# Detection of GPS Spoofing

Joseph Le
*Purdue University, West Lafayette, Indiana, 47906, USA*

## I. Nomenclature

| | | |
|---|---|---|
| $\rho$ | = | pseudorange |
| $\rho_c$ | = | Corrected pseudorange |
| $x, y, z$ | = | Receiver position in ECEF |
| $x^i, y^i, z^i$ | = | $i^{th}$ satellite position |
| $r^i$ | = | Geometric range between the $i^{th}$ satellite and the receiver |
| $u^i$ | = | Unit vector between the receiver and the $i^{th}$ satellite |
| $k$ | = | time step |
| *SDA* | = | *Pseudorange difference calculated values* |
| *LS* | = | *Least Squares calculated values* |
| $\mu$ | = | Mean |
| $N_0$ | = | Noise Floor |
| $\delta t$ | = | Time bias in measurement |

## II. Introduction

The GPS (Global Positioning System) spoofing is the act of interfering with the workings of various parts of the system such that a false location is determined. Since the codes and signal structures for GPS are available to the public, attackers are able to reproduce, alter, and overpower these signals that are received by a receiver and interferes with the algorithm that calculates the positioning. This is typically done by transmitting false signals that assume the codes/signals of various satellites such that the receiver calculates the location that was determined by the transmitter, this desired false location may be a determined location that is spoofed or a random mix of signals that return random or nonsensical on the receiver end [6].

The risks of GPS spoofing vary in many ranges. This may include interference with the power grid, takeover of the control of remotely controlled vehicles [1], and risks in cybersecurity either in civilian cases or government and military applications. These risk not only harms the workings of society but could also physically affect people within it.

Detection of these spoofed transmissions has been a deep field yielding several methods that make use of various properties of signals and the way that the GPS works. Such methods may include detection through correlation of the signals, doppler and electromagnetic properties, and several new cutting-edge methods. The detection of these GPS spoofing signals is important to improve the security, safety, and overall workings of society and advert the risks that these spoofing attacks may contain.

## III. Background and Review of Literature

### Methods of Spoofing

The first method of spoofing is to bombard the target with a simulated GPS signal that can contain false data and has much higher power than that of the true GPS signal. This will cause the receiver to lock onto the false signal rather than the true GPS signal. This works due to the automatic gain control (AGC) [3] units that are present in many receivers which adjust the gain and power that that receiver locks onto as a spoofed signal's power is increased. The simulated signal does not need to be fully synchronized with the authentic GPS signal [9] but commercial receivers with less robust methods of detecting spoofing, will still be susceptible to this attack. This is the simplest method of spoofing, and many modern receivers are able to easily detect this type of spoofing with monitoring of signal parameters, consistency checks, and other measurements available in certain systems such as movement.

Another method of spoofing is to use a receiver to match the characteristics of a GPS signal more closely. The spoofer uses a receiver that may be nearby to the target receiver and collects the same data as target receiver in order to simulates the authentic GPS signal. The spoofer then injects a false signal into the true signal and transmits towards the target receiver at such a low power that it is in the noise floor of the true signal. After the infected signal is received by the target receiver from the spoofing transmitter, the spoofer then slowly increases the power of the false GPS signal to the point that receiver locks onto the false signal rather than the legitimate signal. The spoofer then is able to shift the false signal to a determined code which changes the calculations that are done by the target receiver which changes the location that is determined by the target. The closer this spoofing apparatus is to the target receiver the easier it is to work since the GPS data received by the spoofer (delay, carrier freq., phase, etc.) are much closer to the receiver and requires less work to match. Since the signals that are transmitted match up closely and are synchronized to that of the intended GPS signal, it is hard to detect with the previous methods that were mentioned. This method of spoofing is also known as repeater spoofing.

A more sophisticated method of spoofing, which is also the most complex method, is receiver-based spoofers [9]. Though this method is hardest to detect it is also the hardest to implement, where the position of the target receiver must be known down to the centimeter level in order to perfectly mimic the code and carrier phases in the spoofing signal. Once this information is known, an array of transmitters is utilized which must also match the array manifold of the target receiver in order to circumvent any angle of arrival filters in the receiver. This method of spoofing is very robust and can defeat many spoof detection methods, but this is offset by the complexity and the limited range of this approach to spoofing.

## Dangers and Vulnerabilities in GPS Dependent Systems

A major danger of the effects of GPS spoofing come in the form of vehicle control [2]. Being able to provide false positioning date, a spoofer is capable of hijacking the autopilot system of boats, planes, and other vehicles [5]. This is a concern especially for military vehicles that may have weapons and sensitive data onboard, posing a security issue. Though military GPS signals are encrypted using P codes [1] which hinder the efforts of spoofing, there are still cases of GPS spoofing hijack as occurred in 2011 when Iran captured US drones [4].

Other major attacks that may occur are those on societal infrastructure that depend on GPS data. This includes the telecommunication network [5] which depends on the synchronization between the clocks in the towers which reduces the interference between the towers. A spoofer could disturb the synchronization between certain towers which renders the tower inoperative until the attack is halted. Similarly, city power grids depend on the GPS system to synchronize and time stamp measurements to detect variations and other data within that power grid. A spoofing attack could cause instabilities and distortions in the system or cause false readings that may lead to blackouts and damage to the system [5].

## Description of Spoofing Detection Methods and Limitations

The first method of spoofing detection is monitoring the power of the received GPS signals. Within this method, various parameters can be used to monitor the signal, one such parameter is the signal to noise ratio (C/N0) [5]. This method relies on the fairly stable signal to noise ratio that is received from the GPS, while as an attacker tries to interfere with the signal by using a higher power, the signal to noise ratio may change. The simplicity of this method also indicates its robustness. If a spoofer can recreate the ratio that the receiver detects, then they are able to bypass the signal to noise ratio checks. Other power monitoring methods include absolute power monitoring [5] and power variation vs. movement monitoring. Absolute power monitoring has similar vulnerabilities as the signal to noise ratio monitoring, but cross referencing the movements of a receiver to the power variation may be a safer bet. This method is based on the power variations that occur when a receiver is moving (doppler shift), cross referencing this to the movements checks if the GPS signal that is received is legitimate, while a GPS spoofer that does not consider the motion of a receiver may not be able to bypass these checks. A limitation of this method is that not all receivers are free to move, thus leaving stationary receivers vulnerable.

Spatial processing makes use of multiple antennas to cross reference between each other to determine the legitimacy of a GPS signal. This is effective against GPS spoofers that make use of a single antenna to transmit counterfeit signals. Multiantenna spoofing discrimination [9] uses the separation of multiple fixed antennas to detect the phase differences between the antennas, this information can be used to calculate the distances between the antennas with GPS data and compare them to the actual distance between the antennas. A single antenna spoofer transmitting illegitimate signals that are received by an antenna array will result in incorrect orientation calculations which will result in the detection of the spoofer. The limitation of this method is the physical complexity of the array of antennas which cannot be used on mobile receivers or smaller receiver facilities.

Signal Quality Monitoring (SQM), which was used originally for multipath detection, focuses on the PRN codes that are sent out by GPS satellites. As a spoofing signal slowly takes over the tracking of an authentic signal, the phases in the code will also slowly shift and separate from each other [10], this will then affect the correlated signal where the signal from a spoofer will match up with the signal from a GPS signal. SQM employs two tests (ratio and delta) [9] which detect any type of “flatness” peaks in the correlated signals of the received signals and abnormal asymmetry in said signal. These tests are done through vector tracking which monitors the PRN phase, doppler shift, and carrier phase. The method keeps track of these values and compares them to the signal that is transmitted by a spoofer to determine the authenticity of the signal, if the doppler shift and PRN code phase end being nonzero after the algorithm is implemented it is considered a counterfeit signal. A limitation of this method is that it assumes that the receiver has already locked onto an authentic signal peak before being attacked by a spoofer, also, since SQM is used for multipath monitoring, this method is unable to detect the difference between a signal that is being reflected and a signal that is being spoofed.

The methods that will be used to detect spoofing will be explained in depth and detail in later sections of this paper. This methods uses the pseudorange differences to determine the presence of spoofing. The mathematical analysis will be included for the pseudorange difference method and will be implemented on a publicly available spoofing dataset from the University of Texas at Austin called the TEXBAT dataset.

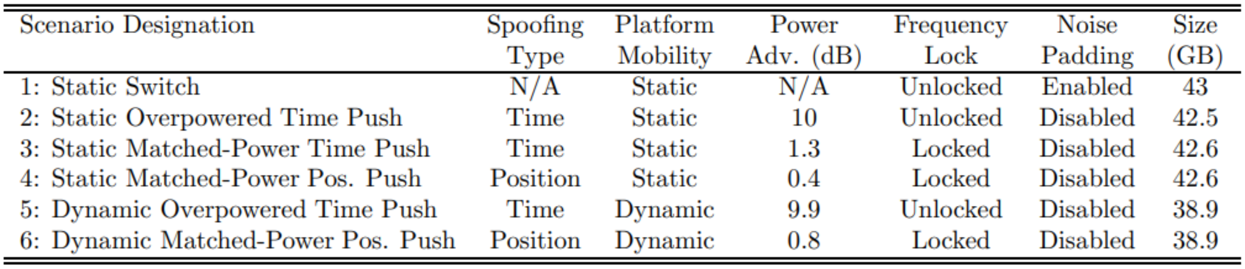
## IV. The TEXBAT Dataset

### A. TEXTBAT Dataset

The Texas Spoofing Test Battery (TEXBAT) dataset [11] was created by the University of Texas at Austin to be the standard for testing GPS/GNSS spoofing detection and authentication algorithms. The data set has 8 different scenarios compiled that represent different methods of spoofing a GPS signal. The dataset was created to help improve methods of detecting spoofing in civil GPS signals without major changes required in the space sector (changes in the transmission signal from the satellites) [11] and made available to the public to be able to test new methods as future developments are found.

### B. Sequence of Spoofing

The first 6 scenarios are defined as seen below in table 1, while the scenarios 7 and 8 are refined and extended versions of scenario 6. Additional features of scenario 7 include spoofing signal having synchronized carrier phase cycle signals as seen in figure 1 where both the clean static signal and the spoofed signal are matched perfectly. This reduces the numbers of possible detection algorithms that use differences or change in the carrier phase to detect any types of spoofing. As noted in [12] the spoofing signal injection starts at around 110 seconds into the dataset. Each spoofing scenario is governed by a set of conditions: Spoofing Type, Platform mobility, Power Advantage, Frequency Lock, and Noise Padding. Spoofing types have three possible states: N/A means there is an instantaneous switch from the authentic signal to the spoofed signal without any injection methods, this type of exhibited by scenario 1; Time means there is a time push where the spoofing signal offsets the time of the receiver calculation at a certain power level, scenarios 2, 3, and 5 (along with 7 and 8) display this spoofing type; Position is the spoofing type where the position of the receiver calculation is offset rather than the time, scenarios 4 and 6 display this spoofing type. Platform Mobility is the mobility of the receiver, static refers to a static receiver and dynamic means the receiver can move. Power Advantage refers to the power ratio between the spoofed signal and the authentic signal. Frequency Lock refers to the carrier phase alignments from the spoofer and scenarios 3, 4 and 6 display this feature. Noise padding is either excluded or included in the spoofing signal and refers to the added random noise to “pad” out the spoofing signal and is enabled for scenario 1 only.



| Scenario Designation | Spoofing Type | Platform Mobility | Power Adv. (dB) | Frequency Lock | Noise Padding | Size (GB) |
|---|---|---|---|---|---|---|
| 1: Static Switch | N/A | Static | N/A | Unlocked | Enabled | 43 |
| 2: Static Overpowered Time Push | Time | Static | 10 | Unlocked | Disabled | 42.5 |
| 3: Static Matched-Power Time Push | Time | Static | 1.3 | Locked | Disabled | 42.6 |
| 4: Static Matched-Power Pos. Push | Position | Static | 0.4 | Locked | Disabled | 42.6 |
| 5: Dynamic Overpowered Time Push | Time | Dynamic | 9.9 | Unlocked | Disabled | 38.9 |
| 6: Dynamic Matched-Power Pos. Push | Position | Dynamic | 0.8 | Locked | Disabled | 38.9 |

Table 1 – Scenario Summary [17]

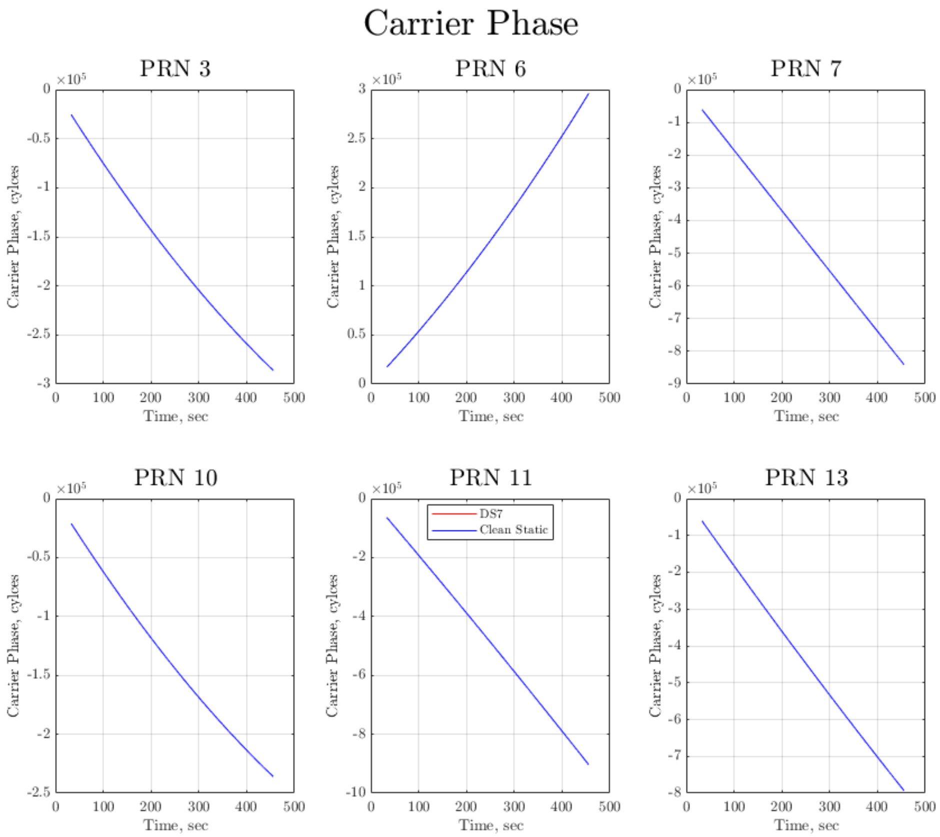


Figure 1 – Carrier Phase

For the purposes of this paper, scenario 7, a refined version of scenario 3 [12] will be used. This scenario is a power-matched time push where the spoofing type is time, the receiver is static, the frequency is locked, the power advantage is 1.3 dB, and the noise padding is enabled. The differences between scenario 3 and scenario 7 are the reduced noise, some error corrections, and a higher quality combination technique, otherwise the two scenarios' dataset are effectively identical. The time sequence for the spoofing and authentic signals are as follows (in seconds):

[0 – 110] – No spoofing present, nearly identical to clean static data

[110 – 130] – Spoofing signal injected for each present GPS signal. The spoofing signal is injected by adding to the in-phase component as the phase slowly increases for the spoofer while the in-phase component stays constant throughout the interval. The amplitude of the spoofing signal is also increased over time. This is done such that the added phasor has a constant amplitude over this interval as the phase of the spoofing signal changes. This method of spoofing allows the spoofer to take over the carrier and code tracking loops without having to align with navigation data bits.

[130 – 150] – The spoofing phasor will remain constant for this time interval (as it was at the end of the previous time interval). The measurements (carrier phase, C/N0, pseudorange, and doppler frequency) fairly close to the static data with slight differences.

[150 – 400] – Each spoofing signal's phase is constant but the amplitude changes linearly over the time interval and the code phase changes at a rate of 1.2 m/s which causes a change in the measure clock offset in the receiver. As noted before that DS7 is a frequency locked so the doppler frequency will remain the same. Also noted is that the carrier phase of the spoofing signal is lined up almost perfectly to the authentic signal.

[400 – 468] – The phase of the spoofing signals is held constant as well as the amplitude, however the code phase of the spoofing signal still increases at about 1.2 m/s. By the end of the sequence, the receiver clock bias will have been offset which concludes the spoofing attack.

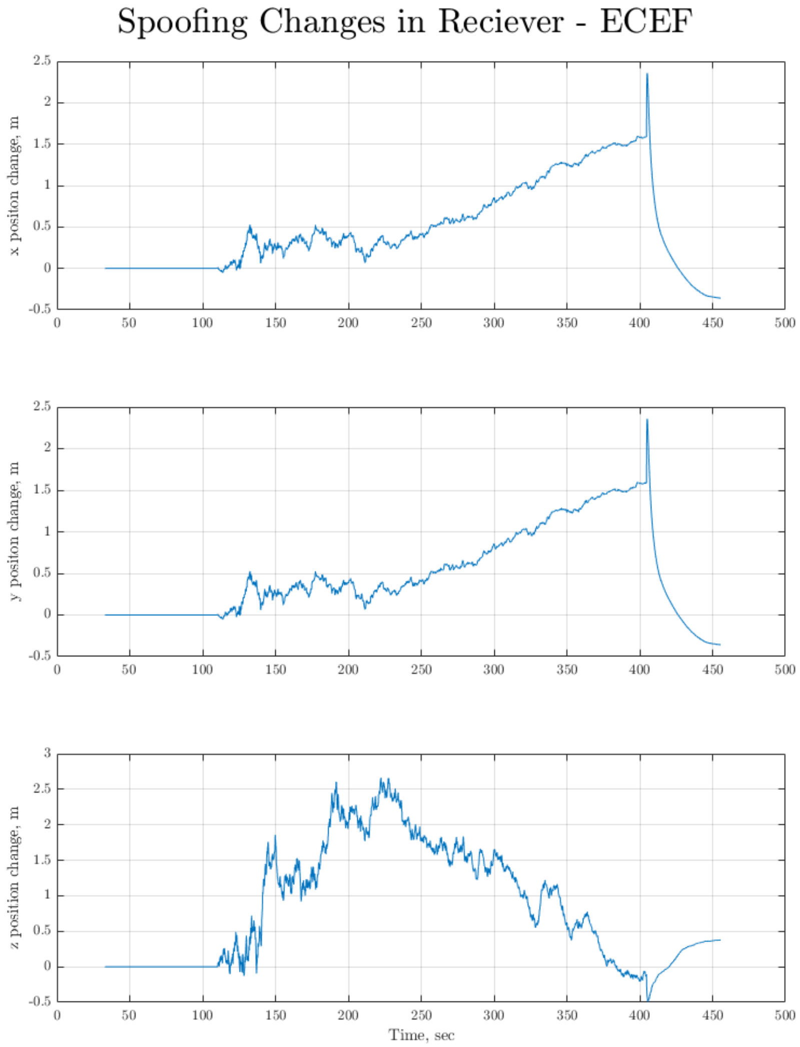


Figure 2 – Navigation Solution in ECEF

## V. Detailed Review of Two Detection Algorithms

### A. Detection via Pseudorange Differences

The main method that will be detailed and used on the TEXBAT data in this paper is the method present by [11] which makes use of two different calculations of the navigation solution, the normal least squares method and the introduced pseudorange double difference method. This method obtains a navigation solution for the position of the receiver using pseudorange differences rather than the typical least squares method that is used in most applications including the TEXBAT dataset navigation solutions. The algorithm requires the navigation solution from both the Least Squares method and the Pseudorange method that will be compared against each other to create the spoofing condition. Though the TEXBAT dataset already includes the navigation solution using the least squares method the least square will be calculated as explained in the [11] for the sake of following their methodology.

To begin the algorithm used in this method, a few equations and variables must first be defined. The first variable is the geometric distance between the receiver, p, and a satellite, i:

$$R\left(\mathbf{X}^i, \mathbf{X}_p\right) = \sqrt{\left(x^i - x_p\right)^2 + \left(y^i - y_p\right)^2 + \left(z^i - z_p\right)^2} \tag{1}$$

Where $\mathbf{X}^i$ is the satellite position in the Earth-centered Earth-Fixed (ECEF) frame. Next, the pseudorange is defined as:

$$\rho_k^i = R\left(\mathbf{X}_k^i, \mathbf{X}_{p,k}\right) + \left(\delta t_{r,k} - \delta t_{s,k}^i\right) \times c + \left(\delta t_{ion,k}^i + \delta t_{trop,k}^i\right) \times c + \varepsilon_k^i \tag{2}$$

$$\rho^i_{k+1} = R\left(\mathbf{X}^i_{k+1}, \mathbf{X}_{p,k+1}\right) + \left(\delta t_{r,k+1} - \delta t^i_{s,k+1}\right) \times c + \left(\delta t^i_{ion,k+1} + \delta t^i_{trop,k+1}\right) \times c + \varepsilon^i_{k+1} \tag{3}$$

Where $\delta t_r$ and $\delta t_s^i$ are the receiver and satellite clock biases respectively and the $\delta t_{ion}^i$ and $\delta t_{trop}^i$ are the ionospheric and tropospheric delays respectively. Once these variables have been defined. The single difference can be taken between the first timestep, k, and the next timestep, k+1, for the entire series for each satellite:

$$\Delta\rho^i_{k+1,k} = \rho^i_{k+1} - \rho^i_k = R\left(\mathbf{X}^i_{k+1}, \mathbf{X}_{p,k+1}\right) - R\left(\mathbf{X}^i_k, \mathbf{X}_{p,k}\right) + \left(\delta t_{r,k+1} - \delta t_{r,k}\right) \times c + \left(\varepsilon c^i_{k+1} - \varepsilon c^i_k\right) \tag{4}$$

$$\Delta\rho^j_{k+1,k} = \rho^j_{k+1} - \rho^j_k = R\left(\mathbf{X}^j_{k+1}, \mathbf{X}_{p,k+1}\right) - R\left(\mathbf{X}^j_k, \mathbf{X}_{p,k}\right) + \left(\delta t_{r,k+1} - \delta t_{r,k}\right) \times c + \left(\varepsilon c^j_{k+1} - \varepsilon c^j_k\right) \tag{5}$$

For a single pseudorange the double difference can be taken between one satellite and another as seen below:

$$\begin{aligned} \Delta\rho^{ij}_{k+1,k} &= \Delta\rho^i_{k+1,k} - \Delta\rho^j_{k+1,k} \\ &= \left[R\left(\mathbf{X}^i_{k+1}, \mathbf{X}_{p,k+1}\right) - R\left(\mathbf{X}^i_k, \mathbf{X}_{p,k}\right)\right] - \left[R\left(\mathbf{X}^j_{k+1}, \mathbf{X}_{p,k+1}\right) - R\left(\mathbf{X}^j_k, \mathbf{X}_{p,k}\right)\right] + \left(\varepsilon c^i_{k+1} - \varepsilon c^i_k\right) - \left(\varepsilon c^j_{k+1} - \varepsilon c^j_k\right) \end{aligned} \tag{6}$$

In this example, the range between the satellite and the receiver at time k is known (iterated for at each time step) thus, when taking the Taylor expansion for the equation, the expansion is only needed for the range between the satellite and receiver at time k+1. Thus the expansion around the position at time k results in:

$$R\left(\mathbf{X}^i_{k+1}, \mathbf{X}_{p,k+1}\right) \approx R\left(\mathbf{X}^i_{k+1}, \mathbf{X}_k\right) + u^i_{x,k+1,k}\Delta x + u^i_{y,k+1,k}\Delta y + u^i_{z,k+1,k}\Delta z \tag{7}$$

Where the $u^i_{v,k+1,k}$ are the line of sight unit vectors from the receiver to the satellite which are calculated by taking the difference between the receiver position and the satellite position and dividing it by its norm. Next, the Taylor expansion of the range at time step k+1 about the position at time k is taken:

$$\begin{aligned} &R\left(\mathbf{X}^i_{k+1}, \mathbf{X}_k\right) - R\left(\mathbf{X}^j_{k+1}, \mathbf{X}_k\right) + u^i_{x,k+1,k}\left(x_{k+1,SDA} - x_k\right) + u^i_{y,k+1,k}\left(y_{k+1,SDA} - y_k\right) + u^i_{z,k+1,k}\left(z_{k+1,SDA} - z_k\right) \\ &- u^j_{x,k+1,k}\left(x_{k+1,SDA} - x_k\right) - u^j_{y,k+1,k}\left(y_{k+1,SDA} - y_k\right) - u^j_{z,k+1,k}\left(z_{k+1,SDA} - z_k\right) \\ &= \Delta\rho^i_{k+1,k} - \Delta\rho^j_{k+1,k} + \left[R\left(\mathbf{X}^i_k, \mathbf{X}_{p,k}\right) - R\left(\mathbf{X}^j_k, \mathbf{X}_{p,k}\right)\right] - \left(\varepsilon c^i_{k+1} - \varepsilon c^i_k\right) + \left(\varepsilon c^j_{k+1} - \varepsilon c^j_k\right) \end{aligned} \tag{8}$$

Since these equations are for the difference between only two satellite signals, the equations must be changed to accommodate for the other n-1 satellites which results in n-1 equations in the form:

$$\mathbf{M} \begin{bmatrix} x_{k+1,SDA} - x_k \\ y_{k+1,SDA} - y_k \\ z_{k+1,SDA} - z_k \end{bmatrix} = \mathbf{L} \tag{9}$$

Where M is the difference between the unit vector of the first satellite and the unit vectors of the other satellites which results in a (n-1)x3 matrix. L is the difference between the pseudorange single-difference for the first satellite summed with the range at time k subtracted by the ranges at time k+1 along with the error which results in a (n-1)x1 matrix. These matrices are seen below:

$$\begin{aligned} \mathbf{M} &= \mathbf{M}_1 - \mathbf{M}_2 \\ &= \begin{bmatrix} u^1_{x,k+1,k} & u^1_{y,k+1,k} & u^1_{z,k+1,k} \\ u^1_{x,k+1,k} & u^1_{y,k+1,k} & u^1_{z,k+1,k} \\ \vdots & \vdots & \vdots \\ u^1_{x,k+1,k} & u^1_{y,k+1,k} & u^1_{z,k+1,k} \end{bmatrix} - \begin{bmatrix} u^2_{x,k+1,k} & u^2_{y,k+1,k} & u^2_{z,k+1,k} \\ u^3_{x,k+1,k} & u^3_{y,k+1,k} & u^3_{z,k+1,k} \\ \vdots & \vdots & \vdots \\ u^n_{x,k+1,k} & u^n_{y,k+1,k} & u^n_{z,k+1,k} \end{bmatrix} \end{aligned}$$

$$\begin{aligned}\mathbf{L} &= \mathbf{L}_1 - \mathbf{L}_2 \\ &= \begin{bmatrix} \Delta\rho^1_{k+1,k} + R\left(\mathbf{X}^1_k, \mathbf{X}_{p,k}\right) - R\left(\mathbf{X}^1_{k+1}, \mathbf{X}_k\right) - \left(\varepsilon c^1_{k+1} - \varepsilon c^1_k\right) \\ \Delta\rho^1_{k+1,k} + R\left(\mathbf{X}^1_k, \mathbf{X}_{p,k}\right) - R\left(\mathbf{X}^1_{k+1}, \mathbf{X}_k\right) - \left(\varepsilon c^1_{k+1} - \varepsilon c^1_k\right) \\ \vdots \\ \Delta\rho^1_{k+1,k} + R\left(\mathbf{X}^1_k, \mathbf{X}_{p,k}\right) - R\left(\mathbf{X}^1_{k+1}, \mathbf{X}_k\right) - \left(\varepsilon c^1_{k+1} - \varepsilon c^1_k\right) \end{bmatrix} \\ &- \begin{bmatrix} \Delta\rho^2_{k+1,k} + R\left(\mathbf{X}^2_k, \mathbf{X}_{p,k}\right) - R\left(\mathbf{X}^2_{k+1}, \mathbf{X}_k\right) - \left(\varepsilon c^2_{k+1} - \varepsilon c^2_k\right) \\ \Delta\rho^3_{k+1,k} + R\left(\mathbf{X}^3_k, \mathbf{X}_{p,k}\right) - R\left(\mathbf{X}^3_{k+1}, \mathbf{X}_k\right) - \left(\varepsilon c^3_{k+1} - \varepsilon c^3_k\right) \\ \vdots \\ \Delta\rho^n_{k+1,k} + R\left(\mathbf{X}^n_k, \mathbf{X}_{p,k}\right) - R\left(\mathbf{X}^n_{k+1}, \mathbf{X}_k\right) - \left(\varepsilon c^n_{k+1} - \varepsilon c^n_k\right) \end{bmatrix}\end{aligned}$$

To solve this equation, an additional row will be added to both M and L. Where the added row for $M_1$ and $M_2$ is the unit vector for the first satellite and the added row for $L_1$ and $L_2$ is the similar pseudorange difference and other parameters for the first satellite. This results in a nx3 matrix for M and nx1 matrix for L.

$$\left\{\begin{bmatrix} \mathbf{0}^{\mathbf{M}}_{1\times 3} \\ \mathbf{M}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{0}^{\mathbf{M}}_{1\times 3} \\ \mathbf{M}_2 \end{bmatrix}\right\} \begin{bmatrix} x_{k+1,SDA} - x_k \\ y_{k+1,SDA} - y_k \\ z_{k+1,SDA} - z_k \end{bmatrix} = \left\{\begin{bmatrix} \mathbf{0}^{\mathbf{L}}_{1\times 1} \\ \mathbf{L}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{0}^{\mathbf{L}}_{1\times 1} \\ \mathbf{L}_2 \end{bmatrix}\right\} \tag{10}$$

$$\mathbf{0}^{\mathbf{M}}_{1\times 3} = \begin{bmatrix} u^1_{x,k+1,k} & u^1_{y,k+1,k} & u^1_{z,k+1,k} \end{bmatrix}$$

$$\mathbf{0}^{\mathbf{L}}_{1\times 1} = \left[\Delta\rho^1_{k+1,k} + R\left(\mathbf{X}^1_k, \mathbf{X}_{p,k}\right) - R\left(\mathbf{X}^1_{k+1}, \mathbf{X}_k\right) - \left(\varepsilon c^1_{k+1} - \varepsilon c^1_k\right)\right]$$

Next, the following relationships are also defined and known to be true:

$$\begin{bmatrix} \mathbf{0}^{\mathbf{M}}_{1\times 3} \\ \mathbf{M}_1 \end{bmatrix} \begin{bmatrix} x_{k+1,SDA} - x_k \\ y_{k+1,SDA} - y_k \\ z_{k+1,SDA} - z_k \end{bmatrix} = \begin{bmatrix} \rho^1_{k+1} - R\left(\mathbf{X}^1_{k+1}, \mathbf{X}_k\right) - \delta t_{r,k+1} \times c - \varepsilon c^1_{k+1} \\ \rho^1_{k+1} - R\left(\mathbf{X}^1_{k+1}, \mathbf{X}_k\right) - \delta t_{r,k+1} \times c - \varepsilon c^1_{k+1} \\ \vdots \\ \rho^1_{k+1} - R\left(\mathbf{X}^1_{k+1}, \mathbf{X}_k\right) - \delta t_{r,k+1} \times c - \varepsilon c^1_{k+1} \end{bmatrix} \tag{11}$$

$$\left[\rho^i_k - R\left(\mathbf{X}^i_k, \mathbf{X}_{p,k}\right) - \varepsilon c^i_k\right] - \left[\rho^j_k - R\left(\mathbf{X}^j_k, \mathbf{X}_{p,k}\right) - \varepsilon c^j_k\right] = 0 \tag{12}$$

Combining equations (10), (11), and (12):

$$\begin{bmatrix} \mathbf{0}^{\mathbf{M}}_{1\times 3} \\ \mathbf{M}_2 \end{bmatrix} \begin{bmatrix} x_{k+1,SDA} \\ y_{k+1,SDA} \\ z_{k+1,SDA} \end{bmatrix} = \begin{bmatrix} \mathbf{0}^{\mathbf{M}}_{1\times 3} \\ \mathbf{M}_2 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} + \begin{bmatrix} \rho^1_{k+1} - R\left(\mathbf{X}^1_{k+1}, \mathbf{X}_k\right) \\ \rho^2_{k+1} - R\left(\mathbf{X}^2_{k+1}, \mathbf{X}_k\right) \\ \vdots \\ \rho^n_{k+1} - R\left(\mathbf{X}^n_{k+1}, \mathbf{X}_k\right) \end{bmatrix} - \begin{bmatrix} \delta t_{r,k+1} \times c \\ \delta t_{r,k+1} \times c \\ \vdots \\ \delta t_{r,k+1} \times c \end{bmatrix} - \begin{bmatrix} \varepsilon c^1_{k+1} \\ \varepsilon c^2_{k+1} \\ \vdots \\ \varepsilon c^n_{k+1} \end{bmatrix} \tag{13}$$

This is the final equation for the pseudorange method that will be iterated through similar to the Least squares method to find the position of the receiver at time k+1 with the input of the position at time k for the pseudorange difference method. To compare this answer, the least square method can also be defined with similar matrices:

$$\rho^i_{k+1} = R\left(\mathbf{X}^i_{k+1}, \mathbf{X}_{p,k+1}\right) + \delta t_{r,k+1} \times c + \varepsilon c^i_{k+1} \tag{14}$$

$$\mathbf{G}_e \begin{bmatrix} x_{k+1,LS} \\ y_{k+1,LS} \\ z_{k+1,LS} \\ \delta t_{r,k+1} \times c \end{bmatrix} = \mathbf{G}_e \begin{bmatrix} x_k \\ y_k \\ z_k \\ 0 \end{bmatrix} + \begin{bmatrix} \rho^1_{k+1} - R\left(\mathbf{X}^1_{k+1}, \mathbf{X}_k\right) \\ \rho^2_{k+1} - R\left(\mathbf{X}^2_{k+1}, \mathbf{X}_k\right) \\ \vdots \\ \rho^n_{k+1} - R\left(\mathbf{X}^n_{k+1}, \mathbf{X}_k\right) \end{bmatrix} - \begin{bmatrix} \varepsilon c^1_{k+1} \\ \varepsilon c^2_{k+1} \\ \vdots \\ \varepsilon c^n_{k+1} \end{bmatrix} \tag{15}$$

Where $G_e$ is the modified $M_2$ matrix from equation (10) along with a column of ones to make the $G_e$ have dimensions nx4. This equation can be solved iteratively as well which results in the navigation solution using the least squares method.

Below are the results from both of the methods described above. The metric plotted is the deviation in the x position in ECEF of both the pseudorange difference and least squares method over the time frame described in [12].

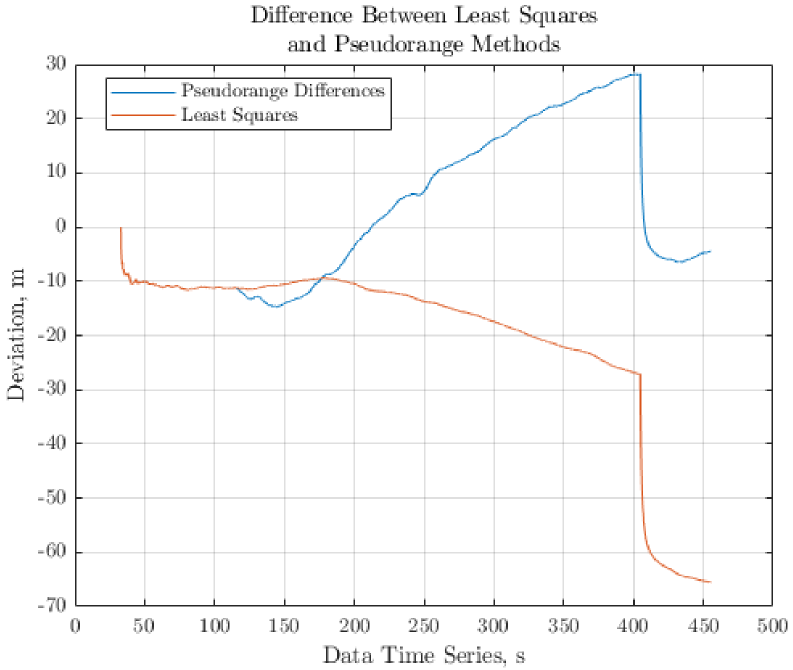


Figure 4 – Deviation of Least Squares and Pseudorange Methods

As seen in the figure above, it is clearly illustrated that there is spoofing the incoming signal. In [11], Ke Liu et al, specify that the condition for spoofing is separation in the deviation between the least squares and pseudorange difference methods. This can clearly be seen between 100 and 120 seconds where [12] states that the spoofing signal is injected into the authentic signal. Ke Liu et all also explains that this difference is caused by the receiver clock bias which is reinforced by the fact that the DS7 dataset moves the receiver clock bias through the spoofing attack [12] which is illustrated in figure 5. The processing required for this section made use of the algorithms presented in [11], spoofing data from [12] and [13], and the ephemeris data for the satellite positions from the International GNSS Service (IGS) [15]. A toolbox for MATLAB was also used to extract the data from [15] and process into necessary parameters (ie. satellite positions, ephemeris information for delay calculations etc.). All of these calculations are shown in the appendix, showing the code required to generate the results from above and the implementation of the algorithms proposed in [11].

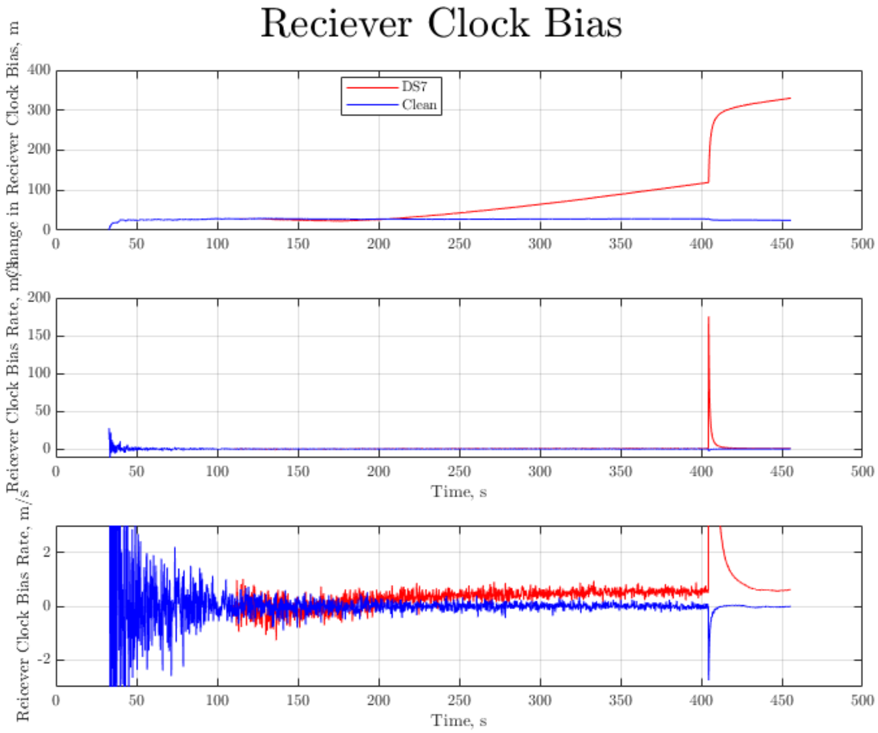


Figure 5 – Receiver Clock Bias

## VI. Conclusion

This paper introduced several methods that are used to spoof a GPS signal and described the possible dangers that may occur without the use of spoofing detection in certain systems as well as actual real-world cases where these dangers came into fruition. Several methods were also introduced that use mathematical methods and data processing to detect whether a signal is being spoofed or if the signal is authentic. These ranged from very complex solutions such as using antenna arrays to very simple data processing methods including statistical methods.

Using real dataset from the  TEXBAT dataset and the broadcast ephemeris data from IGS was an important step in implementing the algorithm that was detailed in the last section. Implementing the algorithm on real data illustrated the effectiveness of the algorithm. The algorithm that was implemented was simple but robust as seen in the results of the implementation, showing that the method actually worked and gave very clear results that did not lie in the margin of error.

## References

[1] O'Hanlon, B. W., Psiaki, M. L., Bhatti, J. A., Shepard, D. P., and Humphreys, T. E., "Real-time GPS spoofing detection via correlation of encrypted signals," Navigation, vol. 60, 2013, pp. 267–278.

[2] Kerns, A. J., Shepard, D. P., Bhatti, J. A., and Humphreys, T. E., "Unmanned Aircraft Capture and control via GPS spoofing," Journal of Field Robotics, vol. 31, 2014, pp. 617–636.

[3] Akos, D. M., "Who's afraid of the spoofer? GPS/GNSS spoofing detection via Automatic Gain Control (AGC)," Navigation, vol. 59, 2012, pp. 281–290.

[4] Zhang, H., Peng, S., Liu, L., Su, S., and Cao, Y., "Review on GPS spoofing-based time Synchronisation attack on Power System," IET Generation, Transmission & Distribution, vol. 14, 2020, pp. 4301–4309.

[5] Ahmad, M., Farid, M. A., Ahmed, S., Saeed, K., Asharf, M., and Akhtar, U., "Impact and detection of GPS spoofing and countermeasures against spoofing," 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2019.

[6] Chapman, A., "GPS Spoofing," Tufts Available: https://sites.tufts.edu/eeseniordesignhandbook/files/2017/05/Red_Chapman.pdf.

[7] Troglia Gamba, M., Truong, M. D., Motella, B., Falletti, E., and Ta, T. H., "Hypothesis testing methods to detect spoofing attacks: A test against the TEXBAT datasets," GPS Solutions, vol. 21, 2016, pp. 577–589.

[8] T.E. Humphreys, J.A. Bhatti, D.P. Shepard, and K.D. Wesson, "The Texas Spoofing Test Battery: Toward a Standard for Evaluating GPS Signal Authentication Techniques," Proc. ION GNSS, Nashville, TN, 2012.

[9] Jafarnia-Jahromi, A., Broumandan, A., Nielsen, J., and Lachapelle, G., "GPS vulnerability to spoofing threats and a review of antispoofing techniques," International Journal of Navigation and Observation, vol. 2012, 2012, pp. 1–16.

[10] Zhang, X., Li, H., Yang, C., and Lu, M., "Signal Quality monitoring-based spoofing detection method for global navigation satellite system vector tracking structure," IET Radar, Sonar & Navigation, vol. 14, 2020, pp. 944–953.

[11] Liu, K., Wu, W., Wu, Z., He, L., and Tang, K., "Spoofing detection algorithm based on pseudorange differences," Sensors, vol. 18, 2018, p. 3197.

[12] Humphreys, T., "TEXBAT Data Sets 7 and 8," The University of Texas at Austin, Mar. 2016.

[13] Radionavigation Laboratory Available: https://rnl.ae.utexas.edu/index.php?option=com_content&view=article&id=289%3Atexas-spoofing-test-battery-texbat&catid=50&Itemid=27.

[14] Spanikp, "SPANIKP/GNSS-toolbox: Useful functions to handle processing of GNSS data," GitHub Available: https://github.com/spanikp/GNSS-toolbox.

[15] "Daily Broadcast Ephemeris Data," International GNSS Service Available: https://igs.org/data/#:~:text=The%20IGS%20creates%20daily%20broadcast,ephemeris%20messages%20for%20each%20day.

[16] Gamba, M. T., Motella, B., and Pini, M., "Statistical test applied to detect distortions of GNSS signals," 2013 International Conference on Localization and GNSS (ICL-GNSS), 2013.

[17] Humphreys, T., Bhatti, J., Daniel Shepard, and Wesson, K., “The Texas Spoofing Test Battery: Toward a Standard for Evaluating GPS Signal Authentication Techniques,” The University of Texas at Austin, Austin, TX.

## Appendix

```
%% Spoofing Detection (Detection Algorithms)

addpath(genpath(fileparts(which('pathfile.m'))))
interr = 'latex';
% interr = 'none';
set(groot,'defaulttextinterpreter',interr);
set(groot, 'defaultAxesTickLabelInterpreter',interr);
set(groot, 'defaultLegendInterpreter',interr);
c = 299792458;
L1freq = 1575.42e6;
Re = 6378137;
if ~exist('clean')
    spoofingdetection
end
if ~exist('brdcind')
    brdc = loadRINEXNavigation('G','data','brdc2580.12n');
end
%%
[~,brdcind] = ismember(prnnums,brdc.sat);
GPStime = ORT - dR/c; GPStime = mod(GPStime,24*3600);
GPStime2 = [ORT_weeksec(:,1), ORT_weeksec(:,2)-dR/c];
wgs = wgs84Ellipsoid;
UTECEF = [mean(x_cs),mean(y_cs),mean(z_cs)];
[UTGEO(1),UTGEO(2),UTGEO(3)] = ECEF2GEODETIC(UTECEF);
alpha = [0.1490D-07  0.2235D-07 -0.1192D-06 -0.1192D-06];
beta = [0.1085D+06  0.6554D+05 -0.1966D+06 -0.1311D+06];
timeind = 7;
for i = 1:length(prnnums)
    % sat position from ephemeris
    eph = brdc.eph{brdcind(i)}(:,timeind);
    a = eph(22);
    ecc = eph(20);
    TGD(i) = eph(37);
    svbias(i) = eph(12);
    [posT{i},aux{i}] = getSatPosGPS(ORT_weeksec,eph);
    Ek = aux{i}(:,timeind);
    posR{i} =
rotPos(posT{i}(:,1),posT{i}(:,2),posT{i}(:,3),prange(:,i)/c);
    LLA{i} = ecef2lla(posR{i},'WGS84');
%      [AER{i}(:,1),AER{i}(:,2),~] =
ecef2aer(posR{i}(:,1),posR{i}(:,2),posR{i}(:,3),UTGEO(1),UTGEO(2),UTGE
O(3),wgs);
    iono(:,i) =
klobmodel(UTGEO(1),UTGEO(2),az_cs(:,i),el_cs(:,i),GPStime,alpha,beta);
    trop(:,i) =
saastamoinen(.44,300,1011,UTGEO(1),UTGEO(2),UTGEO(3),az_cs(:,i),el_cs(
:,i));
```

```
    trel(:,i) =  getSVRelativityClockCorrection('G',a,ecc,Ek);
end

n = length(prnnums);
prange_corr = prange - iono - trop - c*(TGD - svbias - trel);
prange_corr_cs = prange_cs - iono - trop - c*(TGD - svbias - trel);

for i = 1:length(prnnums)
    prange_sdiff(:,i) = diff(prange_corr(:,i));
end

LS = navsolLS([x_cs(1)-40;y_cs(1)-
64;z_cs(1)+20],prange_corr_cs,posR,GPStime,prnnums);
start = [LS(1,1),LS(1,2),LS(1,2)];
est = navsolPD(GPStime,prnnums,posR,prange_corr,start,dR,"PD");
est_cs = navsolPD(GPStime,prnnums,posR,prange_corr_cs,start,dR,'PD');

plot(plottime,(est(:,1)-LS(:,1)))
hold on
plot(plottime,(est_cs(:,1)-LS(:,1)))
title(["Difference Between Least Squares" ,...
    "and Pseudorange Methods"])
legend('Pseudorange Differences','Least Squares','Location',"best"  )
grid on
xlabel('Data Time Series, s')
ylabel('Deviation, m')

% Pseudorange Differencing method for solving navigation solution at all
time steps
function XHAT =
navsolPD(GPStime,prnnums,posR,prange_corr,start,dR,option)
n = length(prnnums);
xk = start;
XHAT(1,:) = xk;
stop = length(GPStime)-1;
for i = 1:length(prnnums)
    prange_sdiff(:,i) = diff(prange_corr(:,i));
end
for t = 1:stop
    delta = 1;
%         while delta > .05
    for i = 1:6
        for i = 1:length(prnnums)
            rkp(i,1) = norm(xk - posR{i}(t,:));
            rkp2(i,1) = norm(xk - posR{i}(t+1,:));
            ukp{t,i} = (xk - posR{i}(t+1,:))/rkp(i);
            M2(i,:) = ukp{t,i};
            pcomp = rkp(i) + dR(t);
            yk(i,1) = prange_corr(t,i) - pcomp;
        end
        G = [M2,ones(n,1)];
        if isequal(string(option),"LS")
            xkpxk = inv(G'*G)*G'*yk;
        elseif isequal(string(option),"PD")
            xkpxk = inv(M2.'*M2)* M2.' * yk;
        end
        xk = xkpxk(1:3)' + xk;
        delta = norm(xkpxk(1:3));
    end
    XHAT(t+1,:) = xk;
end
end

function estimate = navsolLS(gECEF,prange_corr,sat_pos,GPStime,prnset)

y = cell(length(GPStime),1);
H = cell(length(GPStime),1);
delR = cell(length(GPStime),1);
XHAT = cell(length(GPStime),1);
Xstar = cell(length(GPStime),1);

% iterative process for calculating navigation solution for all time
steps
for p = 1:length(GPStime)
    Xstar{p} = [gECEF;1];
end
for i = 1:length(GPStime) % for each time step
    delta = 100;
    %%%% Iterated Part
    k = 1;
    while delta > 0.05
        if k > 1
            old = delR{i};
        end
        for j = 1:length(prnset)
            range_comp = norm(sat_pos{j}(i,:) - Xstar{i}(1:3)');
            prange_comp = range_comp+Xstar{i}(4);
            y{i}(j,:) = prange_corr(i,j) - prange_comp; %1
            H{i}(j,:) = [(Xstar{i}(1:3)'-
sat_pos{j}(i,:))/range_comp,1]; %2
        end
        delR{i} = (H{i}'*H{i})\H{i}'*y{i};
        XHAT{i} = Xstar{i}+delR{i};
        Xstar{i} = XHAT{i};
        estimate(i,:) = XHAT{i}';
        if k > 1
            delta = norm(delR{i}(1:3));
        end
        k = k+1;
    end
end
```