A GNSS Internal Spoofing Generator using Vector Tracking-Based Receiver

Qian MENG and Li-Ta HSU
Interdisciplinary Division of Aeronautical and Aviation Engineering,
The Hong Kong Polytechnic University, Hong Kong

BIOGRAPHY (IES)

Qian MENG is currently a postdoctoral fellow with Interdisciplinary Division of Aeronautical and Aviation Engineering, Hong Kong Polytechnic University, Hong Kong. He received his Bachelor of Engineering in Automation and PhD. degree in Guidance, Navigation and Control from Nanjing University of Aeronautics and Astronautics, China, in 2013 and 2018, respectively. He was a visiting PhD student at Centre of Transport Studies, Imperial College London in 2017. His current research interests focus on GNSS software-defined receiver, Integrity monitoring and Signal processing in autonomous driving. He is a member of Institute of Navigation.

Li-Ta HSU received the B.S. and Ph.D. degrees in aeronautics and astronautics from National Cheng Kung University, Taiwan, in 2007 and 2013, respectively. He is currently an assistant professor with Interdisciplinary Division of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, before he served as post-doctoral researcher in Institute of Industrial Science at University of Tokyo, Japan. In 2012, he was a visiting scholar in University College London, U.K. His research interests include GNSS positioning in challenging environment and localization for pedestrian, autonomous driving vehicle and unmanned aerial vehicle. Dr. Hsu currently are members of ION and IEEE and serve as a member of editorial board and reviewer in professional journal related to GNSS.

ABSTRACT

Spoofing will seriously threaten the application of global navigation satellite system (GNSS) like autonomous vehicle. A research spoofing generator will contribute to assess the threat of spoofing attack and help the anti-spoofing research. But the recent commercial of the shelf (COTS) spoofing generator are expensive and the technology implementation is sophisticated and complicated. To address the above problem and promote the GNSS safety-critical applications, a spoofing generator using vector tracking-based software-defined receiver is proposed in this paper. The spoofing generator aims to modify the raw signals by cancelling the actual signal. The connections between the spreading code and carrier and the states of the victim/spoofed receiver are established through vector-tracking. The actual signal can be predicted effectively and the spoofing signal will be generated at the same time. The experiment test results show that the spoofing attack signal can mislead the victim receiver to the designed trajectory effectively. Neither the tracking channels nor the positioning observations has 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 good quality of concealment. With the superiority of programmability and diversity, it is believed that the proposed spoofing generator method based on an open source software-defined receiver with vector tracking architecture has a great value for GNSS anti-spoofing researches.

INTRODUCTION

Autonomous vehicles require an extremely accurate, robust, and reliable navigation system [1,2]. Global navigation satellite system (GNSS) receivers are heavily relied upon in current vehicular navigation systems. However, it has been well-known that GNSS is vulnerable to interference, such as multipath, jamming and spoofing [3]. The impacts of multipath and jamming can result in a positioning error of several tens of meters or even cause the malfunction of GNSS receivers [4,5]. Different from multipath and jamming, spoofing signals are intentionally designed to mislead GNSS receivers by generating fabricated synchronized navigation signals leading to fake navigation solutions. Spoofing seriously limits the GNSS applications related to life safety such as autonomous

https://doi.org/10.33012/2019.17069

vehicles [6]. Although the GNSS receiver has own function to detect and exclude fault called receiver autonomous integrity monitoring (RAIM), the redundancy observations and consistency check still limit its performance to anti-spoofing [7,8].

Protecting GNSS from spoofing is critical to autonomous vehicle navigation and understanding the spoofing mode is the first step to realize anti-spoofing. The basic mode of traditional spoofing technologies is to broadcast a spoofing signal to the victim receiver, which we name it as external spoofing in this paper. The correlation peaks of actual and spoofing signals are overlapped. These spoofing methods are not easy to succeed for two reasons. One is that the victim receiver will receive the actual signal synchronously, and the received signal mixes actual signal and spoofing signal will get complicated. The other reason is that to oppress the actual signal, it is necessary to modify some parameters in the spoofing signal, like amplitude, code delay and so on. But the tracking channels will detect the abnormal changes easily. Most of the recent anti-spoofing methods dependent on actual signal features to detect this kind of spoofing signal. However, once the GNSS-denial circumstance is carried out, e.g. jammer or urban canyon environment, or the hackers in cyberattacks infiltrate the electronic control units of GNSS positioning chip and modify the raw signals between antenna and baseband processing block [9,10,11]. The actual signal will be cancelled and only the spoofing signal is left (We name this spoofing mode as internal spoofing). There's no actual signal any longer. This novel spoofing solution is hazardous and the recent anti-spoofing technologies are less able to overwhelm it.

To generate the spoofing signal, the methods can be broadly divided into meaconing, simulator-based spoofing and receiver-based spoofing [12,13]. In meaconing the GNSS 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 GNSS simulator can be used to replicate the signals as they would appear at a chosen location, misleading the receiver to produce an incorrect PVT solution. However, besides the high cost of a commercial signal generator, with the development of new signals, channel structures and navigation message coding rules, implementing a sophisticated signal generator from scratch is a huge, difficult and time-consuming task. In most cases the spoofing signals are generated on a software-defined receiver (SDR) by modifying the recorded actual signals. In recent researches, a way to convert a vector tracking based SDR into a GNSS software transceiver is proposed to reuse the sophisticated and optimized infrastructure of the software receiver for the signal generator [14]. This approach makes it possible to realize a 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.

Inspired by the above GNSS transceiver, a GNSS internal spoofing generator is proposed in this paper. The generator is implemented using the vector tracking architecture on a SDR platform. The function implementation is shown in Figure 1. Firstly, the generator will track the actual signal synchronously to get the ephemeris, open-sky satellites, 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 and generate the spoofing signal component. Finally, the cancellation signal component and spoofing signal component will be combined as the attack signal. The hacker will be able to plant the attack signal to the raw signal.

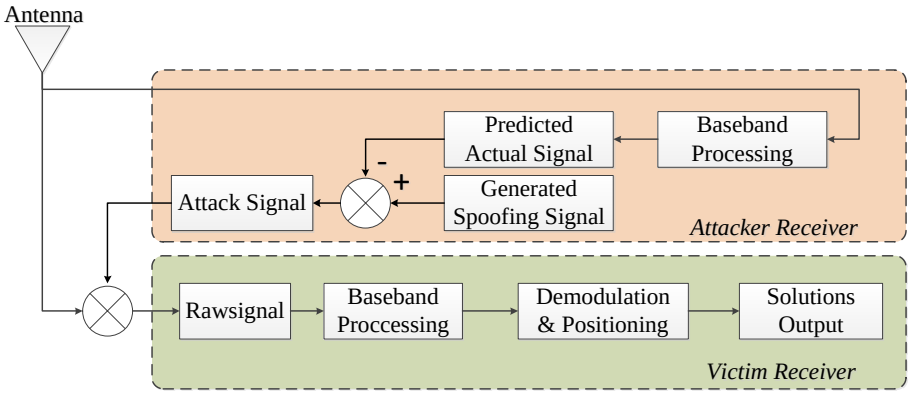

Figure 1 Functional diagram of internal spoofing generator

The rest of the paper is organized as follows: The design of vector tracking is introduced firstly. After that the details about the actual signal prediction and spoofing signal generation are given. Then, the experiment test evaluates the performance of the proposed spoofing method. As it is undeniable that there is an actual and urgent need to research on spoofing generator, the above spoofing generator, implemented based on an open source SDR with a vector tracking architecture, will help the research on spoofing defenses in the future.

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 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 environment originally, e.g. increased capabilities against weak signal or high dynamic conditions. In recent years, with the increasing development of intelligent transportation system and location-based service in urban canyon areas, vector-tracking shows more potential superiorities. For example, vector-tracking is applied to multipath or non-line-sight reception mitigation in signal processing stage [15,16]. 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.
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. And on the other side, 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).

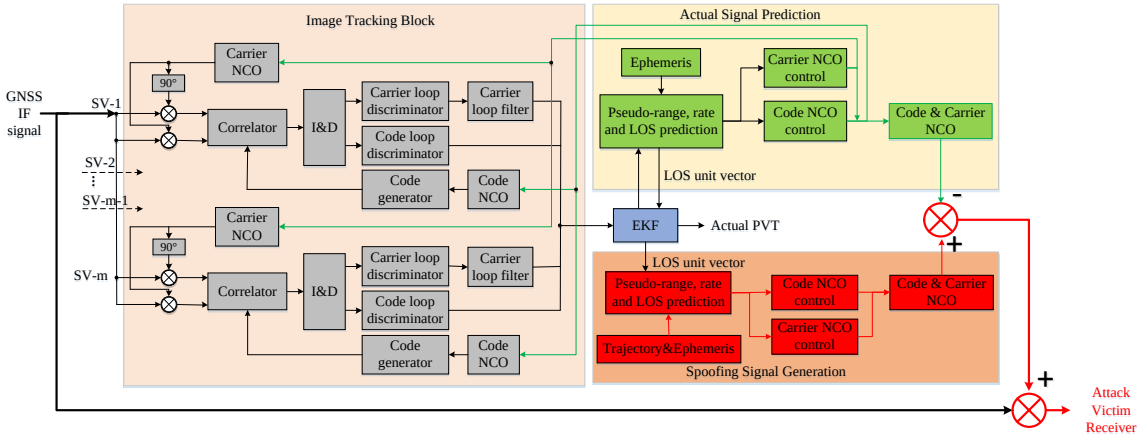

Figure 2 Implementation architecture of Spoofing generator based on vector tracking

The EKF estimates the actual PVT based on its system propagation and the measurements. After obtaining the navigation solution, the pseudorange 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 $\Delta\mathbf{p} = \left[\Delta p_x, \Delta p_y, \Delta p_z \right]$ and $\Delta\mathbf{v} = \left[\Delta v_x, \Delta v_y, \Delta v_z \right]$ 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 errors in the units of meters and meters per second, respectively. The system propagation at epoch k is:

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

where

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

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

In equation (2), τ is the update interval of the EKF. The superscript and subscript, "-" and "+", denote the system state before and after measurement update, respectively. The symbol " ^ " represents the EKF estimates.
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. 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.

ACTUAL SIGNAL PREDICTION AND SPOOFING SIGNAL GENERATION

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

$$\tilde{f}_{code,k+1}^{j}=f_{CA}\left[1-\frac{\tilde{\rho}_{k+1}^{j}-\hat{\rho}_{k}^{j}}{c\tau}\right] \tag{8}$$

where $\tilde{\rho}_{k+1}^{j}$ and $\hat{\rho}_{k}^{j}$ are the predicted pseudorange at epoch $k+1$ and the estimated pseudorange at epoch k. f_{CA} is the code chipping rate (1.023 MHz for GPS L1 C/A); c is the speed of light. The predicted pseudorange is calculated using

$$\tilde{\rho}_{k+1}^{j}=\left\|\tilde{\mathbf{r}}_{u,k+1}-\mathbf{r}_{k+1}^{j}\right\|+\delta\hat{\rho}_{sv,c}^{j}+\delta\hat{\rho}_{I}^{j}+\delta\hat{\rho}_{T}^{j}-\hat{b}_{clk} \tag{9}$$

where $\mathbf{r}_{k+1}^{j}$ is the satellite position at epoch $k+1$, which is known from the broadcast ephemeris. $\tilde{\mathbf{r}}_{u,k+1}$ is the predicted receiver position respectively, which can be calculated based on the estimated position and clock bias at the previous epoch. $\delta\hat{\rho}_{sv,c}^{j}$, $\delta\hat{\rho}_{I}^{j}$ and $\delta\hat{\rho}_{T}^{j}$ are the pseudorange errors caused by satellite clock error, ionospheric delay and tropospheric delay, respectively. $f_{code,k+1}^{j}$ 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 pseudorange rate at epoch $k+1$ as follows:

$$\tilde{f}_{doppler,k+1}^{j}=-\dot{\tilde{\rho}}_{k+1}^{j}\frac{f_{L1}}{c} \tag{10}$$

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

$$\tilde{\dot{\rho}}_{k+1}^{j}=\left(\mathbf{v}_{sv,k+1}^{j}-\tilde{\mathbf{v}}_{u,k+1}\right)\mathbf{l}^{j}+\hat{d}_{u,clk}-d_{sv,clk}^{j} \tag{11}$$

where; $\tilde{\mathbf{v}}_{u,k+1}$ and $\mathbf{v}_{sv,k+1}^{j}$ 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 $d_{sv,clk}^{j}$ are the estimated receiver clock drift and the j^{th} satellite clock drift, respectively, both in meters per second.

Then the measurement vector of EKF at epoch $k+1$ can be got

$$\Delta\rho^{j}=\Delta\tau^{j}\cdot\frac{c}{f_{CA}} \tag{12}$$

$$\Delta\dot{\rho}_{k+1}^{j}=f_{doppler}^{j}\frac{c}{f_{L1}}-\left(\mathbf{v}_{sv,k+1}^{j}-\tilde{\mathbf{v}}_{u,k+1}\right)\mathbf{l}^{j}-\hat{d}_{u,clk}+d_{sv,clk}^{j} \tag{13}$$

where $\Delta\tau^{j}$ is the code discriminator output in chips, $f_{Doppler}^{j}$ 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 pseudorange and pseudorange rate are calculated as:

$$\tilde{\rho}_{spoof,k+1}^{j}=\left\|\mathbf{r}_{trj,k+1}-\mathbf{r}_{k+1}^{j}\right\|+\delta\hat{\rho}_{sv,c}^{j}+\delta\hat{\rho}_{I}^{j}+\delta\hat{\rho}_{T}^{j}-\hat{b}_{clk} \tag{14}$$

$$\tilde{\dot{\rho}}_{spoof,k+1}^{j}=\left(\mathbf{v}_{sv,k+1}^{j}-\mathbf{v}_{trj,k+1}\right)\mathbf{l}^{j}+\hat{d}_{u,clk}-d_{sv,clk}^{j} \tag{15}$$

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 referred to the Reference [14], which including a 4th degree spline interpolation and a second extrapolation. $\hat{b}_{clk}$ is got from the EKF state vector.

ATTACK SIGNAL GENERATION

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

$$\hat{D}_{nav,actual}^{j}=r_{IF}\ .*\ C_{prompt}^{j}\ .*\ Carr_{\cos}^{j} \tag{16}$$

where r_{IF} is the raw signal, C_{prompt}^{j} and $Carr_{\cos}^{j}$ are the code and carrier in the prompt branch of satellite *j* channel. Using $\hat{D}_{nav,actual}^{j}$ 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

$$\hat{D}_{nav,spoof}=\begin{cases}1, & if\ I_p>0\\ -1, & if\ I_p<0\end{cases}\qquad where\ I_p=\sum\left(r_{IF}\ .*\ C_{prompt}\ .*\ Carr_{\cos}\right) \tag{17}$$

About the amplitude, a simple method to get the signal amplitude mentioned in [17] is accepted as

$$\hat{A}^{j}=\frac{\sum_{1}^{N_{sample}}\left(r_{IF}\ .*\ C_{prompt}^{j}\ .*\ \hat{D}_{nav,actual}^{j}\ .*\ Carr_{\cos}^{j}\right)}{\sum_{1}^{N_{sample}}\left(C_{prompt}^{j}\ .*\ \hat{D}_{nav,actual}^{j}\ .*\ Carr_{\cos}^{j}\right)^{2}} \tag{18}$$

Finally, the attack signal is combined with the predicted actual signal component and generated spoof signal component as

$$r_{attack}=r_{spoof}-r_{actual} \tag{19}$$

EXPERIMENT AND ANALYSIS

Experimental tests were conducted to evaluate the performance of the proposed spoofing generator. The actual signal was collected statically in an open-sky environment in Hong Kong. GPS signals were collected using a Nottingham Scientific Ltd. (NSL) Stereo front-end for post-processing by the developed software and method. The sampling frequency and IF of the front-end are 26 MHz and 6.5 MHz, respectively. The spoofing attack began to attack the GPS signal from the 2nd second with a velocity of 15m/s, 15m/s and 15m/s in an ECEF coordinates. The attacked signal was processed on a software-defined receiver with conventional tracking architecture.

The proposed method is implemented on the software-defined receiver 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 [18]. The MATLAB software and the corresponding vector tracking open source codes can be downloaded on the GPS Toolbox website at: https://www.ngs.noaa.gov/gps-toolbox. In the spoofing architecture, the update interval of the EKF is one millisecond.

Performance in tracking

The tracking results are analyzed in this section. Three scenarios are considered in this experiment: 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 also combined with the generated spoofing signal component. Figure 3, Figure 4 and Figure 5 show the outputs of prompt branch, DLL discriminator and PLL discriminator in tracking. In every figure, the above three scenarios are presented from top to bottom.

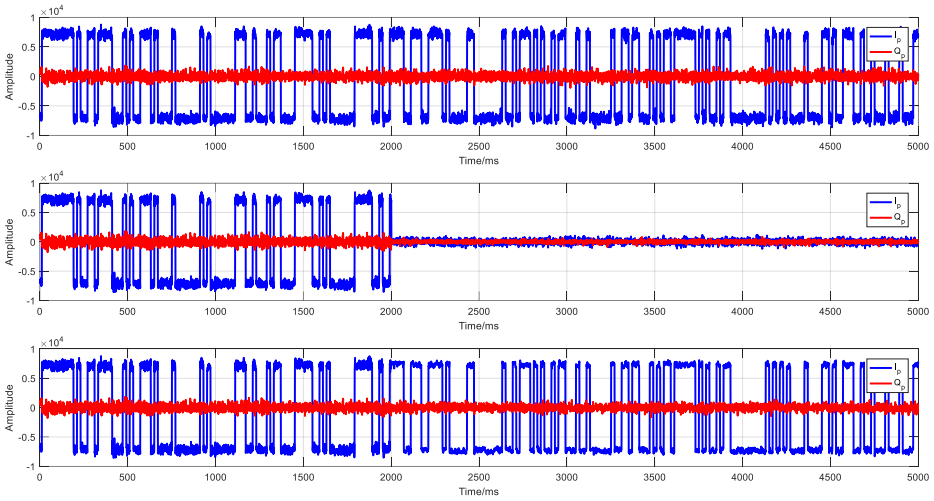

Figure 3. I_p and Q_p outputs of PRN-10 tracking in three different scenarios of signal tracking. From top to bottom: (top) when no attacks exist, (middle) actual signal cancelled and (bottom) actual signal cancelled and spoofing signal modulated.

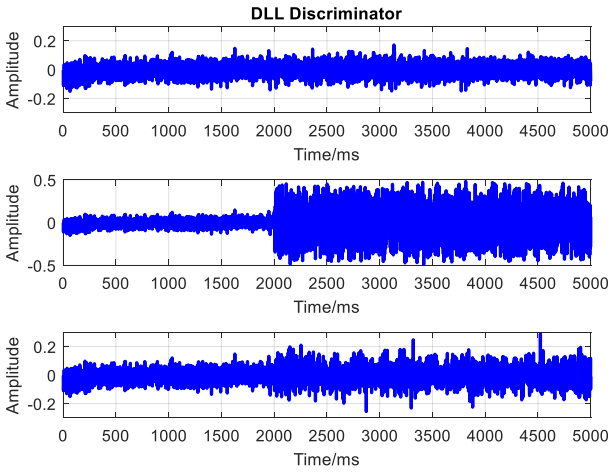

Figure 4 DLL discriminator in the three scenarios

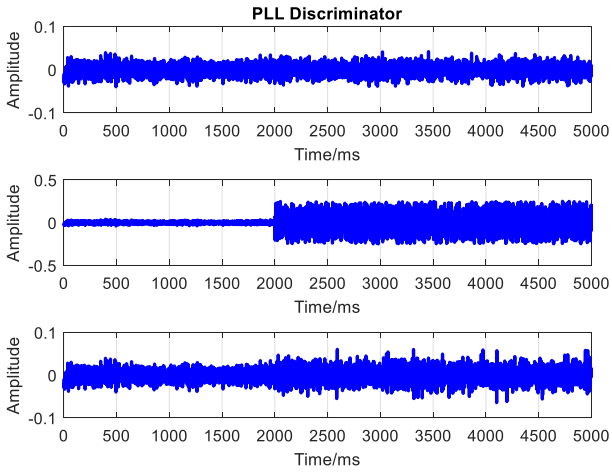

Figure 5 PLL discriminator in the three scenarios

The results of the 2nd scenario show the results after the actual signal was cancelled. Both the code loop and carrier loop lost locked immediately. There are only noises in the correlations of In-phase branch (I_p) and Quadrature (Q_p) branch. The actual signal has been demodulated and cancelled ideally. Meanwhile, the tracking results of the 3rd scenario have no obvious difference compared with those of the 1st scenario. There was no outlier or loss of lock in the code loop or carrier loop from Figure 4 and Figure 5. The amplitude of the correlation outputs of the prompt branch has no significant change from raw signal to attack signal.

The above results are encouraging as there's no abnormal change in the tracking channel after the raw signal are attacked. All the anti-spoofing methods based on tracking loop check cannot detect the spoofing attack under this circumstance. The spoofing attack has a good function of crypticity.

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 conventional spoofing attack. The positioning outputs after the attack was modulated are shown in Figure 6. As the attack was triggered at the 2nd second, only the positioning results during the 2~5 second are shown in the figure.

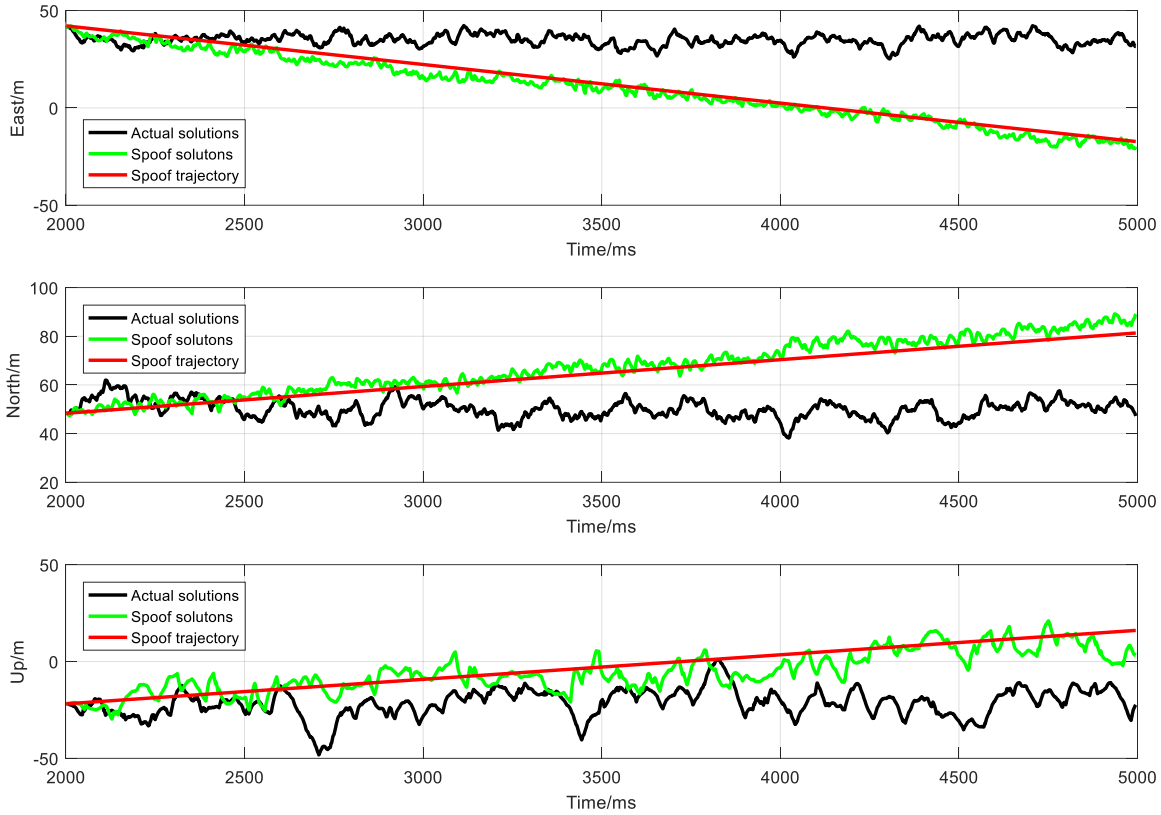

Figure 6 Positioning results under spoofing attack

The black line, green line and red line represent the actual solution, spoof solution and spoof trajectory in East-North-Up coordinate, respectively. As shown in the figure, the actual positioning results verified that the receiver is static. However, the positioning results after spoofing attack was misled to the spoofing trajectory in accordance with expectation. The victim receiver began to move with a consecutive velocity. As the proposed spoofing attack method can cover the open-sky satellites and all the observations are consistent, the anti-spoofing method based on position domain like RAIM can not detect the spoofing easily.

Compared to that of the up component, the positioning results in east and north component matched the spoofing trajectory better. It is easy to be accepted as that the positioning accuracy in horizontal direction is better and we care more about the horizontal results in autonomous vehicle.

CONCLUSIONS

A GNSS spoofing generator using VT-based SDR is proposed in this paper. The generator is implemented by cancelling the actual code with the spoofing code through internal attacking the raw signal. 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, navigation data and so on. The preliminary test results shown that the spoofing attack can work effectively. The receiver was misled to the spoofing trajectory successfully. The anti-spoofing methods in track channel or positioning domain are hard to detect this spoofing as there's no abnormal change in the tracking results or positioning solutions. The threaten of this spoofing mode to autonomous vehicles is hazardous once all the open-sky GNSS satellites are spoofed. The researchers will focus on the evaluation of the proposed spoofing method to dynamic victim receiver in urban environment and test the defense effect of different anti-spoofing technologies in the future work.

ACKNOWLEDGMENTS

The authors acknowledge the support of the Hong Kong Polytechnic University on the project P0013910, “Security Enhancement of Positioning Sensors on Connect Autonomous Vehicle”.

REFERENCES

1. Bonnefon, J. F., Shariff, A., & Rahwan, I. (2016). The social dilemma of autonomous vehicles. *Science*, 352(6293), pp1573-1576.
2. Claybrook, J., & Kildare, S. (2018). Autonomous vehicles: No driver… no regulation?. *Science*, 361(6397), pp36-37.
3. Takefuji, Y. (2018). Connected vehicle security vulnerabilities [commentary]. IEEE Technology and Society Magazine, 37(1), pp15-18.
4. Ioannides, R. T., Pany, T., & Gibbons, G. (2016). Known vulnerabilities of global navigation satellite systems, status, and potential mitigation techniques. *Proceedings of the IEEE*, 104(6), pp1174-1194.
5. Hsu, L. T. (2018). Analysis and modeling GPS NLOS effect in highly urbanized area. *GPS solutions*, 22(1), pp7.
6. Broumandan, A., & Lachapelle, G. (2018). Spoofing Detection Using GNSS/INS/Odometer Coupling for Vehicular Navigation. Sensors, 18(5), pp1305.
7. Meng, Q., Liu, J., Zeng, Q., Feng, S., & Xu, R. (2019). Improved ARAIM fault modes determination scheme based on feedback structure with probability accumulation. *GPS Solutions*, 23(1), pp16.
8. Meng, Q., Liu, J., Zeng, Q., Feng, S., & Xu, R. (2019). Impact of one satellite outage on ARAIM depleted constellation configurations. *Chinese Journal of Aeronautics*. 32(4), pp 967-977.
9. Petit, J., & Shladover, S. E. (2014). Potential cyberattacks on automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(2), pp546-556.
10. Hahn, D. A., Munir, A., & Behzadan, V. (2019). Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges. *IEEE Intelligent Transportation Systems Magazine*. DOI: 10.1109/MITS.2019.2898973
11. Wyglinski, A. M., Huang, X., Padir, T., Lai, L., Eisenbarth, T. R., & Venkatasubramanian, K. (2013). Security of autonomous systems employing embedded computing and sensors. *IEEE micro*, 33(1), pp80-86.
12. Kuusniemi, H., Blanch, J., Chen, Y. H., Lo, S., & Enge, P. (2017). Feasibility of Fault Exclusion Related to Advanced RAIM for GNSS Spoofing Detection. *Proceedings of ION GNSS+ 2017*. Portland, Oregon, September 2017, pp. 2359-2370.
13. Psiaki, M. L., & Humphreys, T. E. (2016). GNSS spoofing and detection. Proceedings of the IEEE, 104(6), 1258-1270.
14. Maier, Daniel S., Frankl, Kathrin, Pany, Thomas. (2018). The GNSS-Transceiver: Using Vector-tracking Approach to Convert a GNSS Receiver to a Simulator; Implementation and Verification for Signal Authentication. Proceedings of ION GNSS+ 2018, Miami, Florida, September 2018, pp. 4231-4244.
15. Hsu, L-T. Integration of Vector Tracking Loop and Multipath Mitigation Technique and its Assessment. *Proceedings of ION GNSS+ 2013*, Nashville, TN, September 2013, pp. 3263-3278.
16. Hsu, L. T., Jan, S. S., Groves, P. D., & Kubo, N. (2015). Multipath mitigation and NLOS detection using vector tracking in urban environments. *GPS Solutions*, 19(2), 249-262.
17. Hsu, Li-Ta, Shau-Shiun Jan, Chih-Cheng Sun, and Yao-Cheng Lin. "A new algorithm for the signal cancellation of GIOVE-A L1B & GPS L1 Signal." *In Proceedings of international symposium on GPS/GNSS*. 2007.
18. Xu, B., & Hsu, L. T. (2019). Open-source MATLAB code for GPS vector tracking on a software-defined receiver. *GPS solutions*, 23(2), 46.