate the performance of the proposed spoofing generator. The actual signal was collected in a field experiment in Hong Kong and the experimental vehicle platform is shown in Figure 3. The antenna was mounted on the roof of the vehicle. The hardware related to signal collection and processing are shown in Figure 4. NovAtel SPANCPT was used to provide a reference trajectory. GPS signals were collected using a Nottingham Scientific Ltd. (NSL) stereo front-end for post-processing on a mobile workstation. The sampling frequency and intermediate frequency (IF) of the front-end are 26 MHz and 6.5 MHz, respectively. The victim receiver processed the signal with a traditional tracking architecture and least squared positioning mode. The trajectory design, spoofing signal performance in positioning and channel tracking at the transition moment, and the spoofing detection results are analyzed in the following subsections.

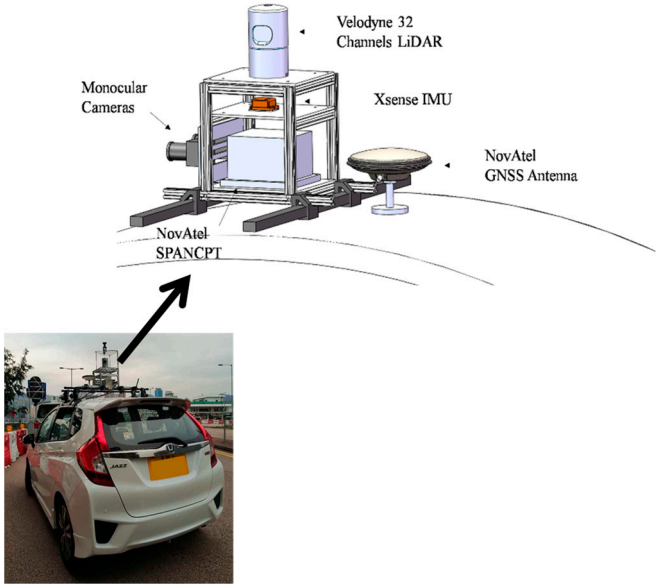

Figure 3. Experimental vehicle platform.

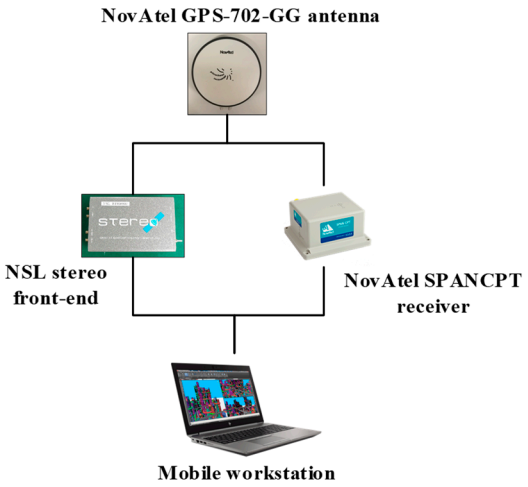

Figure 4. Hardware for signal collect and processing.

The proposed method is implemented on the SDR platform with a vector tracking architecture developed by the Positioning and Navigation Lab, Interdisciplinary Division of Aeronautical and Aviation Engineering (AAE), Hong Kong Polytechnic University [36]. The MATLAB software and the corresponding vector tracking open source codes can be downloaded on the GPS Toolbox website [40]. The modular procedure flowchart of the proposed generator execution is show in Figure 5.

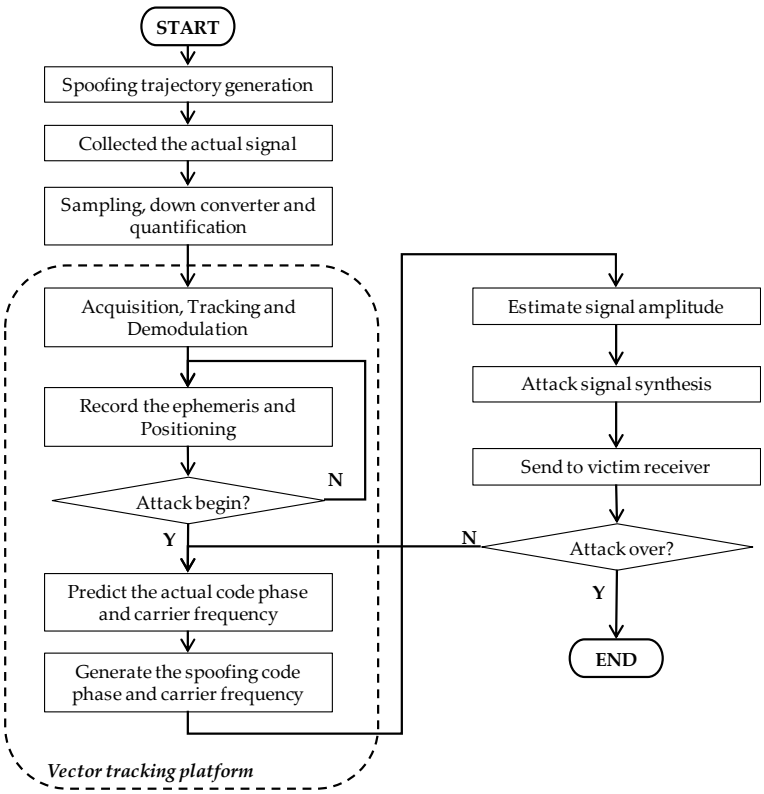

Figure 5. Flowchart of the spoofing generator based on vector tracking.

4.1. Trajectory Design

The detailed test trajectory is shown in Figure 6. The actual kinematic automobile signal was collected along the Shing Fung Road near the Kai Tak Cruise Terminal, Hong Kong. The black line is the actual trajectory. It started from the Kai Tak Cruise side, then crossed the bridge and turned to the southeast. Finally, the experiment terminated near the Hong Kong Children's Hospital. The vehicle

kept static for about 30 s before moving with a moderate speed along the coast. The whole period was about 115 s, including 115,000 positioning epochs.

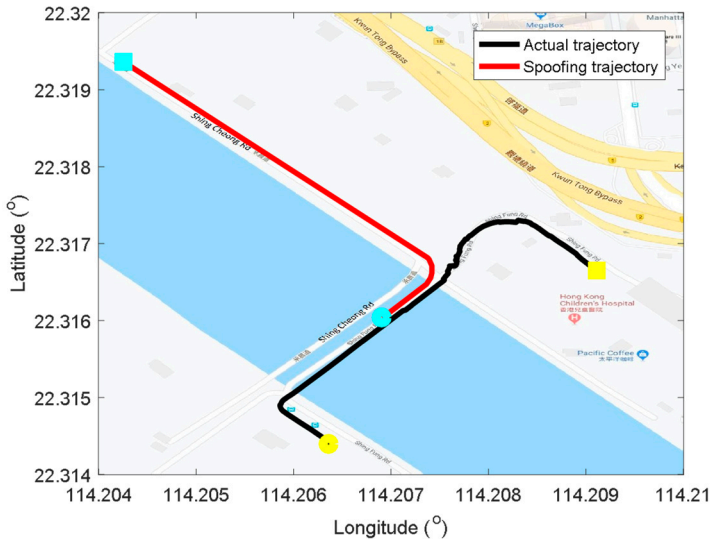

Figure 6. Sketch map for kinematic automobile trajectory. The black and red line are actual and spoofing trajectories, respectively. The starting/terminal points are shown in circle/square points of yellow and blue, respectively.

The spoofing trajectory was designed on the Google map and also plotted in the same figure as the red line. It is better to use actual roads to generate the spoofing trajectory to meet the physical road constraints of the navigation map in autonomous vehicles. It is easy to connect the spoofing trajectory with the actual trajectory at intersections. As shown in the figure, the spoofing attack was launched from the end of the bridge and aimed to guide the automobile to the Shing Cheong Road, which is parallel to the actual test road but turn to northwest at the end of the bridge. The spoofing attack was launched from the 70th second.

4.2. Performance in Positioning

The act and purpose of spoofing is not only to affect the victim receiver to output the wrong positioning solutions, but also to mislead the receiver to the spoofing trajectory. Actually, the hazard of this type of spoofing attack is much more serious compared to those of the conventional overlapped spoofing attack. The positioning outputs before and after the spoofing attack are shown in Figure 7, also plotted on a Google map.

It is within expectations that the victim receiver was spoofed off its actual trajectory successfully and turned to the Shing Cheong Road at the end of bridge. Then, it kept on working with the established trajectory. What needs to be explained is that the positioning errors under the actual signal in the last half part became bigger due to the interference caused by buildings around the hospital, while the positioning errors under the spoofing signal were small and stable thanks to a relatively open sky along the coast. It is also worth remembering that the spoofing signal generation should consider the impact of surrounding buildings to keep its fidelity, which is considered in our future work. The positioning errors related to the spoofing trajectory are also given in Figure 8, which are given in East–North–Up (ENU) coordinates. The positioning errors are defined as the differences of positioning results and the spoofing trajectory.

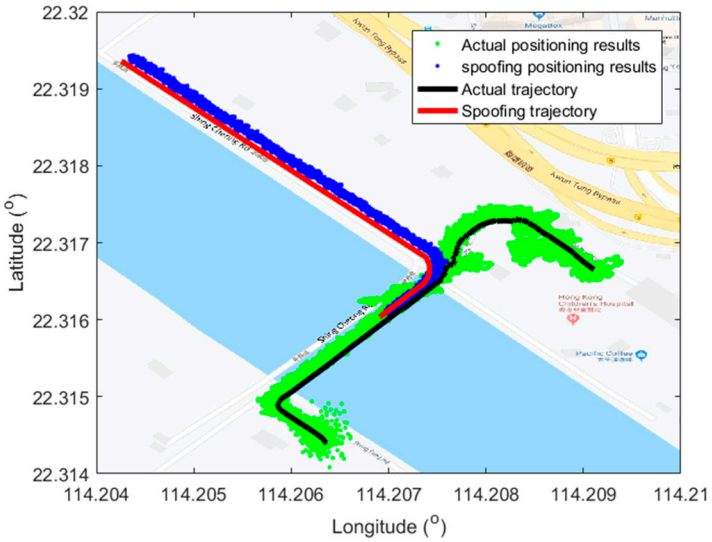

Figure 7. Positioning results plotted in Google map. The green points, blue points, black line, and red line are positioning results under actual signal, positioning results under spoofing signal, the actual trajectory, and spoofing trajectory, respectively.

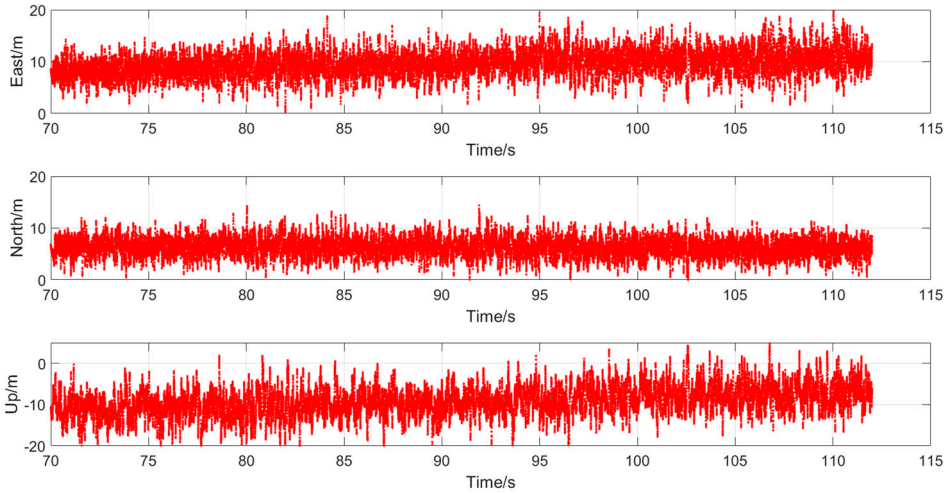

Figure 8. Positioning errors under spoofing attack. From top to bottom: positioning errors in east, north, and up component, respectively.

As shown in the Figure 7, the values of errors in the three position components kept relatively stable during the whole attack period. This verified the pseudo-range consistency of the whole visible satellites. The superiority of the proposed method was fully shown as the spoofing could cover the visible satellites. Compared to that of the up component, the positioning results in the east and north components matched the spoofing trajectory a little better. This is expected as the positioning accuracy in the horizontal direction is usually better than the vertical direction. Nevertheless, one should note that in positioning and navigation of autonomous vehicles, the horizontal results are of more interest.

4.3. Performance in Channel Tracking

To evaluate the performance of spoofing signal further, the tracking results at the transition moment are analyzed in this subsection. Three scenarios are considered in this analysis: (1) actual signal tracking, in which no attack exists; (2) attack with only actual signal cancellation, in which the attack signal only includes the predicted actual signal component; (3) attack with spoofing signal modulated, in which the attack signal not only includes the predicted actual component, but is also

combined with the generated spoofing signal component. The tracking results lasted 6 s, including 3 s before spoofing and 3 s after spoofing. The transition point was the 70th second. Figure 9, Figure 10, and Figure 11, respectively, show the outputs of prompt branch, delay lock loop (DLL) discriminator, and phase lock loop (PLL) discriminator in tracking. In every figure, the above three scenarios are presented from top to bottom. Particularly, in the 3rd scenario, the results before and after spoofing are plotted in different colors.

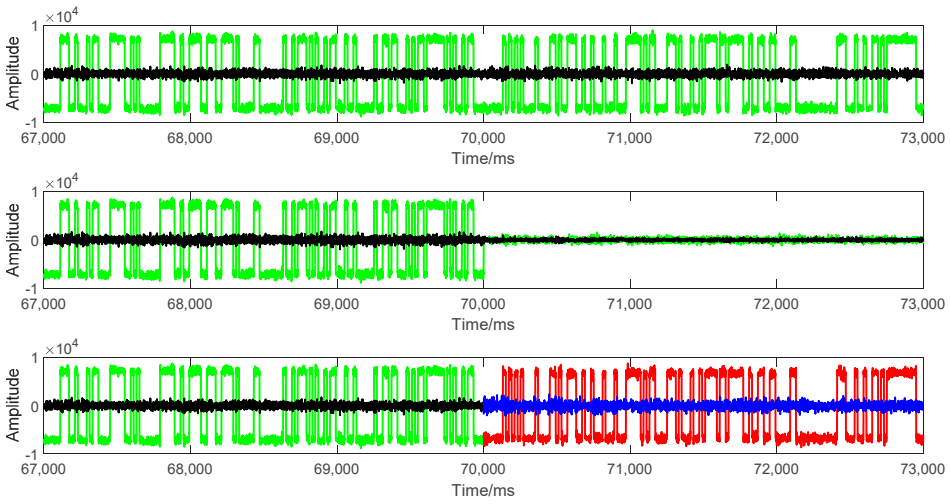

Figure 9. In-phase branch (I_p) and quadrature branch (Q_p) outputs of PRN-10 tracking in three different scenarios of signal tracking. The y-axis is the amplitude of coherent integration in 1 millisecond. From top to bottom: (top) when no attacks exist, (middle) actual signal cancelled, and (bottom) actual signal cancelled and spoofing signal modulated. Green and block points represent the I_p and Q_p outputs of actual signal, respectively. Red and blue points represent the I_p and Q_p outputs of spoofing signal, respectively.

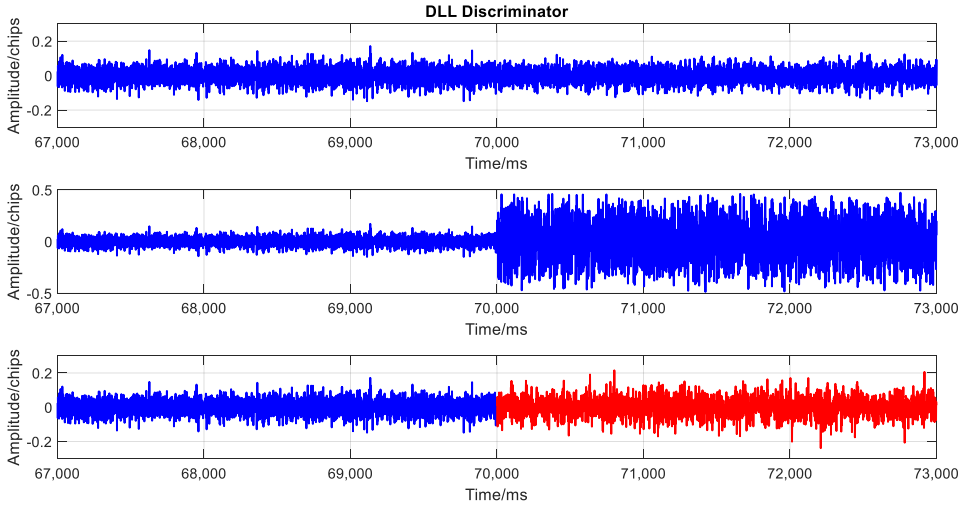

Figure 10. Delay lock loop (DLL) discriminator in the three scenarios. From top to bottom: (top) when no attacks exist, (middle) actual signal cancelled, and (bottom) actual signal cancelled and spoofing signal modulated. Blue and red points represent the outputs of actual and spoofing signal, respectively.

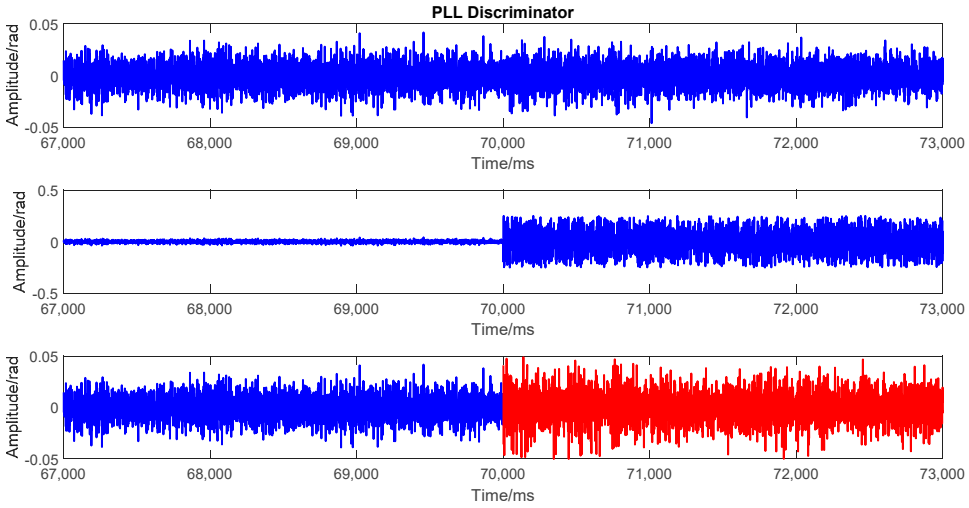

Figure 11. Phase lock loop (PLL discriminator in the three scenarios. From top to bottom: (top) when no attacks exist, (middle) actual signal cancelled, and (bottom) actual signal cancelled and spoofing signal modulated. Blue and red points represent the outputs of actual and spoofing signal, respectively.

The 2nd scenario shows the results after the actual signal was cancelled. Both the code loop and carrier loop lost lock immediately. There were only noises in the correlations of in-phase branch (I_p) and quadrature (Q_p) branch. The actual signal was demodulated and cancelled ideally. A good non-overlapped spoofing attack can be launched in this scenario.

Meanwhile, the tracking results of the 3rd scenario had no obvious difference compared with those of the 1st scenario. There was no outlier or out of lock in the code loop or carrier loop seen from Figures 10 and 11. The amplitude of the correlation outputs of the prompt branch had no significant change from the actual signal to the attack signal, which means that the signal power kept stable at the transition moment.

4.4. Hidden Characteristic for Spoofing Detection

It seems that the hidden function is the most important characteristic for spoofing attack, especially at the transition moment. The above positioning and tracking results are encouraging from this aspect as there is no abnormal change in the tracking channel after the raw signal are attacked. All the changes at the transition moment are within the receiver normal limits. The victim receiver after spoofing attack can be positioned normally with the spoofing trajectory. The anti-spoofing scheme will not be triggered in this non-overlapped scenario. The machine learning methods would not available as there is no classical spoofing features for training.

Moreover, the other widely-used methods that aim to check the pseudo-range consistency to detect spoofing attack will not be effective for the proposed spoofing approach. These methods are generally applied in the positioning domain and are based on RAIM or pseudo-range residual detection. Spoofing attacks on only one or several satellites, or spoofed signals inconsistent in different channels are easily exposed to this kind of consistency detection; however, they are ineffective when all signals are spoofed. Figures 12–14 show three representative parameters around the transition point for consistency checking. Figure 12 shows the pseudo-range residuals in all channels. Figure 13 is the test statistics based on sum of the squares of the residual errors (SSE). Figure 14 shows the maximum slope for the geometry in RAIM. The detailed calculation method of the above parameters can be found in [41].

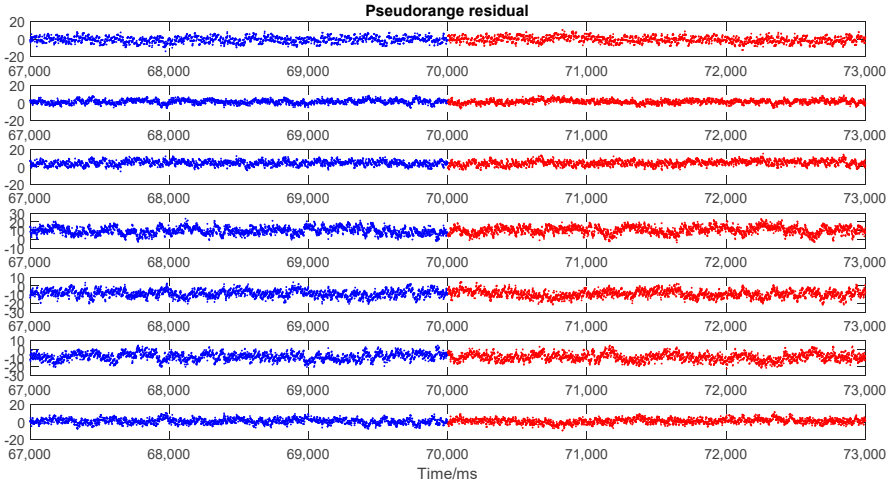

Figure 12. Pseudo-range residuals in every tracking channels (for 7 satellite observations). The blue and red points represent the outputs of actual and spoofing signal, respectively.

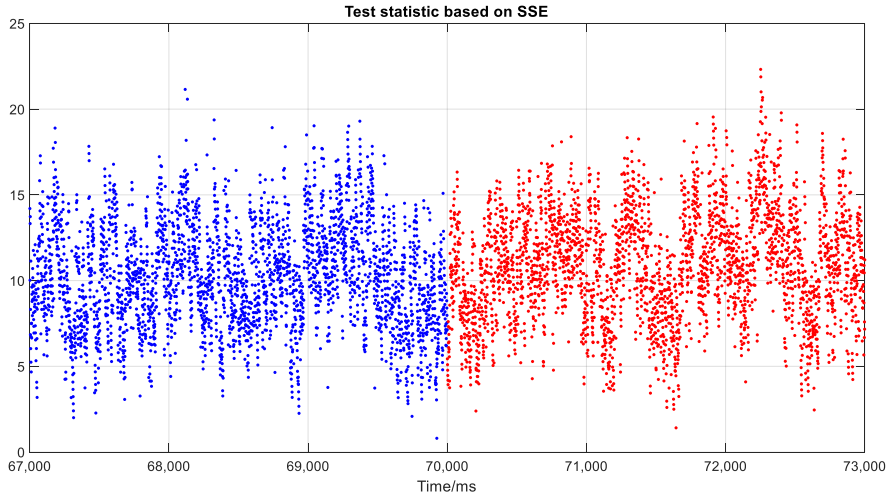

Figure 13. Test statistics based on squares of the residual errors (SSE). Blue and red points represent the outputs of actual and spoofing signal, respectively.

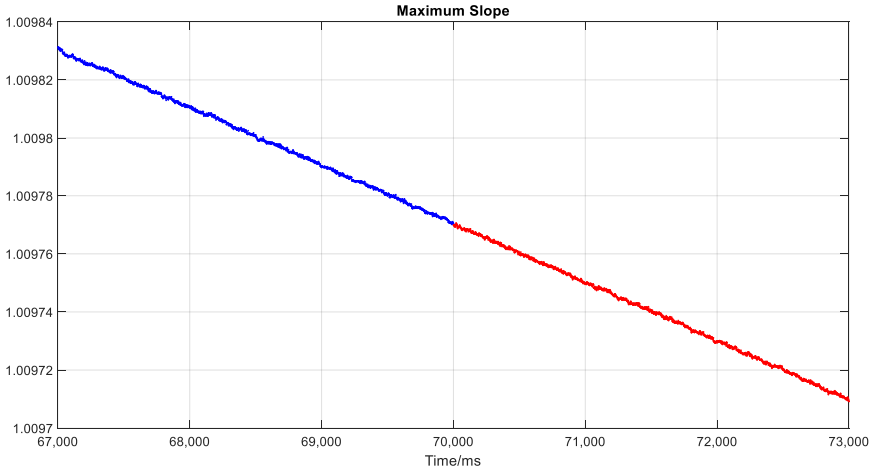

Figure 14. Maximum slope of geometry in receiver autonomous integrity monitoring (RAIM). Blue and red points represent the outputs of actual and spoofing signal, respectively.

As shown in Figure 12, although the residuals in different channels were different, there was no abnormal change around the transition point. The vector tracking proved its effectiveness as the LOS consistency could be guaranteed exactly for all visible satellites. Thus, spoofing detection based on checking consistency of pseudo-range residuals was incapable of detection of the spoof attack.

Test statistics and maximum slope are important parameters for classical RAIM fault detection and protection level check. The spoofing detection alarm in RAIM will be triggered only when the test statistics exceeds a threshold. As shown in Figure 13, there was no obvious change before and after the transition point, and the threshold was hard to be set in this circumstance. The maximum slope shown in Figure 14 also kept the same trend after the spoofing attack began, which verified the time consistency of the geometry matrix further.

5. Discussion

In the above experimental test and performance evaluation, the spoofing generator shows superiority in signal features and observation consistency. As the actual signal component has been blocked and the spoofing signal component is closely similar to that of the actual signal, it is difficult to detect this attack based on the resulting differences of tracking channels between neighbored epochs or the snapshot consistency at the present epoch.

Compared to the traditional spoofing methods, another advantage of the proposed spoofing generation method is that it is trajectory driven. The superiority of vector-tracking is well utilized to covert the spoofing trajectory to the code and carrier trends of all open sky satellites. The traditional spoofing methods cannot spoof the victim receiver to the deliberate destination as planned. As shown in Figure 15, it is the attack results under a classical repeater, which is also known as meaconing. This attack recorded the actual GNSS signal and replayed after a set delay. This kind of attack is easy to be implemented and may work well in a very short time. However, the spoofing trajectory is uncertain and easy to notice due to the urban road constraints. On the other hand, once the spoofing signal does not cover whole open sky satellites perfectly, as shown in Figure 16, it also failed to guide the victim receiver along the designed trajectory.

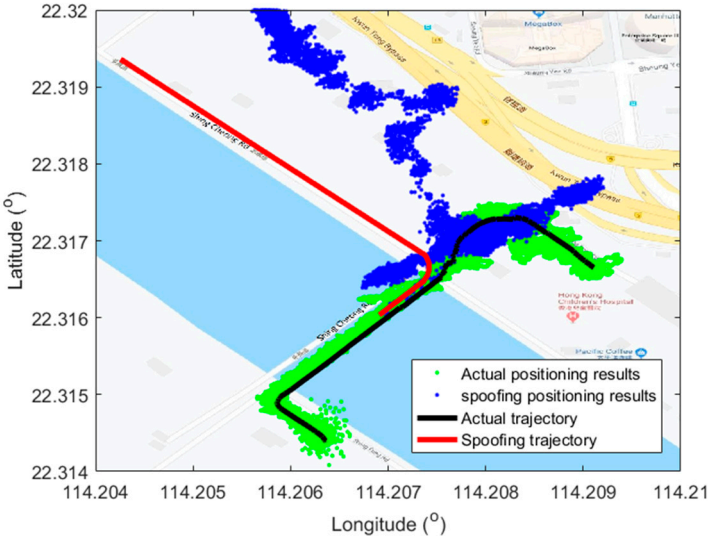

Figure 15. Positioning results under repeater attack. The green points, blue points, black line, and red line are positioning results under actual signal, positioning results under repeater spoofing signal, the actual trajectory, and spoofing trajectory, respectively.

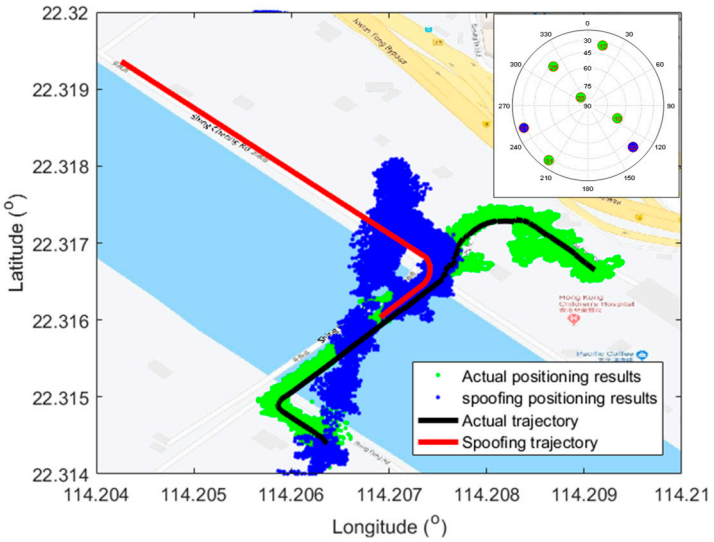

Figure 16. Positioning results under the scenario where PRN 21 and PRN 31 were not spoofed. The sky plot of satellites is shown in the top right corner, where green and blue numbers represent the satellites spoofed and not spoofed, respectively.

The limitation of the proposed spoofing generator is that this kind of spoofing is based on actual signals. It needs to track the actual signal for a period of time to calculate the visible satellites, the corresponding ephemeris, the signal power, and other useful channel features. Besides, it is applicable for non-overlapped scenarios and under only GNSS available circumstances. The actual signal arriving at the victim receiver needs to be blocked to avoid the overlapped uncertainty. The information supported from other sensors or antenna is not considered in this spoofing attack scheme. What cannot be ignored is its reliance on the vector tracking receiver. In the case that vector tracking cannot guarantee its performance, the performance of the proposed spoofing attack will be compromised as well. It is believed that advanced filtering technologies [42,43] and model selection methods [44,45] will help to improve the tracking of actual signals and prediction of spoofing signals in challenging environments.

The above results are based on the assumption that the non-overlapped scenario has been created. The researchers are researching on the non-overlapped scenario implementation based on 3DMA in urban environments and will investigate methods that can rapidly detect this advanced type of spoofing in the future work.

6. Conclusions

A GPS spoofing generator using vector tracking-based SDR is proposed in this paper. With the help of a non-overlapped scenario, the internal nulling spoofing attack is carried out by modifying the actual signal and cancelling the actual component with the spoofing component. With the superiority of SDR vector tracking architecture, it is easy to convert the spoofing trajectory to the corresponding code and carrier. The modified signal still maintains the actual amplitude, satellite ephemeris, and other important signal features. The test results show that the spoofing attack can work effectively, and the receiver was misled to the spoofed trajectory successfully. The spoofing detection methods in track channel or positioning domain have difficulty detecting this spoofing as the spoofing signal keeps high consistency in tracking features and observation pseudo-ranges. There is no abnormal change in the tracking results or positioning solutions. The threat of this spoofing mode to autonomous vehicles is hazardous once all the visible GPS satellites are spoofed.

As it is undeniable that there is an actual and urgent need to research on spoofing generators, the above spoofing generator, implemented based on an open source SDR with a mature vector tracking architecture, will help the research on spoofing defenses in the future.

Author Contributions: Conceptualization and supervision, L.-T.H.; writing and methodology, Q.M.; investigation and resources, B.X.; reviewing and editing, X.L.; reviewing and editing A.E.-M.

Funding: This research project "Security Enhancement of Positioning Sensors on Connected Autonomous Vehicles" is funded by Hong Kong Polytechnic University, grant number P0013910 (ZVP9).

Acknowledgments: The authors would like to thank Guohao Zhang for data collection.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bonnefon, J.-F.; Shariff, A.; Rahwan, I. The social dilemma of autonomous vehicles. *Science* **2016**, *352*, 1573–1576. [CrossRef] [PubMed]
2. Claybrook, J.; Kildare, S. Autonomous vehicles: No driver . . . no regulation. *Science* **2018**, *361*, 36–37. [CrossRef] [PubMed]
3. Takefuji, Y. Connected vehicle security vulnerabilities. *IEEE Technol. Soc. Mag.* **2018**, *37*, 15–18. [CrossRef]
4. Blum, R.; Dötterböck, D.; Pany, T. Investigation of the Vulnerability of Mobile Networks against Spoofing Attacks on their GNSS Timing-receiver and Developing a Meaconing Protection. In Proceedings of the 2019 International Technical Meeting of The Institute of Navigation, Reston, VA, USA, 28–31 January 2019; pp. 345–362.
5. Ioannides, R.T.; Pany, T.; Gibbons, G. Known Vulnerabilities of Global Navigation Satellite Systems, Status, and Potential Mitigation Techniques. *Proc. IEEE* **2016**, *104*, 1174–1194. [CrossRef]
6. Hsu, L.T. Analysis and modeling GPS NLOS effect in highly urbanized area. *GPS Solut.* **2018**, *22*, 7. [CrossRef]
7. Shin, B.; Park, M.; Jeon, S.; So, H.; Kim, G.; Kee, C. Spoofing Attack Results Determination in Code Domain Using a Spoofing Process Equation. *Sensors* **2019**, *19*, 293. [CrossRef] [PubMed]
8. Broumandan, A.; Lachapelle, G. Spoofing Detection Using GNSS/INS/Odometer Coupling for Vehicular Navigation. *Sensors* **2018**, *18*, 1305. [CrossRef] [PubMed]
9. Meng, Q.; Liu, J.; Zeng, Q.; Feng, S.; Xu, R. Improved ARAIM fault modes determination scheme based on feedback structure with probability accumulation. *GPS Solut.* **2019**, *23*, 16. [CrossRef]
10. Meng, Q.; Liu, J.; Zeng, Q.; Feng, S.; Xu, R. Impact of one satellite outage on ARAIM depleted constellation configurations. *Chin. J. Aeronaut.* **2019**, *32*, 967–977. [CrossRef]
11. Tesla Model 3 Spoofed off the highway—Regulus Navigation System Hack Causes Car to Turn on Its Own. Available online: https://www.regulus.com/blog/tesla-model-3-spoofed-off-the-highway-regulus-researches-hack-navigation-system-causing-car-to-steer-off-road/ (accessed on 1 July 2019).
12. Kuusniemi, H.; Blanch, J.; Chen, Y.H.; Lo, S.; Enge, P. Feasibility of Fault Exclusion Related to Advanced RAIM for GNSS Spoofing Detection. In Proceedings of the ION GNSS+ 2017, Portland, OR, USA, 25–29 September 2017; pp. 2359–2370.
13. Psiaki, M.L.; Humphreys, T.E. GNSS spoofing and detection. *Proc. IEEE* **2016**, *104*, 1258–1270. [CrossRef]
14. Maier, D.S.; Frankl, K.; Pany, T. The GNSS-Transceiver: Using Vector-tracking Approach to Convert a GNSS Receiver to a Simulator; Implementation and Verification for Signal Authentication. In Proceedings of the ION GNSS+ 2018, Miami, FL, USA, 24–28 September 2018; pp. 4231–4244.
15. Meng, Q.; Hsu, L.T. A GNSS Internal Spoofing Generator using Vector Tracking-Based Receiver. In Proceedings of the ION GNSS+ 2019, Miami, FL, USA, 16–20 September 2019. in press.
16. Liu, K.; Wu, W.; Wu, Z.; He, L.; Tang, K. Spoofing Detection Algorithm Based on Pseudo-range Differences. *Sensors* **2018**, *18*, 3197. [CrossRef] [PubMed]
17. Hsu, L.T.; Gu, Y.; Kamijo, S. 3D building model-based pedestrian positioning method using GPS/GLONASS/QZSS and its reliability calculation. *GPS Solut.* **2016**, *20*, 413–428. [CrossRef]
18. Hsu, L.T.; Gu, Y.; Huang, Y.; Kamijo, S. Urban pedestrian navigation using smartphone-based dead reckoning and 3-D map-aided GNSS. *IEEE Sens. J.* **2015**, *16*, 1281–1293. [CrossRef]
19. Hahn, D.A.; Munir, A.; Behzadan, V. Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges. *IEEE Intell. Transp. Syst. Mag.* **2019**. [CrossRef]
20. Humphreys, T.E. Detection Strategy for Cryptographic GNSS Anti-Spoofing. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 1073–1090. [CrossRef]
21. O'Hanlon, B.W.; Psiaki, M.L.; Bhatti, J.A.; Shepard, D.P.; Humphreys, T.E. Real-Time GPS Spoofing Detection via Correlation of Encrypted Signals. *Navigation* **2013**, *60*, 267–278. [CrossRef]

22. Maier, D.; Frankl, K.; Blum, R.; Eissfeller, B.; Pany, T. Preliminary Assessment on the Vulnerability of NMA-based Galileo Signals for a Special Class of Record & Replay Spoofing Attacks. In Proceedings of the IEEE/ION PLANS 2018, Monterey, CA, USA, 23–26 April 2018; pp. 63–71.
23. Tanil, C.; Khanafseh, S.; Joerger, M.; Pervan, B. An INS Monitor to Detect GNSS Spoofers Capable of Tracking Vehicle Position. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *54*, 131–143. [CrossRef]
24. Xu, R.; Ding, M.; Qi, Y.; Yue, S.; Liu, J. Performance Analysis of GNSS/INS Loosely Coupled Integration Systems under Spoofing Attacks. *Sensors* **2018**, *18*, 4108. [CrossRef]
25. Liu, Y.; Li, S.; Fu, Q.; Liu, Z. Impact Assessment of GNSS Spoofing Attacks on INS/GNSS Integrated Navigation System. *Sensors* **2018**, *18*, 1433. [CrossRef]
26. Dampf, J.; Pany, T.; Bär, W.; Winkel, J.; Mervart, L.; Ávila-Rodríguez, J.; Hein, G. Real World Spoofing Trials and Mitigation. *Inside GNSS* **2017**, *12*, 55–65.
27. Borio, D.; Gioia, C. A sum-of-squares approach to GNSS spoofing detection. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 1756–1768. [CrossRef]
28. Liu, Y.; Li, S.H.; Xiao, X.; Fu, Q.W. INS-aided GNSS spoofing detection based on two antenna raw measurements. *Gyroscopy Navig.* **2016**, *7*, 178–188. [CrossRef]
29. Yang, C.; Pany, T.; Soloviev, A. An Implementation of Variable IF Tracking Loop (VITAL) and Initial Test Results. *Navigation* **2017**, *64*, 515–533. [CrossRef]
30. Guo, Y.; Miao, L.; Zhang, X. Spoofing Detection and Mitigation in a Multi-correlator GPS Receiver Based on the Maximum Likelihood Principle. *Sensors* **2019**, *19*, 37. [CrossRef] [PubMed]
31. Shafiee, E.; Mosavi, M.R.; Moazedi, M. Detection of Spoofing Attack using Machine Learning based on Multi-Layer Neural Network in Single-Frequency GPS Receivers. *J. Navig.* **2018**, *71*, 169–188. [CrossRef]
32. Li, W.; Huang, Z.; Lang, R.; Qin, H.; Zhou, K.; Cao, Y. A Real-Time Interference Monitoring Technique for GNSS Based on a Twin Support Vector Machine Method. *Sensors* **2016**, *16*, 329. [CrossRef]
33. Wang, F.; Li, H.; Lu, M. GNSS Spoofing Detection and Mitigation Based on Maximum Likelihood Estimation. *Sensors* **2017**, *17*, 1532. [CrossRef]
34. Tao, H.; Li, H.; Lu, M. A Method of Detections' Fusion for GNSS Anti-Spoofing. *Sensors* **2016**, *16*, 2187. [CrossRef]
35. Xu, B.; Jia, Q.; Luo, Y.; Xu, B.; Hsu, L.-T. Intelligent GNSS LOS/Multipath/NLOS Classifiers based on Correlator, RINEX and NMEA-level Measurements. *Remote Sens.* **2019**, *11*, 1851. [CrossRef]
36. Xu, B.; Hsu, L.-T. Open-source MATLAB code for GPS vector tracking on a software-defined receiver. *GPS Solut.* **2019**, *23*, 46. [CrossRef]
37. Hsu, L.T. Integration of Vector Tracking Loop and Multipath Mitigation Technique and its Assessment. In Proceedings of the ION GNSS+ 2013, Nashville, TN, USA, 16–20 September 2013; pp. 3263–3278.
38. Hsu, L.T.; Jan, S.S.; Groves, P.D.; Kubo, N. Multipath mitigation and NLOS detection using vector tracking in urban environments. *GPS Solut.* **2015**, *19*, 249–262. [CrossRef]
39. Hsu, L.T.; Jan, S.; Sun, C.; Lin, Y. A new algorithm for the signal cancellation of GIOVE-A L1B & GPS L1 Signal. In Proceedings of the International Symposium on GPS/GNSS, Sydney, Australia, 4–6 December 2007.
40. Xu, B.; Hsu, L.-T. Open Source MATLAB Code for GPS Vector Tracking on a Software-Defined Receiver. Available online: https://www.ngs.noaa.gov/gps-toolbox/GPS_VT_SDR.htm (accessed on 1 July 2019).
41. Borre, K. GPS Easy suite II. *Inside GNSS* **2009**, *2*, 48–51.
42. Kotecha, J.H.; Djuric, P.M. Gaussian sum particle filtering. *IEEE Trans. Signal Process.* **2003**, *51*, 2602–2612. [CrossRef]
43. Martino, L.; Elvira, V.; Camps-Valls, G. Group Importance Sampling for particle filtering and MCMC. *Digit. Signal Process.* **2018**, *82*, 133–151. [CrossRef]
44. Martino, L.; Read, J.; Elvira, V.; Louzada, F. Cooperative parallel particle filters for online model selection and applications to urban mobility. *Digit. Signal Process.* **2017**, *60*, 172–185. [CrossRef]
45. Urteaga, I.; Bugallo, M.F.; Djurić, P.M. Sequential Monte Carlo methods under model uncertainty. In Proceedings of the 2016 IEEE Statistical Signal Processing Workshop, Palma de Mallorca, Spain, 26–29 June 2016; pp. 1–5.