공학박사학위논문

A study on GNSS receiver performance analysis using various interference scenarios

다양한 교란 시나리오를 이용한 GNSS 수신기 성능 분석에 대한 연구

2020년 2월

서울대학교 대학원

기계항공공학부

신범주

A study on GNSS receiver performance analysis using various interference scenarios

다양한 교란 시나리오를 이용한 GNSS 수신기 성능분석에 대한 연구

지도교수 기창돈

이 논문을 공학박사 학위논문으로 제출함
2019년 11월

서울대학교 대학원
기계항공공학부
신범주

신범주의 공학박사 학위논문을 인준함
2019년 12월

위 원 장 _____(인)

부위원장 _____(인)

위 원 _____(인)

위 원 _____(인)

위 원 _____(인)

ABSTRACT

A study on GNSS receiver performance analysis using various interference scenarios

Beomju Shin

School of Mechanical and Aerospace Engineering

The Graduate School

Seoul National University

The security and safety aspects of global navigation satellite systems have been receiving significant attention from researchers and the general public, because the use of GNSS has been increasing in modern society. In this situation, the importance of GNSS safety and security is also increasing. The most dangerous type of interference is a spoofing because if the receiver captures a spoofing signal, the navigation solution can be controlled by the spoofer. In this paper, I analyzed the characteristics of the main spoofing parameters that determines the success or failure of spoofing process when the spoofing signal is injected into the receiver. I also proposed a CCEE. It determines the spoofing result according to the various spoofing parameter. Also the correlation between spoofing parameters could be explained by estimating the boundary value and line using CCEE. In addition, spoofing success and failure could be distinguished in the spoofing parameter space using CCEE results.

When the covert capture is performed at the receiver, the two correlation peaks of authentic and covert capture signals are generated on the code domain. The relative velocity (Doppler difference value) of the two signal peaks determines the time of total spoofing process. In general, the timing at which the DLL tracking lock point is switched from the authentic signal to the spoofing signal is different according to the visible satellite. This raises the value of WSSE. In order to minimize this, the spoofing should be performed in a short time by determining the optimal sweep direction. In a 3D situation, triangles are defined using a particular visible satellites, and the circumcenter direction of the triangle on the victim becomes the optimal direction, and the relative speed of the authentic and the covert capture signal for the visible satellite be maximized on the optimal covert capture direction.

To simulate the proposed methods, we defined the covet capture scenarios and generated the IF data to simulate the intended scenarios. Then, using the corresponding IF data, signal processing was performed through SDR. Through this, it was confirmed that the spoofing is successfully performed as intended scenarios through the optimal spoofing parameters generated through CCEE, and the covert capture process time is noticeably minimized through the optimal sweep direction.

Key word: GNSS, GNSS receiver, GNSS interference, spoofing, DLL, WSEE, victim trajectory

Student number: 2014-30355

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1. Introduction

1.1. Research Motivation

The security and safety aspects of global navigation satellite systems (GNSSs) have been receiving significant attention from researchers and the general public, because the use of GNSSs has been increasing in modern society [1, 23-25]. Because the power of a GNSS signal coming from the ground is very low, the signal is exposed to different types of radio interferences [2]. Moreover, in contrast to military signals, safety and security issues are not considered for civilian GNSS signals [34-35]. A civilian signal is not encrypted, and the details of such a signal are open [3, 28]. In other words, anyone can intentionally transmit a fake signal to deceive the user. There are many error source that could be harmful to GNSS receiver measurements [42]. To improve the performance of GNSS solution, there are many approaches such as signal modernization [43], multipath mitigation [44-47], accurate ionospheric modeling [48-51] or tropospheric modeling [52-54].

Some of the types of intentional interferences include jamming, meaconing, and spoofing [4-6]. These interferences are more harmful than aforementioned error sources [55,56]. The aim of jamming is to prevent a user from receiving the authentic signal by transmitting another signal with a significantly greater power than that of the authentic signal. A meaconing attack involves transmitting another signal collected at a different location or time. If a meaconing attack is successful, the receiver would end up providing navigation information, such as the location and time, at which the meaconing signal was collected. The most dangerous type of interference is a spoofing attack. If the

receiver captures a spoofing signal, the navigation solution can be controlled by the spoofer [7, 22].

In Dec. 2011, a drone of U.S was captured by Iranian engineer. Finally, the drone was landing into Iranian territory. This accident shocked many researcher and peoples. Also, this means that the spoofing attack is realistic. And it could lead to dangerous and threatening consequences.

There are a lot of researches about anti-jamming and anti-spoofing. But when interference signals such as jamming, meaconing or spoofing are received in GNSS receiver, there is a lack of research on the phenomena occurring in the GNSS receiver. In this thesis, the basic studies in GNSS receivers is presented like what happened in the auto correlation function in delay lock loop or what is the condition to succeed the spoofing process as well as the fundamental study of spoofing process when interference signals are received in the GNSS receiver.

1.2. Related research

There are two main technological approaches for spoofing researches: spoofing attacks and anti-spoofing techniques. Many spoofing attack tests have been conducted over the past few years. A portable GPS spoofer was developed, and a spoofing attack test was demonstrated for a target receiver [8]. Although this experiment was conducted with a very short distance between the spoofer and the target receiver, it was possible to develop a practical spoofer with low cost. Moreover, successful spoofing tests were carried out against an unmanned aerial vehicle [9], a ship [10], and a mobile device [11]. These studies have shown that spoofing attacks could be executed in real situations. Moreover, many anti-spoofing techniques have been studied for receiver security and safety. A

maximum likelihood estimation-based positioning technique was applied to the detection of spoofing signals and correction of navigation solution [12]. In another study, a cross-correlation approach between two GNSS receivers was used to detect the spoofing signal [13]. In Ref [14], an extended coupled amplitude delay lock loop (DLL) architecture was applied to spoofing detection. A pseudorange difference-based anti-spoofing algorithm was introduced [15]. In Ref [16], spoofing detection was performed using a machine learning algorithm such as a neural network. In other studies, antenna-aided techniques [17] and inertial measurements unit-aided techniques [18] have been developed. In [31-33], cryptographic and authentication techniques was developed for anti-spoofing. The signal quality monitoring technique (SQMT) was proposed to detect the spoofing signal [36]. Also, signal power comparison [38-39], spatial distribution properties [40-41]. In [58], spoofing detection algorithm in GPS L1 is analyzed using signal strength. In [61], various spoofing counter-measures are presented and analyzed such as navigation message monitoring and authentication, detecting of delayed synthesized signal, antenna arrays based monitoring.

Although the aforementioned studies report on spoofing attacks and anti-spoofing techniques, few have analyzed the conditions and circumstances required for a successful spoofing attack. In [19], the spoofing attack results were presented considering the time, position, and power offset. However, only the effects of the spoofing parameters on the spoofing attack results were studied. In ref [9], a spoofing signal with a 10 dB greater power than that of the authentic signal was transmitted to successfully deceive a drone. However, to avoid as much as possible the detection of a spoofing signal at the victim receiver, it is better to transmit the signal with the minimum power possible for a successful spoofing attack. The signal power condition of successful induction for the target receiver

is analyzed using the proposed receiver tracking loops in [37]. Also, code delay of spoofing signal effect on GPS L1 signal is analyzed in [59]. They shows that the spoofing signal within the 1chip affects the tracking loop of receiver and its measurements. Shin analyzed the effects of spoofing signal on GPS receiver about auto correlation function, signal strength, tracking error, pseudo range and positioning result [60].

1.3. Outline of the Dissertation

This dissertation presents the study of analysis in GPS receiver when the interference signals are received with authentic GPS signal. Especially, for the sweep type disturbance signal on the receiver code domain, the ACF is modeled in some way, and the conditions for changing the tracking point of the receiver lock are analyzed.

After the introduction in the first capture, the GPS signal structure and signal processing structure of GPS receiver are described. Also GPS interference signals such as jamming or spoofing are briefly explained in the second capture.

In third capture, auto correlation function (ACF) modeling is presented and which express the situation that the authentic signal and interference signal are received at the GPS receiver. The ACF could be categorized in five shape according to the peak distance between the authentic signal and interference signal. Then, tracking point transition condition and correlation between interference signal parameters are carefully analyzed. Also, covert capture effectiveness equation (CCEE) is derived and boundary value, line and surface are introduced. In the fourth capture, optimal capture direction is introduced. For this, Doppler error is firstly defined and principle of optimal capture direction is explained. In

fifth capture, cover capture simulation is presented using the intermediate frequency data generation module and software defined receiver (SDR). To demonstrate the proposed method, various simulation results are analyzed. In sixth capture, a method for controlling the position, navigation and timing (PNT) of a victim's receiver by directing victim's path to the intended location is presented. Finally, a conclusion and future suggestions are presented.

1.4. Contributions

In this paper, I analyzed the conditions for a successful spoofing attack in the code domain. The spoofing parameters considered in this study are the spoofing signal strength [26], spoofing sweep velocity (Doppler offset) [27], DLL order, and bandwidth. With the increase in the spoofing signal strength or DLL bandwidth, the probability of a successful spoofing attack increases. If the sweep velocity increases, the probability of a successful spoofing attack is reduced because of the increase in the Doppler offset between the authentic signal and the spoofing signal. However, for a specific spoofing signal, it is difficult to determine whether a spoofing attack would be successful when the bandwidth is more than a certain level. It is also difficult to determine the correlation between each parameter for a successful spoofing attack. In this research, we develop a covert capture effectiveness equation (CCEE) for the entire spoofing process. Generally, to determine whether a specific spoofing signal would be successful, it is necessary to perform an iterative DLL calculation during the entire spoofing process. The concept of the CCEE is to reduce the number of iterative DLL tracking calculations during the spoofing process by increasing the integration time. Moreover, I express the entire DLL calculation process in the form of an nth

order polynomial. The spoofing attack results could be obtained in one single calculation through the CCEE. Next, the optimal covert capture direction is analyzed to reduce the covert capture process time. It is the most basic to decide the sweep direction of spoofing signal in the progress of victim direction. In this study, the Doppler error between the original signal and the deceptive signal was considered for the best covert capture direction to reduce that time. It also briefly described how to induce the victim to the intended position using spoofing signal after the victim receiver tracks the spoofing signal.

Following are the contributions of this study:

- I develop the CCEE that can be used to express the entire spoofing process in the form of nth order polynomial.
- I obtain the spoofing results in one single calculation using the CCEE and determine the correlation between each of parameters based on the boundary line which distinguishes between successful and unsuccessful spoofing attacks.
- For a particular receiver, the minimum power of a spoofing signal for a successful spoofing attack could be estimated via the CCEE.
- Optimal covert capture direction is analyzed to reduce the spoofing process time.

Chapter 2. Background

2.1. GPS receiver fundamental

2.1.1. GPS signal structure

The GPS signal broadcasted from the satellite could be expressed as follows:

$$s(t) = \sqrt{2P}D(t)x(t)\cos(2\pi f_{L1}t + \phi_{L1}) \quad (2\text{-}1)$$

where $\sqrt{2P}$ indicate the signal power, $D(t)$ is the navigation message data, $x(t)$ is the code, and $\cos(2\pi f_{L1}t + \phi_{L1})$ denotes the carrier. The GPS signal is generated using a modulation method of the direct sequence spread spectrum (DSSS) with pseudorandom noise (PRN). Through this, all satellites could share the same frequency band, an interference is reduced, and enable signal transition, reception and distance calculation. This multiplexing approach is also known as code division multiple access (CDMA). The GPS signal is generated like figure 2-1 [21]. Figure 2-1 shows the structure of GPS signal. The GPS signal composes of a carrier of 1575.42MHz and a C / A code of 1.023Mcps based on the L1 band. The C / A code is a 1.023MHz pseudorandom noise code, in binary form that repeats every 1ms. Figure 2-2 shows the GPS signal generation process. The receiver could utilize the C / A code to know which satellite signals are

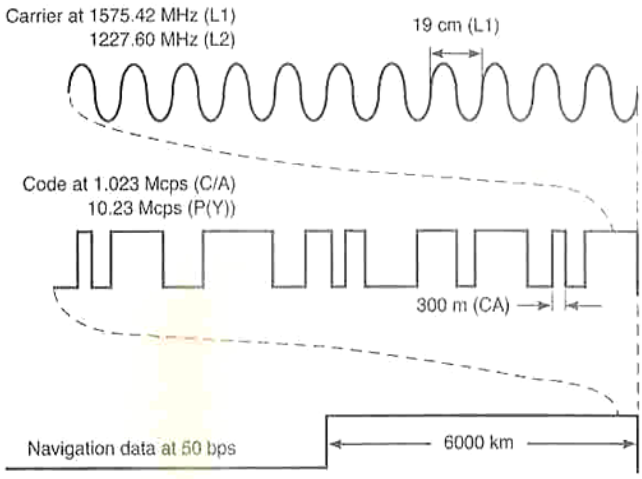


Figure 2-1. GPS signal configuration

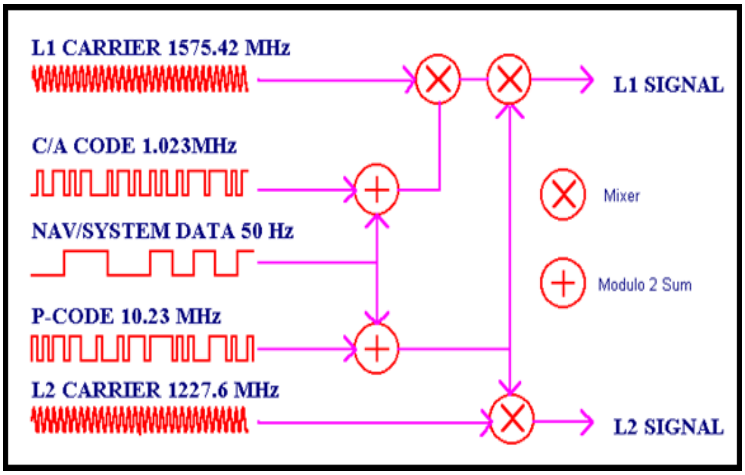


Figure 2-2. GPS signal generation

currently being received in present. In the frequency domain, the GPS L1 C / A code signal appears as a signal with a main lobe of 2.046 MHz bandwidth at the center frequency of 1575.42 MHz. The navigation message contained in the C / A code conveys the information necessary for navigation to the receiver. Information contained in the navigation message includes satellite orbit information, correction information and other system parameters.

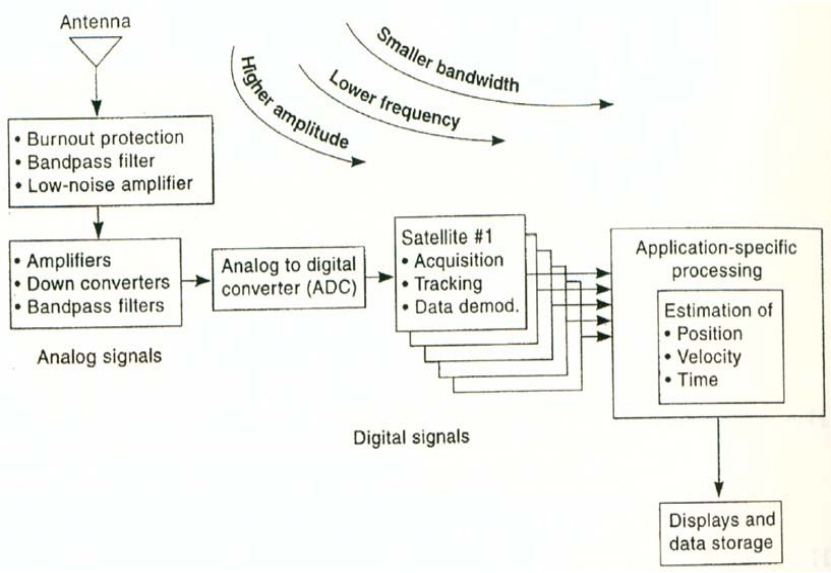


Figure 2-3. Block diagram of GPS receiver

2.1.2. Signal processing structure of GPS receiver

GPS aims to provide positioning, navigation, and timing. To do this, time and range between user and satellite are calculated. The system performance is determined by the accuracy of range and time estimates. Especially in the case of GPS receiver, there are many factors such as measurement accuracy, measurement update rate, signal acquisition and signal reacquisition performance, signal tracking performance, signal tracking threshold, multipath error performance, channel bias, interference cancellation performance, anti-spoofing, etc. These all factors affect the performance of the receiver.

When a GPS signal first comes in through an antenna, it passes through a bandpass filter and a low noise amplifier to remove noise other than the band of interest signal. Next, it goes through a down converter to bring the signal down to intermediate or baseband for actual digital processing, and converts the analog

signal into a digital signal through the ADC. This converted digital signals are subjected to signal acquisition, signal tracking, data decoding, and navigation calculation in the each of modules. Figure 2-3 shows the block diagram of GPS receiver [21].

2.1.3. Signal acquisition

Signal acquisition is the process of finding out what satellite signals exist before tracking the GPS signal. In this process, an approximate code start point and a Doppler shift are estimated. Since it is not known which PRN signals are currently being received from a plurality of GPS satellites, estimated value satisfying the certain range are obtained through the signal acquisition process. 2D search process is usually performed for the GPS signal acquisition. Figure 2-4 presents the 2 dimensional plane consisting of code phase and Doppler

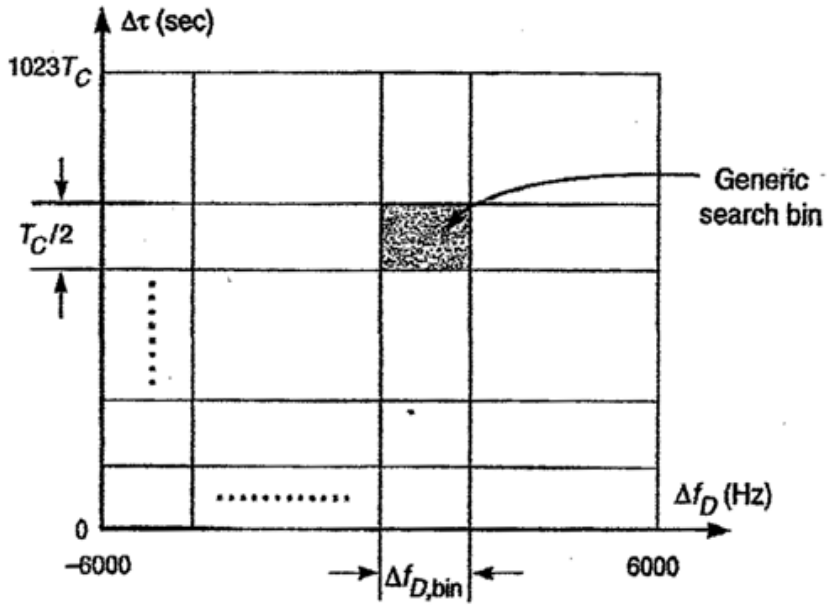


Figure 2-4. 2D search of GPS signal

shift [21]. All PRN candidates are searched to find the code starting point and Doppler shift. The starting point of the PRN code is determined by the distance between the satellite and the user, and the Doppler value is generated by the velocity difference between the satellite and the user. For each search cell, a replica code is generated based on the corresponding code start point and Doppler, and the correlation with the input signal is calculated. If the correlation value exceeds the predetermined threshold, the code start point, Doppler, and PRN of the cell are taken. In general, the search interval for the code is 0.5 chip, and in the case of Doppler, the search is performed at 500 Hz intervals based on 1 ms integration time. However, this search may be performed in a narrower range (interval) depending on the RF environment.

2.1.4. Signal tracking

Signal tracking is the continuous tracking of precise code start points and Doppler from approximate code start points and Doppler obtained as a result of signal acquisition.

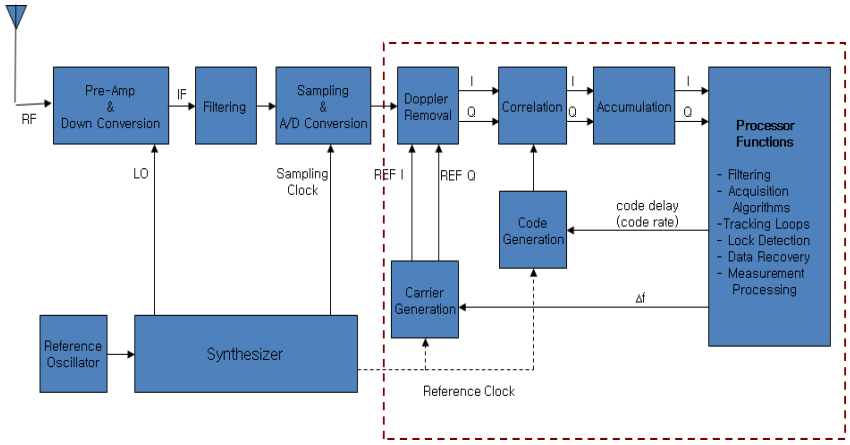


Figure 2-5. Structure of GNSS hardware receiver

In the figure 2-5, which shows the hardware structure by module, the inside of the large box marked with a dotted line shows where the signal tracking is performed. Digital sample signals entered through ADC could be first multiplied by the carrier signal generated by the carrier generator to remove the Doppler. It is then multiplied by the replica code signal generated by the code generator and then accumulated for a set predefined integration time (PIT). Accumulated in-phase and quadrate-phase signals are passed to the processor, which processes these values to provide proper control inputs.

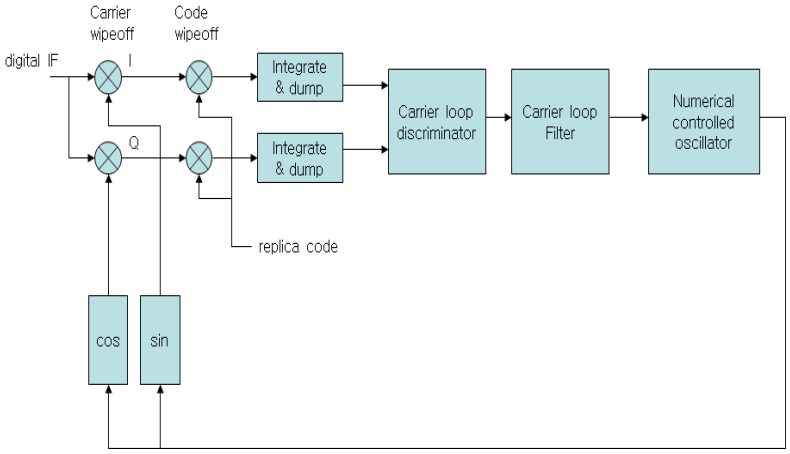


Figure 2-6. Tracking loop structure of GNSS receiver

The figure 2-6 shows the structure of the signal tracking loop. The input value of the signal tracking loop is the signal containing the error at the code start point, carrier frequency, and carrier phase. These input signals are multiplied by the replica signal made by the NCO. This value goes through the process of accumulating for the PIT in the correction filter, which is entered as a descriminator. The descriminator calculates the estimated error of the code start point, carrier Doppler, and carrier phase. The tracking loop filter again estimates the control input to the NCO from the estimated error entered by the descriminator.

NCO stands for numerically controlled oscillator, which produces a frequency output corresponding to the input value. Signal tracking targets code and carrier waves and each waves are estimated through a special form feedback loop. Delay locked loop (DLL) is for code tracking loops frequency locked loop (FLL) for carrier tracking loops to minimize carrier phase offsets. FLL have low precision but robust characteristics in noise and dynamic environments, and phase lock loop (PLL) have high precision but are sensitive to noise and dynamic environments. In carrier tracking loop of GPS receiver, FLL or PLL is used alone, and is implemented in various ways, such as FPLL composite structure or FLL/PLL switching structure. The following figure 2-7 and 2-8 illustrate the structure of the carrier tracking loop and the code tracking loop.

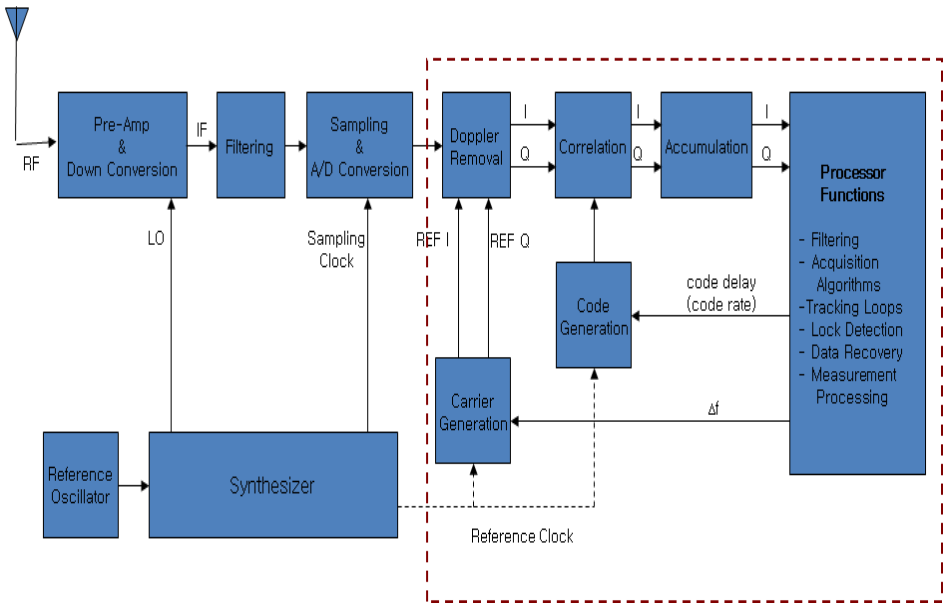


Figure 2-7. Carrier Tracking Loop Structure

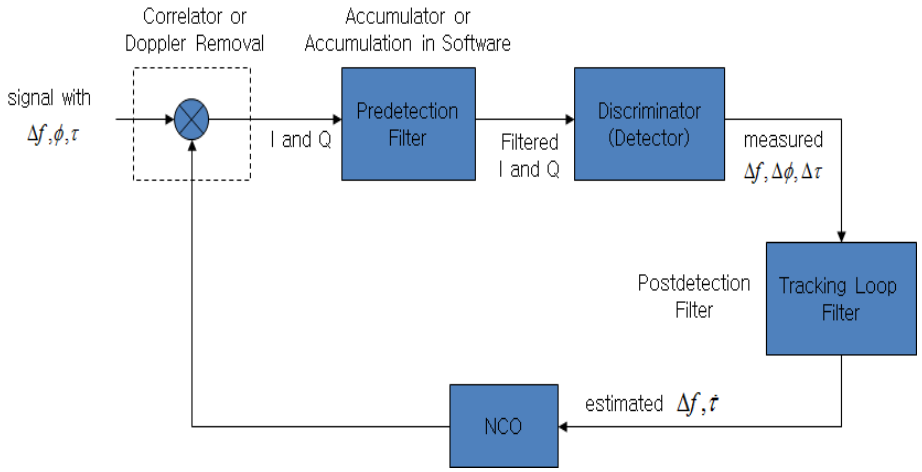


Figure 2-8. Code-tracking loop structure

2.1.5. Navigation Message Decoding

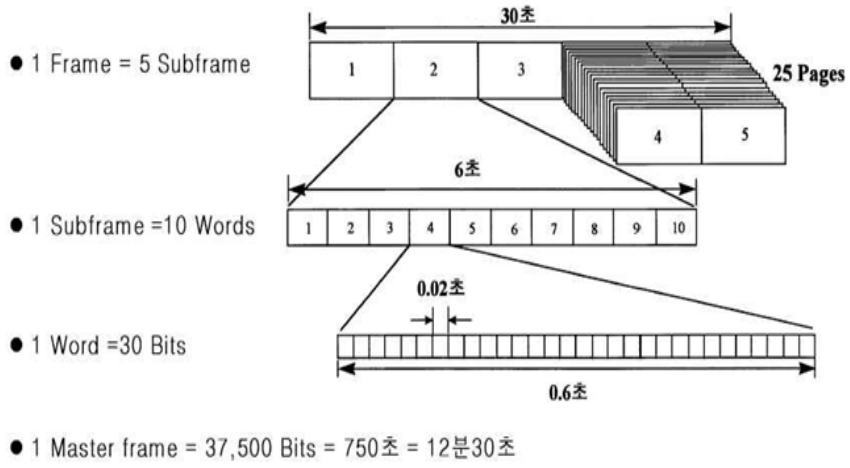


Figure 2-9. Navigation Data Structure

Navigation data include information necessary for the user to accurately calculate the position of each satellite and information about the transmission time of each navigation signal, as well as information on switching GPS time to UTC and several calibration data. Navigation data are structured as figure 2-9. First, a

frame exists that represents the primary data unit and is 30 seconds long. Each frame consists of five subframe (6 seconds), each subframe consists of 10 words (0.6 seconds), and each word has 30 navigation data bits (20 ms). The subframe No. 4 and No. 5 of each frame has 25 versions, so the master frame, the total unit of data that is transmitted repeatedly, becomes 25 frames, or 12 minutes and 30 seconds long. The TLM of word 1 and HOW of word 2 are transmitted first at the beginning of each subframe. For TLM, the pre-amplifier set at bits 1 to 8 is broadcast and the receiver detects the start point of the subframe. HOW includes information such as time of week and subframe numbers that give information about the signal transmission time. Word 3 through 10 broadcasts satellite clock error compensation information, satellite location information, and other information for each subframe. It is not different from the demodulation process in general communication, which is the decoding process of navigation data. Once the subframe start point is located using the preamble contained in the TLM and the subframe number is identified in the HOW, the data structure of the subframe is defiend in advance, so each data bit can be stored in that data. The first thing to be done is to locate the bit inverted position. In our software defined receiver (SDR), signal tracking is done in 1 ms. And the count at that point is stored. The position of the bit reversal could be determined by finding out where the bit reversal occurs and checking if it occurs periodically in 20 ms at that point. The starting point of the frame is determined using the preamble. The preamble to the GPS signal is 10001011. Decoding module in SDR searches the subframe start point by scanning the preamble to the stored bit. Also, after decoding the frame number through HOW word, bitstream is stored in the structure system.

2.1.6. Pseudorange model and range calculation

To calculate a user's position, at first the distance between the user and the satellite is calculated. Pseudorange is a measure distance with several errors in the actual distance, and the model for that distance is shown below. This distance contains several error source, and also includes user clock error. For this reason, this distance is called pseudorange.

$$\rho^i = d^i + B + b^i + I^i + T^i + \delta R^i + M^i + \varepsilon^i \tag{2-2}$$

ρ^i : *jth GPS pseudorange*
d^i : *geometric distance between* jth *GPS satellite and receiver*
B: *clock error of receiver for GPS time*
b^i : *clock error of* jth *GPS satellite for GPS time.*
I^i : *ionosphere delay error of jth GPS pseudorange*
T^i : *tropospheric delay error of jth GPS pseudorange*
δR^i : *jth GPS satellite position error*
M^i :*multiple path error of* jth GPS *signal*
ε^i : *jth GPS pseudorange noise*

In the navigation calculation module, the location and speed of the receiver are calculated using the satellite location information and various calibration information obtained through the navigation message decoding, and the calculation results of the distance measurements. The following models of distance measurements are used for navigation calculation process.

$$\rho^i = d^i + B + \varepsilon^i \tag{2-3}$$

In the following, the process of deploying navigation equations and calculating navigation solution by using the pseudorange measurement model presented above. $\overline{R_u}$ is the location vector of the receiver, $\overline{R^j}$ is location vector of the jth GPS satellite, $\overline{d}$ is the distance vector from the receiver position to the *jth* GPS satellite location , and $\hat{e}^j$ is the unit vector looking at the *jth* GPS satellite from the receiver. These vectors allow the measurement of the distance to be expressed as follows.

$$\rho^i = (\overline{R^j} - \overline{R_u}) \cdot \hat{e}^j + B \tag{2-3}$$

$$\hat{e}^j = \frac{\overline{d}}{\|d\|} \tag{2-4}$$

To move all remaining terms to the right to move the unknown receiver position vector $\overline{R_u}$ and receiver clock error B to the left in the distance vector expression is as follows

$$\overline{R_u} \cdot \hat{e}^j - B = \overline{R^j} \cdot \hat{e}^j - \rho^j \tag{2-5}$$

With distance measurements taken from the number of m GPS satellite signals, the above expressions could be obtained as below. All expressions could be aggregated into a single matrix and expressed as follows.

$$\begin{bmatrix} \hat{e}^{1^T} & -1 \\ \hat{e}^{1^T} & -1 \\ \vdots & \\ \hat{e}^{m^T} & -1 \end{bmatrix} \begin{bmatrix} \overline{R_u} \\ B \end{bmatrix} = \begin{bmatrix} \overline{R_u} \cdot \hat{e}^{1^T} - \rho^1 \\ \overline{R_u} \cdot \hat{e}^{2^T} - \rho^2 \\ \vdots \\ \overline{R_u} \cdot \hat{e}^{m^T} - \rho^m \end{bmatrix} \tag{2-6}$$

A simple representation $H \cdot X = Z$ of the above matrix is an estimation problem for the state variable vector X. Since the size of X is 4 by 1, it is possible to estimate the by using the least square method when *m* is greater than *4*. Estimates of through the least square method are made by means of the following formula

$$\hat{X} = (H^T H)^{-1} H \cdot Z \tag{2-7}$$

Using the configuration of these navigation equations and the process of estimating navigation, we now look at the actual process of calculating navigation solution from GPS receivers. When the navigation equation was first configured, only the satellite position was calculated, but there is no information about the receiver position, making it impossible to calculate the unit line vector. Therefore, the initial receiver position is assumed appropriately to be calculated. Next, the navigation equation is constructed using satellite location vectors and distance measurements. Navigation is now conducted through the least square method, estimated and checked for convergence. If not converging, re-compute the unit

line of sight vector based on the newly calculated receiver position, and repeat the calculation. If the navigation converges, the convergent value is taken as the final navigation solution. Calculation of user velocity is done in the same way as position calculation using Doppler measurements from the receiver. Doppler measurements could be generated by simply converting the carrier NCO values obtained from the signal tracking from the receiver into units of speed, and the calculation process is the same method as position calculation. One difference is that the line of sight vector is already known in the position calculation and this value can be used immediately.

2.2. GNSS interferences and attack strategies

2.2.1. Types of GNSS interferences

Generally, there are three GNSS interferences: jamming, meaconing, and spoofing. Jamming is a method that blocks reception of authentic signals by transmitting signals larger than original signals in the same band, making location impossible. The meaconing attack indicates that receives a GPS signal from a repeater and then amplifies and transmits it. Repeater is a device that receives a GPS signal and transmits it by amplifying its signal power. Because the repeater amplifies and transmits the received GPS signal without any operation, if the target receiver receives the signal, the positioning results becomes the repeater location. Spoofing attack is a method that causes receivers to calculate incorrect position, velocity and time by manipulating the code start point using a GPS signal structure that opens to the public. In this case, when the spoofing signal is received from the receiver, the spoofing signal passes through the code tracking point of

the authentic signal by adjusting the signal strength so that the spoofing signal is received higher than the authentic signal, resulting in a larger peak than that of authentic signal, and therefore tracking the spoofing signal rather than the authentic signal. In this regard, it may be arranged as shown in the table below.

Table 2-1. Interference types and its signal strength offset and results

Interference type	Signal strength offset	results
jamming	26dB	PNT solution impossible
meaconing	26dB	PNT solution same with repeater
spoofing	3dB	False PNT solution

The form of interferences according to the signal strength are divided as shown in Figure 2-10 [29]. That is, if the user is located near a spoofer and the strength of the received spoofing signal is greater than 26 dB compared to the authentic signal, a phenomenon such as jamming effect occurs. In addition, if the signal strength offset between authentic signal and the spoofing signal is greater than 1~2 dB and less than 26 dB of the signal, the spoofing attack phenomenon occurs. If the distance between the spoofer and the user receiver is far, the relative signal strength of the spoofing signal is similar or smaller than the authentic signal, it is expected that the existing signal will be maintained.

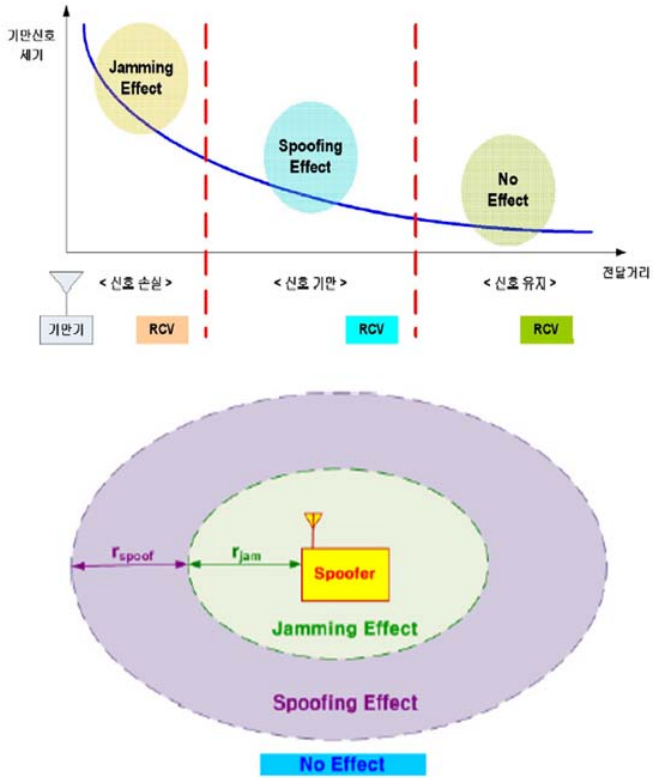


Figure 2-10. Effects on the Receiver by interference signal strength

2.2.2. Interference attack strategies

2.2.2.1. Overt type attack

An overt type attack is a method of sending out strong signals from spoofer, allowing users to receive signals from spoofer rather than authentic signals. The two signals received from the user receiver in the event of such an attack are shown in figure 2-11 [68]. In the corresponding figure, the x-axis is the code delay and the y-axis is the frequency axis. The z-axis is the power of the signal. A peak occurs when the code delay and frequency (Doppler) of the

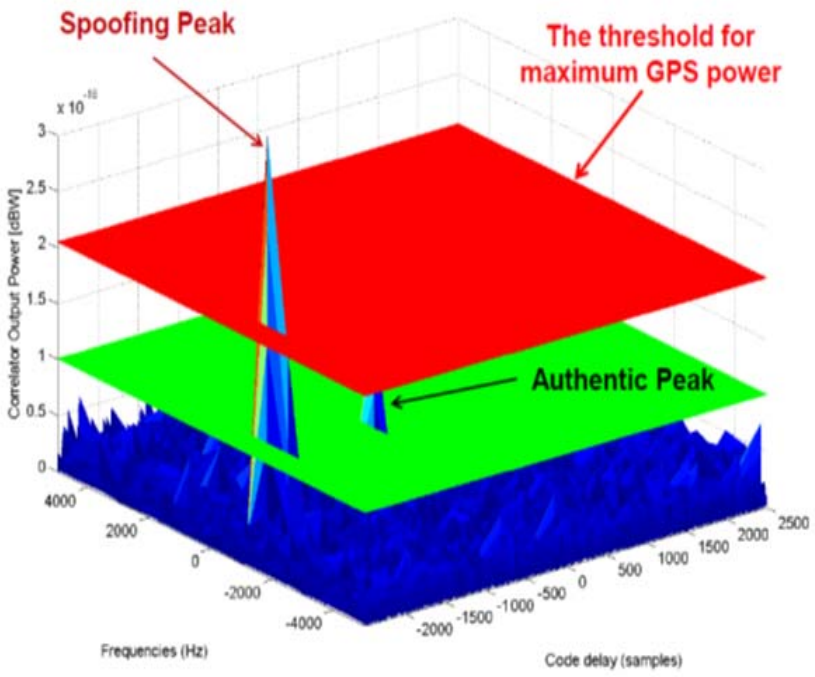


Figure 2-11. Impact on receiver by overt attack

incoming signal and the replica signal generated by the receiver are correct. A blue wave is a noise floor when only the authentic signal exists. In this case, the authentic signal is tracked from the receiver without any problems. If the user receiver exists near the spoofer, as shown in figure 2-10, and the signal strength of the spoofing signal is relatively greater than the authentic signal, the overall noise floor is raised like a green plane by the spoofing signal. In order to maintain the signal tracking of the authentic signal at the receiver, power above a certain threshold must be maintained, but an increase in the overall noise floor will cause the SNR of the existing signal to fall. In this case, signal tracking is not maintained and tracking lock is lost. Reacquiring a signal from the receiver will result in the acquisition of code delay and Doppler values around the spoofing signal and

eventually tracking the spoofing signal. Ultimately, if the above occurs on all channels, the user's position will be estimated as the location controlled by spoofer.

2.2.2.2. Covert type attack

A covert type attack is a strategy in which a spoofer uses a spoofing signal to calculate the wrong position, velocity and time from the receiver. What makes covert attacks different from overt type attacks is that the tracking lock of the user's receiver remains in sweep process. This is because the relative signal strength of a signal intended for spoofing is generally weaker than that of overt type attack, so the SNR of a user receiver does not fall below a certain threshold. Assuming that the spoofer knows the location and speed of the user receiver, the code delay and the Doppler estimates of the specific satellite signal being tracked by the user receiver could be known. In this situation, the spoofer could take away the tracking point by changing the code delay while keeping the Doppler value with same. Figure 2-12 is a simulation of the covert type spoofing process. In the picture on the left, blue dot lines mean the authentic signal and red dot lines indicate the spoofing signal. The spoofing signal from 1 to 5 approaches from right to left, showing the process of taking away the tracking point of the user receiver. In addition, the strength of the spoofing signal and the authentic signal according to each number is shown in the figure 2-12 on the right [8]. Initially, the spoofing signal approaches a signal weaker than the authentic signal, but increases the signal strength of the spoofing signal near the point at which the code delay is matched. If the signal strength is raised and the code delay point of the spoofing signal is continuously changed, the user receiver will track the spoofing

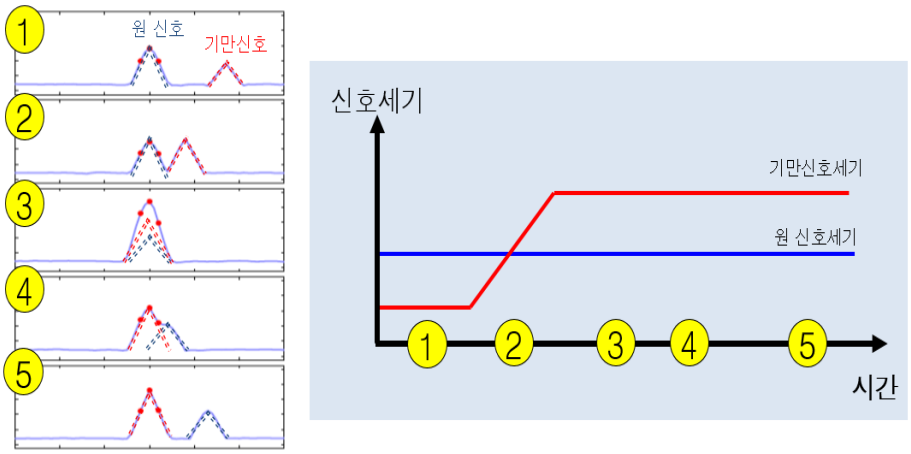


Figure 2-12. Changes in signal tracking points in the user receiver with covert type signal strength

Table 2-2. Effect of spoofing parameters on spoofing results

	Covert capture parameter	Capture success probability (parameter value ↑)
Signal	Signal power	increase
	Sweep velocity	decrease
Receiver	Bandwidth	increase

signal because it is supposed to track the larger signal. The spoofing signal keeps moving to the right, so it becomes increasingly distant from the authentic signal. For all channels, when the covert type strategies are successful, the induced user position becomes out that is generated by spoofing.

In this research, we define four spoofing parameter such as spoofing

signal power, spoofing velocity, receiver bandwidth and DLL loop filter order. The characteristic of each spoofing parameter is like table 2-2. Existing studies have analyzed the characteristics of the spoofing parameters, but the correlation of each parameter could not be presented. And previous researches does not suggest the boundary value of each parameter to determine the success or failure of spoofing process. Therefore, it is difficult to know how much signal to transmit in order for spoofing to succeed.

Chapter 3. Covert Capture Effectiveness Equation

3.1. Authentic and spoofing signal ACF model

This chapter contents are published in the MDPI sensors [57].

In this section, the authentic and spoofing signal models are presented. To generate a spoofing signal aligned with the authentic signal, the spoofer should estimate the position and velocity of the target receiver. Figure 3-1 shows a brief illustration of a spoofing scenario. First, the spoofer estimates the position and velocity of the victim receiver using radar [6]. The spoofer can then calculate the aligned spoofing signal by compensating for the spoofer processing delay and transmission delay. Moreover, the power of the spoofing signal should be greater than that of the authentic signal. Therefore, it is necessary to compensate for the propagation loss depending on the distance between the spoofer and the victim receiver. The signal received at the victim receiver antenna can be represented using a complex baseband model as follows:

$$s(t) = C[t-\tau_a(t)]\exp(j\phi_a(t)) + \sqrt{P_s}C[t-\tau_s(t)]\exp(j\phi_s(t)) + n(t) \quad (3\text{-}1)$$

Here,

- $s(t)$ denotes the total received signal;
- C denotes the pseudorandom code;
- $\tau_a(t)$ is the code phase of the authentic signal;
- $\tau_s(t)$ is the code phase of the spoofing signal;
- P_s is the spoofing power advantage;

- $\phi_a(t)$ is the carrier phase of the authentic signal;
- $\phi_s(t)$ is the carrier phase of the spoofing signal;
- $n(t)$ is the complex zero-mean white Gaussian noise (AWGN).

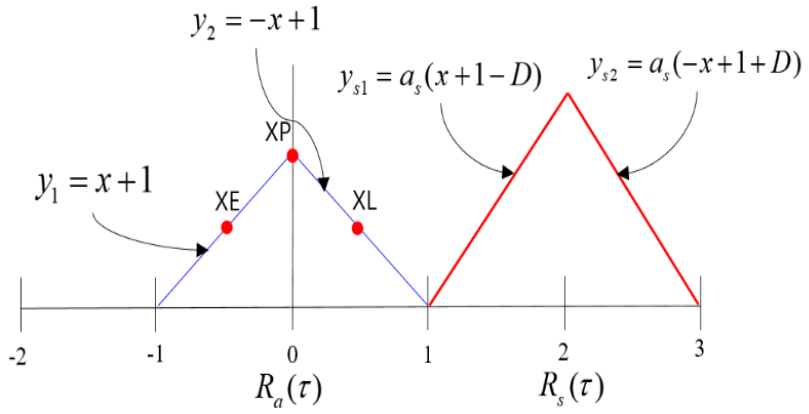


Figure 3-1. ACFs of the authentic and spoofing signal models.

In the receiver, a correlation process is implemented to track the input signal $s(t)$. Figure 3-1 shows the ACF model of $s(t)$. The blue triangle indicates the ACF of the authentic signal, whereas the red triangle indicates the ACF of the spoofing signal. The horizontal axis represents the chip offset, and the vertical axis represents the normalized correlator output (the amplitude of the authentic signal is 1). The parameters, shown in figure 3-1, can be expressed as follows:

$$
\begin{aligned}
&y_1 = x+1, \quad -1 < \tau \le 0 \\
&y_2 = -x(\tau)+1, \quad 0 < \tau < 1 \\
&R_a(\tau) = y_1 + y_2 \\
&y_{s1} = a_s(x+1-D), \quad -1+D < \tau \le 0+D \\
&y_{s2} = a_s(-x+1+D), \quad 0+D < \tau < 1+D \\
&R_s(\tau) = y_{s1} + y_{s2} \\
&R(\tau) = R_a(\tau) + R_s(\tau) \\
&D = \tau_a - \tau_s
\end{aligned}
\tag{3-2}
$$

Here,

- y_1 indicates the left line of the ACF of the authentic signal;
- y_2 indicates the right line of the ACF of the authentic signal;
- $R_a(\tau)$ is the ACF of the authentic signal;
- y_{s1} indicates the left line of the ACF of the spoofing signal;
- y_{s2} indicates the right line of the ACF of the spoofing signal;
- a_s is the slope of the ACF of the spoofing signal;
- $R_s(\tau)$ is the ACF of the authentic signal;
- $R(\tau)$ is the ACF of the total signal;
- D is the difference in the code phases between authentic and spoofing signals;
- XE is the accumulation result with the replica code separated 0.5 chip early;
- XP is the accumulation result with the replica code;
- XL is the accumulation result with the replica code separated 0.5 chip late.

The XP could be written as [20]:

$$XP[n] = R(\Delta\tau_a[n]) \cdot \frac{\sin(\pi\Delta f_a[n] \cdot T)}{\pi\Delta f_a[n] \cdot T} \exp(j\Delta\phi_a[n]) + R[\Delta\tau_s[n]] \cdot \frac{\sin(\pi\Delta f_s[n] \cdot T)}{\pi\Delta f_s[n] \cdot T} \exp(j\Delta\phi_s[n]) \quad (3\text{-}3)$$

Here,

- $\Delta\tau_a[n]$ is the code phase difference between the local replica and the authentic signal;
- $\Delta\tau_s[n]$ is the code phase difference between the local replica and the spoofing signal;
- $\Delta f_a[n]$ is the Doppler frequency difference between the local replica and the spoofing signal;
- $\Delta f_s[n]$ is the Doppler frequency difference between the local replica and the spoofing signal;
- $\Delta\phi_a[n]$ is the carrier phase difference between the local replica and the authentic signal;
- $\Delta\phi_a[n]$ is the carrier phase difference between the local replica and the spoofing signal.

In the next section, we explain the change in the replica code phase, τ, depending on the success of failure of a spoofing attack. In our simulation, we assume that the code phase and Doppler frequency of the replica are perfectly aligned with the authentic signal before the spoofing signal approaches. This implies that $\Delta\tau_a[1]$ and $\Delta f_a[1]$ are zero.

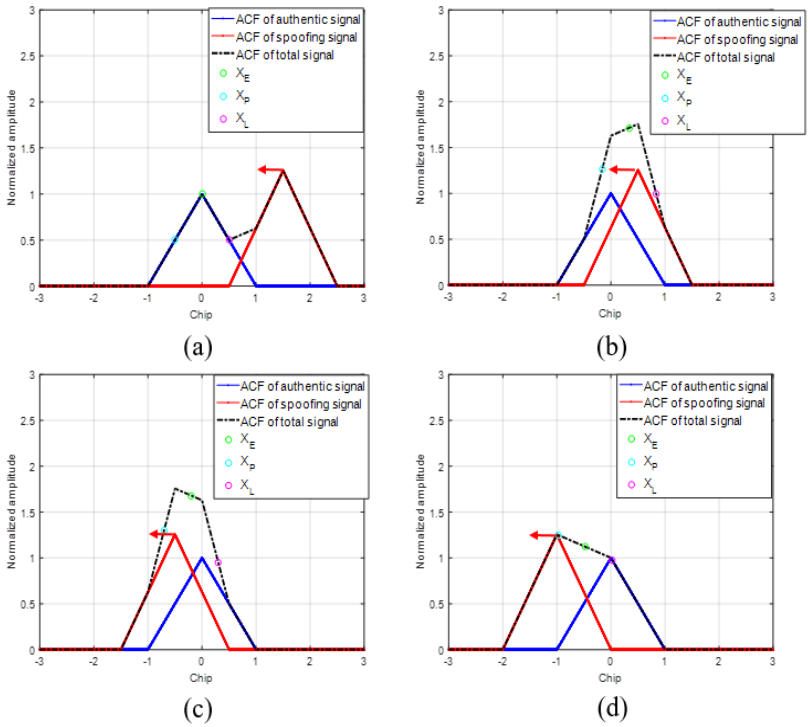


Figure 3-2. ACF variation with respect to the difference in the code phases between the authentic and spoofing signals.

3.2. Spoofing scenario simulation using ACF model

Using the ACF models explained in Section 2, we conduct the spoofing simulation. We assume that the authentic signal is stationary and that the spoofing signal is moving from right to left with a static velocity. This simulation is done without noise. In general, the DLL discriminator is used to calculate the feedbackoutput using XE and XL and thereby track the incoming signal. The replica code phase gradually aligns with the code start point of the incoming signal during DLL code tracking. In our spoofing simulation, the DLL initially tracks the authentic signal. When the spoofing signal approaches and overlays with the authentic signal, the ACF changes. Figure 3-2 shows the sequential ACF variation

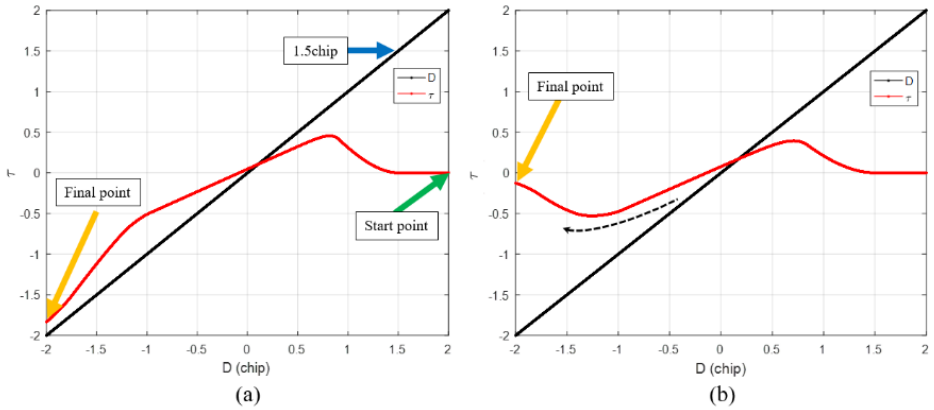


Figure 3-3. Calculated histories of the local replica code phase in case of (a) successful spoofing attack; (b) spoofing attack failure.

during the spoofing simulation. The cyan circle indicates the XP, which is the prompt of the DLL. The position of XP in each figure is different. The position of XP is determined by the shape of the ACF and the positions of previous XP and DLL settings. Figures 3-3 (a) and (b) show the τ histories of the two spoofing simulations. The only difference between the two simulations is the receiver bandwidth. In Figure 3-3, the black lines indicate the code phase distance between the authentic and spoofing signals. Because the authentic signal is fixed at zero, this can be considered the position of the spoofing signal relative to the authentic signal in the code domain. We now focus on Figure 3-3 (a). The green arrow indicates the start point of τ. At the start of the simulation, τ is zero. The spoofing signal approaches the authentic signal from a distance of 2 chips. When the spoofing signal reaches a distance of 1.5 chips, τ starts to gradually increase, as the spoofing signal starts to affect XL. A blue arrow indicates the point where τ is increasing. The peak point of the total ACF $R(\tau)$ is always the same as that of the spoofing ACF $R_a(\tau)$. Therefore, τ moves to the peak point of $R(\tau)$ until the discriminator output becomes zero. After the spoofing signal passes the

authentic signal, the peak point of $R(\tau)$ is located on the negative side, as shown in Figure 3-3(c). Finally, τ follows the spoofing signal. The orange arrow represents the final value of the τ. In the case of Figure 4(a), τ follows the spoofing signal, and therefore, the spoofing attack is successful. Figure 3-3 (b) shows the other spoofing simulation case. In Figure 3-3 (b), the final value of τ returns to zero. τ seems to chase the spoofing signal, as indicated using the dotted black line, but eventually returns to its location. This implies that the spoofing attack is a failure. The difference between the two simulations is that the bandwidths of the receivers used are different. The receiver bandwidth in the first simulation is 5 Hz, whereas it is 3 Hz in the second. As shown in the simulation results, the greater the bandwidth of the receiver, the more vulnerable it is to a spoofing attack. Moreover, the higher the strength of the spoofing signal, the higher is the probability of a successful spoofing attack. The faster the spoofing signal sweeps, the more likely it is that the spoofing attack will fail. Table 3-1 lists the changes in the spoofing attack results with respect to increases in the bandwidth, signal strength, and sweep velocity. However, it is difficult to determine how strong a signal should be for a successful spoofing attack. It is also difficult to obtain a correlation between the different parameters for a successful spoofing attack.

Table 3-1. Relationship between spoofing parameters and spoofing results.

parameters	Spoofing attack success probability
increase in the spoofing signal strength	increase
Increase in spoofing signal sweep velocity	decrease
increase in the DLL bandwidth	increase

3.3. Development of spoofing process equation

3.3.1. conventional approach for tau calculation

XP is calculated through DLL using the ACF and previous XP. The first-order DLL can be expressed as follows:

$$\begin{aligned} &\Delta\tau[n] = \frac{XE[n] - XL[n]}{2} \\ &\tau[n+1] = \tau[n] - \omega_0 \cdot T \cdot \Delta\tau[n], \\ &\omega_0 = 0.25 \cdot B \end{aligned} \tag{3-4}$$

where $\Delta\tau$, B, and T indicate the discriminator output, integration time, and bandwidth, respectively. In general, the spoofing attack results can be obtained by determining which signal the DLL is tracking after the spoofing signal completely sweeps the authentic signal. In other words, if the integration time of the receiver is 1 ms, it is necessary to repeatedly calculate the equation thousands of times to obtain the spoofing attack results. This calculation can be expressed as follows:

$$
\begin{aligned}
&k = \frac{\omega_0 \cdot T}{2} \\
&\tau[2] = \tau[1] - \omega_0 \cdot T \cdot \Delta\tau[1] = \tau[1] - k \cdot \{R_1(\tau[1] - \frac{1}{2}) - R_1(\tau[1] + \frac{1}{2})\} \\
&\tau[3] = \tau[2] - \omega_0 \cdot T \cdot \Delta\tau[2] = \tau[2] - k \cdot \{R_2(\tau[2] - \frac{1}{2}) - R_2(\tau[2] + \frac{1}{2})\} \\
&\quad \vdots \\
&\tau[n] = \tau[n-1] + \omega_0 \cdot T \cdot \Delta\tau[n-1] = \tau[n-1] - k \cdot \{R_{n-1}(\tau[n-1] - \frac{1}{2}) - R_{n-1}(\tau[n-1] + \frac{1}{2})\} \\
&\tau[n] = \tau_1 - (n-1) \cdot k \cdot \sum_{m=1}^{n-1} \{R_m(\tau[m] - \frac{1}{2}) - R_m(\tau[m] + \frac{1}{2})\}
\end{aligned}
\tag{3-5}
$$

For a specific spoofing attack scenario, a lot of computations are required to calculate the final replica code phase $\tau[n]$. Moreover, it is necessary to know τ and ACF at all previous epochs. Thus, the final τ value can be written as follows:

$$
\tau[n] = f(\tau[1],\ \tau[2],\ \tau[3]\ ,,,\ \tau[n-1],\ R_1, R_2, R_3,,, R_{n-1}) \tag{3-6}
$$

3.3.2. proposed approach for τ calculation

In this subsection, we propose a method to compute the spoofing attack results by calculating each epoch at a certain chip interval (CI). The entire spoofing process is summarized in a mathematical equation, i.e., the SPE, and the spoofing results are obtained by one calculation using the SPE. Figure 3-4 shows the results of the τ estimation with respect to the CI. The blue lines indicate the calculation results of τ per 1 ms. The red circles indicate the calculation results of τ per specific CI. The equation for calculating the integration time in terms of the chip interval can be expressed as follows:

$$T = \frac{300}{V_s} \cdot CI \ , \tag{3-7}$$

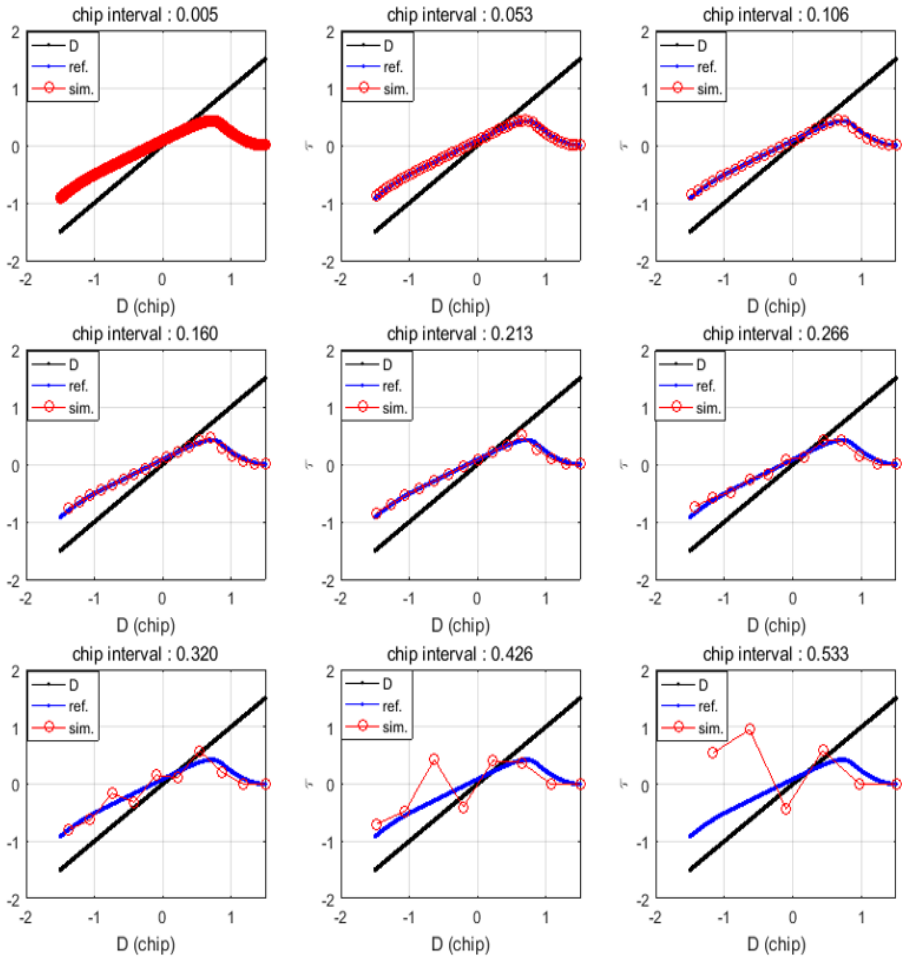


Figure 3-4. Replica code phase histories for various CI settings. The blue lines indicate the calculated replica code phase values of the original calculation. The red lines indicate the calculated replica code phase with respect to the CI.

Table 3-2. Integration time calculation according to CI.

Chip interval (chip)	Integration time (second)
0.005	0.02
0.053	0.2
0.106	0.4
0.160	0.6
0.213	0.8
0.266	1
0.320	1.2
0.426	1.6
0.533	2

where V_s denotes the spoofing sweep velocity (m/s), and the number 300 indicates the wavelength of the C/A code in meter-scale. Table 3-2 lists the calculated integration times with respect to each CI in case of the spoofing sweep velocity is 80 m/s. τ error decreases with the decrease in the CI. However, additional calculations are required to estimate the final τ when CI is low. In our research, we set the CI to 0.125 considering the complexity of the equation and τ error.

3.3.3. Spoofing attack success or failure criteria

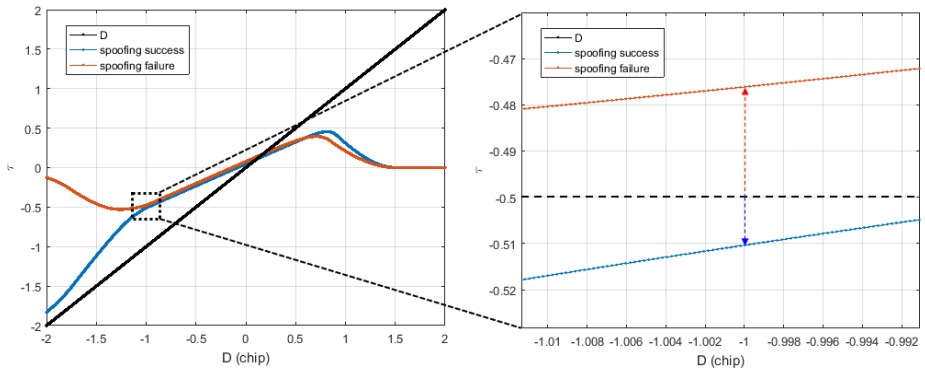


Figure 3-5. Different τ values at D is -1 according to the spoofing attack success or failure.

Figure 3-5 shows the τ results of DLL when the spoofing signal sweeps the authentic signal with constant velocity. The blue line indicates the case of a successful spoofing attack, whereas the red line indicates the case of a failed spoofing attack. Generally, the success or failure of a spoofing attack can be determined from the type of signal the DLL tracks when the spoofing signal completely sweeps the authentic signal. In both the simulations, the only difference is the bandwidth of the receiver.

The success or failure of a spoofing attack can be determined by looking at the absolute value of τ at the point where D is −1. Figure 3-5 (b) shows the region enclosed in the black box shown in Figure 3-5 (a). If the spoofing attack is successful, the absolute value of τ exceeds 0.5 at the point where D is −1, and if it fails, the absolute value of the prompt is lower than 0.5.

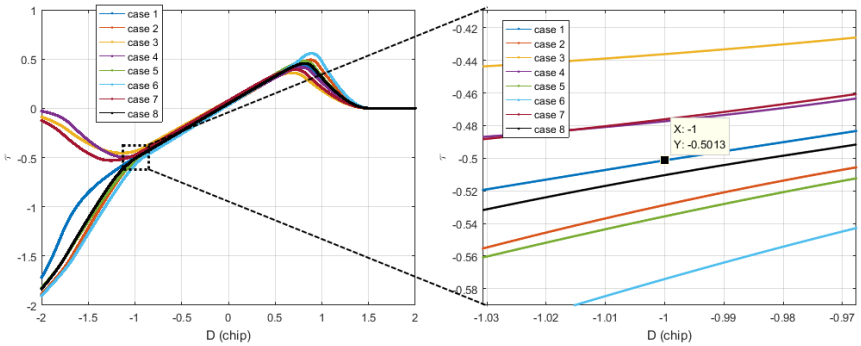


Figure 3-6. Different τ values at D is -1 according to the various spoofing attack scenarios.

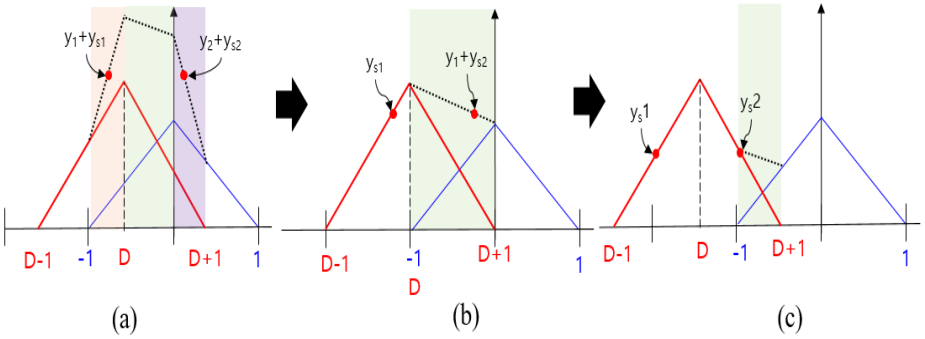


Figure 3-7. ACF change according to the spoofing signal in case that spoofing signal strength is larger than authentic signal strength.

Figure 3-6 shows the τ estimates for various spoofing parameters listed in Table 3-3 It is noteworthy that the absolute value of τ at D is −1. As shown in Figure 3-6, if the absolute value of τ exceeds 0.5 chip when D is −1, the DLL tracks the spoofing signal.

The following analysis shows that the criterion used for determining the spoofing result is reasonable. If the spoofing sweep velocity is very low or if the bandwidth is very high in the spoofing simulation, there will be sufficient time or

Table 3-3. τ estimates for various spoofing parameters.

Case	Spoofing signal strength offset (dB)	Sweep velocity (m/s)	Bandwidth	Spoofing results	$\lvert\tau\rvert$ at D = -1
1	1.5	50	3	Success	0.5013
2	1.5	50	5	Success	0.5287
3	1.5	70	3	Failure	0.4362
4	1.5	70	5	Failure	0.4774
5	2	50	3	Success	0.5357
6	2	50	5	Success	0.5741
7	2	70	3	Failure	0.4761
8	2	70	5	Success	0.5104

control input for the DLL to track the peak point of the ACF. In this case, the discriminator output would become zero and XP would be located at the point where XE equals XL. Figure 8 shows a series of snapshots where the discriminator output is zero with respect to the ACF. The τ value for the case, shown in Figure 3-7 (a), can be derived as follows:

$$
\begin{aligned}
&XE = XL \\
&y_1 = x+1 \\
&y_2 = -x+1 \\
&y_{s1} = a_s(x+1-D) \\
&y_{s2} = a_s(-x+1+D) \\
&y_1 + y_{s2} = y_2 + y_{s2} \\
&x+1+a_s(x+1-D) = -x+1+a_s(-x+1+D) \\
&(a_s+1)x + a_s - a_sD + 1 = -(a_s+1)x + a_s + a_sD + 1 \\
&(a_s+1)(\tau - \frac{1}{2}) + a_s - a_sD + 1 = -(a_s+1)(\tau + \frac{1}{2}) + a_s + a_sD + 1 \\
&(a_s+1)\tau - \frac{1}{2}(a_s+1) + a_s - a_sD + 1 = -(a_s+1)\tau - \frac{1}{2}(a_s+1) + a_s + a_sD + 1 \\
&2(as+1)\tau = 2asD \\
&\tau = \frac{a_sD}{a_s+1}
\end{aligned}
\tag{3-8}
$$

Moreover, the τ values for the cases, shown in Figures 3-7 (b) and (c), can be derived in a similar manner as follows:

$$
\begin{aligned}
&y_{s1} = y_1 + y_{s2} \\
&a_s(x+1-D) = a_s(-x+1+D) + x + 1 \\
&a_s(\tau - \frac{1}{2}) + a_s - a_sd = (-a_s+1)(\tau + \frac{1}{2}) + a_s + a_sd + 1 \\
&\tau = \frac{2a_sd + \frac{3}{2}}{2a_s - 1}
\end{aligned}
\tag{3-9}
$$

$$
\begin{aligned}
&y_{s1} = y_{s2} \\
&a_s(x+1-D) = a_s(-x+1+D) \\
&a_s(\tau - \frac{1}{2}) + a_s - a_sD = -a_s(\tau + \frac{1}{2}) + a_s + a_sD \\
&\tau = D
\end{aligned}
\tag{3-10}
$$

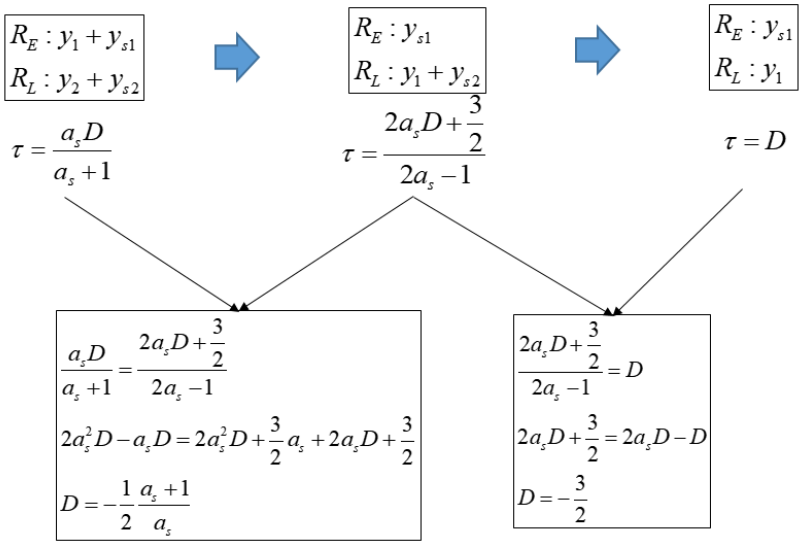


Figure 3-8. Equation of τ and D in case that the spoofing signal strength is larger than authentic signal and XE is same with XL.

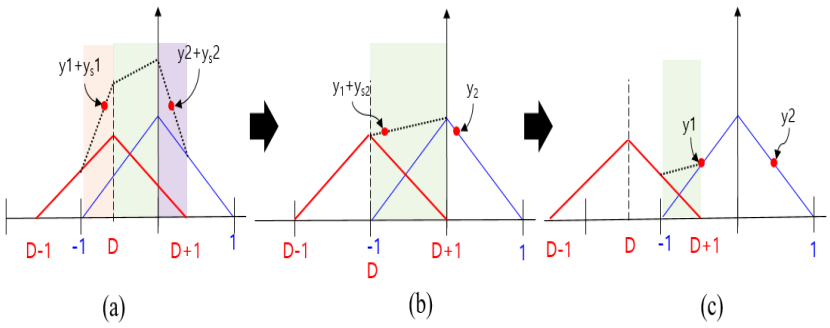


Figure 3-9. ACF change according to the spoofing signal in case that spoofing signal strength is lower than authentic signal strength.

Figure 3-8 shows the summary of Equations (3-11) to (3-13). For any ACF, shown in Figure 3-9, τ can be estimated using a_s and D when the spoofing sweep velocity is very low or when the bandwidth is considerable. Moreover, it is possible to calculate D corresponding to the different equations of τ. Figure 10 shows the ACF change with respect to the spoofing signal when the spoofing signal strength is lower than the authentic signal strength. We can derive an

equation to calculate τ in the same manner as above. The τ value for the cases, shown in Figures 10(a)–(c), can be derived as follows:

$$
\begin{aligned}
& y_1 + y_{s1} = y_2 + y_{s2} \\
& a_s(x+1-D) + x + 1 = a_s(-x+1+D) - x + 1 \\
& (a_s+1)(\tau - \frac{1}{2}) + a_s - a_s D + 1 = -(a_s+1)(\tau + \frac{1}{2}) + a_s + a_s D + 1 \\
& \tau = \frac{a_s D}{a_s + 1}
\end{aligned}
\tag{3-11}
$$

$$
\begin{aligned}
& y_1 + y_{s2} = y_2 \\
& a_s(-x+1+D) + x + 1 = -x + 1 \\
& (-a_s+1)(\tau - \frac{1}{2}) + a_s + a_s D + 1 = -(\tau + \frac{1}{2}) + 1 \\
& \tau = \frac{\frac{3}{2}a_s + a_s D}{a_s - 2}
\end{aligned}
\tag{3-12}
$$

$$
\begin{aligned}
& y_1 = y_2 \\
& x + 1 = -x + 1 \\
& (\tau - \frac{1}{2}) + 1 = -(\tau + \frac{1}{2}) + 1 \\
& \tau = 0
\end{aligned}
\tag{3-13}
$$

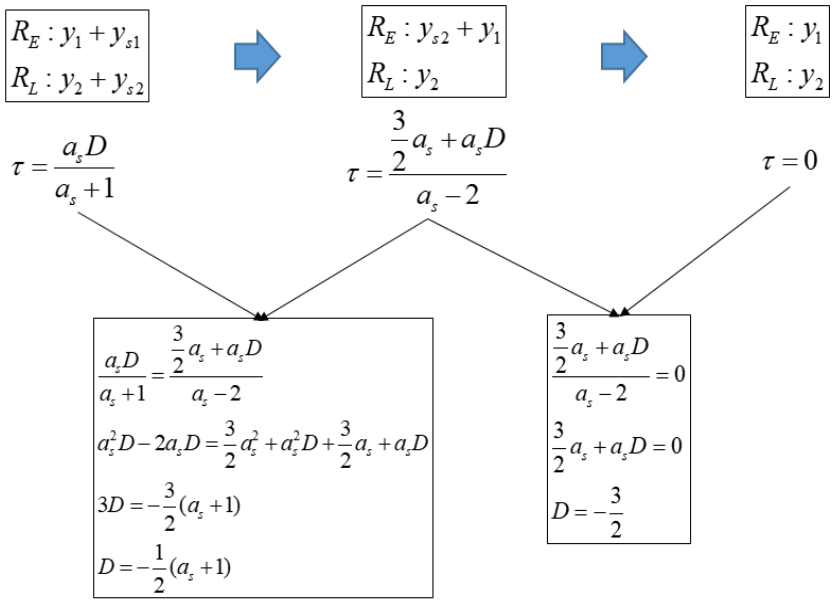


Figure 3-10. Equation of τ and D in case that the spoofing signal strength is lower than authentic signal and XE is same with XL.

Figure 3-10 shows the summary of Equations (3-11) to (3-13). Figure 3-11 shows a graphical representation of Equations (3-9) to (3-13). The blue lines indicate spoofing attack success, whereas the red lines indicate spoofing attack failure. If a_s is greater than 1, τ follows the blue line, and if a_s is lower than 1, it follows the red line. The minimum condition for a successful spoofing attack is that the strength of the spoofing signal should be greater than that of the authentic signal. When a_s is 1, D at the time of transition, from (b) to (c) in Figure 3-8, is −1, and the absolute value of τ becomes 0.5, as follows:

$$D = -\frac{1}{2}\cdot\frac{a+1}{a} = -\frac{1}{2}\cdot\frac{1+1}{1} = -1$$
$$\tau = \frac{2aD+\frac{3}{2}}{2a-1} = \frac{2\cdot 1\cdot(-1)+\frac{3}{2}}{2\cdot 1-1} = -\frac{1}{2} \tag{3-14}$$

Thus, we could regard that the τ value is a boundary value when D is -1. This implies that the absolute value of τ should exceed 0.5 chip before D approaches −1 for a successful spoofing attack.

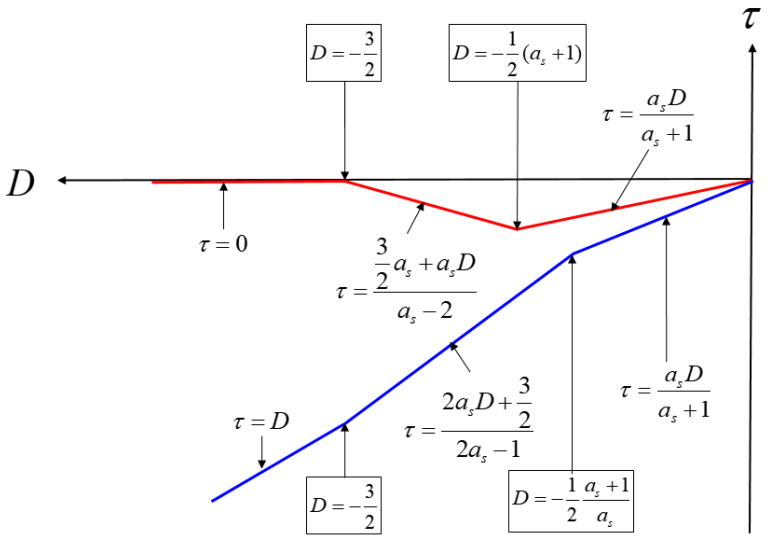


Figure 3-11. Summary of equations (8) to (13).

3.3.4. Derivation of SPE

Figure 3-12 shows the ACF variation per 0.125 CI. Assuming that the spoofing signal shifts from right to left, the spoofing signal affects the DLL discriminator when D approaches within 1.5 chip. Moreover, to determine the spoofing results, we only need to calculate τ until D reaches −1. Therefore, if

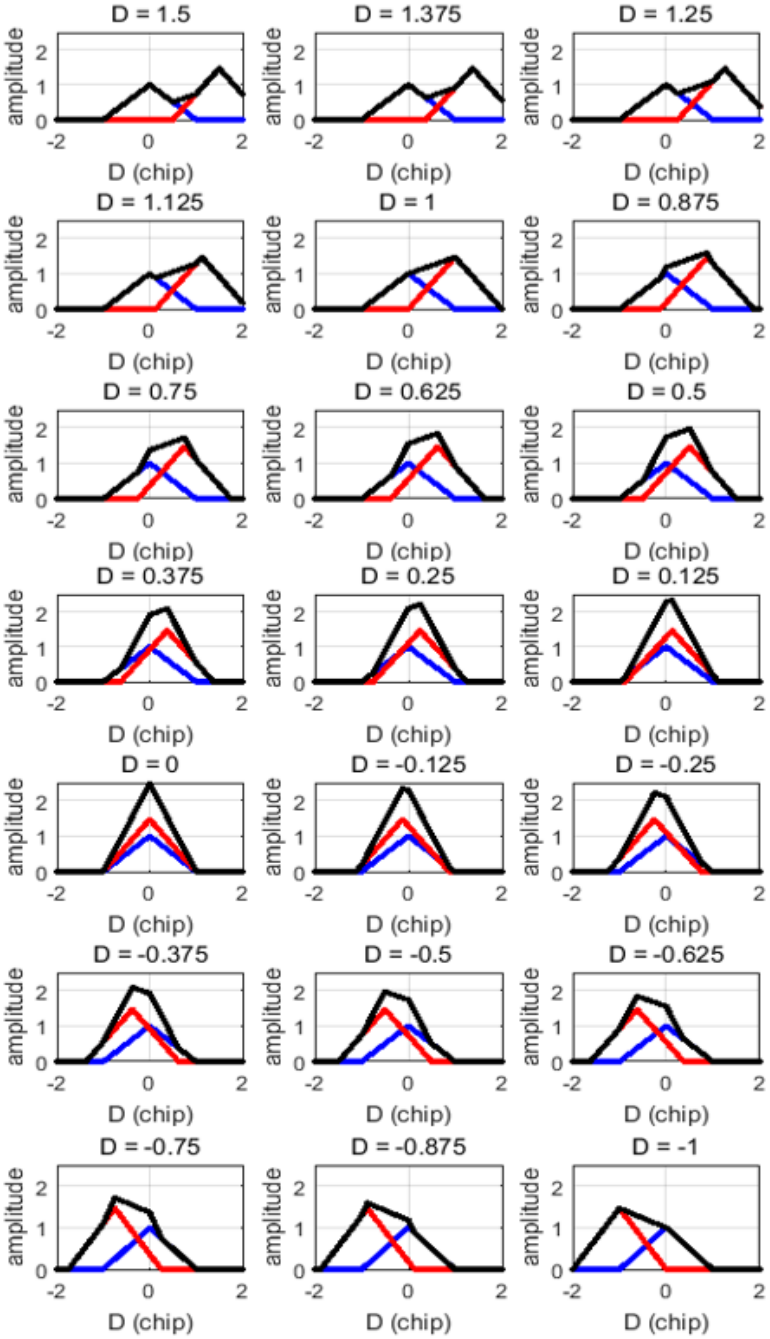


Figure 3-12. ACF variation in case that CI is 0.125.

Table 3-4. Range of $\tau[k]$ and ACF model of XE and XL according to the D[k].

k	D[k]	Range of $\tau[k]-0.5$	Range of $\tau[k]+0.5$	ACF model of X_E	ACF model of X_L
1	1.375	-1~0	0.375~1	y_1	y_2+y_{s1}
2	1.25	-1~0	0.25~1	y_1	y_2+y_{s1}
3	1.125	-1~0	0.125~1	y_1	y_2+y_{s1}
4	1	-1~0	0~1	y_1	y_2+y_{s1}
5	0.875	-1~-0.25	-0.125~0.875	y_1	y_2+y_{s1}
6	0.75	-0.25~0	0.75~1	y_1+y_{s1}	y_2+y_{s2}
7	0.675	-0.375~0	0.625~1	y_1+y_{s1}	y_2+y_{s2}
8	0.5	-0.5~0	0.5~1	y_1+y_{s1}	y_2+y_{s2}
9	0.375	-0.625~0	0.375~1	y_1+y_{s1}	y_2+y_{s2}
10	0.25	-0.75~0	0.25~1	y_1+y_{s1}	y_2+y_{s2}
11	0.125	-0.875~0	0.125~1	y_1+y_{s1}	y_2+y_{s2}
12	0	-1~0	0~1	y_1+y_{s1}	y_2+y_{s2}
13	-0.125	-1~-0.125	0~0.875	y_1+y_{s1}	y_2+y_{s2}
14	-0.25	-1~-0.25	0~0.75	y_1+y_{s1}	y_2+y_{s2}
15	-0.375	-1~0.375	0~0.625	y_1+y_{s1}	y_2+y_{s2}
16	-0.5	-1~-0.5	0~0.5	y_1+y_{s1}	y_2+y_{s2}
17	-0.625	-1~0.625	0~0.375	y_1+y_{s1}	y_2+y_{s2}
18	-0.75	-1~-0.75	0~0.25	y_1+y_{s1}	y_2+y_{s2}
19	-0.875	-1~0.875	0~0.125	y_1+y_{s1}	y_2+y_{s2}

CI is 0.125 chip, it is possible to determine the spoofing attack results by a total of 19 calculations. In Equation (3-19), all the previous τ values and ACF are required to calculate $\tau[n]$. The covert capture effectiveness equation (CCEE) can be used to calculate τ at the point where D is -1 to determine whether the spoofing attack is a successful one or not by only one calculation. We set the CI

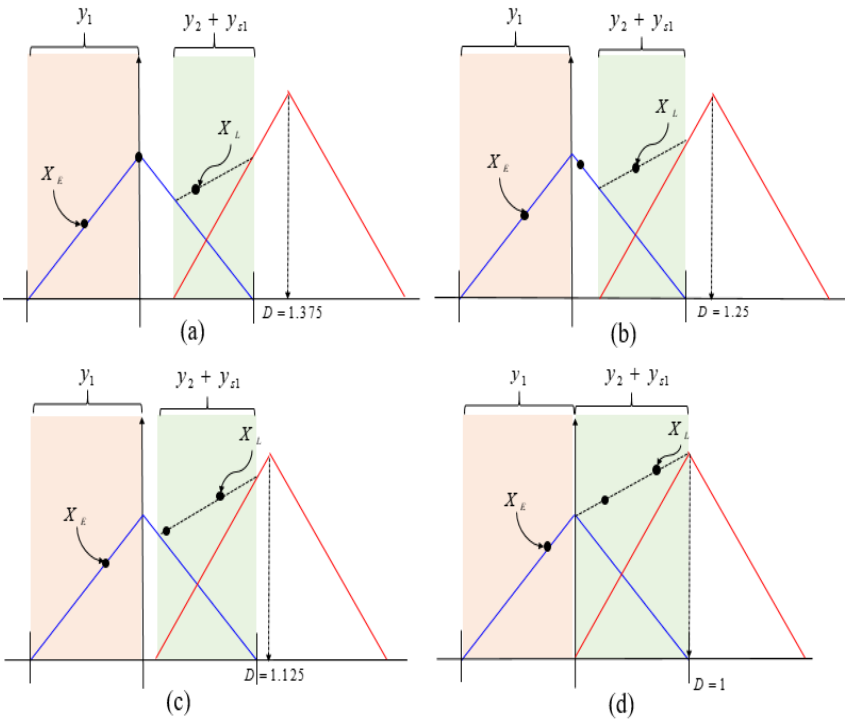


Figure 3-13. ACF snapshots with k from 1 to 4.

to 0.125. Thus, the ACF sequences are generated, as shown in Figure 3-12. However, it is not possible to specify the previous τ values. This is because τ value at a certain D changes according to the spoofing parameters. However, the range of τ can be defined for each D value for τ to be close to −0.5 chip when D is −1. The spoofing attack results can be determined by checking whether the absolute value of τ at D = −1 exceeds 0.5 chip or not. In our case, $\tau[19]$ is the final τ value. For $\tau[19]$ to be close to −0.5, $\tau[18]$ must be in a specific range. Moreover, $\tau[17]$ must be in a specific range for $\tau[18]$ to be in the defined range. Thus, we can define each range according to the D value of the entire process. Table 3-4 lists the range of $\tau[k]$ for each $D[k]$. If each $\tau[k]$ is within the defined

range for each $D[k]$, $\tau[19]$ will be calculated close to −0.5. There are ACF models of XE and XL for each *D*. Figure 3-13 shows the ACF snapshots for *k* ranging from 1 to 4. If $\tau[4]-0.5$ and $\tau[4]+0.5$ are in the defined range, XE and XL can be calculated using $y_1(\tau[4]-0.5)$ and $y_2(\tau[4]+0.5)+y_{s2}(\tau[4]+0.5)$, respectively. $\tau[1]$, $\tau[2]$, $\tau[3]$, and $\tau[4]$ can be expressed as follows:

$$\begin{aligned}\tau[1] &= -k\cdot\{R_1(\tau[1]-\frac{1}{2})-R_1(\tau[1]+\frac{1}{2})\}\\ &= -k\cdot\{y_1(-\frac{1}{2})-y_2(\frac{1}{2})-y_{s1}(\frac{1}{2})\}\\ &= k(\frac{3}{2}a_s-D[1])\end{aligned} \tag{3-15}$$

$$\begin{aligned}\tau[2] &= \tau[1]-k\{y_1(\tau[1]-\frac{1}{2})-y_2(\tau[1]+\frac{1}{2})-y_{s1}(\tau[1]+\frac{1}{2})\}\\ &= k\{(a_s-2)(\frac{3}{2}a_s-a_sD[1])k+(2\cdot\frac{3}{2}a_s-a_s(D[1]+D[2])\}\end{aligned} \tag{3-16}$$

$$\begin{aligned}\hat{\tau}[3] &= k\{(a_s-2)^2(\frac{3}{2}a_s-a_sD[1])k+(a_s-2)(\frac{3}{2}a_s-a_sD[1])k\\ &+(a_s-2)(\frac{3}{2}a_s+\frac{3}{2}a_s-a_sD[2]-aD[2])k+\\ &(3\cdot\frac{3}{2}a_s-a_s(D[1]+D[2]+D[3])\}\end{aligned} \tag{3-17}$$

$$\begin{aligned}\hat{\tau}[4] &= k\{(a_s-2)^3(\frac{3}{2}a_s-a_sD[1])k^3+2(a_s-2)^2(\frac{3}{2}a_s-a_sD[1])k^2\\ &+(a_s-2)^2(\frac{3}{2}a_s+\frac{3}{2}a_s-a_sD[1]-a_sD[2])k^2+(a_s-2)(\frac{3}{2}a_s-a_sD[1])k\\ &+(a_s-2)(\frac{3}{2}a_s+\frac{3}{2}a_s-a_sD[1]-a_sD[2])k+(a_s-2)(\frac{3}{2}a_s+\frac{3}{2}a_s+\frac{3}{2}a_s\\ &-a_sD[1]-a_sD[2]-a_sD[3])k\\ &+4\cdot\frac{3}{2}a_s-a_s(D[1]+D[2]+D[3]+D[4])\}\end{aligned} \tag{3-18}$$

If τ is developed until k is 19, CCEE is complete. CCEE has the following form like:

$$\tau[19] = f(a_s, Vs, B). \tag{3-19}$$

Although the CCEE looks complicated, it is possible to generalize the equation. Thus, we can obtain the CCEE regardless of CI. The inputs to the CCEE are the spoofing signal strength, spoofing sweep velocity, and receiver bandwidth. When the calculation of CCEE is performed, $\tau[19]$ is calculated for D = −1 by just one calculation. The success or failure of the spoofing attack is determined by the absolute value of τ .

3.4. Analysis of CCEE simulation results

3.4.1. CCEE performance analysis

To verify the performance of the SPE, we compared the estimated SEP results with the original DLL results. Table 3-5 presents the various spoofing signal parameters and τ results in the cases of using the original DLL and CCEE at D = −1. τ_{1ms} indicates the estimated replica code phase obtained using the original DLL, the integration time of which is 1 ms. τ_{SPE} is the estimated replica code phase obtained using the SPE. The calculation time required by the CCEE is significantly lower than that required by the original DLL, because τ_{SPE} can be calculated in just one calculation using the SPE. We can see that the replica code phase values estimated using the two methods are very similar. Figure 3-14 shows the τ_{SPE} difference value with respect to CI. The τ_{SPE} difference values

Table 3-5. Various spoofing parameters and τ results in case of using original DLL and SPE.

Case	Spoofing signal strength offset (dB)	Sweep velocity (m/s)	Bandwidth	τ_{SPE}	Difference value $\sqrt{(\tau_{1ms} - \tau_{SEP})^2}$
1	1.5	55	3	-0.4903	0.0008
2	1.5	60	3	-0.4777	0.0035
3	1.5	60	5	-0.5030	0.0013
4	1.5	65	5	-0.4931	0.0004
5	2	60	3	-0.507	0.0002
6	2	65	3	-0.4937	0.0003
7	2	65	5	-0.5217	0.004
8	2	70	3	-0.4797	0.0027
9	2	70	5	-0.5104	0.0008
10	2.5	65	3	-0.5231	0.0022
11	2.5	80	3	-0.4787	0.0028
12	2.5	80	5	-0.5136	0.0011

decrease with the decrease in CI. Thus, the CCEE difference value is due to the reduction in the number of DLL calculations during the spoofing attack process.

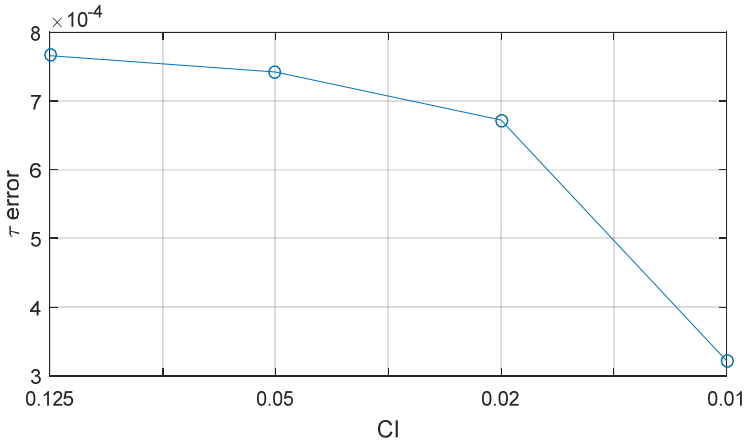


Figure 3-14. CCEE difference value with respect to CI when D is −1.

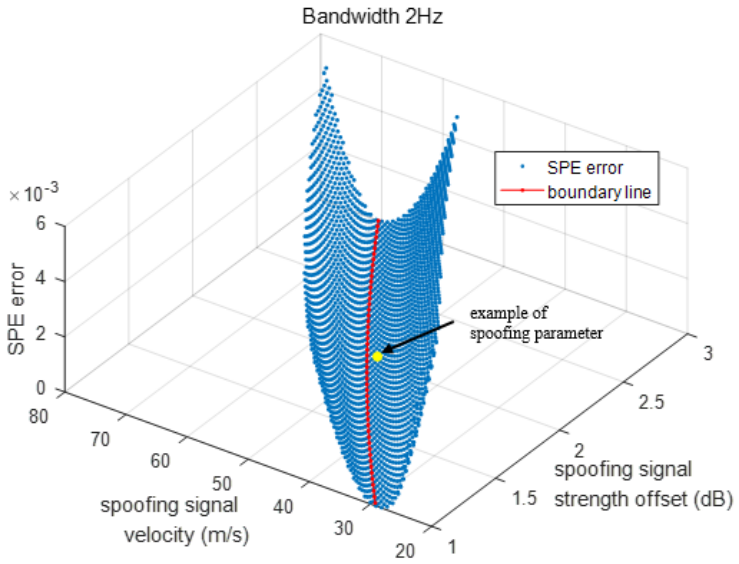


Figure 3-15. CCEE difference value with respect to the spoofing signal strength and velocity in three dimensions.

Figure 3-15 shows the difference value distribution of the CCEE with respect to the spoofing signal strength and sweep velocity on a fixed bandwidth. At the yellow point, the values of the spoofing parameters, namely the signal strength

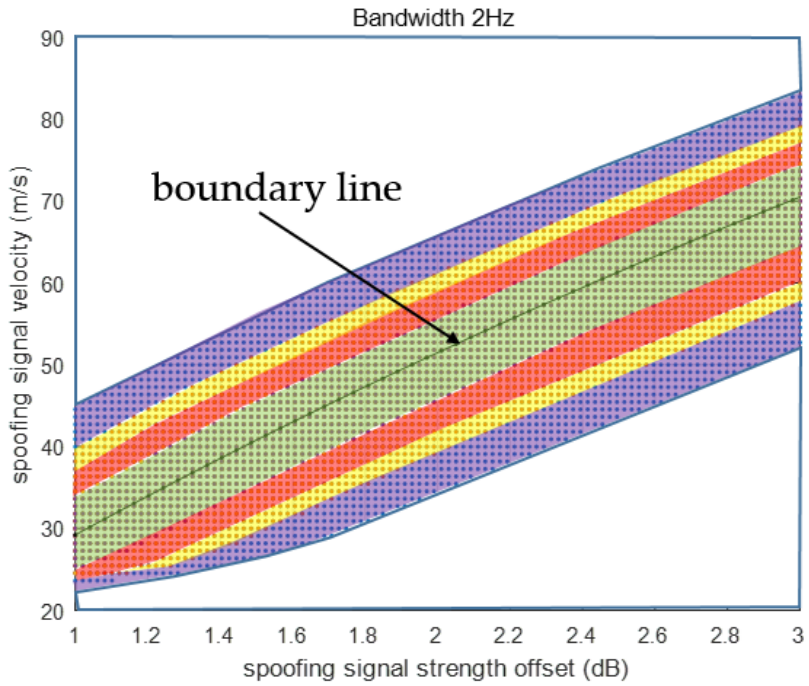


Figure 3-16. CCEE difference value according to spoofing signal strength and velocity.

offset, sweep velocity, and bandwidth, are 2 dB, 50 m/s, and 2 Hz, respectively. When τ_{SPE} is calculated for the above set of spoofing parameter values using the SPE, the error in τ_{SPE} is the Z axis value corresponding to the yellow point. The CCEE performance is the best around the boundary line. The success and failure of the spoofing attack can be divided based on the boundary line. In other words, τ_{SPE} of the spoofing parameters on the boundary line is −0.5. τ_{SPE} error increases as the distance from the boundary value and spoofing parameter increases. A large difference value indicates that the previous τ values are outside the defined range in the τ_{SPE} calculation process. This means that the functions used to calculate XE and XL vary with respect to the already defined

Table 3-6. Estimated $\tilde{a}_s$ values according to the spoofing parameters.

Number	Sweep velocity (m/s)	Bandwidth (Hz)	$\tilde{a}_s$ (dB)
1	40	2	1.46
2	45	2	1.69
3	50	2	1.92
4	55	2	2.17
5	60	2	2.42
6	65	2	2.69
7	70	2	2.97
8	75	2	3.26
9	80	2	3.56
10	85	2	3.87
11	90	2	4.20
12	95	2	4.54
13	100	2	4.09

XE and XL. Figure 3-16 shows the CCEE difference value distribution in two dimensions. The CCEE error is lowest around the boundary line. The spoofing parameters are divided using different colors with respect to the CCEE error size.

3.4.2. Determination of boundary line and surface using SPE

The boundary line and surface that divide the spoofing attack success and failure can be estimated using the SPE. The input parameters of the CCEE are the

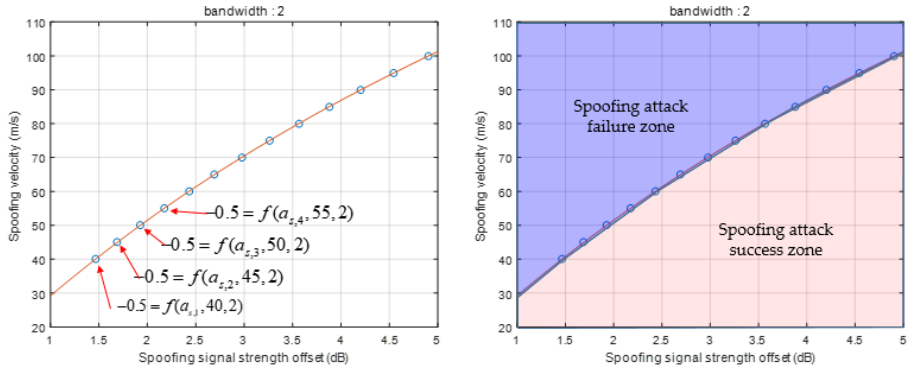
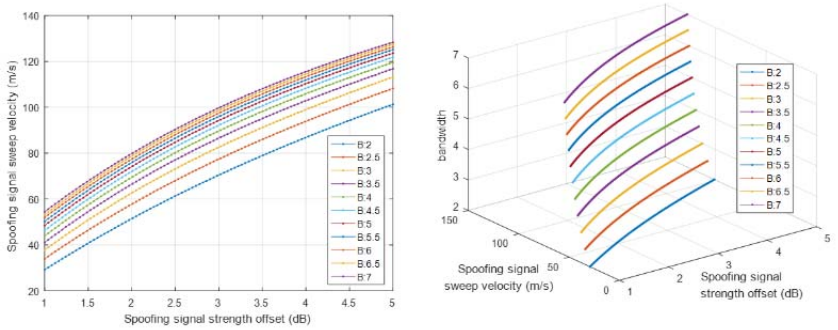


Figure 3-17. (a) a_s estimation using SPE. (b) Determination of spoofing attack success and failure by boundary line.

spoofing signal strength, spoofing signal sweep velocity, and DLL bandwidth. In Equation (3-20), if we set each variable as follows:

$$-0.5 = f(a_s, 40, 2), \tag{3-20}$$

only one variable, i.e., a_s, remains, and the CCEE becomes an equation to calculate a_s. We use MATLAB solver to obtain $\tilde{a}_s$ which is an estimated value of a_s obtained using equation (3-20). This means that the CCEE result of the spoofing parameters, ($\tilde{a}_s$, 40, 2), becomes −0.5. Therefore, the spoofing parameter, ($\tilde{a}_s + \varepsilon$, 40, 2), will result in a successful spoofing attack. The other spoofing parameter, ($\tilde{a}_s - \varepsilon$, 40, 2), will result in a failed spoofing attack. ε is a small positive small. Figure 3-17 shows the boundary line dividing the spoofing attack success and failure zones. We obtain the spoofing signal strength values by fixing the other spoofing parameters and τ . Table 5-6 presents the estimated $\tilde{a}_s$ values with respect to the spoofing parameters. The red line in Figure 3-17 (a)



(a) (b)

Figure 3-18. Boundary lines according to the DLL bandwidth (a) boundary lines in two dimension (b) boundary lines in three dimension.

indicates the boundary line. The boundary can be estimated using the spoofing parameters listed in Table 3-6 as follows:

$$\begin{aligned} V_s &= f_{bl}(a_s) \\ &= p_1 \cdot a_s^3 + p_2 \cdot a_s^2 + p_3 \cdot a_s + p_4 \end{aligned} \tag{3-21}$$

where f_{bl} is the function of the boundary line for a bandwidth of 2 Hz. p_1, p_2, p_3, and p_4 are the coefficients at the boundary line. Moreover, this line expresses the correlation between two parameters for a successful spoofing attack. From Equation (3-21), we find that as the spoofing signal strength increases, the spoofing attack becomes successful even with a higher sweep velocity.

The spoofing attack success and failure zone is divided based on the boundary line, as shown in Figure 3-17 (b). The zone above the boundary line indicates spoofing attack failure, whereas the zone below indicates spoofing attack success. Figure 3-18 (a) shows the boundary lines for various bandwidths. We find that the receiver becomes more vulnerable to spoofing attacks as its bandwidth increases. With the increase in the bandwidth, the spoofing attack becomes successful even

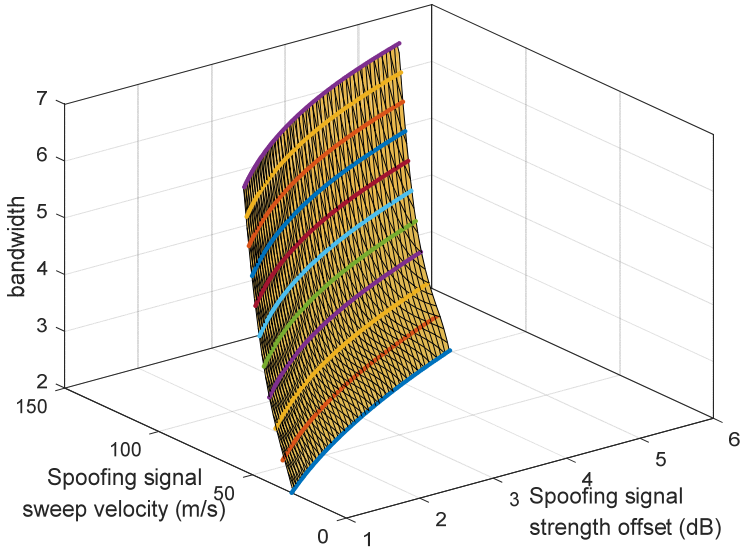


Figure 3-19. Boundary surface.

for a low spoofing signal strength when using a fixed spoofing sweep velocity. Figure 3-18 (b) shows the boundary lines in three dimensions. The boundary surface can be estimated using the boundary lines, as shown in Figure 3-19. The boundary surface can be expressed as follows.

$$\begin{aligned} B &= f_{sf}(a_s, V_s) \\ &= c_1 + c_2 a_s + c_3 V_s + c_4 a_s^2 + c_5 a_s^2 V_s + c_6 a_s V_s^2 + c_7 V_s^3 \\ &+ c_8 a_s^3 V_s + c_9 a_s^2 V_s^2 + c_{10} a_s V_s^3 + c_{11} V_s^4 \end{aligned} \tag{3-22}$$

where f_{sf} indicates the function of the boundary surface. $c_1 \sim c_{11}$ are coefficients of f_{sf}. For specific spoofing parameters, the spoofing attack results

can be determined using f_{sf}. Equation (3-23) is the case for spoofing attack failure, whereas Equation (3-24) is the case for spoofing attack success.

$$B > f_{sf}(a_s, V_s) \tag{3-23}$$

$$B < f_{sf}(a_s, V_s) \tag{3-24}$$

Chapter 4. Optimal sweep direction of covert capture signal

4.1. Maximum Doppler difference value

For GNSS receivers, the code tracking algorithm tracks the code points of a particular PRN. After creating the replica in the receiver, the correlation between the incoming signal and replica signal are calculated. The code starting point of the authentic signal is tracked reliably according to the code tracking algorithm when the signal strength is kept higher than a certain signal strength. In the case of a sweep type spoofing attack, the spoofing signal is shifted in a certain direction, as shown in figure 4-1, from the authentic signal to the spoofing signal, and when the receiver tracks, the user's receiver is completely deceived and the result of PNT controlled according to the spoofing signal.

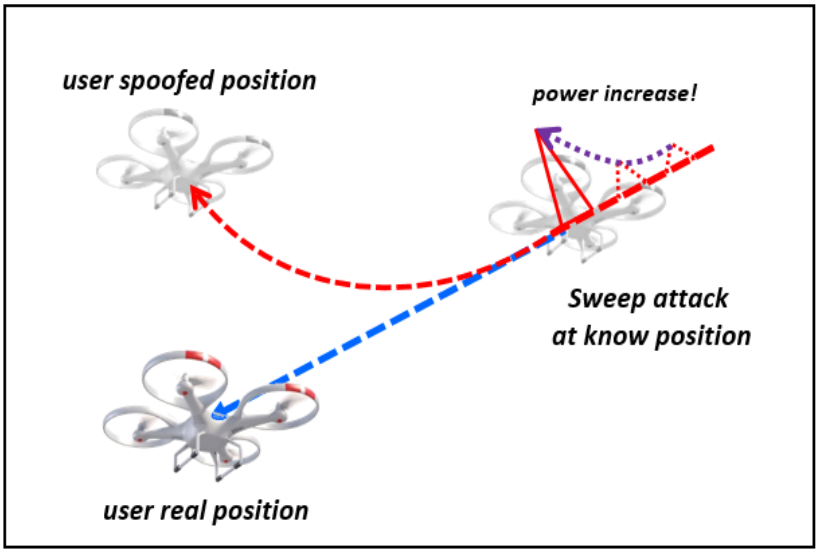


Figure 4-1. Illustration of covert capture process.

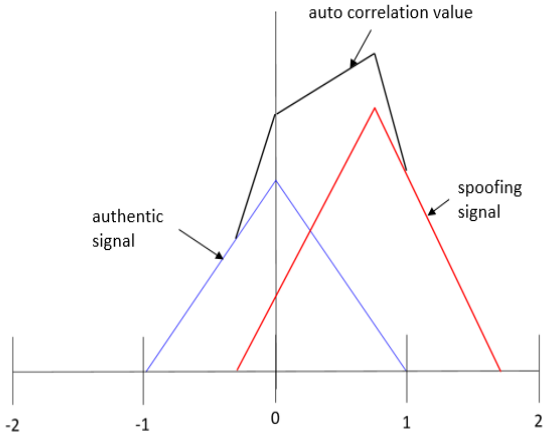


Figure 4-2. Auto correlation function.

In general, the spoofing process is performed on the location domain. In other words, the covert sweep signal generates the spoofing signal for a drone moving in a certain direction, as shown in figure 4-1, causing the spoofing signal to over-position the drone. If a covert sweep attack is normally carried out, the location of the output from the receiver will be presented in a spoofing position that will sweep.

In order for spoofing to be performed to the intended position created by the spoofer, the tracking point on all channels must be shifted from the authentic signal to the spoofing signal. And in order for the spoofing position to be unaffected by the authentic signal, the difference of code start point between the authentic and spoofing signal must be apart at least 2 chips. Figure 4-2 shows the auto correlation function for the authentic signal and the spoofing signal. Red lines are the ACF of spoofing signal, and blue lines indicate the ACF of authentic signal. And black lines shows ACF with total signal. In the figure 4-2, a special form of ACF is formed because each signal overlaps when the code point of the original signal and the deceptive signal are within 2 chip. Figure 4-3 shows when the each

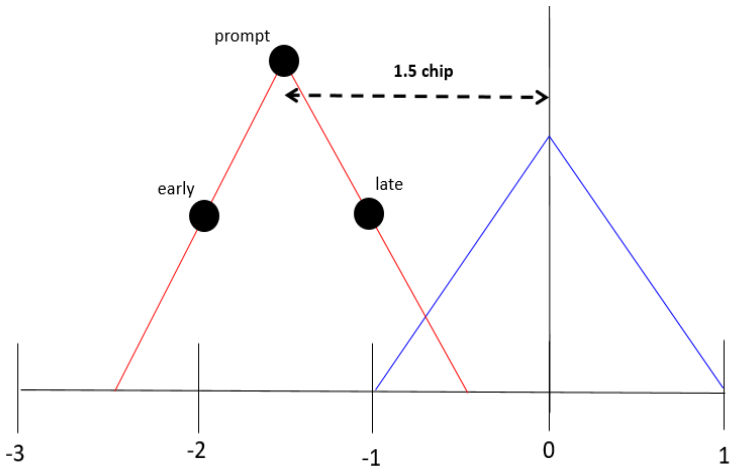


Figure 4-3. 1.5 chip apart of each of code start points.

code start point of authentic and the spoofing signal are 1.5 chip apart. If more than 2 chips of code start points are apart, there will be no overlap between each ACF. Assuming the distance of a chip is 300 meters, the difference in the each of code start points is about 600 meters. And in this case, each ACFs does not affect each other. However, from the receiver's point of view, an early or late point is half-chip delay based on the code start point (prompt), so up to 1.5 chips (450 meters) will not be affected. Figure 4-3 illustrates this situation. In general, assuming that the covert capture signal sweeps the authentic signal in a certain direction, the time that takes place more than 1.5 chips is related to the sweep speed of the spoofing signal. The relative velocity of the authentic signal and the spoofing signal determines the distance between each code start point. If the relative velocity over the code domain is 450m/s, each code start point become apart 1.5 chip in one second.

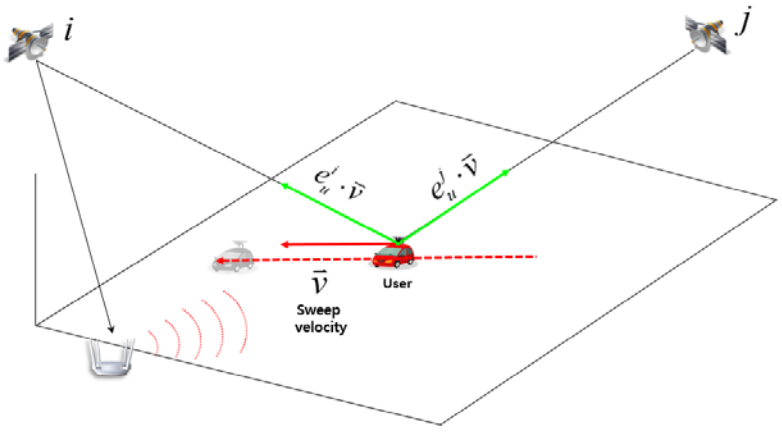


Figure 4-4. Covert capture sweep in position domain.

Figure 4-4 presents the covert capture sweep in position domain. Let's assume that covert capture is conducted with velocity $\vec{v}$ In the position domain. Then, the velocity in code domain becomes $e_u^i \cdot \vec{v}$. In other words, the speed in the code domain is the speed in the position domain multiplied by the line of sight vector in the direction of the satellite. In the same way, the code domain sweep speed in the jth satellite becomes $e_u^j \cdot \vec{v}$. And this can also be represented by the Doppler difference value between the authentic and the spoofing signals. Below equation (4-1) describes this. The Doppler difference value of each signal is defined as follows, and the greater the relative velocity, the greater the Doppler difference value between the two signals. This means how different the Doppler value of the deceptive signal is based on the authentic signal. In other words, the larger the Doppler difference error value, the less time the code points of the two signals are apart in code domain while covert capture signal sweeps.

$$
\begin{aligned}
\Delta f_u^i &= e_u^i \cdot (\frac{\Delta v_{i-u}}{c} \cdot f_0) \\
\Delta f_s^i &= e_s^i \cdot (\frac{\Delta v_{i-s}}{c} \cdot f_0) \\
\Delta f_u^i - \Delta f_s^i &= e_u^i \cdot (\frac{\Delta v_{i-u}}{c} \cdot f_0) - e_s^i \cdot (\frac{\Delta v_{i-s}}{c} \cdot f_0) \\
&= e_u^i \cdot (\frac{\Delta v_i}{c} \cdot f_0 - \frac{\Delta v_i}{c} \cdot f_0) - e_u^i \cdot (\frac{\Delta v_u}{c} \cdot f_0 - \frac{\Delta v_s}{c} \cdot f_0) \\
&= e_u^i \cdot \frac{\Delta v_s}{c} \cdot f_0 \\
&= e_u^i \cdot \frac{\Delta v_s}{\lambda}
\end{aligned}
\tag{4-1}
$$

4.2. Optimal covert capture direction in 2D case

As previously explained, in order to deceive the user's PNT solution, spoofing must be carried out on all visible satellites. In other words, for all channels, the tracking point should be shifted from the authentic signal to the spoofing signal.

In the covert capture process, channel-specific Doppler difference value is determined depending on the sweep direction of covert capture and the geometrical position of the satellite. Let's define V_c as sweep speed on a channel, V_c is calculated as shown in the following formula below with respect to the authentic and the spoofing signal.

$$
\begin{aligned}
V_C &= \overline{e^i} \cdot (\vec{V}^i - \vec{V}_u) - \overline{e^i} \cdot (\vec{V}^i - \vec{V}_s) \\
&= \overline{e^i} \cdot (\vec{V}_s - \vec{V}_u) \\
&= \overline{e^i} \cdot \vec{G}
\end{aligned}
\tag{4-2}
$$

Where $\overline{V^i}$ indicate the velocity of ith satellite, e^i is line of sight vector. And $\overline{G}$ is defined by (11). Finally, the time it takes for the 1.5 chip difference of the code start point between two signals to be determined by V_c. And it takes less time when the dot product of $\overline{G}$ and $\overline{e}$ is small.

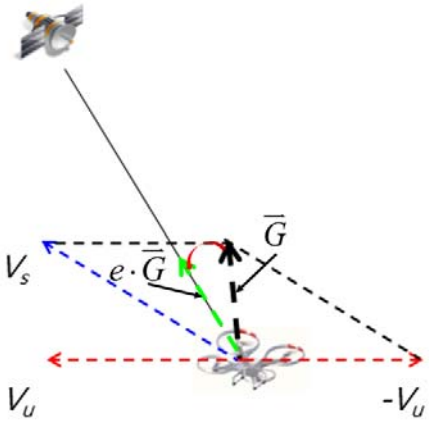


Figure 4-5. Determination of Speed on Code Domain according to User Speed and Covert Capture Direction.

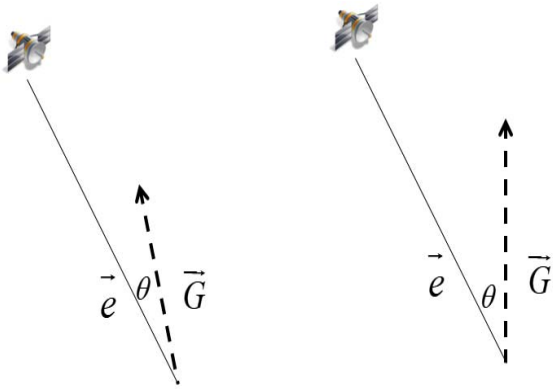


Figure 4-6. Relationship between $\overline{e}$ and $\overline{G}$ direction and the Doppler difference value.

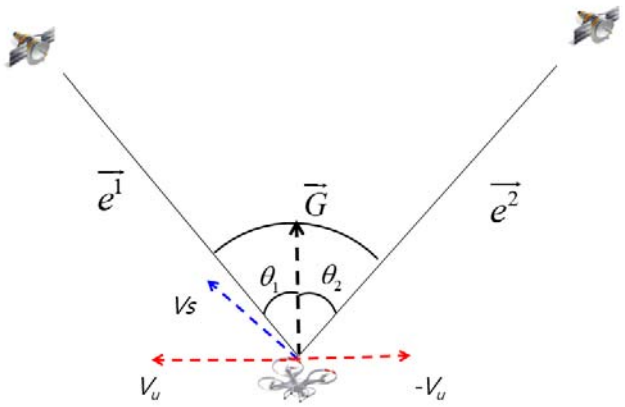


Figure 4-7. Finding the optimal covert capture direction for two satellites.

Since the covert capture must be performed on all channels, the sweep speed at the lowest channel determine the process time of covert capture. Figure 4-5 shows that for a single satellite, the sweep speed is determined in the code domain according to the $\overline{e}$ and $\overline{G}$ direction. In the left picture of figure 4-6, the sweep speed is faster in the code domain than in the right picture. This is because the direction of $\overline{G}$ is more similar to that of $\overline{e}$. Let's assume that

$$\overline{g} = \frac{\vec{G}}{|\overrightarrow{G}|} \tag{4-3}$$

And the dot product between $\overline{e}$ and $\overline{g}$ is expressed as

$$\begin{aligned} \vec{e} \cdot \left|\overline{g}\right| &= \cos(\theta)\left|\vec{e}\right|\left|\overline{g}\right| \\ &= \cos(\theta) \end{aligned} \tag{4-4}$$

That is, the smaller the angle between the two vectors, the faster the speed is in the code domain. In the extreme case, if the direction of the satellite's line of

sight vector is the same as that of $\overline{G}$, the θ becomes zero, and the norm of $\overline{G}$ itself becomes the speed in the codomain.

In a 2D situation, the situation becomes more complicated when there are more than two satellites. As already mentioned, in order for spoofing to be performed normally, it must be successfully carried out on all satellites. In other words, the code domain speed should be fast on all channels. Physically, however, it is impossible to find with maximum code sweep speed for satellites with maximum θ.

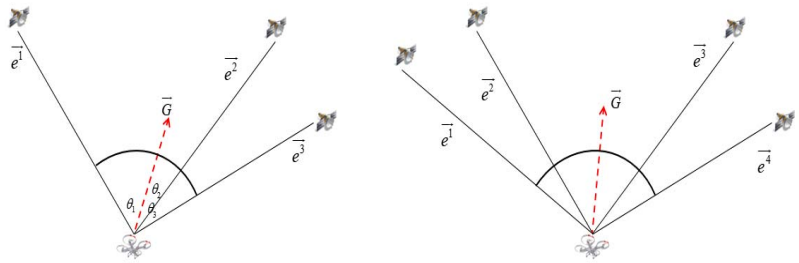


Figure 4-8. Finding the optimal covert capture direction for more than three satellites.

In the 2D situation, finding the optimal covert capture direction for two satellites is shown in Figure 4-8. As mentioned above, the direction of $\overline{G}$ should be set to perform the fastest covert capture on all satellites. In other words, the problem is to first find the satellite having the maximum angle between the $\overline{G}$ and the $\overline{e}$, and then determine the $\overline{G}$ that maximizes the dot product of the $\overline{e}$ and $\overline{g}$. For example, assuming that there are *n* visible satellites, and the spoofing is carried out on the n-1 satellites quickly. But if the spoofing for remaining satellite is not done, then the whole spoofing is not done. The duration of spoofing is determined by the satellite that the lock point changes last. That is,

it is important to finding a direction that minimizes the time for satellites where spoofing is conducted at the latest. If there are two satellites, the optimized direction is middle direction of the two line of sight vectors. In other words, if the $\bar{g}$ direction is directed to a specific satellite, the overall covert capture time will increase as the angle to the opposite satellite increases. The blow equation shows to estimate the $\bar{g}$ in case of two satellites.

$$
\begin{aligned}
&\vec{G} = \vec{V}_s - \vec{V}_u \\
&\vec{g} = \frac{\vec{G}}{\left|\vec{G}\right|} \\
&\vec{e^i} \cdot \vec{g} = \vec{e^j} \cdot \vec{g} \\
&\begin{bmatrix} \vec{e^i}^T \\ \vec{e^j}^T \end{bmatrix} \cdot \vec{g} = \begin{bmatrix} k \\ k \end{bmatrix} \\
&\vec{g} = \begin{bmatrix} \vec{e^i}^T \\ \vec{e^j}^T \end{bmatrix}^{-1} \begin{bmatrix} k \\ k \end{bmatrix} \\
&\vec{g} = \begin{bmatrix} \dfrac{2k \cdot (e^i_{1,1} + e^i_{1,2})}{\sqrt{(2k \cdot (e^i_{1,1} + e^i_{1,2}) + 2k \cdot (e^j_{2,1} + e^j_{2,2}))^2}} \\ \dfrac{2k \cdot (e^j_{2,1} + e^j_{2,2})}{\sqrt{(2k \cdot (e^i_{1,1} + e^i_{1,2}) + 2k \cdot (e^j_{2,1} + e^j_{2,2}))^2}} \end{bmatrix}
\end{aligned}
\tag{4-5}
$$

For three satellites, two low elevation angle satellites are selected and the center direction is optimal. Even, for satellites larger than fore, the same approach is applied. Since only visible satellites should be considered, the covert capture direction is determined by selecting the two satellites with lowest. Assuming that the left is west and the right is east in the 2D situation, the lowest angle satellite in the left-hand direction is assumed to be ith satellite, and the lowest angle in the

right-hand side is the jth satellite. In this case, $\overline{G}$, $\vec{g}$ and $\overrightarrow{g'}$ are defined as below equation, respectively. Since the inner product value of $\vec{g}$ and $\overline{e^i}$ is the same as that of $\vec{g}$, $\overline{e^j}$ and $\overrightarrow{g'}$ could be calculated using the corresponding equation.

$$\overrightarrow{g'} = \begin{bmatrix} \overrightarrow{e^1}^T \\ \overrightarrow{e^2}^T \\ \overrightarrow{e^3}^T \end{bmatrix}^{-1} \begin{bmatrix} k \\ k \\ k \end{bmatrix} \tag{4-6}$$

If the $\overline{G}$ is determined, the optimal covert capture direction in the position domain is calculated as follows

$$\begin{aligned} \vec{G} &= \overrightarrow{V_s} - \overrightarrow{V_u} \\ \overrightarrow{V_s} &= \vec{G} + \overrightarrow{V_u} \end{aligned} \tag{4-7}$$

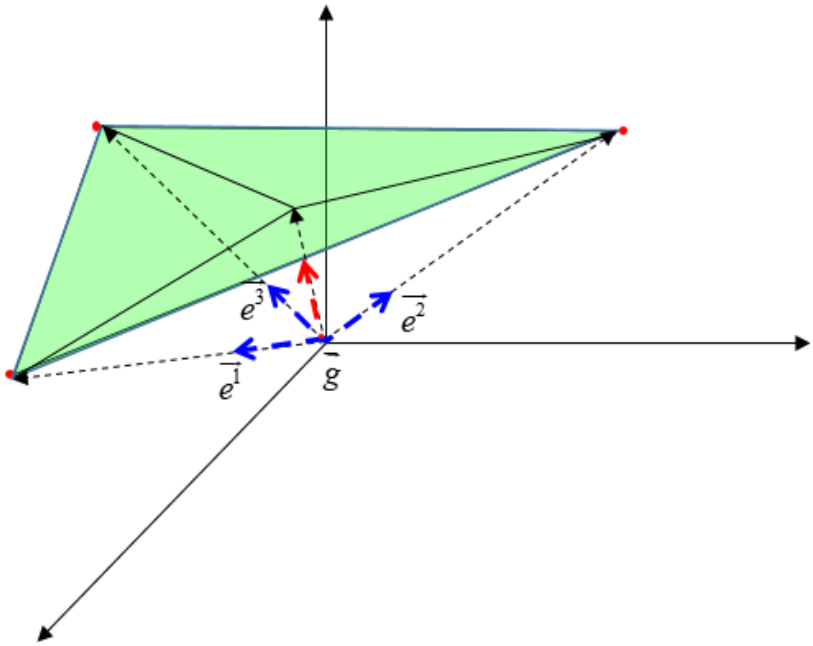


Figure 4-9. Optimal direction in 3D case.

4.3. Optimal covert capture direction in 3D case

Since the actual covert capture sweep is done in the 3D environment, this chapter discusses the optimal direction in 3D situation. Figure 4-9 shows the optimal direction in 3D situation. Unlike the 2D situation, the 3D situation requires all directions to be considered, but the issues given are the same. That is, for every satellite, it is the problem of finding the direction in which it is the largest $\vec{g} \cdot \vec{e}$ value for certain satellites which has largest angle between its line of sight vector and $\vec{g}$. In 2D case, at first, the satellites having lowest elevation angle with both direction was selected to find optimal covert capture direction. In 3D case, we select three satellites with the lowest elevation angles to determine the optimal covert capture direction. The order of determining the optimal covert capture

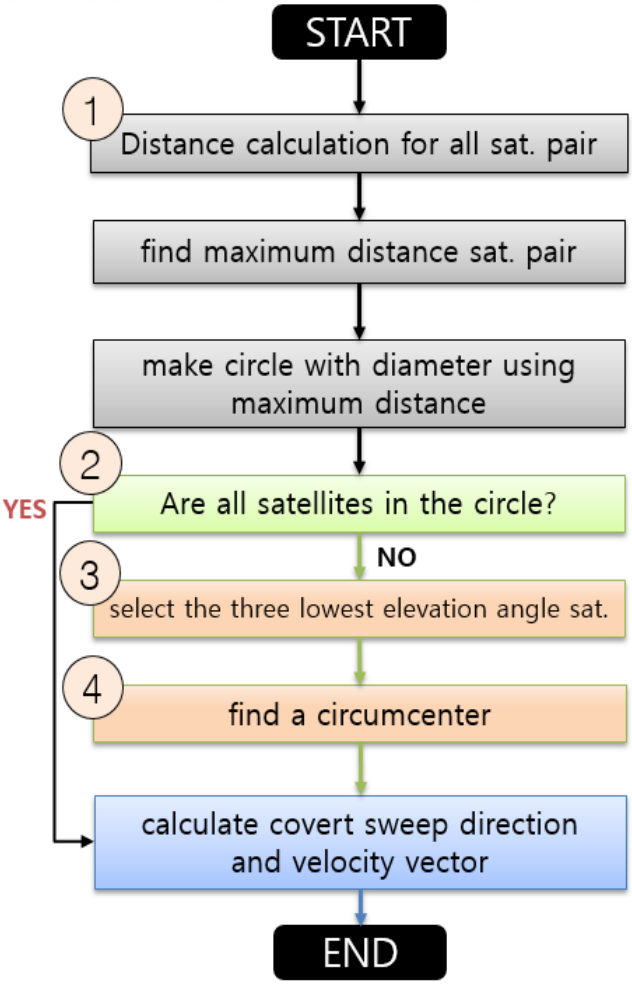


Figure 4-10. Flow chart for optimal covert capture direction in 3D case.

direction in 3D case is as follows in the flow chart below. First, for all satellites, calculate the distance between the two satellites. Then we draw a circle with the diameter of the line, and check that all visible satellites enter the circle. If not, we select the lowest elevation angle satellites in remaining satellites. Then we define a triangle with three satellites as vertices. Then determine the circumcenter of the triangle. Figure 4-9 shows the optimal direction in 3D case. The dot product

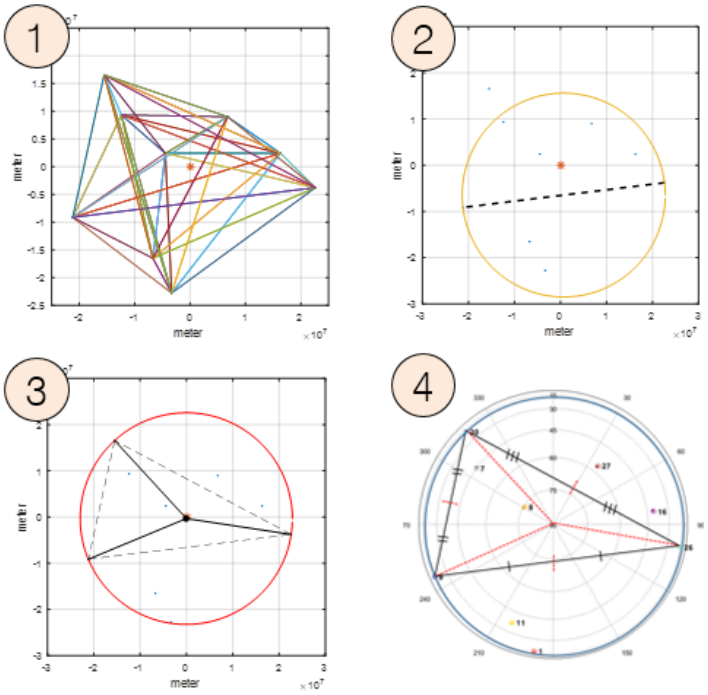


Figure 4-11 Process for optimal capture direction in 3D case.

value between the circumcenter and line of sight vector direction of the three satellites will be the same. That is, in the case of 2D, the optimal direction is the middle direction of two selected satellites, in 3D case, it is in the circumcenter direction when the selected three satellites were triangular. And the each dot product value of $\overline{g}$ and $\overline{e}$ become same in 2D and 3D cases. Figure 4-10 shows flow chart for optimal covert capture direction in 3D case and figure 4-11 present the each block numbering in the figure 4-10.

4.4. Optimal covert capture direction using optimization method

$$\max_{\vec{g}\in R^3} J(\vec{g}) = \arg\min_i (\bar{e}\cdot\vec{g})$$
$$J(\vec{g}) = J(\vec{g}) - k\cdot\frac{\partial J}{\partial g} \tag{4-8}$$

Equation 4-8 shows the optimization equation for optimal covert capture direction. The purpose of this optimization equation is to find the $\bar{g}$ direction producing the largest dot product of $\bar{g}$ and $\bar{e^i}$, and this dot product is the smallest value among all dot products. The simulation results of optimization approach in cases of 2D and 3D are almost similar with the results of 4.3 and 4.4 captures. Accordingly, it could be seen that the analytically obtained directions and the directions obtained by the optimization equation are almost the same.

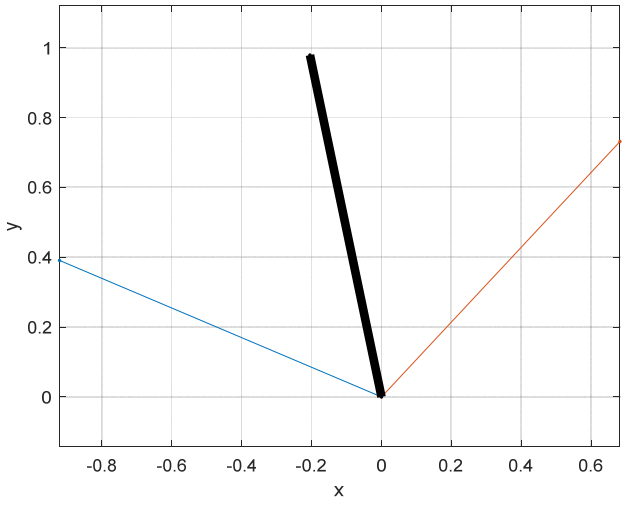


Figure 4-12 2D optimal direction results using optimization approach.

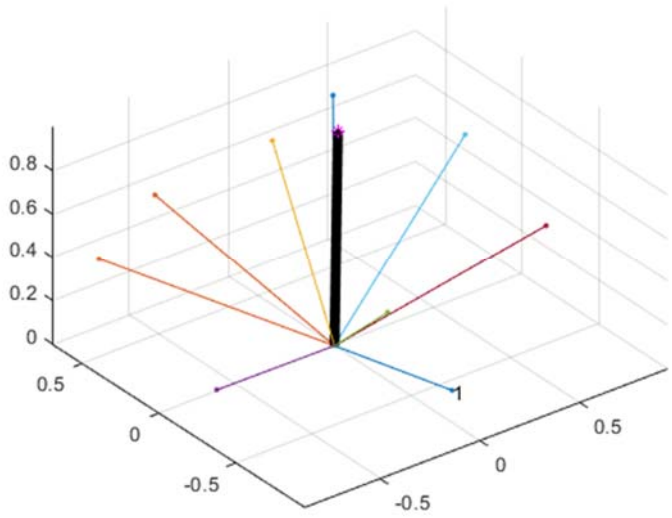


Figure 4-13 3D optimal direction results using optimization approach.

Chapter 5. Covert capture simulation using software defined receiver

5.1. Implementation of GNSS measurement and IF data generation simulator

5.1.1. Pseudorange model

In the covert capture scenarios, the position and velocity for authentic and spoofing signals are determined. Then, GNSS measurements are calculated such as the position and velocity of visible satellites, distance, Doppler, and SNR values. The simulator will generate C/A code, navigation data, carrier and signal noise based on these values. Eventually, all of the above values will be combined and the IF signal data will be generated. Then, the receiver conduct the signal processing using the output from IF signal generator and the results from receiver will become the same with measurement made in advance. Figure 5-1 shows a

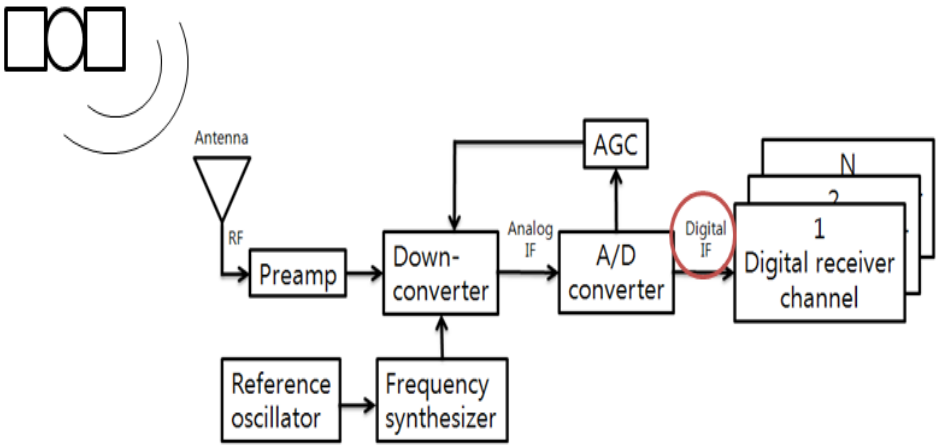


Figure 5-1. Block diagram of receiver

block diagram of a typical GPS receiver. Figure 2-7 shows a block diagram of a typical GPS receiver. Satellite signals transmitted at high altitudes of 22,000 kilometers above the ground reach the receiver's antenna through the atmosphere. The GPS signal received from the antenna has a weak power of about -158 dB, so it is amplified by pre amp. It is also lowered to IF frequency unit through down conversion to facilitate processing of signal data, and is quantified through A/D converter. IF signal data after going through the A/D converter. Therefore, a signal model for quantified IF signals is required, which is as presented in (5-1).

$$S_{IF} = \sum_{i=1}^{N} \{\sqrt{2P_r} D_i(t-r) C_i(t-r) \cos(2\pi(f_{IF} + f_{d,i})t - 2\pi(f_{IF} + f_{d,i})\tau + \psi_0)\} + \eta \quad (5\text{-}1)$$

P_r : *Signal strength in antenna*
D_i : *Navigation message from the ith satellite*
C_i : *C / A code for the ith satellite*
f_{L1} : *L1 frequency* $(1575.42\ MHz)$
$f_{d,1}$: *Doppler of ith satellite*
N : *the number of visible satellite*
ψ_0 : *initial carrier phase*
η : *signal noise*

5.1.2. Simulator structure

By input scenario from scenario generation program, the visual satellite is determined using ephemeris, and the location and speed of the satellite are calculated. And GPS time is also defined to calculate the measurements of distance and Doppler. In our case, the measurement is created in a period of 1 ms.

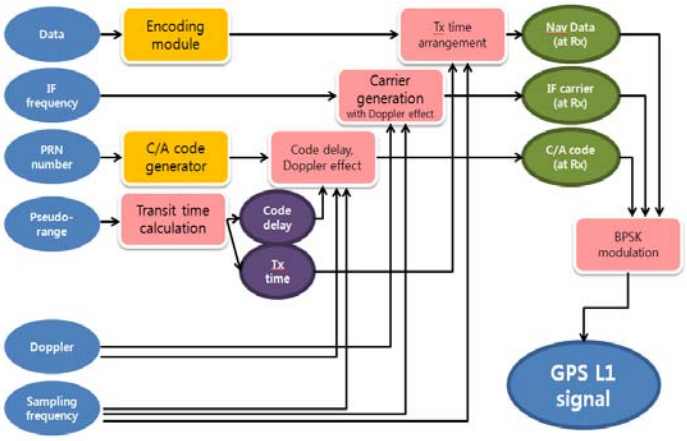


Figure 5-2. Simulator signal generator structure

The figure 5-2 shows the structure of producing a signal about the one satellite. The PRN is determined by the visible satellites. The transmit time is calculate using pseudorange. After the carrier is created, multiply the navigation and the C/A code. Finally, each satellite signals are summed to make final GPS L1 signal.

5.1.3. Signal amplitude calculation in spoofing scenario

In the presence of multiple GPS satellites, the spoofing signal is transmitted by the spoofer, which is the same as if there are multiple GPS satellites. In other words, there will be two different codes for the same PRN. And it can be thought of as two independent satellites with the same PRN. From the above assumptions, the signal strength for a particular satellite can be defined as follows.

$$C/N_{0,i} = \frac{P_i}{(\sum_{j=1, j\neq i}^{n_{sv}} N_{i,j} + \sum_{k=1}^{n_{sv\_spoofed}} N_{i,\mathrm{k}} + N)\cdot T_{\mathrm{int}}} \tag{5-2}$$

$C/N_{0,i}$: C/N_0 *of ith satellite*

P_i : *carrier power of ith satellite*

N : *noise power*

$N_{i,j}$: *noise power due to jth satellite*

n_{sv} : *number of visible satellite*

If the strength of the spoofing signal is greater than that of the GPS signal, the cross correlation value of the GPS signal and the spoofing signal will be greater than the cross correction value of the GPS signal. However, since the difference is small compared to noise power, we assumed that it is the same and the signal strength is implemented in the simulator. And the $C/N_{0,i}$ could be defined as follows

$$\begin{aligned} C/N_{0,i} &= \frac{\frac{n_{sp}^2 A_i^2}{4}}{\{\sum_{j=1, j\neq i}^{n_{sv}} \frac{n_{sp}^2 A_j^2}{4}\{\mathrm{E}\left[C_i(t)C_j(t)\right]\}^2 + \sum_{k=1}^{n_{sv\_spoofing}} \frac{n_{sp}^2 A_k^2}{4}\{\mathrm{E}\left[C_i(t)C_k(t)\right]\}^2 + n_{sp}\,\mathrm{var}(N)\}\cdot T_{\mathrm{int}}} \\ &= \frac{\frac{f_{sp}\cdot T_{\mathrm{int}}\cdot A_i^2}{4}}{\{\sum_{j=1, j\neq i}^{n_{sv}} \frac{f_{sp}\cdot T_{\mathrm{int}}}{4} A_j^2\{\mathrm{E}\left[C_i(t)C_j(t)\right]\}^2 + \sum_{k=1}^{n_{sv\_spoofing}} \frac{f_{sp}\cdot T_{\mathrm{int}}}{4} A_k^2\{\mathrm{E}\left[C_i(t)C_k(t)\right]\}^2 + n_{sp}\,\mathrm{var}(N)\}\cdot T_{\mathrm{int}}} \end{aligned} \tag{5-3}$$

And to calculate the amplitude of *ith* satellite, the amplitude is defined as follows

$$A_i^2 = T_{\text{int}} \cdot \sum_{j=1, j\neq i}^{n_{sv}} C/N_{0,i} \cdot A_j^{\,2} \{\mathrm{E}\left[C_i(t)C_j(t)\right]\}^2$$
$$+T_{\text{int}} \cdot \sum_{k=1}^{n_{sv\_spoofing}} C/N_{0,i} \cdot A_k^{\,2} \{\mathrm{E}\left[C_i(t)C_k(t)\right]\}^2 + \frac{4}{f_{sp}} \cdot \mathrm{var}(N) \cdot T_{\text{int}} \quad (5\text{-}4)$$

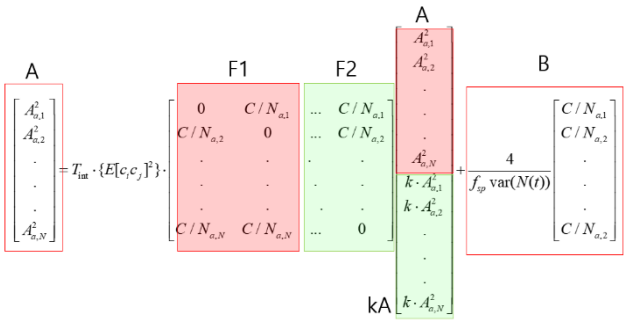
In CCEE, the ratios of the original signal to the spoofing signals are calculated. That is, the ratio of the spoofing signal is calculated with the signal strength of the authentic signal normalized to 1. For GNSS signal simulators, since the signal strength of C/N0 is used as input, it is necessary to change the value of the ratio to the actual signal strength.

$$\begin{bmatrix} A_{a,1}^2 \\ A_{a,2}^2 \\ . \\ . \\ . \\ A_{a,N}^2 \\ A_{s,1}^2 \\ A_{s,2}^2 \\ . \\ . \\ . \\ A_{s,N}^2 \end{bmatrix} = T_{\text{int}} \cdot \{E[c_i c_j]^2\} \cdot \begin{bmatrix} 0 & C/N_{a,1} & ... & C/N_{a,1} \\ C/N_{a,2} & 0 & ... & C/N_{a,2} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ C/N_{a,N} & C/N_{a,N} & ... & C/N_{a,N} \\ C/N_{s,1} & C/N_{s,1} & ... & C/N_{s,1} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ C/N_{s,N} & C/N_{s,N} & ... & 0 \end{bmatrix} \begin{bmatrix} A_{a,1}^2 \\ A_{a,2}^2 \\ . \\ . \\ . \\ A_{a,N}^2 \\ A_{s,1}^2 \\ A_{s,2}^2 \\ . \\ . \\ . \\ A_{s,N}^2 \end{bmatrix} + \frac{4}{f_{sp}\,\mathrm{var}(N(t))} \begin{bmatrix} C/N_{a,1} \\ C/N_{a,2} \\ . \\ . \\ . \\ C/N_{a,N} \\ C/N_{s,1} \\ C/N_{s,2} \\ . \\ . \\ . \\ C/N_{s,N} \end{bmatrix} \quad (5\text{-}5)$$

In the above formula, A_a and A_s indicate the signal strength of the authentic and that of spoofing signal, respectively. Because the ratio of the signal ratio of the authentic signal and the spoofing signal can be calculated through CCEE, it can be changed as follows.

$$\begin{bmatrix} A_{a,1}^2 \\ A_{a,2}^2 \\ . \\ . \\ . \\ A_{a,N}^2 \\ k_1 \cdot A_{a,1}^2 \\ k_2 \cdot A_{a,2}^2 \\ . \\ . \\ . \\ k_N \cdot A_{a,N}^2 \end{bmatrix} = T_{\text{int}} \cdot \{E[c_i c_j]^2\} \cdot \begin{bmatrix} 0 & C/N_{a,1} & ... & C/N_{a,1} \\ C/N_{a,2} & 0 & ... & C/N_{a,2} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ C/N_{a,N} & C/N_{a,N} & ... & C/N_{a,N} \\ C/N_{s,1} & C/N_{s,1} & ... & C/N_{s,1} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ C/N_{s,N} & C/N_{s,N} & ... & 0 \end{bmatrix} \begin{bmatrix} A_{a,1}^2 \\ A_{a,2}^2 \\ . \\ . \\ . \\ A_{a,N}^2 \\ k_1 \cdot A_{a,1}^2 \\ k_2 \cdot A_{a,2}^2 \\ . \\ . \\ . \\ k_N \cdot A_{a,N}^2 \end{bmatrix} + \frac{4}{f_{sp}\ \text{var}(N(t))} \begin{bmatrix} C/N_{a,1} \\ C/N_{a,2} \\ . \\ . \\ . \\ C/N_{a,N} \\ C/N_{s,1} \\ C/N_{s,2} \\ . \\ . \\ . \\ C/N_{s,N} \end{bmatrix} \tag{5-6}$$

where *K* indicates the ratio of *ith* satellites. And this matrix equation could be changed like below matrix.



(5-7)

And this could be expressed as follows

$$A = F_1 \cdot A + F_2 \cdot K \cdot A + B \tag{5-8}$$

$$\begin{aligned} A &= F_1 A + F_2 K A + B \\ A - F_1 A - F_2 K A &= B \\ (I - F_1 - F_2 K) A &= B \\ A &= (I - F_1 - F_2 K)^{-1} \cdot B \end{aligned} \tag{5-9}$$

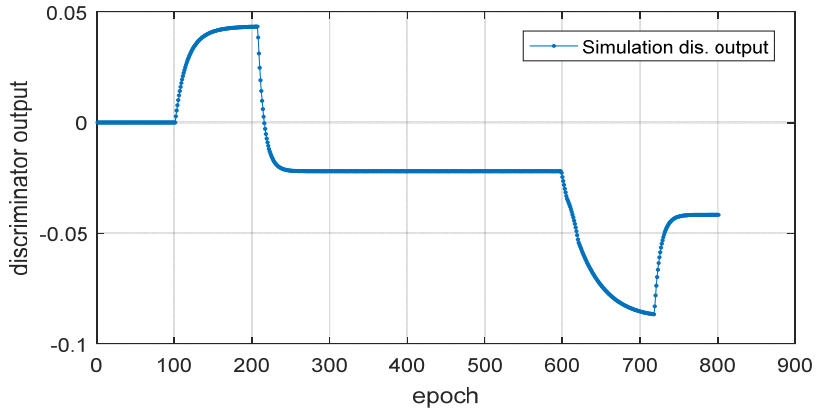


Figure 5-3. Discriminator output in CCEE

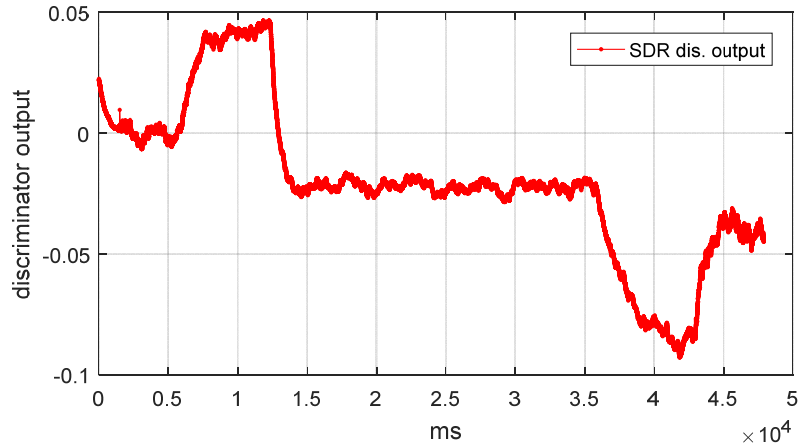


Figure 5-4. Discriminator output in SDR

Finally, the amplitudes of the authentic signals could be obtained through the above equation. And, for the amplitudes of spoofing signals, we can obtain them by multiplying the authentic signal by *K*.

$$A_s = K \cdot A \tag{5-10}$$

$$K = \begin{bmatrix} k_1 & & 0 \\ & \ddots & \\ & & \ddots \\ 0 & & k_N \end{bmatrix} \tag{5-11}$$

Figure 5-3 shows the discriminator output in CCEE. And Figure 5-4 shows the discriminator output in SDR. We can identify that the discriminator values in two case is almost same. In other ward, the early and late point values in two case is also identical. Thus, the signal strength ratio of the authentic and spoofing signal in the CCEE simulation is well implemented in SDR simulation. Figure 5-5 shows the ACF using in the CCEE simulation. And Figure 5-6 shows the ACF in SDR. The ratio (P_a and P_s) of the two signals to be calculated in the SDR was then found to be approximately *1.5*. Thus, the signal strength of each signal was calculated by CCEE and the corresponding results were correctly reflected in the SDR.

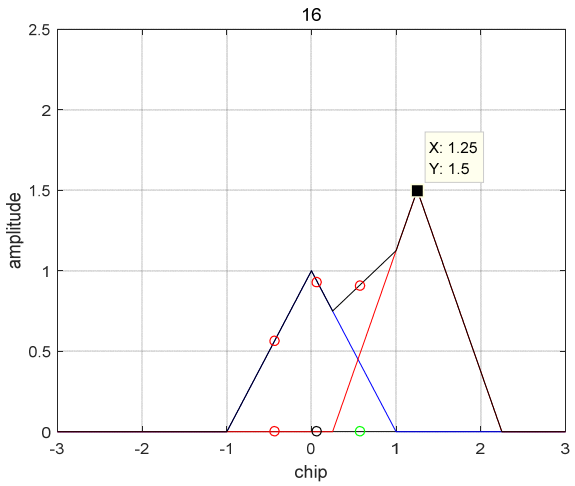


Figure 5-5. ACF in CCEE

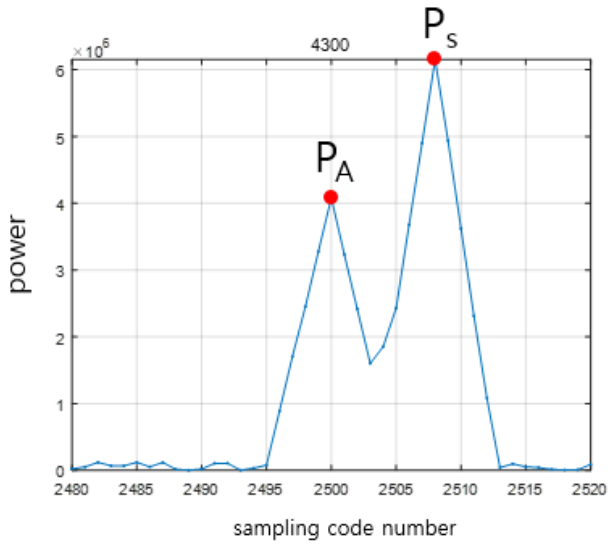


Figure 5-6. ACF in SDR

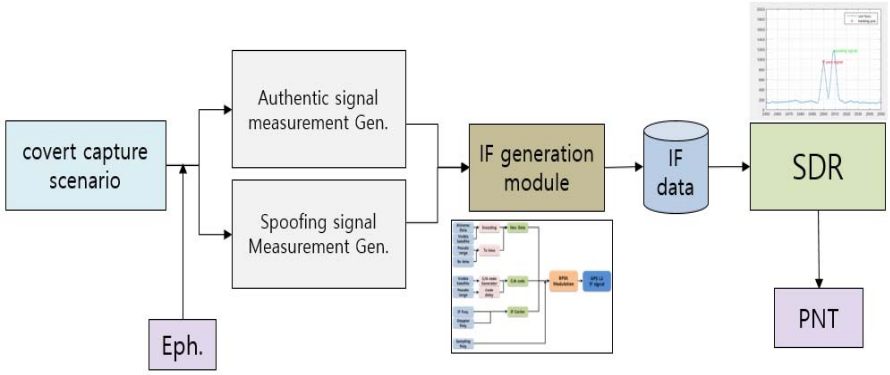


Figure 5-7. Simulator signal generator structure

5.2. CCEE simulation in SDR

Using the optional signal strength obtained by CCEE, the covert capture Scenarios was constructed and verified through SDR. Figure 5-7 shows the overall simulation configuration. At first, it creates a covert capture scenario, then

generates the measurements, and then create an IF file. The SDR then generates

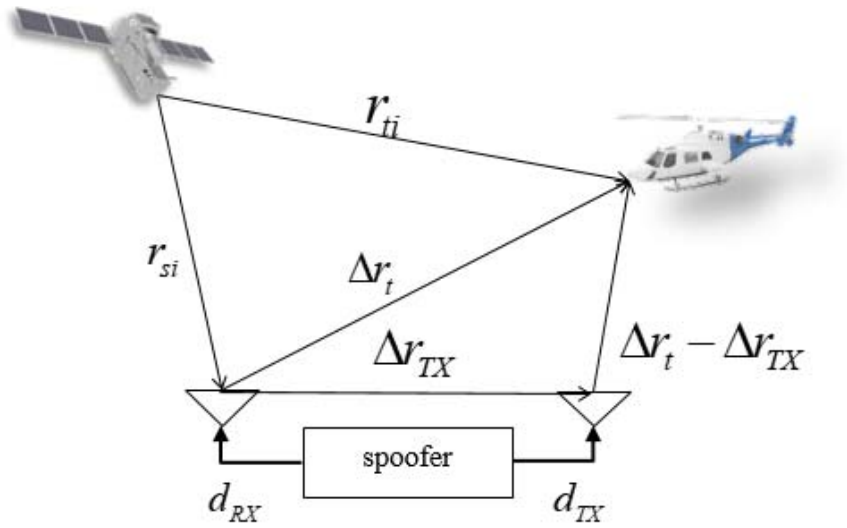


Figure 5-8. Consideration factors for aligned covert capture signal

the final PNT solution by performing signal processing using the IF data generated. In this case, the result of covert capture is analyzed by comparing the SDR result with the measured value. In the SDR simulation, the following assumptions were made to simulate the covert capture sweep scenario.

- Estimate and compensate the calculated delay.
- Random delay could be compensated through compensation.
- The attenuation generated by the transmission of the spoofing signal can be compensated through the signal attenuation model.

$$r_{ti} = r_{si} + c(d_{RX} + d_{TX} + d_s) + \left\| \Delta r_t - \Delta r_{TX} \right\| + \zeta$$
$$\zeta = r_{si} + c(d_{RX} + d_{TX} + d_s) + \left\| \Delta r_t - \Delta r_{TX} \right\| - r_{ti} \quad (5\text{-}12)$$

r_{ti} : *distance between satellite and spoofer*
r_{si} : *distance beetween satellite and victim*
d_{RX} : cable *bias between reception antenna and spoofer*
d_{TX} : cable *bias between transmission antenna and spoofer*
d_s : *processing delay*
$\left\| \Delta r_t - \Delta r_{TX} \right\|$: *distance between spoofer and victim*
ζ : *total compensation delay*

ζ indicates the final compensation delay. If compensation is carried out for all delays, the same code start point with authentic signal can be generated using covert capture signal.

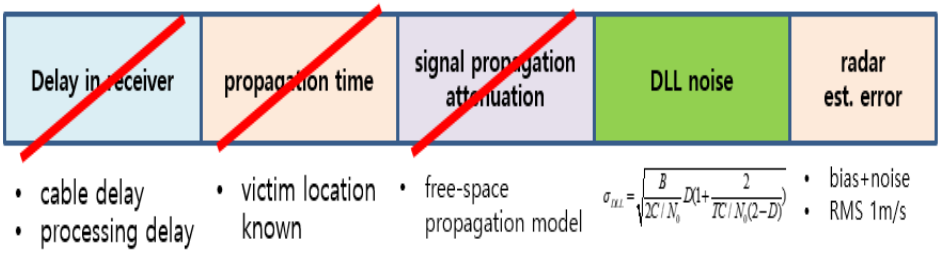


Figure 5-9. Consideration factors for delay and error

Figure 5-9 shows the delays and error terms that should be considered in real-life situations. Calculated delays such as cable or propagation delay are calculated and compensated in ζ, and non-calculated noise is compensated using an expected maximum error value. The purpose of covert capture is to perform on capture in all channels.

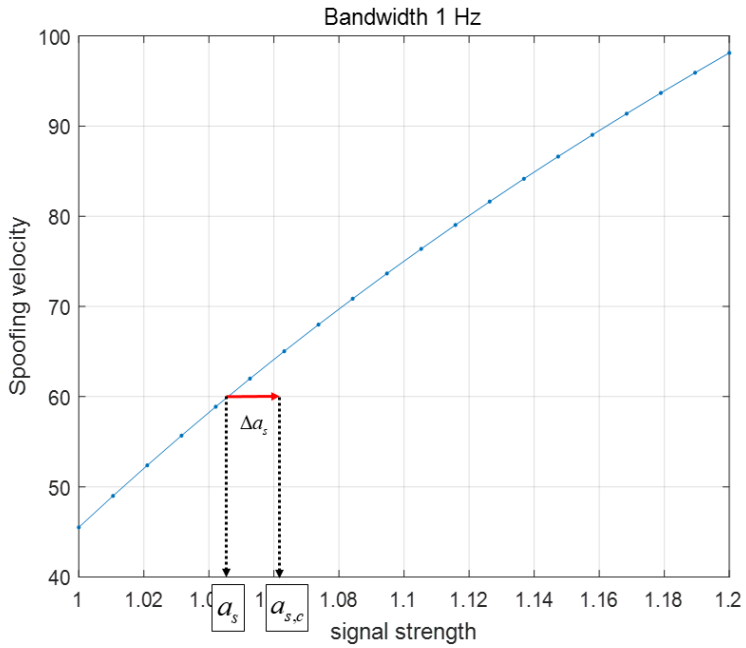


Figure 5-10. Calculation of compensated value using boundary line.

5.2.1. Compensation value calculation for covert capture

In calculating compensation values, this section presents a method for calculating error with random characteristics. The error terms to be considered are DLL noise and radar estimation errors. By using the boundary line estimated by CCEE the compensated value could be calculated.

$$\begin{aligned} -0.5 &= f_{CCEE}(a_s, V_s, B) \\ -0.5 &= f_{CCEE}(a_{s,c}, V_s + 3\sigma_{V_s}, B) \\ \Delta a_s &= a_{s,c} - a_s \end{aligned} \tag{5-13}$$

Let's assume that V_s is the velocity of victim. And let's $3\sigma_{V_s}$ is a radar speed errors. σ_{V_s} indicates the standard deviation of radar estimation error. By

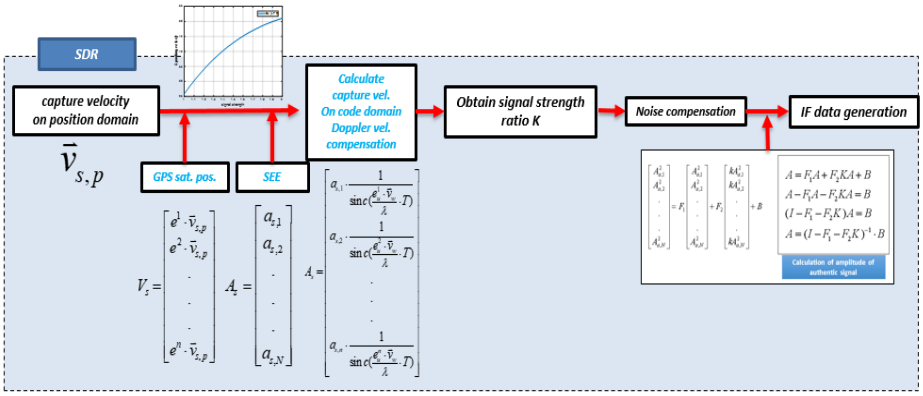


Figure 5-11. Process of IF data generation using compensated signal strength value

multiplying the corresponding standard deviation by three, it could compe sates for the maximum error. Because CCEE provides the optimal spoofin g parameters at boundary values, it is possible to calculate the optimal sig nal strength for the velocity with errors. Thus, the signal strength to be c ompensated is Δa_s.

5.2.2. Compensation value calculation for covert capture

This section analyzes the SDR results for the covert capture scenario. Figure 5-11 shows the calculation of the compensation value in the covert capture scenario obtained through CCEE and the use of it to generate IF data. Compensation for Doppler and noise lamps is carried out in the procedure. The covert capture sweep is shown in Figure 5-12. The blue line shows the user trajectory and red line indicate the trajectory of covert capture signal. Table 5-1 shows a detailed simulation setting for covert capture scenario.

Table 5-1. simulation setting for covert capture scenario.

time	2016.06.1.06.00.04
User speed	27.7 m/s
Sweep speed	121.17 m/s

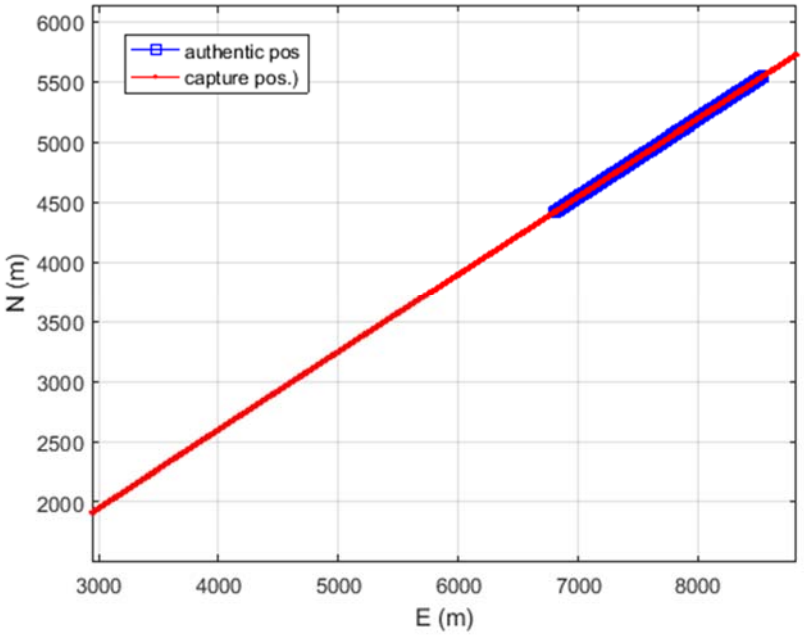


Figure 5-12. User and covert capture trajectory.

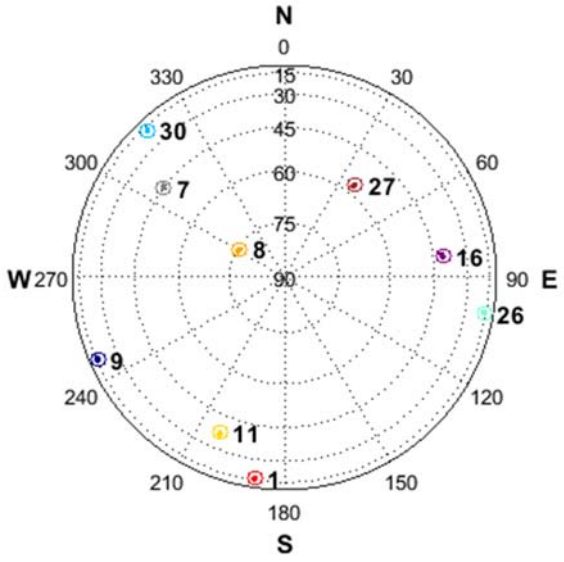


Figure 5-13. Sky plot of covert capture scenario.

Table 5-2. Simulation setting value for covert capture scenario.

PRN	K	K compensated	Authentic C/N0 (dB/Hz)	Capture C/N0 (dB/Hz)	C/N0 Offset (dB/Hz)
1	1.04	1.080	45.7	46.3	**0.67**
7	1.04	1.045	49.3	49.7	**0.38**
8	1.04	1.041	50.5	50.8	**0.35**
9	1.04	1.130	45.6	46.7	**1.06**
11	1.04	1.083	47.7	48.4	**0.69**
16	1.04	1.085	48.6	49.3	**0.70**
26	1.04	1.087	46.0	46.7	**0.73**
27	1.04	1.065	50.0	50.6	**0.54**
30	1.04	1.042	46.2	46.6	**0.35**

Figure 5-13 shows the sky plot of covert capture scenario. The number of visible satellites in this scenario is nine and each PRN number is in the figure. And table 5-2 display the signal strength ratio k, relative Doppler value, compensated k, authentic signal strength, spoofing signal strength and signal strength offset. Figure 5-14 shows the signal strengths of authentic and covert capture signal. Blue bars indicates the authentic signal strength according to the authentic signal. And red bars present the signal strength of covert capture signal. It could be seen that the signal strength difference between the authentic and spoofing signal is not large.

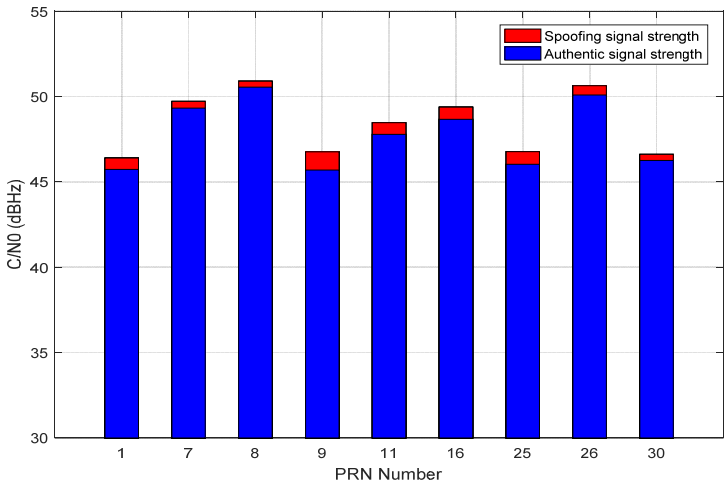


Figure 5-14. Signal strength of authentic and covert capture signal.

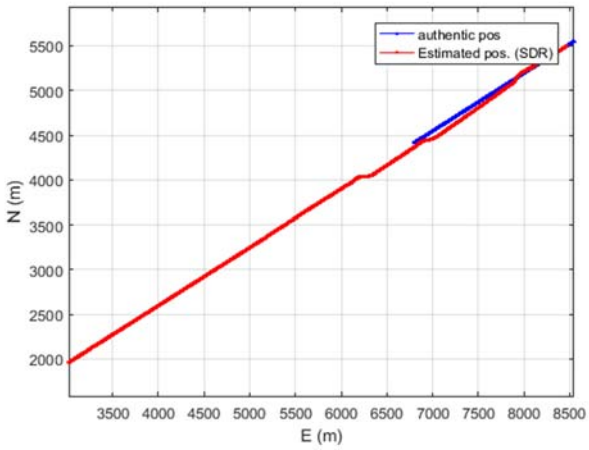


Figure 5-15. Covert capture scenario results.

Figure 5-15 shows the positioning results of SDR. We can identify that the estimate position change authentic trajectory to covert capture trajectory.

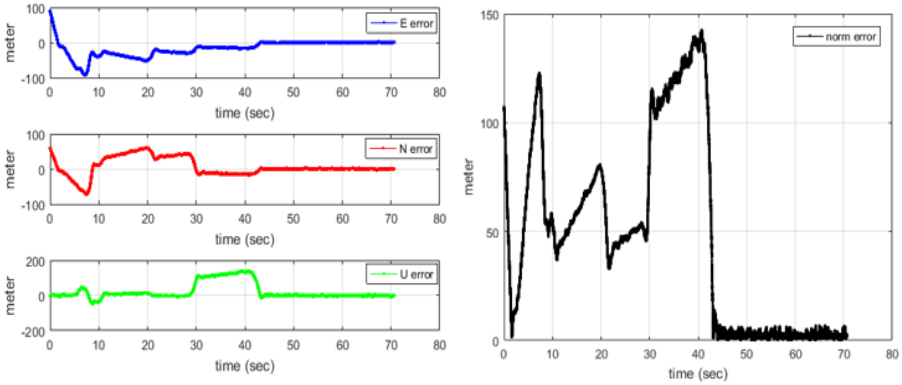


Figure 5-16. Positioning error of Covert capture scenario.

In figure 5-16, the positioning error such as each axis and norm is displayed. Position error is the difference between the intended spoofing position and the estimated position in SDR. A low position error means that the covert capture has been performed well.

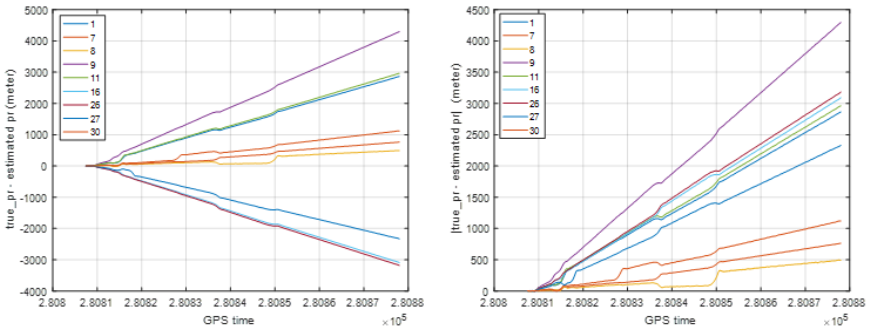


Figure 5-17. Pseudorange difference value of true authentic signal and SDR results.

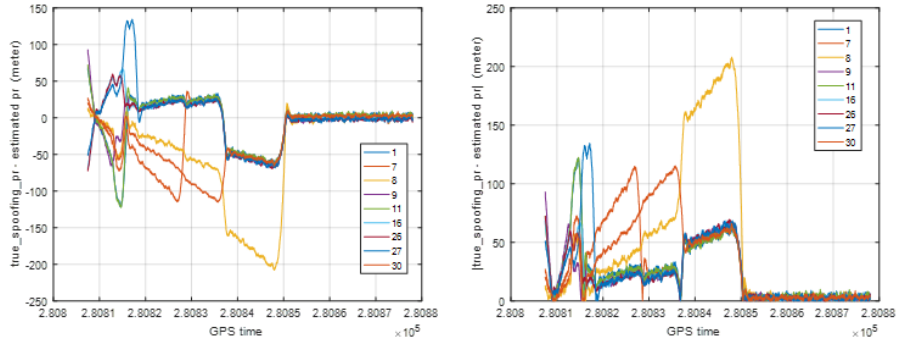


Figure 5-18. Pseudorange difference value of true covert capture signal and SDR results.

Figure 5-17 presents the pseudorange difference value of true authentic and SDR results. It shows that the difference between the authentic pseudorange and the estimated pseudorange in SDR is widening. This means that spoofing is performed well on all channels. Also figure 5-18 shows peudorange difference value of true covert capture signal and SDR results. We can identify that the difference value eventually converges. It indicates that the receiver DLL tracks the covert capture signal.

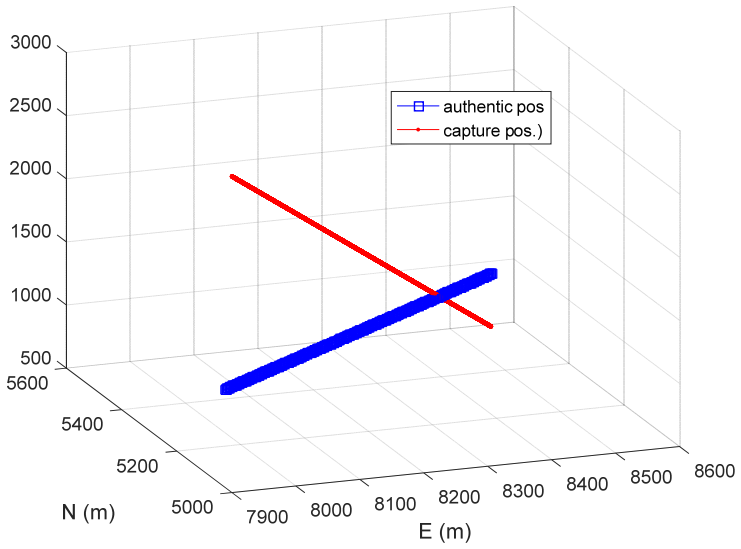


Figure 5-19. Optimal direction of covert capture scenario.

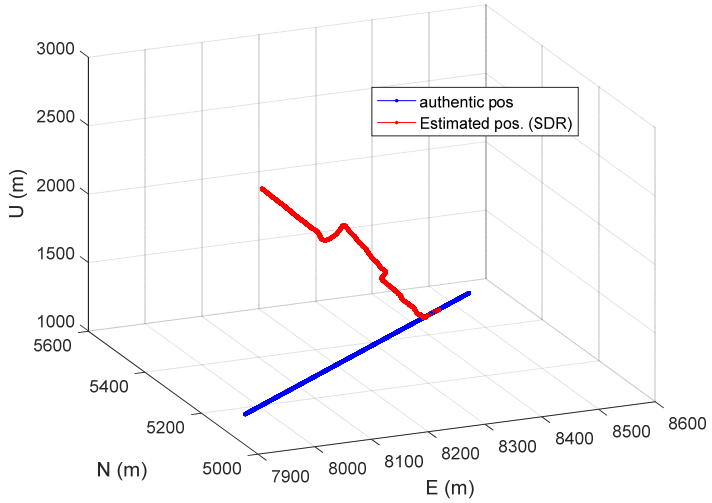


Figure 5-20. SDR results of optimal direction covert capture scenario.

5.3. Optimal covert capture direction simulation in SDR

Figure 5-19 presents optimal direction of covert capture scenario. Blue line indicates the authentic signal trajectory and red line indicates the covert capture signal trajectory. Especially, the covert capture signal is considered the optimal direction to reduce the spoofing process time. It means that the optimal direction induces that code start points of two signals make diverged fast. Figure 5-20 shows the estimated position from SDR and its input is the optimal covert capture direction scenario. It is identified that the covert capture is well performed. In addition, the success or failure of covert capture is independent of the sweep direction.

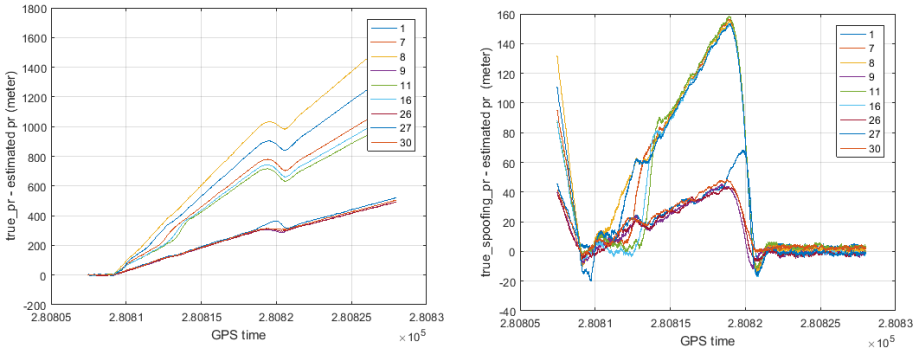


Figure 5-21. Pseudorange difference value of true covert capture signal and SDR results in optimal covert capture direction.

Figure 5-21 shows the pseudorange difference value of true covert capture signal and SDR results in optimal covert capture direction. It can be seen that the difference between the estimated pseudorange of the SDR and the pseudorange of the authentic signal is increasing. On the other hand, it can be seen that the

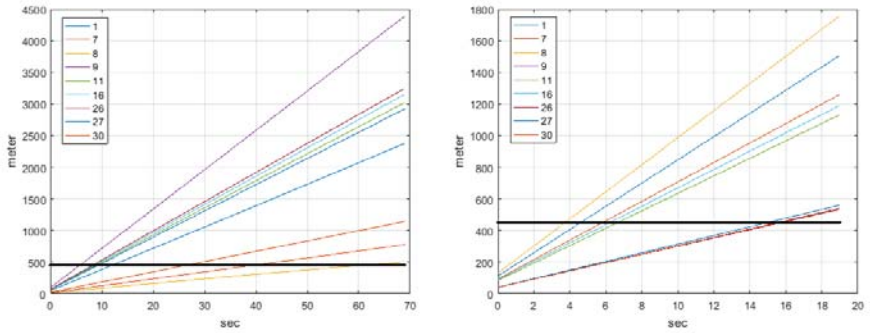


Figure 5-22. Pseudorange difference value of authentic and covert capture signal.

pseudorange of all channels converge with those of the covert capture scenario. This shows that spoofing has been performed well. Figure 5-22 shows the Pseudorange difference value of authentic and covert capture signal. When the difference between the code start point of the original signal and the spoofing signal is 1.5 chip, it is regarded that complete spoofing is done. Table 5-3 presents required time comparison of normal direction and optimal direction. The time for 1.5 chip separation is 3.5 times faster in case of optimal direction of covert capture sweep than that of normal direction. Also, the advantage of optimal direction is that the interval of abnormal WSSE is shorter than that of normal direction. The timing of the transition of the lock point of the DLL from the authentic signal to the spoofing signal is different in all channel, thereby increasing the WSSE during covert capture process. Figure 5-23. Shows the WSSE comparison between normal direction and optimal direction of covert capture scenario.

Table 5-3. Required time comparison of normal direction and optimal direction.

	Normal direction (sec)	Optimal direction (sec)
required time	62.2	17.4

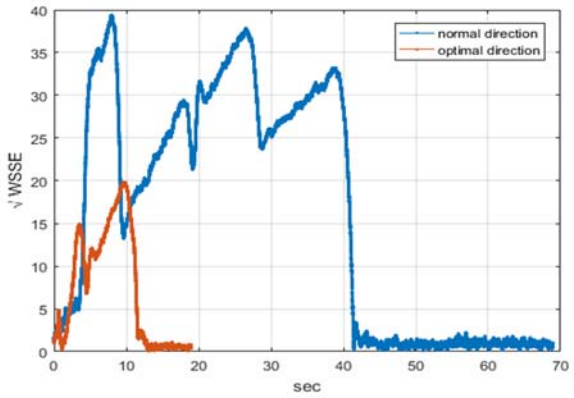


Figure 5-23. WSSE comparison between normal direction and optimal direction of covert capture scenario.

Chapter 6. Changing the user's trajectory using covert capture signal

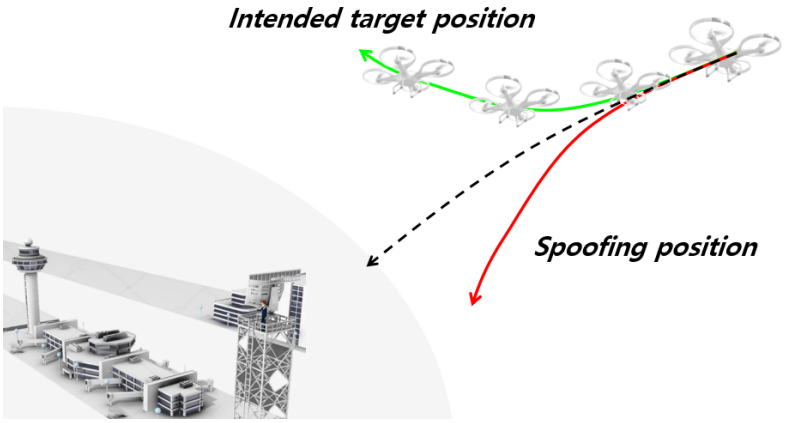


Figure 6-1. Illustration of changing the victim's trajectory using covert capture signal

This chapter discusses how to change the user's trajectory using spoofing signals. There are various researches [63-67] for drone or robot guidance and path planning. This study differs from previous researches in that it uses a spoofing signal to guide a new trajectory for a drone heading to a specific place [62]. Under the assumption that critical facilities are protected from reconnaissance drones or missiles, the target user is spoofed first, and then the covert capture signal is utilized to change the trajectory of the target user [30]. In [9], spoofing test was presented to land a hovering drone using a spoofing signal. Also, the experiment was carried out to change the path of the ship using a spoofing signal in real time [10].

In this paper, it is assumed that the target user is equipped with only one GNSS sensor and is flying in the shortest path toward the target position.

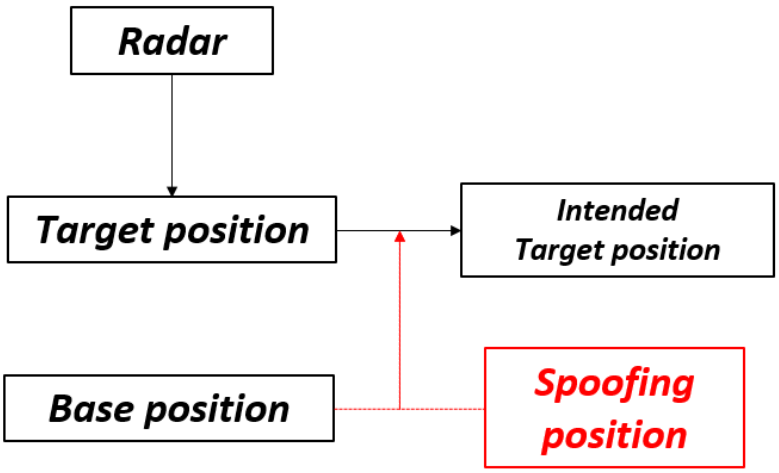


Figure 6-2 Process for the scenario of changing the trajectory of the target user using the covert capture signal.

Figure 6-1 shows the illustration of changing the victim's trajectory using covert capture signal. Using the covert capture signal, if the user position is induced to an opposite direction to the position to be guided, the user moves to the intended position. For example, for a drone hovering at a certain altitude, if experimenter uses the spoofing signal to deceive the altitude that the drone is estimating slightly, try to lower the drone to the ground by increasing the altitude to maintain a certain altitude. It is the same principle. In order to do this, it is necessary to estimate the location of the target user in real time. Therefore, this study also assumes that the position of target user could be estimated in real time using detecting system such as radar, lidar or image processing technology. In order to be induced to the intended position, the covert capture position at a specific time point is calculated and transmitted. As already mentioned, in order to guide the user's position to the intended position, the user's position must be accurately estimated by the radar. The specifications of the radars used recently suggest that the estimation is sufficient with the required accuracy. Assuming that the target user is flying to the

base position, the base station is protected by directing the target user to the intended position through a covert capture signal. Figure 6-3 shows the process for the scenario of changing the trajectory of the target user using the covert capture signal.

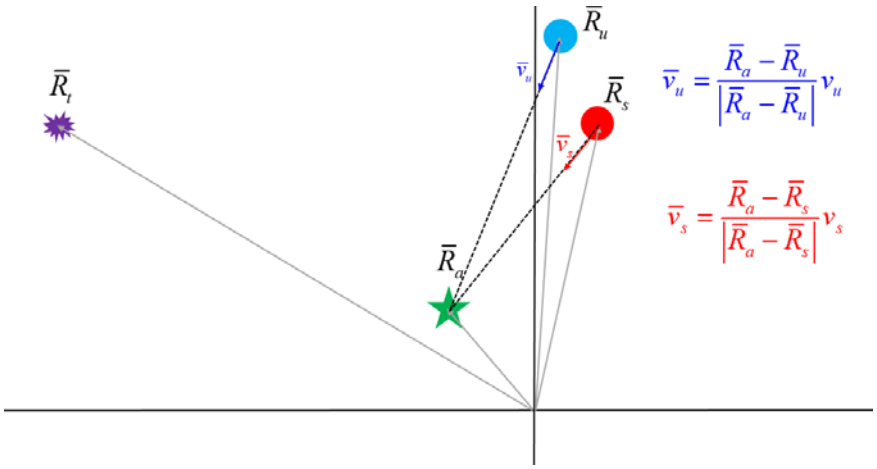


Figure 6-3 Illustration of victim trajectory change (a).

$$\bar{v}_u = \frac{\bar{R}_a - \bar{R}_u}{|\bar{R}_a - \bar{R}_u|} v_u \tag{6-1}$$

The equation (6-1) indicate the user velocity where $\bar{R}_a$ indicates the aiming position and $\bar{R}_u$ presents the user position. v_u is the drone speed. Also this means that drone moves to the aiming position in the shortest distance direction. Equation (6-2) shows the velocity spoofing signal and v_a is the spoofing signal speed.

$$\overline{v}_s = \frac{\overline{R}_a - \overline{R}_s}{\left|\overline{R}_a - \overline{R}_s\right|} v_s \tag{6-2}$$

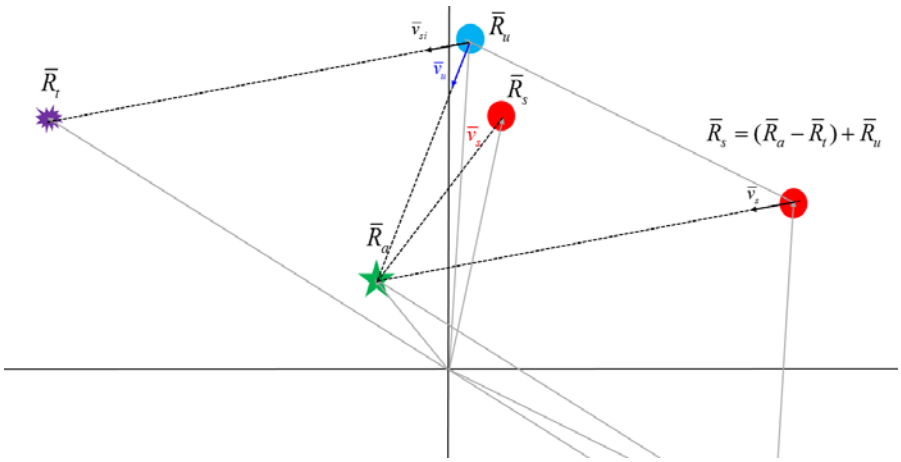


Figure 6-4 Illustration of victim trajectory change (b).

$$\overline{R}_a - \overline{R}_s = \overline{R}_t - \overline{R}_u \tag{6-3}$$

$$\overline{R}_s = (\overline{R}_a - \overline{R}_t) + \overline{R}_u \tag{6-4}$$

Equation (6-3) is required to change the victim's position to the target position. The user's position is changed to the target position, for which the covert capture position is directed to the target position. It is important to note that since the covert capture was successfully performed, the user's position is now changed to the spoofing position, which is expressed as equation (6-4).

$$\overline{R}_{si} = (\overline{R}_a - \overline{R}_t) + \overline{R}_u \tag{6-5}$$

$$\overline{R}_s = (1-\frac{t}{\tau})\overline{R}_u + \frac{t}{\tau}[(\overline{R}_a - \overline{R}_t) + \overline{R}_u] \tag{6-6}$$

In addition, at the beginning of the covert capture, the spoofed location of the current victim and the spoofing location required to send the victim to the target location are different. Let's regard that $\overline{R}_{si}$ is the actual location we need. Equation 6-5 illustrates this. At the beginning of the covert capture, the location of the user is near the location of the actual user, and the process of changing the position to $\overline{R}_{si}$ for a certain time τ could be expressed by Equation (6-6).

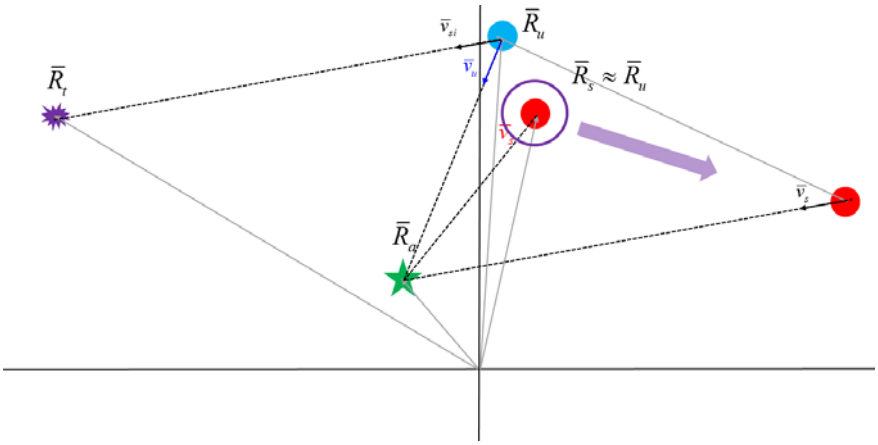


Figure 6-5 Illustration of victim trajectory change (c)

In addition, in the covert capture scenario, the tracking of the user's receiver may be affected by the trajectory of the covert capture. For example, if a Doppler or pseudorange suddenly changes, the tracking process such as DLL or FLL could be stopped. In this study, before creating measurements such as Doppler and pseudorange, the trajectory is first created and the measurement is based on that. Therefore, the trajectory was created based on the heading and velocity constraints.

In relation to this, the figure 6-6 shows velocity constraint of covert capture signal and Figures 6-7 present heading change constraint of covert capture signal.

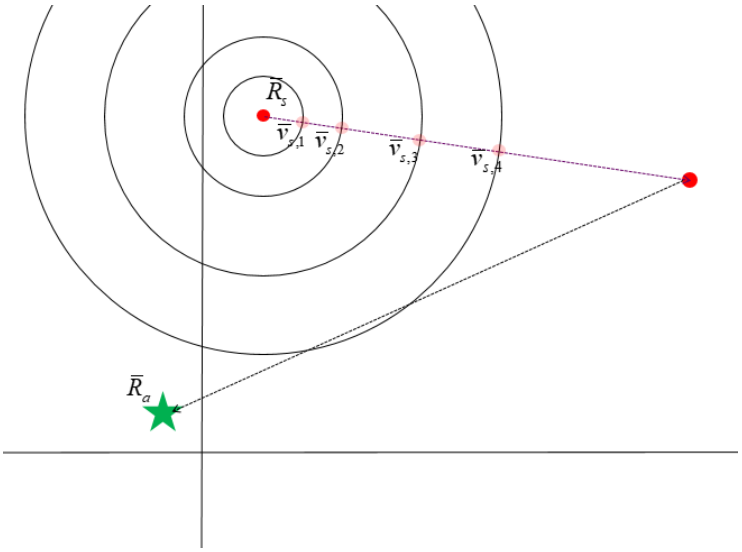


Figure 6-6 Velocity constraint of covert capture signal.

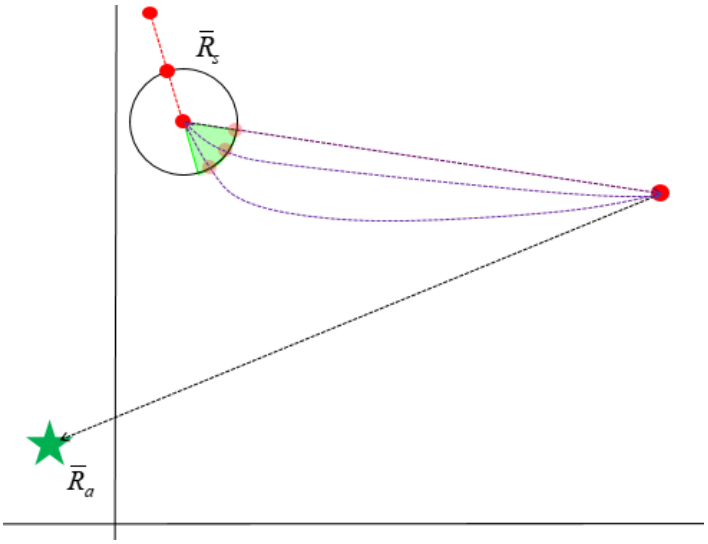
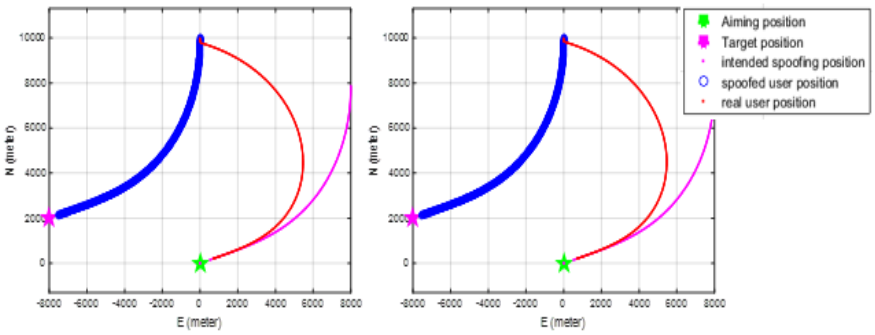


Figure 6-7 Heading change constraint of covert capture signal.



75 Figure 6-8 Victim trajectory change using covert capture signal

(Covert capture velocity: 130km/h)

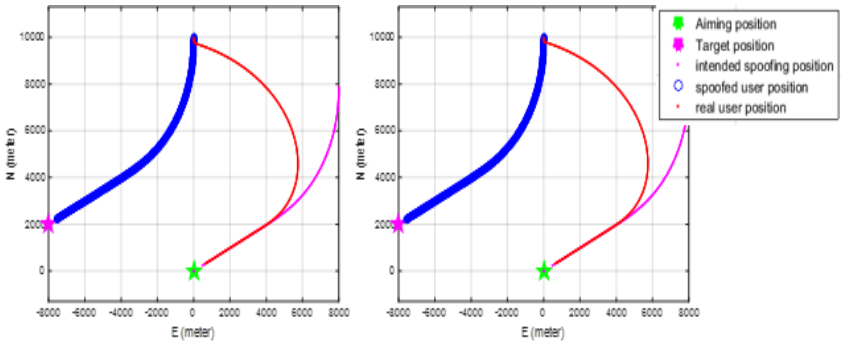


Figure 6-9 Victim trajectory change using covert capture signal.

(Covert capture velocity: 150km/h)

Figures 6-8 shows victim trajectory change using covert capture signal in case that covert capture velocity is 130km/h. Also figure 6-9 presents that victim trajectory change using covert capture signal in case that covert capture velocity is 150km/h. It is identified that the faster the covert capture signal is, the faster it can converge into the integrated spoofing position, $\overline{R}_{si}$.

Chapter 7. Conclusions and future works

7.1. Conclusions

The security and safety aspects of global navigation satellite systems have been receiving significant attention from researchers and the general public, because the use of GNSSs has been increasing in modern society. Analyzing the replica code phase variation due to the reception of the spoofing signal is important for developing spoofing attack or anti-spoofing techniques. In this paper, we propose the CCEE that could be used to calculate the replica code phase following a spoofing attack and determine whether the spoofing attack is successful using the CCEE output. The advantage of the CCEE is that it could theoretically create a minimal spoofing signal condition for a successful spoofing attack. The boundary surface dividing the spoofing attack success or failure is obtained using the CCEE. The boundary surface shows the correlation of how each spoofing parameter affects the code tracking results. This study is meaningful in that it presents a detailed study about the variation in the replica code phase during a spoofing attack process. We expect that the research results would aid the development of spoofing attack or anti-spoofing techniques.

The optimal direction for covert capture was proposed to maximize the Doppler difference value between the authentic and spoofing signal. Through this, each of code start points are diverged quickly and the spoofing process time is reduced. As a result, the abnormal section of WSSE was much minimized than that of normal direction case. It also minimizes the probability of detecting the spoofing attack by the spoofing detection technology or RAIM on the receiver.

To demonstrate the proposed technology, the IF data was generated and it is utilized for input of SDR. In IF data, the spoofing scenario for the optimal spoofing parameters and optimal direction for covert sweep was saved. The SDR results confirm that spoofing process is performed well with the optional power estimated by CCEE. In addition, the spoofing process time is significantly reduced by applying the optimal direction for sweep. Also, the simulation shows that the victim's path could be altered using the covert capture signal.

We expect that the research results would aid the development of spoofing attack or anti-spoofing techniques.

7.2. Future works

- In this paper, we analyzed the effect of the spoofing signal on the local replica code phase using the CCEE. However, for a completely successful spoofing attack, the point of FLL tracking should be moved from the authentic signal to the spoofing signal. In the future, we will focus on spoofing process analysis in the frequency domain.
- In this work, we derive the CCEE using first order Loop filter. However, when applying to the second or third Loop filter through the proposed CCEE, the error of τ was small only in certain sections. It is required to derive the new CCEE for second or third Loop filter.
- In order to change the trajectory of victim using the deceptive signal, research should focus on trajectory change of victim equipped with GPS/INS. In case of GPS/INS, the position would be estimated through KF. Thus, the trajectory of spoofing signal have to be created elaborately so that it does not fall into KF's residual.

Capture 8. Reference

[1]. Psiaki, M. L.; Humphreys, T. E. GNSS Spoofing and Detection. In Proceedings of the IEEE; 2016; Vol. 104, pp. 1258–1270.

[2]. Cavaleri, A.; Motella, B.; Pini, M.; Fantino, M. Detection of spoofed GPS signals at code and carrier tracking level. In Proceedings of the 2010 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), Noordwijk, The Netherlands, 8–10 December 2010; pp. 1–6.

[3]. Tippenhauer, N. O.; Pöpper, C.; Rasmussen, K. B.; Capkun, S. On the requirements for successful GPS spoofing attacks. Proc. 18th ACM Conf. Comput. Commun. Secur. - CCS '11 2011, 75, doi:10.1145/2046707.2046719.

[4]. Hui, H.; Na, W. A study of GPS jamming and anti-jamming. PEITS 2009 - 2009 2nd Conf. Power Electron. Intell. Transp. Syst. 2009, 1, 388–391, doi:10.1109/PEITS.2009.5406988.

[5]. Ng, Y.; Gao, G. X. "Mitigating jamming and meaconing attacks using direct GPS positioning", Proc. IEEE/ION Position, Locat. Navig. Symp. PLANS 2016, 2016, 1021–1026, doi:10.1109/PLANS.2016.7479804.

[6]. A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," J. Field Robot., vol. 31, no. 4, pp. 617–636, 2014.

[7]. Jafarnia-Jahromi, A.; Broumandan, A.; Nielsen, J.; Lachapelle, G. GPS vulnerability to spoofing threats and a review of antispoofing techniques. Int. J. Navig. Obs. 2012, 2012, doi:10.1155/2012/127072.

[8]. Humphreys, T. E.; Ledvina, B. M.; Tech, V.; Psiaki, M. L.; Hanlon, B. W. O.; Kintner, P. M. Assessing the Spoofing Threat : Development of a Portable GPS Civilian Spoofer. Proc. 21st Int. Tech. Meet. Satell. Div. Inst. Navig. (ION GNSS 2008) Sept. 16 - 19, 2008 Savannah Int. Conv. Cent. Savannah, GA 2009, 2314–2325, doi:10.1111/grow.12084.

[9]. Shepard, D. P.; Bhatti, J. A.; Humphreys, T. E. Drone Hack: Spoofing Attack Demonstration on a Civilian Unmanned Aerial Vehicle. GPS World 2012, 23, 30–33.

[10]. Bhatti, J.; Humphreys, T. E. Hostile Control of Ships via False GPS Signals: Demonstration and Detection. Navig. J. Inst. Navig. 2017, 64, 51–66, doi:10.1002/navi.183.

[11]. Wang, K.; Chen, S.; Pan, A. Time and position spoofing with open source projects. Black Hat 2015, 148.

[12]. Wang, F.; Li, H.; Lu, M. GNSS spoofing detection and mitigation based on maximum likelihood estimation. Sensors (Switzerland) 2017, 17, doi:10.3390/s17071532.

[13]. Psiaki, M. L.; O'Hanlon, B. W.; Bhatti, J. A.; Shepard, D. P.; Humphreys, T. E. GPS spoofing detection via dual-receiver correlation of military signals. IEEE Trans. Aerosp. Electron. Syst. 2013, 49, 2250–2267, doi:10.1109/TAES.2013.6621814.

[14]. Manfredini, E. G.; Dovis, F. On the use of a feedback tracking architecture for satellite navigation spoofing detection. Sensors (Switzerland) 2016, 16, doi:10.3390/s16122051.

[15]. Liu, K.; Wu, W.; Wu, Z.; He, L.; Tang, K. Spoofing detection algorithm based on pseudorange differences. Sensors (Switzerland) 2018, 18, 1–20, doi:10.3390/s18103197.

[16]. 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, doi:10.1017/S0373463317000558.

[17]. Daneshmand, S.; Jafarnia-jahromi, A.; Broumandan, A.; Lachapelle, G. A Low-Complexity GPS Anti-Spoofing Method Using a Multi-Antenna Array. 2016.

[18]. Broumandan, A.; Lachapelle, G. Spoofing Detection Using GNSS/INS/Odometer Coupling for Vehicular Navigation. Sensors 2018, 18, 1305, doi:10.3390/s18051305.

[19]. Perdue, L.; Sasaki, H.; Fischer, J. Testing GNSS Receivers to Harden Against Spoofing Attacks. Int. Symp. GNSS 2015.

[20]. Ma, C.; Lachapelle, G.; Cannon, M. E. Implementation of a Software GPS Receiver. Architecture 2003, 2004, 956–970.

[21]. P. Misra and P. Enge, Global Positioning System - signals, Measurement, and Performance, second ed.: Ganga-Jamuna Press, 2006.

[22]. M. L. Psiaki, et al., "GNSS lies, GNSS truth: Spoofing detection with two-antenna differential carrier phase," GPS World, vol. 25, no. 11, pp. 36–44, Nov. 2014.

[23]. Grant, A., "GPS Jamming and the Impact on Maritime Navigation," Journal of Navigation, Vol. 62, No. 2, 2009.

[24]. International Marine Contractors Association, "Guidelines for the Design and Operation of Dynamically Positioned Vessels," 2007. Available at: http:// www.imca-int.com/media/73055/imcam103.pdf.

[25]. Thomas, M., Norton, J., Jones, A., Hopper, A., Ward, N., Cannon, P., Ackroyd, N., Cruddace, P., and Unwin, M., "Global Navigation Space Systems: Reliance and Vulnerabilities,"

[26]. Tim Klimasewski, Lisa Perdue, "Testing GPS Susceptibility to Spoofing Attacks", DATT Summit National Harbor, MD April 25-28, 2016.

[27]. S.-H. Seo, B.-H. Lee, S.-H. Im, G.-I. Jee, "Effect of spoofing on unmanned aerial vehicle using counterfeited gps signal", Journal of Positioning Navigation and Timing, vol. 4, no. 2, pp. 57-65, 2015.

[28]. “Vulnerability assessment of the transportation infrastructure relying on the Global Positioning System," Tech. rep., John A. Volpe National Transportation Systems Center, 2001.

[29]. 조성룡, et al ,” 의사거리 측정치를 이용하는 기만신호 검출 기법의 성능 비교”, 한국 군사 과학 기술 학회지 13 (5), 2010, 793 ~ 800

[30]. Juhwan Noh, et al,"Tractor Beam: Safe-hijacking of Consumer Drones with Adaptive GPS Spoofing", ACM Transactions on Privacy and Security, Vol. 22, No. 2, April 2019.

[31]. Heng, L.; Work, D.B.; Gao, G. Cooperative GNSS authentication. Reliability from unreliable peers. Inside GNSS 2013, 8, 70–75.Sensors 2016, 16, 2051 21 of 22

[32]. De Castro, H.V.; van der Maarel, G.; Safipour, E. The possibility and added-value of authentication in future Galileo open signal. In Proceedings of the 23th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2010), Portland, OR, USA, 21–24 September 2010.

[33]. Cheng, X.-J. ; Xu, J.-N. ; Cao, K.-J.;Wang, J. An authenticity verification scheme based on hidden messages for current civilian GPS signals. In Proceedings of the Fourth International Conference on Computer Sciences and Convergence Information Technology, Seoul, Korea, 24–26 November 2009; pp. 345–352.

[34]. Pini, M.; Motella, B.; Pilos, L.; Vesterlund, L.; Blanco, D.; Lindstrom, F.; Maltoni, C. Robust On-board ship equipment: The TRITON Project. In Proceedings of the 10th International Symposium Information on Ships, Hamburg, Germany, 4–5 September 2014.

[35]. UT Austin Researchers Successfully Spoof an $80 Million Yacht at Sea. 2013. Available online http://www.utexas.edu/news/2013/07/29/ut-austin-researchers-successfully-spoof-an-80-million-yacht-at-sea/

[36]. Manfredini, E.G.; Dovis, F.; Motella, B. Signal Quality monitoring for discrimination between spoofing and environmental effects, based on multidimensional ratio metric tests. In Proceedings of the 28th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2015), Tampa, FL, USA, 14–18 September 2015.

[37]. Huang, L.; Lv, Z.C.; Wang, F.X. Spoofing pattern research on GNSS receivers. J. Astronaut. 2012, 33, 884–890.

[38]. Dehghanian, V.; Nielsen, J.; Lachapelle, G. GNSS spoofing detection based on signal power measurements: Statistical analysis. Int. J. Navig. Obs. 2012, 7, 1–8.

[39]. Jafarnia-Jahromi, A.; Broumandan, A.; Nielsen, J.; Lachapelle, G. Pre-despreading authenticity

[40]. verification for GPS L1 C/A signals. Navig. J. Inst. Navig. 2014, 61, 1–11. Borio, D. PANOVA tests and their application to GNSS spoofing detection. IEEE Trans. Aerosp. Electron. Syst. 2013, 49, 381–394.

[41]. Zhang, Y.T.;Wang, L.;Wang,W.Y.; Lu, D.;Wu, R.B. Spoofing jamming suppression techniques for GPS based on DoA estimating. In Proceedings of the China Satellite Navigation Conference (CSNC 2014), Nanjing, China, 21–23 May 2014.

[42]. P. W. Bradford, J. Spilker, and P. Enge, Global positioning system: theory and applications vol. 1, 1996.

[43]. B. Bailey, "GPS Modernization Update & Program Plans," in Proceedings of the 13th Meeting of National Space-Based Positioning, Navigation, and Timing Advisory Board, 2014.

[44]. Comp CJ, Axelrad P (1998) An adaptive SNR-based carrier phase multipath mitigation technique. Trans Aerospace Electron Syst 34(1):264–276

[45]. Cohen C, Parkinson B (1991) Mitigating multipath error in GPS-based attitude determination. Advances in the astronautical sciences, AAS guidance and control conference, Keystone. Univelt, San Diego, pp 74–78

[46]. Byun SH, Hajj GA, Yang LE (2002) Development and application of GPS signal multipath imulator. Radio Sci 37(6):10-1–10-23

[47]. Bilich A, Larson KM (2007) Mapping the GPS multipath environment using the signal-to-noise ratio (SNR). Radio Sci 42(6):RS6003

[48]. Hernandez-Pajares Met al,"Distribution and mitigation of higher-order ionospheric effects on precise GNSS processing. J Geophys Res Solid Earth 119(4):3823–3837

[49]. Banville S, et al,"Ionospheric monitoring using “integer-levelled” observations. In: Proceedings of ION GNSS 2012, Nashville, 17–21 Sept 2012. Institute of Navigation, pp 2692–2701

[50]. Komjathy A,et al,"Automated daily processing of more than 1000 ground-based GPS receivers for studying intense ionospheric storms. Radio Sci 40(6):RS6006

[51]. Odijk D "Weighting ionospheric corrections to improve fast GPS positioning over medium distances. In: Proc. ION GPS 2000, September 19–22,pp 1113–1123

[52]. Böhm J,et al,"Troposphere mapping functions for GPS and VLBI from ECMWF operational analysis data. J Geophys Res 111:B02406

[53]. Black H,"An easily implemented algorithm for the tropospheric range correction. J Geophys Res 83(B4):1825–1828

[54]. Hopfield H "Tropospheric effect on electromagnetically measured range: prediction from surface weather data. Radio Sci 6:357–367

[55]. GNSS Threats, Attacks and Simulations, https://www.gps.gov/governance/advisory/meetings/2017-06/buesnel.pdf

[56]. Key, E. L., “Techniques to Counter GPS Spoofing,” Internal memorandum, MITRE Corporation, Feb. 1995

[57]. 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

[58]. Taehee Kim, Jaehoon Kim, Sanguk Lee, “Analysis of Performance of Spoofing Detection Algorithm in GPS L1 signal”, 통신위성우주산업연구회논문지 제 8 권 제 2 호, 2013

[59]. Taehee Kim, Cheon Sig Sin, Sanguk Lee, “Analysis of Effect of Spoofing Signal According to Code delay in GPS L1 Signal”, 통신위성우주산업연구회논문지 제 7 권 제 1 호, 2012

[60]. Mi-young Shin, Sung-Lyong Cho, Jun-Oh Kim, Ki-Won Song, Sang-Jeong Lee, “Analysis of GPS Spoofing Characteristics and Effects on GPS Receiver”, 한국군사과학기술학회지 제 13 권 제 2 호, PP. 296~303, 2010 년 4 월

[61]. C. Gu¨nther, “A survey of spoofing and counter-measures,” Navigation, vol. 61, no. 3, pp. 159–177, 2014.

[62]. 1. Bevacqua, G.; Cacace, J.; Finzi, A.; Lippiello, V. Mixed-initiative planning and execution for multiple drones in search and rescue missions. Proc. ICAPS 2015, 2015–January, 315–323.

[63]. 2. Lin, Y.; Saripalli, S. Sampling-Based Path Planning for UAV Collision Avoidance. IEEE Trans. Intell. Transp. Syst. 2017, 18, 3179–3192, doi:10.1109/TITS.2017.2673778.

[64]. 3. Nageli, T.; Meier, L.; Domahidi, A.; Alonso-Mora, J.; Hilliges, O. Real-time planning for automated multi-view drone cinematography. ACM Trans. Graph. 2017, 36, doi:10.1145/3072959.3073712.

[65]. 4. Noh, J.; Kwon, Y.; Son, Y.; Shin, H.; Kim, D.; Choi, J.; Kim, Y. Tractor beam: Safe-hijacking of consumer drones with adaptive GPS spoofing. ACM Trans. Priv. Secur. 2019, 22, doi:10.1145/3309735.

[66]. 5. Soto, M.; Nava, P. A.; Alvarado, L. E. Drone formation control system real-time path planning. Collect. Tech. Pap. - 2007 AIAA InfoTech Aerosp. Conf. 2007, 1, 606–639, doi:10.2514/6.2007-2770.

[67]. 6. Usenko, V.; Von Stumberg, L.; Pangercic, A.; Cremers, D. Real-time trajectory replanning for MAVs using uniform B-splines and a 3D circular buffer. IEEE Int. Conf. Intell. Robot. Syst. 2017, 2017–September, 215–222, doi:10.1109/IROS.2017.8202160.

[68]. Jafarnia-Jahromi, A. (2013) GNSS Signal Authenticity Verification in the Presence of Structural Interference. PhD Thesis, Report No. 20385, Department of Geomatics Engineering, University of Calgary.

초 록

GNSS는 점점 활용범위가 확장되고 있고, 현재는 대체불가능한 시스템이 되었다. 이런 상황에서 GNSS의 안전 및 보안의 중요성 또한 크게 증가하고 있다. 본 논문에서는 GNSS의 보안에 가장 위협이 되는 기만에 대해서, 기만 신호가 수신기에 주입되었을 때 수신기의 ACF가 어떻게 변화되어 가며 기만 공격을 결정하는 주된 기만파라미터들의 특징에 대해서 분석을 진행하였다. 그리고 기만 신호에 따른 기만 결과를 결정하는 CCEE를 제안하고, 이를 통해서 기만파라미터들의 상관관계에 대해서 분석하였다. 기존에는 무수히 반복된 계산을 통해서 판단 가능한 기만 결과를 CCEE를 통해 한번의 계산으로 결과를 확인하도록 하였다. 또한 CCEE를 이용하여 경계 값과 경계 라인을 정의함으로써, 기만파라미터 공간에서 기만 성공과 실패를 구분할 수 있음이 확인되었다.

수신기에서 기만이 수행될 때, 코드도메인상에서 replica와 cross correlation에 의한 원신호와 기만신호 각각의 correlation peak가 생성된다. 두 신호 peak의 상대속도가 기만이 수행되는 시간을 결정한다. 일반적으로 기만이 수행되는 동안, 각 채널간 DLL tracking lock 지점이 원신호에서 기만신호로 전환되는 시점이 다르다. 이로 인해서 WSSE의 값이 상승하게 된다. 이를 최소화하기 위해서, 최적 기만 sweep 방향을 결정함으로써 빠른 시간에 기만을 수행할 수 있음을 확인하였다. 3D 상황에서 특정 가시위성를 이용하여 삼각형을 정의하고, 해당 삼각형의 외심 방향이 최적 방향이 되며,

해당 방향이 기만 수행이 가장 늦게 되는 가시위성에 대한 원신호와 기만신호의 상대속도가 최대가 되는 방향임을 확인하였다.

제안된 방법들을 모사하기 위해서, 기만시나리오를 정의하고, 해당 기만시나리오를 모사하는 IF data를 생성하였다. 그리고, 해당 IF data를 이용하여, SDR을 통해서 신호 처리를 진행하였다. 이를 통해, CCEE를 적용하여 생성한 최적 기만파라미터로 기만이 의도된 데로 수행이 되며, optimal 방향을 통해 기만수행시간이 최소화 됨을 확인하였다.

Key word: GNSS, GNSS receiver, GNSS interference, spoofing, DLL, WSEE, victim trajectory

학번: 2014-30355

감사의 글

박사학위를 받는다고 생각하니, 그토록 꿈꿔왔던 일이지만, 부끄러운 생각이 많이 듭니다. 하지만, 이제 사회로 나가는 첫출발이니 지금까지의 배움을 바탕으로 힘차게 살아보려고 합니다.

먼저 지도교수님이신 기창돈교수님께 감사를 드립니다. 부족한 제자를 인내와 사랑으로 지도해주셨습니다. 교수님의 지도로부터 저 또한 많이 성장했다고 믿습니다. 사회에 나가서도 교수님의 가르침을 기억하며 최선을 다해 살도록 하겠습니다. 또한 심사위원장으로 수고해주신 김유단 교수님께 감사 드리며, 심사위원으로 멀리까지 찾아오신, 허문범 박사님, 박준표 박사님, 전상훈 박사님께 감사 드립니다.

GNSS 연구실 구성원들께 감사를 드립니다. 먼저 종원이에게 감사를 드립니다. 처음 연구실에 들어와서 아무것도 모를 때, SDR 및 의사위성 전수를 성심성의 것 해줘서, 연구하는데 많이 도움이 되었습니다. 그리고 오종이에게 감사를 드립니다. 같은 방에서 생활하면서, 연구와 연구외적으로 많은 의지가 되었습니다. 심사위원으로 까지 오셨던 전상훈박사님께 감사 드립니다. 기만과제 할 때, 형의 도움으로 무사히 마쳤다고 생각합니다. 그리고 수신기에 대해서도 많이 배웠습니다. 형께 배운 것이 거름이 되어 학위논문이 완성된 것 같습니다. 민혁이에게 감사 합니다. 졸업 연구를 진행하면서, 같이 논의했던 것들이 큰 힘이 되었습니다. 앞으로

학위를 기만주제로 할지, 수신기 알고리즘으로 할지는 모르지만, 연구가 잘 풀려서 순조롭게 졸업하시기를 바랍니다. 또한 졸업동기인 정범, 선경, 동욱에게 감사 드립니다. 같이 졸업을 준비하면서 많은 의지가 되었습니다. 졸업과 취업 모두 축하 드립니다. 또한 이미 졸업하여 사회에 나가신, 송준솔, 노희권, 김연실, 한덕화 박사님께 감사 드리고, 큐브위성 연구로 바쁜 한준이, 301동 식구 부겸, 호준이, 302동 식구 영환이, 그리고 새롭게 정밀연 막내로 들어온 종주, 마지막으로 같이 졸업하는 다니엘에게 감사를 드립니다. 또한, 학교생활을 같이 했던 민규, 문기, 민호, 주영과 동기인 민우형께 감사드립니다.

KIST 이택진 박사님께 감사 드립니다. 박사님을 만난 것은 저에게 너무 큰 행운인 것 같습니다. 박사학위 최종심사가 끝났을 때, 정말 진심으로 축하해 주신 것이 기억이 납니다. 또한 학위 과정 동안 너무 많은 배려를 해주셨지요... 진심으로 감사 드립니다. 정호에게 감사합니다. 같이 KIST에서 생활하면서 정말 큰 힘이 되었습니다. 빠른 시기에 박사 졸업할 수 있으시길 기원합니다. 그리고 새롭게 팀에 합류하게 된 동현이, 창수에게도 감사 합니다.

빛의 자녀 교회 김형민 목사님께 감사 드립니다. 제가 박사학위를 하게 된 출발점이 목사님의 설교말씀 이였습니다. 항상 변함없는 모습으로 가르치고 이끌어 주셔서 제가 많이 성장한 것 같습니다. 그리고, 저의 목자이셨던, 구일/은경 목자님께 감사 드립니다. 같이 나눔 하면서, 많은 의지가 되고 힘이 되었습니다. 특히 구일 목자님께

참 목자가 무엇인가에 대해 배웠습니다. 감사 드립니다.

부모님께 감사 드립니다. 변함없이 저를 믿어 주시고, 격려해주셔서 감사 드립니다. 무엇보다 부모님 모두 건강하시길 바랍니다. 이제 아이들과 많이 찾아 뵙도록 하겠습니다. 그리고 형과 형수님께 감사 드립니다. 가끔 형과 나누는 전화가 정말 큰 힘이 되었습니다.

마지막으로 사랑하는 아내인 정은이에게 감사 드립니다. 박사 학위 동안 너무 많이 고생시켜서 미안합니다. 옆에서 큰 힘이 되어 주었습니다. 이제 평생 보답하며 살겠습니다. 그리고 사랑하는 아들 은찬, 딸 기쁨이에게도 감사 드립니다.

6년이라는 학위과정 동안 정말 많은 분들의 도움을 받았습니다. 지금까지 해온 것을 바탕으로, 배우는 자세로 열심히 살아보도록 하겠습니다. 감사합니다.

2020년 1월