TIME DELAY OF ARRIVAL BASED ORBIT DETERMINATION OF GEOSYNCHRONOUS SIGNALS OF OPPORTUNITY

by

Siddharth S. Subramanyam

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Aeronautics and Astronautics

School of Aeronautics and Astronautics
West Lafayette, Indiana
May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. James L. Garrison, Chair

School of Aeronautics and Astronautics

Dr. Carolin Frueh

School of Aeronautics and Astronautics

Dr. Rashmi Shah

NASA Jet Propulsion Laboratory

Approved by:

Dr. Gregory Blaisdel

To my friends and family

ACKNOWLEDGMENTS

I would like to thank begin by thanking all the members of my lab whom I've had the pleasure to work with: Harris, Archana, Seho, Eric, Priyankar, Abi, Elisa, Kevin, Justin, and an extra thanks to Ben, whose contributions would be longer than this thesis if I had to list them out. In addition to being some of the smartest people I know, they've also made me feel extremely welcome.

Additionally, I'd like to thank my friend Nick, who helped me do a day-long data collection by driving me around Lafayette. For the main distributed experiment, a huge thanks goes out to all those that graciously hosted a receiver station: my friend Yasmine, the faculty of Mississippi State and CU Boulder, and Dr. Shah and her family. This experiment would not have been possible without you, and I'm still in awe as to how smoothly it all went. I would like to thank my girlfriend, Shree, for her endless support and assistance with editing my thesis for grammar.

Lastly, I would like to thank my family for being the main reason why I've come this far. For helping me set up the VPN, which was paramount to the success of coordinating six receiver stations spread out across the continental US, and for always taking a keen interest in my research, I thank my father, Subra. I would like to thank my mother, Hema, for all the intangibles that kept me motivated and able to complete this work. Finally, a huge thanks goes to my sister, Twisha, who helped numerous times with the initial rover experiments, which proved that this concept was feasible in the first place.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

c	speed of light (m/s)
dt_r	receiver clock offset from GNSS time (s)
$dT^{(s)}$	satellite clock offset from GNSS time (s)
D	delay observable (m)
ΔD	difference between measured and computed delay (m)
ϵ_P	models measurement noise in pseudorange equation (m)
ν	models measurement noise in linearization (m)
$\mathbf{H}$	design matrix, system of linearized delays
$I^s_{r,\mathrm{i}}$	ionospheric delay (m)
$J(x)$	cost function (m^2)
Ω_G	longitude of transmitter in GEO
σ	standard deviation for TDOA measurement or $\vec{R}_T$ calculation
$P^{(s)}_{r,\mathrm{i}}$	pseudorange measurement (m).
$\vec{R}_T$	vector position of transmitter (m)
$\vec{R}_0$	vector position of primary receiver (m)
$\vec{R}_\mathrm{i}$	vector position of receiver i (m)
$\rho^{(s)}_r$	true range from the satellite's to the receiver's antenna (m)
$T^{(s)}_r$	tropospheric delay (m)
T_{RIC}	transformation matrix to RIC frame
τ	time for signal to travel a set distance (s)
$\Delta\tau$	time delay for the same signal reaching separate receivers (m)
$\hat{u}^T_\mathrm{i}$	unit vector between $\vec{R}_\mathrm{i}$ and $\vec{R}_T$ (m)
$\mathbf{y}$	system of ΔD measurements (m)
x	substitution of $\Delta\vec{R}_T$ for linear system (m)

ABBREVIATIONS

ADC	Analog to Digital Converter
DOA	Difference of Arrival
DOP	Dilution of Precision
ECEF	Earth-Centric, Earth-Fixed
ECI	Earth-Centric Inertial
EHT	Event Horizon Telescope
GEO	Geostationary Orbit
GLONASS	Russian GNSS constellation
GPS	Global Positioning System
GPSDO	GPS Disciplined Oscillator
LEO	Low Earth Orbit
LHCP	Left-Hand Circular Polarized
LO	Local Oscillator
NMEA	National Marine Electronics Association
NUC	Mini PC from Intel
OD	Orbit Determination
POD	Precise Orbit Determination
PPS	Pulse Per Second
PVT	Position, Velocity, Time
QPSK	Quadrature Phase-shift-keying
RIC	Radical/In-track/Cross-track
SDARS	Satellite Digital Audio Radio Service
SDR	Software Defined Radio
SSH	Sea Surface height
SWE	Snow Water Equivalent
SoOp	Signals of Opportunity
TCXO	Temperature Compensated Crystal Oscillator
TDM	Time Division Multiplexed

TDOA	Time Difference of Arrival
TDRS	Tracking and Data Relay Satellite
TLE	Two-Line Elements
TOF	Time of Flight
UHD	USRP Hardware Driver
USRP	Software Defined Radio made by Ettus
VLBI	Very Long Baseline Interferometry
VPN	Virtual Private Network

ABSTRACT

Earth science observations are crucial for our understanding of the Earth's climate, water cycle, land, and atmosphere. Signals of Opportunity (SoOp) has recently emerged as an innovative method for producing these observations. SoOp reuses existing satellite communication signals for science measurements. A key factor in the accuracy of SoOp measurements, is the accuracy with which the transmitting satellite's position can be determined. This thesis developed a distributed network of receivers, which performed time delay of arrival (TDOA) measurements, to solve for the position of a transmitting satellite, using their existing signals. These results were used to characterize the sensitivity of the calculated satellite position, to the TDOA measurement error.

1. INTRODUCTION

1.1 Motivation

Signals of Opportunity (SoOp) reflectometry is a method of microwave remote sensing in bands allocated to Space to Earth communications, which utilizes existing, non-cooperative, satellite transmissions [[1]]. SoOp techniques have evolved, with new and improved Earth science observations of the land, ocean, and cryosphere. Some examples of this are snow water equivalent (SWE), which uses a differential phase measurement in P-band, and sea surface height (SSH), which uses wideband (>500MHz) signals. These measurements, however, require accurate knowledge of the source transmitter position with meter-level accuracy. Current orbit determination techniques cannot achieve this level of precision without the addition of specialized hardware on the transmitting source. By definition, SoOp does not rely on modifications to the source hardware of the signal transmitter. Very Long Baseline Interferometry (VLBI) techniques offer a method to apply localization to these transmitters, as exhibited commonly in the radio astronomy community.

Orbit determination serves as a critical piece of data for various satellite applications. For example, many remote sensing satellites, in addition to those used in SoOp, rely on accurate orbit determination in order to properly characterize absolute and relative distances to the earth. Ocean altimetry [[2]] is such a case where remote sensing can be used to find the relative distance between the satellite and the ocean at an instant [[3]]. However, without precise OD, these measurements will be meaningless, as the orbit and location of the satellite are unknown.

1.2 Traditional Orbit Determination Methods

Classically, orbit determination is done through a series of ranging measurements. These ranging measurements can be obtained through various means, such as ground laser ranging [[4]], onboard radar altimeters, or optical right ascension and declination measurements. A common approach involves some form of active sensing by pointing these sensors toward the desired object, bouncing an emitted signal off the object in question, and obtaining a delay

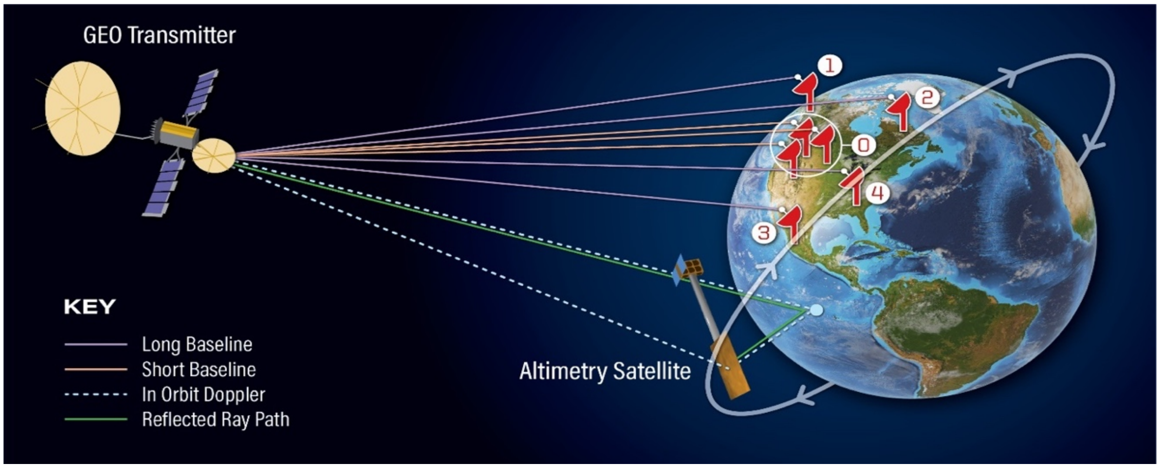

Figure 1.1. SoOp Orbit Determination

$(\Delta)\tau$ in the time it takes said signal to return. Using the physical relationship that these waves travel at the speed of light (c), we can calculate a pseudorange (ρ). Optical orbit determination is another common method, in which a telescope will attempt to capture a spacecraft within its field of view over a period of time [[5]]. The telescope's right ascensions and declination are then used to obtain an orbit via Gauss's method. Additionally, ranging data and either least squares or Kalman filtering can be used to improve the estimation of the orbit.[[6]]. Another common method, especially for satellites in LEO, is to simply use GPS. This, however, is limited in the region in which it can be used as well as the accuracy available to commercial satellites travelling at high speeds in LEO [[7]].

While these methods have been refined over time and are useful for tracking all manner of debris, they suffer from issues of sensor tasking when it comes to tracking a large number of satellites. These methods are also susceptible to tracking error, especially over great distances, such as for objects in GEO [[8]]. New developments in traditional line of orbit determination typically involve enhancing pointing accuracy or making trade-offs by using waves of different frequencies [[9]]. Similarly, with optical observations, there can be errors in actively tracking or centering the object in question within the field of view.

1.3 Pseudorange Observables

Pseudorange (ρ) is a scalar quantity that is calculated from our observed $\Delta\tau$. *Haines, et al.* [[10]] showed that we can formulate an inverse-GPS type problem by incorporating TDRS's Ku-Band signal to generate a pseudorange. However, like ranging measurements, this requires a two-way signal with a *cooperative source.* In GPS, pseudorange is extracted through correlation of a known code. This allows the receiver to acquire the time of signal transmission to calculate the delay ($\Delta\tau$). This is also considered a *cooperative* form of communication. In many scenarios, it would be useful to perform orbit determination on *non-cooperative* sources, mainly in cases where it is impossible or difficult to resolve carrier phase ambiguity and therefore extract a transmit time from a signal.

1.4 Signals of Opportunity

SoOp is a bistatic method of remote sensing which takes advantage of existing signals of interest already present in the available EM spectrum. It is possible to use the existing transmitted signal form satellite, such as a communication satellite, as a SoOp for the sake of POD.

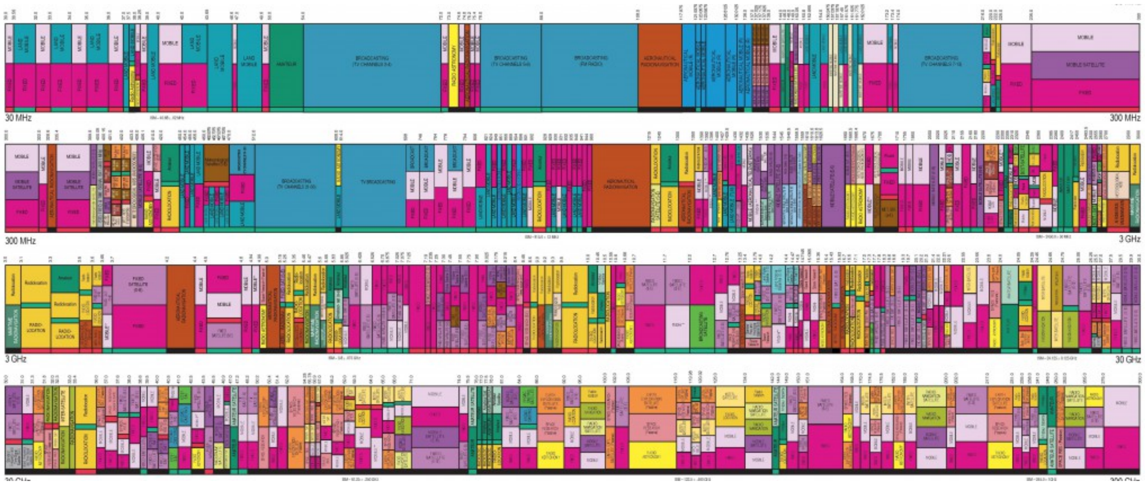

Figure 1.2. Frequency Allocation Chart, SoOp Candidate frequencies

1.5 Baseline Interferometry

For non-cooperative sources, it is possible to difference pairs of receivers by cross-correlation to obtain a differential delay which can be used to resolve carrier phase ambiguity. [[11]] described two potential uses of this differenced pseudorange. The first involves obtaining an algebraic solution for the incident angle at which the signal arrives. This is achieved by assuming a perfectly uniform wavefront that reaches one receiver after the other. From this, it is possible to solve for the incident angle given a known distance between the receivers. The downside to this methodology is a significant loss of precision for calculating the incident angle, especially due to sensitivities of small angles at large distances. This is exacerbated if there is poor geometry of the satellite in relation to the receivers [[12]]. Furthermore, the assumption of a perfectly linear wavefront only holds when the baseline is no more than 650 meters; this further constrains this approach which is already sensitive to the geometry of the receivers.

1.5.1 Very Long Baseline Interferometery (VLBI)

Very Long Baseline Interferometry (VLBI) is a method of obtaining difference-of-arrival (DOA) measurements by taking advantage of the large geometric distance between the receivers to try to create the difference in the arrival of the signal. One prominent usage of VLBI was in the 2019 black hole imaging experiment, which used VLBI across radio telescopes on a planetary scale in order to image the M87 black hole [[13]]. Time synchronization and phase synchronization were critical in order to perform the necessary cross-correlation for the experiment. Therefore, it is insightful to study the techniques used to achieve this in order to apply them to a DOA application for orbit determination.

Time synchronization is initially achieved by aligning all radio telescopes in the Event Horizon Telescope (EHT) network to GPS time. Phase synchronization is achieved with each radio telescope possessing its own frequency reference oscillator [[14]]. The reference oscillator for this experiment is a hydrogen maser, which is a standard used in VLBI astronomy due to its extremely low Allan variance. It should be noted, though, that the integration time for the samples in this experiment are on the order of 10 seconds, and as such, the Allan

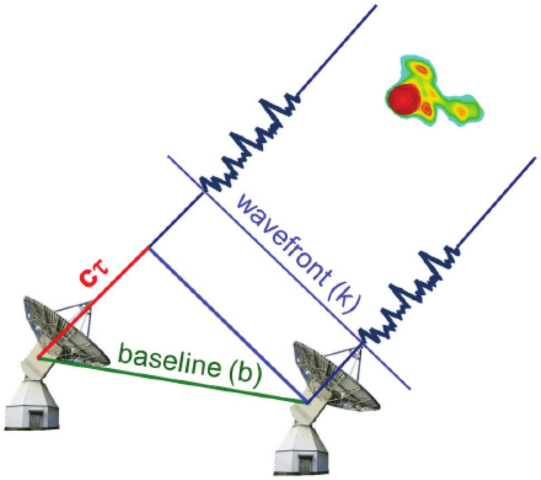

Figure 1.3. VLBI Measurement

variance for this integration span is 1.5×10^{-14} for a hydrogen maser frequency reference. Furthermore, the frequency at which signals are being measured is on the order of 230 GHz, or two orders of magnitude higher frequency (two orders of magnitude lower wavelength) than that of the S-band DOA application in this paper. The significance of this is explained in section 3.2.2.

Another variation of baseline interferometery involves obtaining a Euclidean basis for the differential pseudoranges over large distances for the separated receivers [[15]]. These differential pseudoranges are then solved in a system of equations or some numerical method, such as least squares or weighted least squares, to obtain a numerical solution for the satellite position [[11]]. Over longer baselines, this differential pseudorange gains better geometry for converging a solution. This VLBI approach was applied by [[9]] to augment C-band ranging data to perform orbit determination on geostationary satellites. As part of their experiment, a VLBI-only approach was characterized. They found that they were able to compute positions of these satellites with sub-meter along-track accuracy. Furthermore, they used distant radio sources such as quasars to attempt to calibrate the bias of the system before listening in to an artificial radio source.

2. THEORETICAL DERIVATIONS

2.1 Problem Setup

The proposed method of locating an object in orbit assumes it is transmitting a non-cooperative signal at the time of position determination. To determine the location of a transmitting satellite, $\vec{R}_T$, in relation to the center of the earth (in either ECEF or ECI coordinates), we set up a GPS-like navigation solution using a series of receivers on the ground.

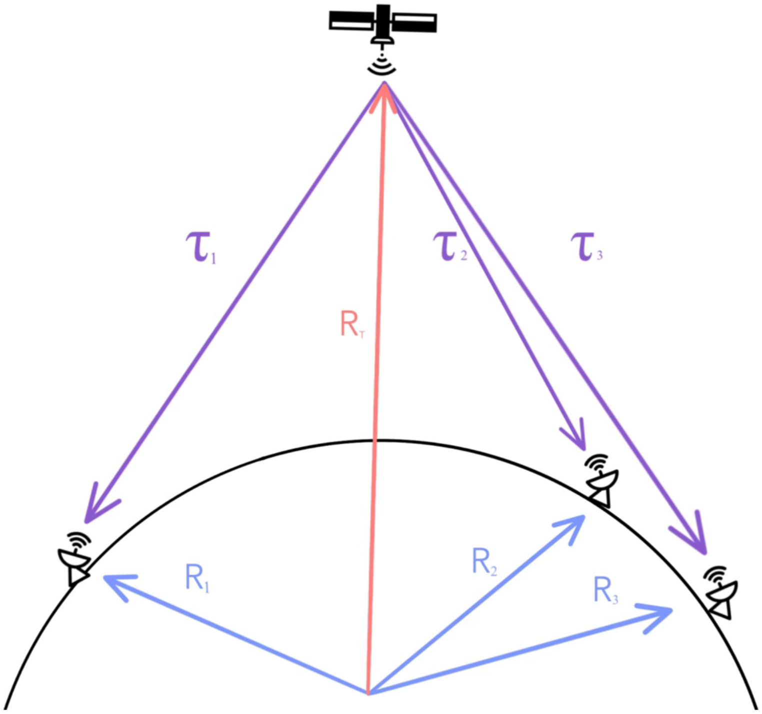

Figure 2.1. Receivers Relative to Transmitting Satellite

In the above diagram, we can see an example for a set of four receivers on the ground, distributed on a continental scale for better solution geometry. The location of these receivers ($\vec{R}_\mathrm{i}$ where i $= 1, 2, 3...$) is already known in the respective frame where we are trying to solve for $\vec{R}_T$.

Assume a pair of receivers that are simultaneously listening to the transmitting satellite. The receivers are assumed to have perfect time and phase synchronization. By cross-correlating the simultaneously received signal, we can obtain a scalar time difference between when the signal was received at each receiver. This time delay τ_{i} is directly related to the geometric path length D between primary receiver $\vec{R}_0$, to the transmitter, and then to receiver $\vec{R}_{\mathrm{i}}$. Formally expressed as the following:

$$D_{\mathrm{i}} = c\Delta\tau_{\mathrm{i}} = \left\|\vec{R}_T - \vec{R}_0\right\| - \left\|\vec{R}_T - \vec{R}_{\mathrm{i}}\right\| \tag{2.1}$$

Where c is the speed of light, relating the time delay to the path delay of the signal. As noted before, the time delay τ_{i} requires a pair of receivers to generate the cross-correlation. In order to fully solve for the three unknown position components, three independent measurements are necessary. A minimum of four receivers are necessary, with one being the baseline with respect to which all the other measurements are calculated. This yields three independent measurements for the time delay between each pairing of receivers with respect to the baseline. In this case, the above equation can be repeated as follows:

$$\begin{aligned} D_1 = c\Delta\tau_1 = \left\|\vec{R}_T - \vec{R}_0\right\| - \left\|\vec{R}_T - \vec{R}_1\right\| \\ D_2 = c\Delta\tau_2 = \left\|\vec{R}_T - \vec{R}_0\right\| - \left\|\vec{R}_T - \vec{R}_2\right\| \\ D_3 = c\Delta\tau_3 = \left\|\vec{R}_T - \vec{R}_0\right\| - \left\|\vec{R}_T - \vec{R}_3\right\| \end{aligned} \tag{2.2}$$

Given that $\vec{R}_T$ is a three dimensional vector, the above system of equations has three equations and three unknowns and can be solved in order to compute the position of the transmitting satellite $\vec{R}_T$ exactly in an ideal system.

The point positioning problem for $\vec{R}_T$ is solved by first linearizing the path delay observation equation and then solving a least squares optimization. First we assume that the actual observable, D, is the sum of a modeled observation plus an error or noise term ν:

$$\begin{aligned} D_{\text{observed}} &= D_{\text{model}} + \text{ noise} \\ &= D(x, y, z) + \nu \end{aligned} \tag{2.3}$$

By considering provisional values (x_0, y_0, z_0), we perform a first-order Taylor expansion of equation 2.3, ignoring second-order and higher terms. These provisional values are initialized with a guess in the least squares optimizer:

$$\begin{aligned} D(x,y,z) &\cong D(x_0,y_0,z_0) + (x-x_0)\frac{\delta D}{\delta x} + (y-y_0)\frac{\delta D}{\delta y} + (z-z_0)\frac{\delta D}{\delta z} \\ &= D_{\text{computed}} + \Delta x\frac{\delta D}{\delta x} + \Delta y\frac{\delta D}{\delta y} + \Delta z\frac{\delta D}{\delta z} \end{aligned} \tag{2.4}$$

Or, in the notation of $\vec{R}_T$, we get the following after differentiating equation 2.1:

$$\begin{aligned} D(\vec{R}_T) &\cong D(\vec{R}_{T_0}) + (\vec{R}_T - \vec{R}_{T_0})\frac{\delta D}{\delta \vec{R}_T} \\ &= D_{\text{computed}} + \Delta\vec{R}_T\left(\frac{\vec{R}_T - \vec{R}_0}{\left\|\vec{R}_T - \vec{R}_0\right\|} - \frac{\vec{R}_T - \vec{R}_{\text{i}}}{\left\|\vec{R}_T - \vec{R}_{\text{i}}\right\|}\right) \\ &= D_{\text{computed}} + \Delta\vec{R}_T\left(\hat{u}_0^T - \hat{u}_{\text{i}}^T\right) \end{aligned} \tag{2.5}$$

Notably, the differentiated delay equation becomes the difference of the respective unit vectors. The residual observation is defined as the difference between the actual observation D_{observed} and the computed observation using the provisional x, y, z values in D_{computed}.

$$\begin{aligned} \Delta D &\equiv D_{\text{observed}} - D_{\text{computed}} \\ &= \Delta x\frac{\delta D}{\delta x} + \Delta y\frac{\delta D}{\delta y} + \Delta z\frac{\delta D}{\delta z} + \nu \\ &= \Delta\vec{R}_T\left(\frac{\vec{R}_T - \vec{R}_0}{\left\|\vec{R}_T - \vec{R}_0\right\|} - \frac{\vec{R}_T - \vec{R}_{\text{i}}}{\left\|\vec{R}_T - \vec{R}_{\text{i}}\right\|}\right) + \nu \\ \Delta D_{\text{i}} &= \Delta\vec{R}_T\left(\hat{u}_0^T - \hat{u}_{\text{i}}^T\right) + \nu \end{aligned} \tag{2.6}$$

This final equation builds into a system for each unique delay observable:

$$\begin{pmatrix} \Delta D_1 \\ \Delta D_2 \\ \Delta D_3 \\ \vdots \\ \Delta D_n \end{pmatrix} = \begin{pmatrix} \hat{u}_0^T - \hat{u}_1^T \\ \hat{u}_0^T - \hat{u}_2^T \\ \hat{u}_0^T - \hat{u}_3^T \\ \vdots \\ \hat{u}_0^T - \hat{u}_n^T \end{pmatrix} \Delta\vec{R}_T + \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \vdots \\ \nu_n \end{pmatrix} \tag{2.7}$$

This constitutes a linear system which can be expressed as:

$$\mathbf{y} = \mathbf{H}x + \nu \tag{2.8}$$

Where:

$$\mathbf{y} = \begin{pmatrix} \Delta D_1 \\ \Delta D_2 \\ \Delta D_3 \\ \vdots \\ \Delta D_n \end{pmatrix}, \mathbf{H} = \begin{pmatrix} \hat{u}_0^T - \hat{u}_1^T \\ \hat{u}_0^T - \hat{u}_2^T \\ \hat{u}_0^T - \hat{u}_3^T \\ \vdots \\ \hat{u}_0^T - \hat{u}_n^T \end{pmatrix}, x = \Delta\vec{R}_T, \nu = \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \vdots \\ \nu_n \end{pmatrix} \tag{2.9}$$

2.2 Least Squares Solution

Let us consider a solution for the linearized observation equations that is denoted as $\hat{x}$. The residuals of this would be defined as the difference between the actual observations and the new, estimated model for the observations. Rearranging equation 2.8, we get the following:

$$\hat{\nu} = \mathbf{y} - \mathbf{H}\hat{x} \tag{2.10}$$

The cost function of a least squares solution is to minimize the square of this residual ν:

$$J(x) = \sum_{\mathrm{i}=1}^{n} {\nu_\mathrm{i}}^2 = \nu^\mathrm{T}\nu = (\mathbf{y} - \mathbf{H}x)^\mathrm{T}(\mathbf{y} - \mathbf{H}x) \tag{2.11}$$

$$\min\left(\sum_{\mathrm{i}=1}^{n}(\mathbf{y} - \mathbf{H}x)^\mathrm{T}(\mathbf{y} - \mathbf{H}x)\right) \tag{2.12}$$

If x is varied a small amount than $J(x)$ should also vary, except where the solution converges; thus, the cost function is minimized. The following shows the derivation of the update term to the least squares process, assuming that this small perturbation $\delta J(\hat{x})$ is 0:

$$
\begin{aligned}
\delta J(\hat{x}) &= 0 \\
\delta\left\{(\mathbf{y}-\mathbf{H}\hat{x})^{\mathrm{T}}(\mathbf{y}-\mathbf{H}\hat{x})\right\} &= 0 \\
\delta(\mathbf{y}-\mathbf{H}\hat{x})^{\mathrm{T}}(\mathbf{y}-\mathbf{H}\hat{x}) + (\mathbf{y}-\mathbf{H}\hat{x})^{\mathrm{T}}\delta(\mathbf{y}-\mathbf{H}\hat{x}) &= 0 \\
(-\mathbf{H}\delta x)^{\mathrm{T}}(\mathbf{y}-\mathbf{H}\hat{x}) + (\mathbf{y}-\mathbf{H}\hat{x})^{\mathrm{T}}(-\mathbf{H}\delta\hat{x}) &= 0 \\
(-2\mathbf{H}\delta x)^{\mathrm{T}}(\mathbf{y}-\mathbf{H}\hat{x}) &= 0 \\
\left(\delta x^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\right)(\mathbf{y}-\mathbf{H}\hat{x}) &= 0 \\
\delta x^{\mathrm{T}}\left(\mathbf{H}^{\mathrm{T}}\mathbf{y}-\mathbf{H}^{\mathrm{T}}\mathbf{H}\hat{x}\right) &= 0 \\
\mathbf{H}^{\mathrm{T}}\mathbf{H}\hat{x} &= \mathbf{H}^{\mathrm{T}}\mathbf{y}
\end{aligned}
\tag{2.13}
$$

Where finally:

$$x = \Delta\vec{R_T} = \left(\mathbf{H^T H}\right)^{-1}\mathbf{H^T y} \tag{2.14}$$

and $\vec{R}_T$ can be converged on until the cost function is minimized:

$$\vec{R_{Tnew}} = \Delta\vec{R_T} + \vec{R_{Told}} \tag{2.15}$$

2.3 Dilution of Precision

Similar to the GPS solution, we can generate a 3 × 3 geometry matrix analogous to Dilution of Precision (DOP):

$$G = \left(\mathbf{H}^T\mathbf{H}\right)^{-1} \tag{2.16}$$

When transformed to the satellite frame, we can obtain the DOP for the computed solution in the cross-track, in-track, and radial directions. This matrix also gives an indication of whether the ”geometry” of the solution is good.

The rotation into a RIC (radial/in-track/cross-track) is described as follows, specifically in application to a satellite in a geostationary orbit, given its location at specified longitude Ω_G:

$$\mathbf{T}_{RIC} = \begin{bmatrix} \cos(\Omega_G) & -\sin(\Omega_G) & 0 \\ \sin(\Omega_G) & \cos(\Omega_G) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.17}$$

This transformation holds for a transmitting satellite in a geosynchronous orbit with zero inclination, which the transmitters in this thesis are assumed to be. With the least squares solver, more than three measurements can be incorporated to constrain the solution further by making it an over-determined system. This is why, relative to the satellite, the receivers should be distributed as far apart as possible to maximize the differences in delay between each pair of receiver.

The DOP matrix can then be transformed into the RIC frame:

$$G_{RIC} = \mathbf{T}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{T}^T \tag{2.18}$$

Now, the components of the DOP matrix can yield a more physically meaningful quantity. The diagonal of this matrix contains respective DOP values for radial, in-track, and cross-track components.

2.4 Cross-Correlation and Phase

The key operator necessary to obtain the previously mentioned time delay of arrival (τ) is the cross-correlation between the two signals. Furthermore, it is useful to measure the offset of a signal from a reference over time by extracting its phase offset.

A receiver computes an approximation of a cross-correlation between two signals as:

$$R_{12}(\tau) = \int_{T_I} x_1(t)x_2^*(t-\tau)dt \tag{2.19}$$

The autocorrelation is correlation of the time series with itself:

$$R_{11}(\tau) = \int_{T_I} x_1(t)x_1^*(t-\tau)dt \tag{2.20}$$

The peak magnitude of this function represents the time delay (τ) at which signal x_2 is with relation to x_1. This is the value with which we calculate the path delay observable mentioned in the problem formulation.

2.5 XM Radio Signal

XM Radio is a form of Satellite Digital Audio Radio Service (SDARS) that is available in the United States and Canada. It is currently operated by Sirius XM Radio Inc. In this thesis, we use the XM radio signal as a proof-of-concept for a non-cooperative transmitting source whose absolute location can be estimated using a TDOA approach. This section will describe the XM constellation along with the signal structure and the way it is applied for orbit determination.

The XM radio system is primarily based off of two high-powered Boeing 702 SDARS satellites called “Rhythm” (XM-4) and “Blues” (XM-3) that are located at 115 deg W and 85 deg W respectively. These satellites are located in a geostationary orbit 35,786 km above the equator and therefore remain stationary relative to the surface of the earth, providing uninterrupted transmission. While the TDOA orbit determination can be applied to a transiting satellite, XM provided a good candidate for this demonstration, as it removed the need for planning collection times. Furthermore, the XM satellites are widebeam signals covering all of the United States in a single beam, thus allowing long baselines in this experiment.

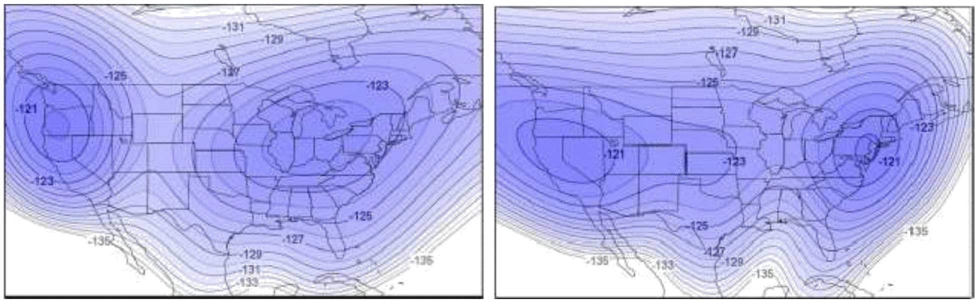

Figure 2.2. XM Satellite Coverage for XM-3 and XM-4 respectively[16]

In addition to these two high-power satellite transmitters, various terrestrial repeaters exist, concentrating around population centers and accommodating terrain and elevation occlusion. These are of concern because of potential interference by the terrestrial repeaters.[17].

2.5.1 Frequency and Polarization

XM radio is allocated spectrum between 2332.5 MHz to 2345.0 MHz (S-band). Each satellite transmits two carriers for a total of four carriers for the system. The downlink frequency breakdown is shown in figure 2.3. Ensemble A and Ensemble B are observed for both satellites. Both of these channels contain half of the total downlink throughput of the XM system. Each of these two channels contains half of the total capacity of the XM system. The satellites occupy an area to the left and the right, whereas the terrestrial repeaters broadcast in the middle. All downlink signals on XM are Left-Hand Circular Polarized (LHCP) for simplicity of the antenna system

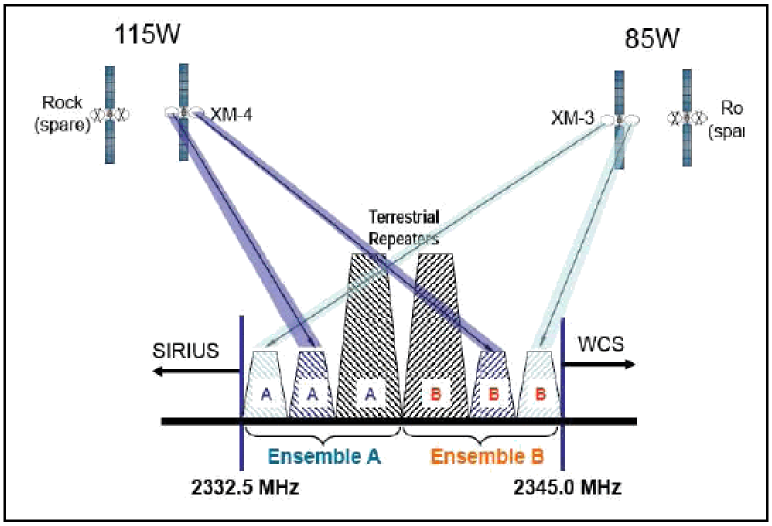

Figure 2.3. XM Frequency Allocation [16]

The four satellite downlink channels have a bandwidth of 1.866 MHz each. Each channel is distinguished by a dip in power in a corresponding chart. They contains a single quadrature phase-shift keying (QPSK) modulated time-division multiplexed (TDM) signal and a symbol period of 1.64 Msps (or 3.28 Mbps). The downlink is error-corrected with a stream and block error correction coding, resulting in an actual information rate of each TDM of 2.048 Mbps. Furthermore, the two satellites signals use a different encrypting scheme and interleaving pattern such that even though they contain the same content, they show up as independent signals. When both ”ensembles” are combined, this provides a more robust and error free transmission. For the experiment discussed in this thesis, ensemble B is used for signal acquisition.

2.5.2 Filtering

In the orbit determination experiment, both XM-3 and XM-4’s signals are acquired at the same time. A center frequency of 2.34125 GHz was used, with a bandwidth of 4 GHz on either side to acquire the B channel of XM-3 and XM-4. In order to process the signals separately, they were filtered in post-processing using a low-pass filter. This was designed as an equal ripple filter with 500 taps using the MATLAB filter design toolbox:

```
    fd = fdesign.lowpass(’N,Fp,Fst’,500, 1.6e+6/2 ,1.82e+6/2,Fs);
    dd = design(fd, ’equiripple’);
```

The sampling frequency is 4MHz, representing the full bandwidth collected on either side of the center frequency. The following magnitude response was produced as an example of the filtering the left channel.

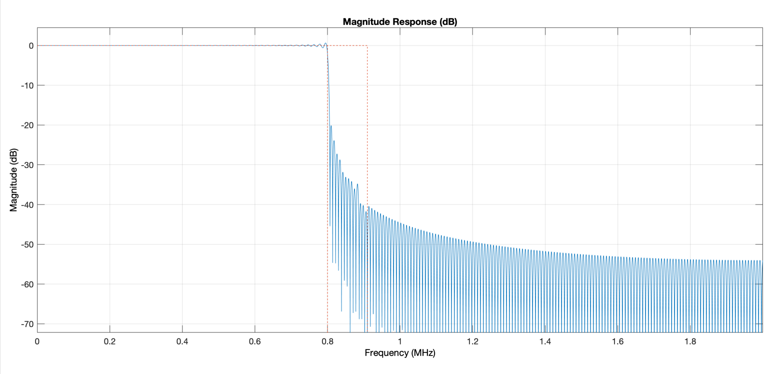

Figure 2.4. Magnitude Plot of Filter

This filter is applied to a sample block of data from XM-3 and XM-4. We can observe how both signals can be parsed separately, and the cross-correlation function is better resolved as a magnitude sinc function.

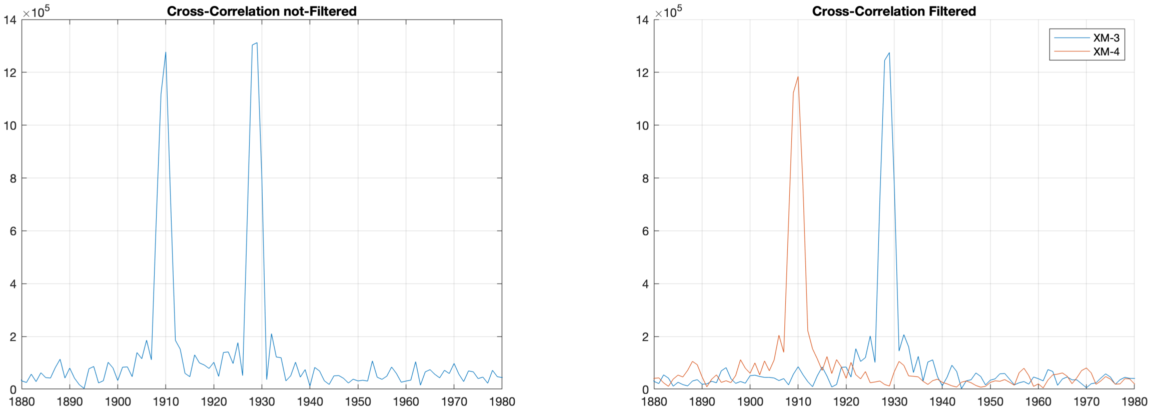

Figure 2.5. XM-3 and XM-4 Filtered out from total signal

2.6 GNSS-SDR - Time and Location Lock

One of the key components to precisely cross-correlating a pair of receivers is accurate time synchronization across continental scales, as well as an accurate position lock. In order to better understand how we might achieve both these goals, GNSS-SDR was studied as a potential software front-end interface to obtain an accurate ground position lock as well as time synchronization. The goal of this software is to produce clock and phase bias values in order to compensate for the arrival time of a signal. A complete study of the software was done and its software architecture and operation were documented.

2.6.1 Introduction

GNSS-SDR is an open source software platform, meant to run on SDR hardware. It is compatible with Ettus's USRP series and is designed to acquire and process all common GNSS constellation signals available today. Its goal is to provide an user-friendly abstraction layer between the USRP software library and the user. This allows the user to easily select and fine-tune the data recorded, as well as process observables into an easy format.

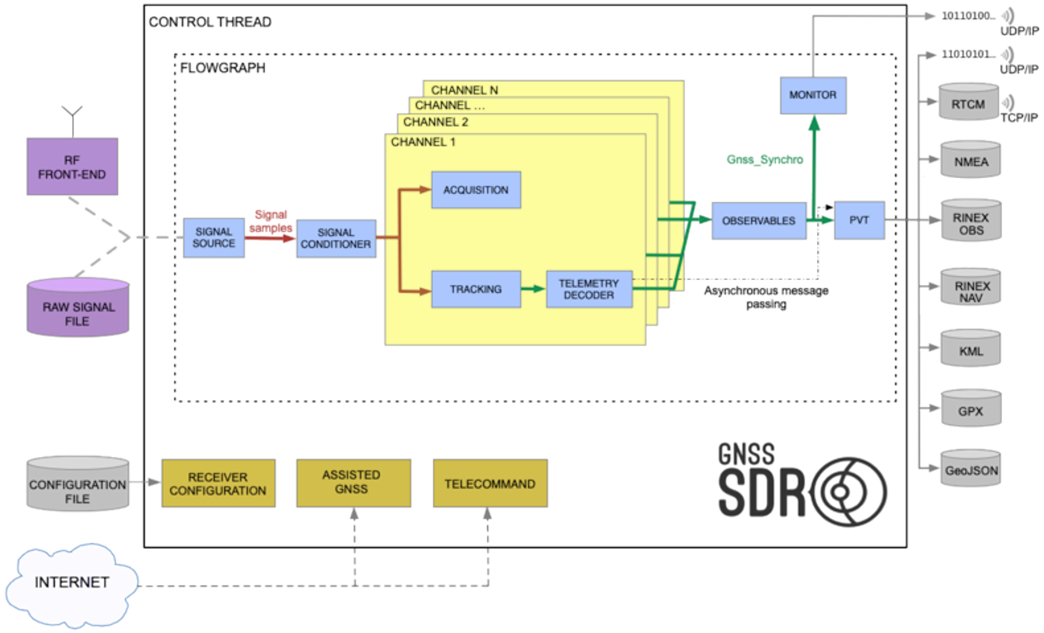

Figure 2.6. GNSS-SDR Software architechture Block Diagram [8]

2.6.2 Observables and Pseudorange Measurement

The goal of the Observables block is to collect and process synchronization data from all the active channels. It will then generate basic GNSS measurements such as pseudorange, carrier phase, and Doppler shift. The pseudorange measurement is made by differencing the time at which the transmission was sent (at satellite time) to the time at which it is received on the ground (in ground time). This includes several delays and biases such as clock offsets.

For satellite s, the pseudorange $P_r^{(s)}$ can be expressed in terms of the signals reception time $\bar{t}_r$ and signal transmit time $\bar{t}^{(s)}$, as well as the following biases and offsets:

$$P_{r,\mathrm{i}}^{(s)} = \quad \rho_r^{(s)} + c\left(dt_r\left(t_r\right) - dT^{(s)}\left(t^{(s)}\right)\right) + I_{r,\mathrm{i}}^{(s)} + T_r^{(s)} + \epsilon_P \tag{2.21}$$

GNSS-SDR attempts to set a common reception time across all channels from which it collects, resulting in the calculation of relative pseudorange with respect to a chosen reference satellite. The reference selection algorithm chooses from among the available satellites the most recent $\mathbf{TOF}_{ref}$ which will be the nearest satellite with an associated reference time t_{ref}. From this, all other travel times for each satellite s are calculated based on this reference as:

$$\tau^{(s)} = \Delta\mathrm{TOW}^{(s)} + \Delta t_r^{(s)} + \tau_{\mathrm{ref}} = \mathrm{TOW}^{(s)} - \mathrm{TOW}_{\mathrm{ref}} + t_r^{(s)} - t_{r_{\mathrm{ref}}} + \tau_{\mathrm{ref}} \tag{2.22}$$

The pseudorange calculation is illustrated in this diagram:

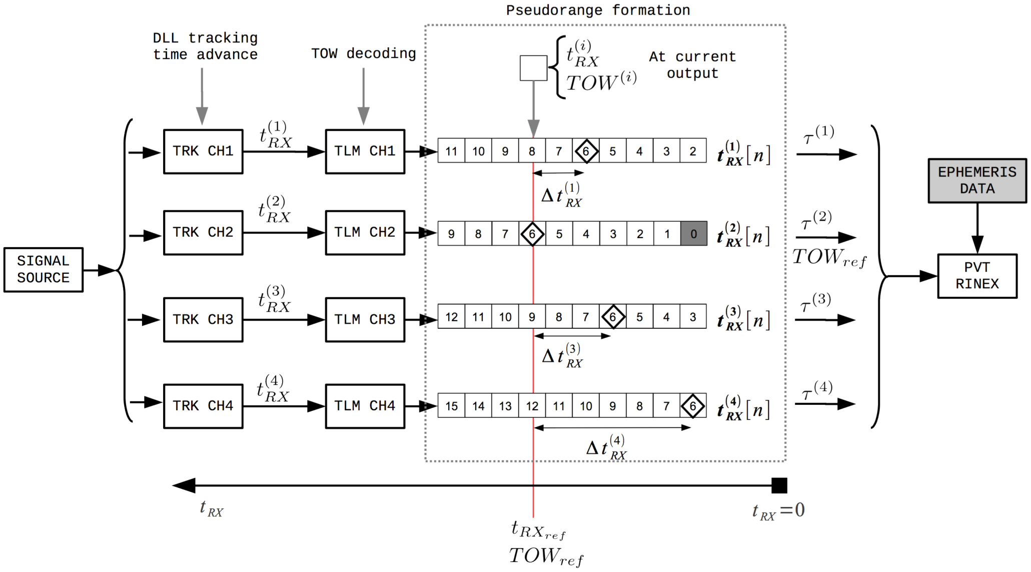

Figure 2.7. GNSS-SDR pseudorange multi-channel timing

2.6.3 PVT Solution

The default positioning mode for GNSS-SDR is *Single Point Positioning* or

```
PVT.positioning_mode=Single
```

in the configuration file.

This mode uses an iterative weighted least squares method to converge to a point solution, as shown by the equation below:

$$\hat{\mathbf{x}}_{\mathrm{i+1}} = \hat{\mathbf{x}}_{\mathrm{i}} + \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\left(\mathbf{y} - \mathbf{h}\left(\hat{\mathbf{x}}_{\mathrm{i}}\right)\right) \tag{2.23}$$

Where our $\mathbf{H}$ matrix is given by:

$$\mathbf{H} = \begin{pmatrix} -\mathbf{e}_r^{(1)^T} & 1 \\ -\mathbf{e}_r^{(2)^T} & 1 \\ -\mathbf{e}_r^{(3)^T} & 1 \\ \vdots & \vdots \\ -\mathbf{e}_r^{(m)^T} & 1 \end{pmatrix}, \quad \text{where } \mathbf{e}_r^{(s)} = \frac{\mathbf{r}^{(s)}\left(t^{(s)}\right) - \mathbf{r}_r\left(t_r\right)}{\|\mathbf{r}^{(s)}\left(t^{(s)}\right) - \mathbf{r}_r\left(t_r\right)\|} \tag{2.24}$$

GNSS-SDR makes use of RTKLIB-PVT, which is an open source GNSS positioning package. This library performs weighting $\mathbf{W}$ in the following manner:

$$\begin{gathered} \mathbf{W} = \operatorname{diag}\left(\sigma_1^{-2}, \sigma_2^{-2}, \sigma_3^{-2}, \ldots, \sigma_m^{-2}\right) \\ \sigma_s^2 = F^{(s)} R_r \left(a_\sigma^2 + \frac{b_\sigma^2}{\sin\left(El_r^{(s)}\right)}\right) + \sigma_{bclock,s}^2 + \sigma_{\text{ion},s}^2 + \sigma_{trop,s}^2 + \sigma_{clias}^2 \end{gathered} \tag{2.25}$$

2.6.4 Results of GNSS-SDR Usage

GNSS-SDR overall provided an effective way to locate a potential receiver station. It provided a simple way to easily link with the USRP device and thereby provide an end-to-end solution from signal to PVT-solution. For this setup, a Ettus B210 USRP was used with a connected L-Band antenna. The software was run on an Intel NUC mini-PC running Ubuntu. Various tests were done to assess the performance and capabilities of the software. The software seemed to be able to process up to eight channels of GNSS-satellites live before triggering significant overflow events from the USRP. These eight channels worked in single and multi-constellation configurations and were tested across GPS, GLONASS, and Galileo.

The above image shows the accuracy of GNSS-SDR given three, five, or seven GPS satellite channels to use for its PVT solution. Furthermore, the covariance matrix was analyzed for these results, and a calculated geometric dilution of precision was plotted over a short time frame for each of the three cases.

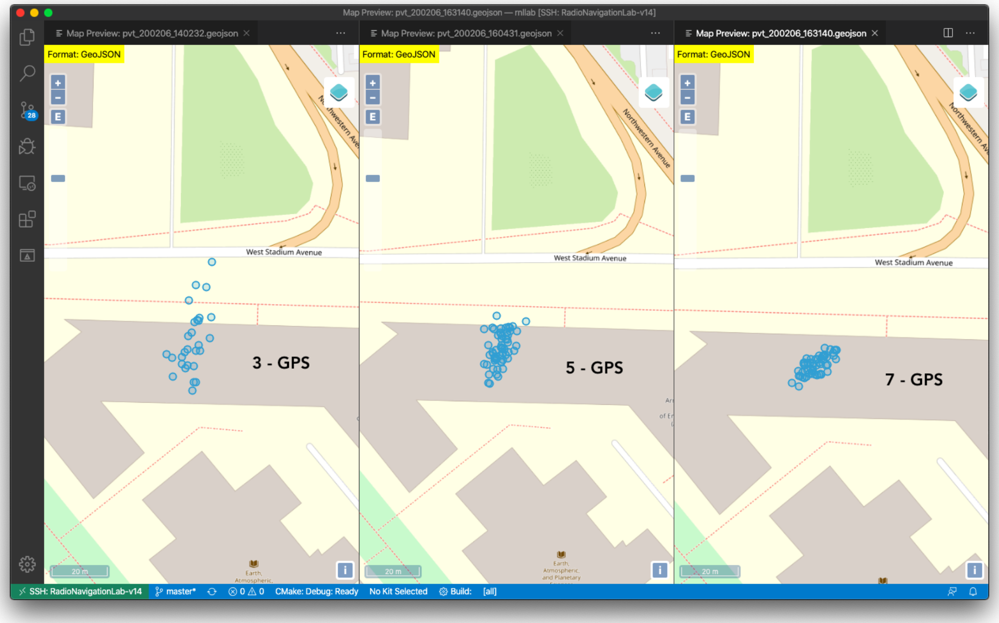
Map Preview: pvt_200206_163140.geojson — milab [SSH: RadioNavigationLab-v14]
Map Preview: pvt_200206_140232.geojson
Map Preview: pvt_200206_160431.geojson
Map Preview: pvt_200206_163140.geojson
Format: GeoJSON
Format: GeoJSON
Format: GeoJSON
West Stadium Avenue
Northwestern Avenue
3 - GPS
5 - GPS
7 - GPS
20 m
SSH: RadioNavigationLab-v14
master*
0 △ 0
CMake: Debug: Ready
No Kit Selected
Build:
[all]

Figure 2.8. GPS accuracy given multiple Channels

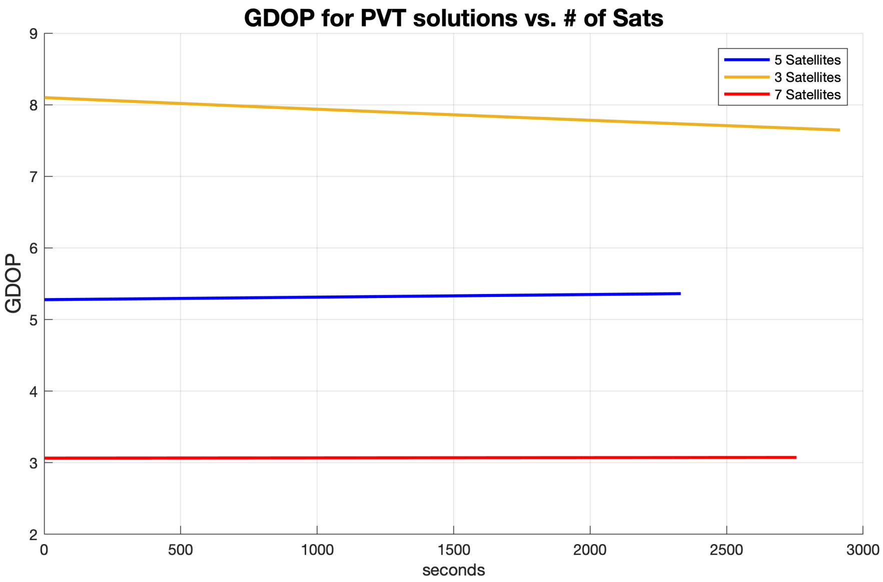

Figure 2.9. GPS accuracy given multiple Channels

3. INSTRUMENT DESIGN

As derived previously, the primary measurement needed to solve the vector system of equations for the transmitter position in equation 2.2 is the time delay between each pair of receiver stations. It was shown that this time delay can be calculated by cross-correlating a signal measured independently by a pair of receivers. A receiver was designed to solve the challenges of making this measurement across very long distances. This receiver also had to be simple, robust, and replicable as in most of the experiments conducted, the receiver was either taken out on a mobile platform or shipped far enough away that it was impractical to to service the instrument if a fault occurred.

3.1 Receiver Station Overview

The main structure of the receiver is laid out in figure ??. The primary elements of the system are the main PC (Intel NUC), B210 USRP, circuit components, and the antennas. All power is derived from the PC. The USRP is powered by its provided USB 3 data and power cable. The amplifier and bias tee for the XM antennas are powered via a USB breakout that is receives its 5 volt from the UBS port of the PC itself. All of the components with the exception of the antennas are kept sheltered inside.

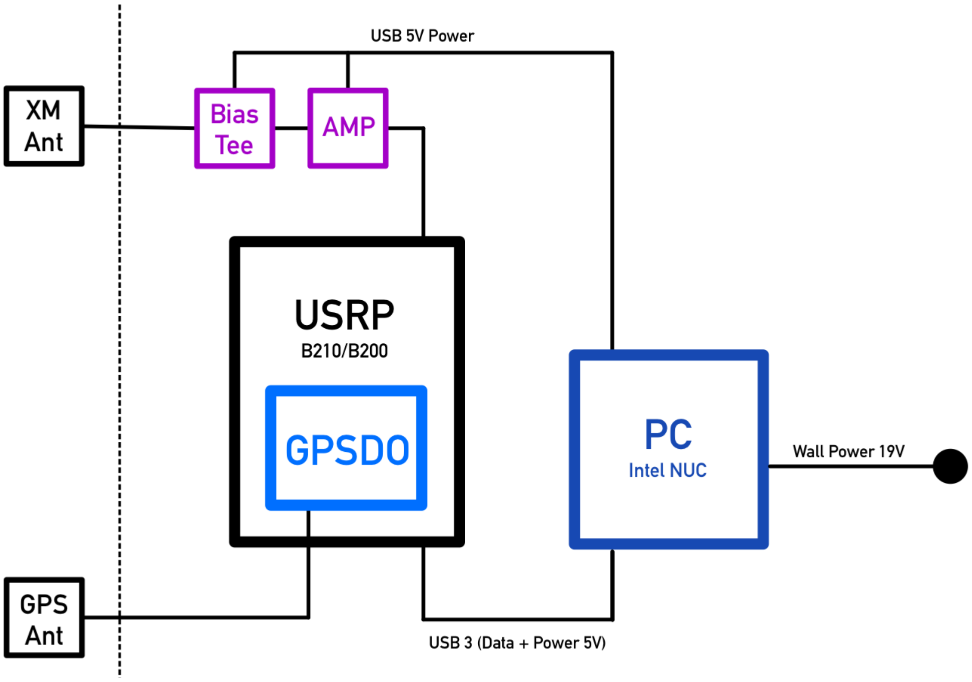

Figure 3.1. Diagram of receiver station

A full setup for a receiver looks approximately like the following photo:

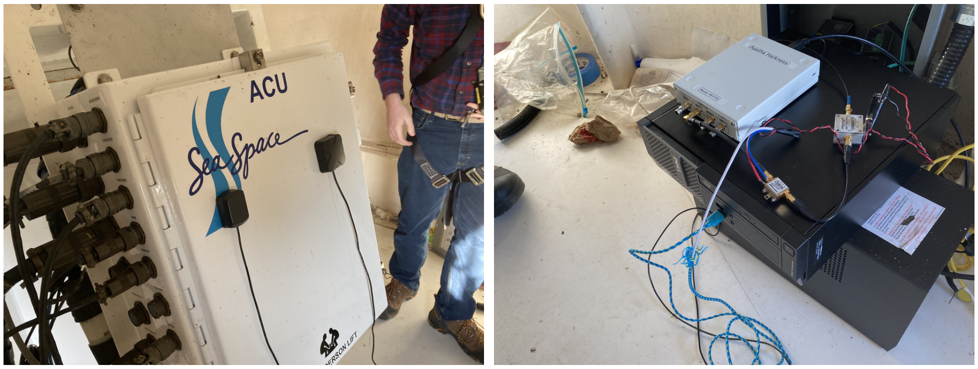

Figure 3.2. Receiver setup at Purdue. GPS and S band antenna mounted on the left. USRP, PC, and power on the right

3.1.1 Computer and Software

The controlling computer was an Intel NUC PC. These are small form factor PCs that allowed the components of the system to be shipped easily. The operating system was Ubuntu 20.04 LTS and ran the USRP Hardware Driver (UHD) version 4.0, which at the time was the latest major revision to the USRP software and featured various improvements in reliability and performance of the USRPs.

3.1.2 USRP

The software defined radio used in this experiment is the USRP B210 from Ettus Research. It is designed for continuous frequency coverage between 70 MHz to 6 GHz. It has a high data throughput and a bandwidth of up to 56 MHz. More importantly, the B series platform allows for a GPS Disciplined Oscillator (GPSDO) module. Ettus provides a temperature controlled GPSDO (TCXO) which provides a pulse per second (PPS) signal which is accurate to within 50 nanoseconds and an Allan deviance of 10^{-11} at 1 second, and a 10 MHz reference. This GPSDO module is relied upon to provide phase and time synchronization over a long distance.

Figure 3.3. B210 USRP by Ettus [18]

3.1.3 Antenna and Circuit

The S-band antenna used in collecting signals from the choice XM-radio satellite is the Abracan AEACBA045015-S2332. This product is a consumer-grade puck antenna meant for SDARS (Satellite Digital Audio Radio Service). Its weather proofing is IP67 rated and it can be mounted almost anywhere.

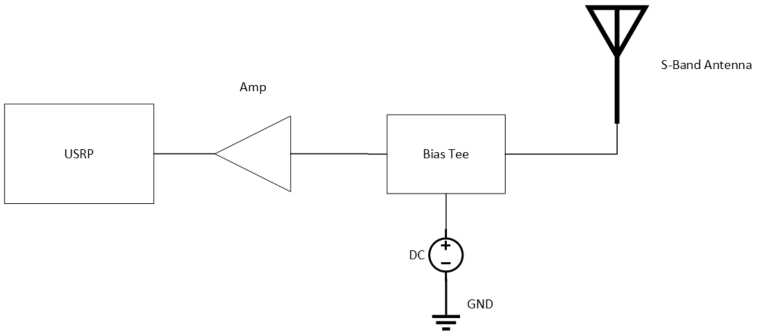

Figure 3.4. Circuit of Front-end

A bias tee is used to power the antenna which consumes 40 mA and operates at 5V. A 40 dB amplifier is applied to the signal output on its way to the USRP to boost signal to noise ratio for better cross-correlation.

3.2 Time Synchronization

In order to perform a TDOA measurement, accurate time and phase synchronization is required across long distances between the receivers. Since it is impractical to use the same clock/frequency source across a wide distance, a method of achieving synchronization using independent time sources needs to be employed either during or after data collection.

Time synchronization is achieved via GPS timing. All USRP devices have provision for a GPSDO to obtain a 1 pulse-per-second (PPS) accurate time lock with GPS time with low variance (on the order of nanoseconds). This is the most reliable form of time synchronization as the USRP software allows querying of GPS time every second. This allows the capability to start two USRPs independent of each other, with separate GPSDOs, and have them operate on the same time reference. The USRPs can start up their respective collection scripts, make all initial configurations, and synchronize their time to GPS time asynchronously, and then wait until a predetermined GPS time before beginning sample collection. This method of using GPS time to time synchronize collection across long distances is standard for DOA and VLBI applications[14] and is able to ensure data collection begins within nanoseconds of each other, i.e., the rated variance for the oscillator.

More specifically, in the experiments presented herein, the above mentioned time synchronization is performed before each block of data is collected.

3.2.1 Time Compensation using GNSS-SDR

In section 2.6, GNSS-SDR was introduced as a method of computing receiver clock bias. In the event that time synchronization cannot be achieved via the built in GPSDO, or some other external reference oscillator, it may be possible to compensate the drifting samples by computing the clock bias of the receiver simultaneously. In this setup, two N300 or four N200/210 USRPs would be needed with one N300 or a pair of N200s at each receiver site. In this scenario, the N series USRPs are needed to collect simultaneous L-band and S-band at each receiver site. The added L-band signal is collected via the front end instead of the GPSDO so that it can be processed via GNSS-SDR to compute clock bias rapidly. This clock bias could then be applied after collection to compensate for time drift.

3.2.2 Time Synchronization Experiment

A set of trials were done to validate the use of existing software libraries in UHD to achieve time synchronization at the start of data collection. The below diagram illustrates the synchronization process in the UHD software.

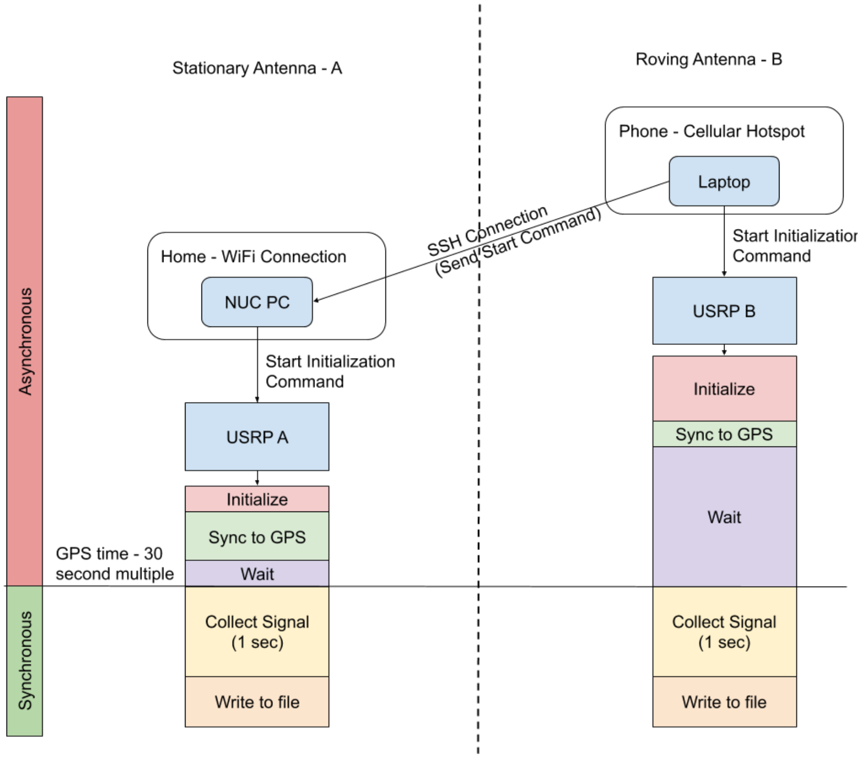

Figure 3.5. Time synchronization process in rover experiment

A pair of receivers were used in this experiment, with one stationary and one taken in a car, moved away from the stationary receiver. The stationary receiver is connected to internet via the computer controlling the USRP (A). The moving antenna set's USRP is connected to a laptop, which is connected to the network via a cellular connection, such as a phone's mobile data hotspot (B). Computer A can be accessed over the network via computer B using an SSH connection. Thus, a command can be sent to USRP A to start collection (via Computer A as a proxy) from computer B. To begin the data collection process, computer B commands USRP B and USRP A (via Computer A over the network) to begin data collection **asynchronously** with the commands being sent about 3-5 seconds apart. Both USRPs then begin their initialization process. As part of the initialization process, the GPSDO acquires a lock on the GPS constellation, and sets the internal clock to GPS time (which then continues to update GPS time every second, with nanosecond accuracy). This initialization also occurs **asynchronously** on both USRPs. On completion of the USRP's initialization, a wait command is set to only begin data collection at the

next 30 second multiple of universal GPS time. Both USRPs, while having performed their configurations and commands asynchronously, will end up being aligned with GPS time. Thus, they can then be commanded to start data collection within nanoseconds of each other by waiting for the next GPS 30-second time multiple. The commanded record time in GPS seconds is always output with the data in order to ensure that both USRPs record at the same 30-second multiple, as well as for post-processing satellite location from a time stamp.

3.3 Phase Synchronization

Phase synchronization or compensation is crucial to cross-correlate the signals received in a DOA application. The method of phase synchronization over long distances of the signal will determine the overall architecture of the experiment.

3.3.1 Phase Synchronization with Accurate Clock Source

The driving source of error for synchronizing the phase of the local oscillator (LO) is the relative Allan variance between the clocks used to drive the data collection at each station. If the stations all use the same clock reference, this difference will not exist. Since this is not practical over large distances, investment must be made into precise clocks to drive an accurate reference signal with enough Allan variance.

We can quantify the needed Allan variance in a clock source for an experiment to have the proper phase synchronization with the following expression [19].

$$\omega \tau \sigma_y(\tau) \ll 1 \tag{3.1}$$

Here, ω is the frequency of the signal being measured (approximately 2.34 GHz), τ represents the integration time desired, and $\sigma_y(\tau)$ is the square root of Allan variance of the desired clock source. For the assumed integration time of 0.004 seconds being used thus far, and approximately a 10^{-11} Allen Deviation for the Built in GPSDO, we achieve a value of 9.36^{-5}. This is considerably less than the 0.023 value that was used in the black hole experiment and suggests the GPSDO built in to the USRP **should be more than enough to achieve phase synchronization**. Otherwise, phase noise is the primary source of error in this measurement.

Achieving this phase synchronization in practice is a bit more challenging due to a variety of technical issues. For example, in the time and phase synchronization experiment mentioned in 3.2.2, while time synchronization was achieved, there was considerable phase drift:

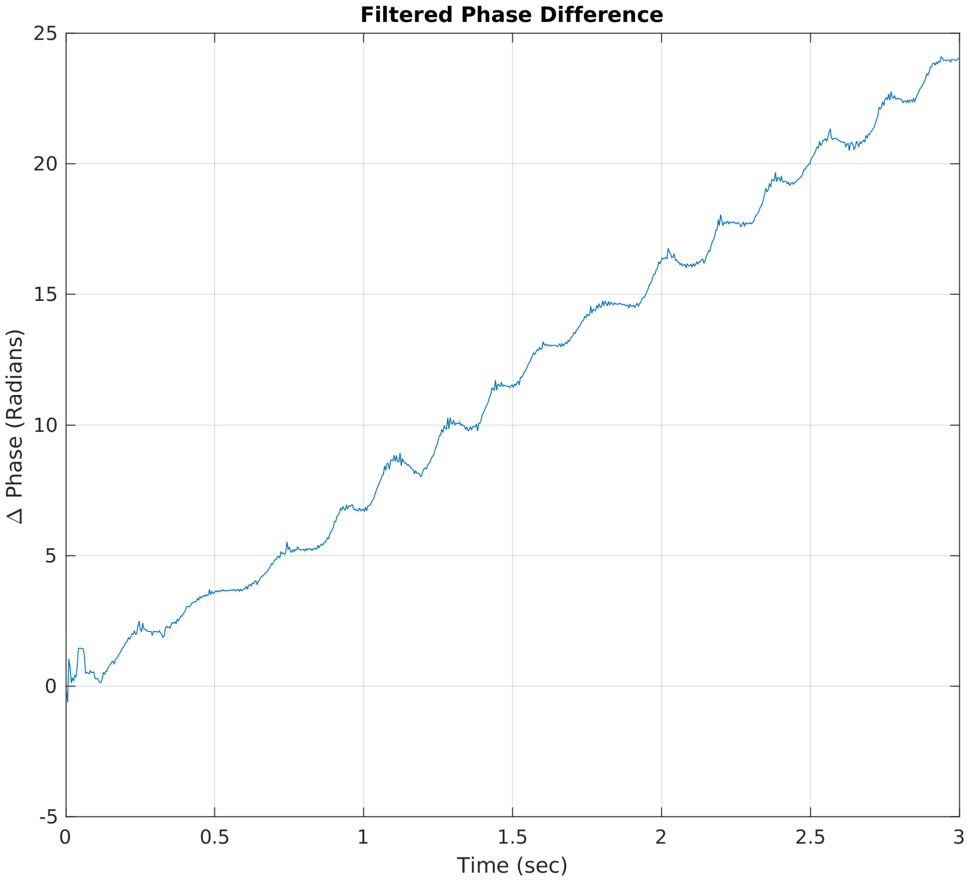

Figure 3.6. Phase Difference over 3 seconds

3.3.2 Phase Synchronization with UHD 4.0

The above section illustrated that it is theoretically possible to achieve meaningful phase synchronization over short periods of time. The initial time and phase synchronization experiment mentioned in section 3.2.2 showed a consistent drift during the course of data collection.

The slope of this drift was unpredictable and also seemed to have an oscillatory component–altogether a very nonlinear behavior that brought the two collected signals out of sync. Using the latest updated software package for the USRPs, UHD 4.0.0.0, as well as a few conditioning practices on the GPSDO, phase synchronization was achieved to a usable limit. The UHD 4 software included newly rewritten firmware for the USRP devices, as well as new features and reliability improvements. An experiment was conducted to determine the improvement in phase drift over time. Measurements were made on four separate days before and after oscillator ”warmup”, during which it powered on for a period of time prior to use. The experiment used a single XM signal that was split into two USRPs each with their own GPSDO and GPS antenna.

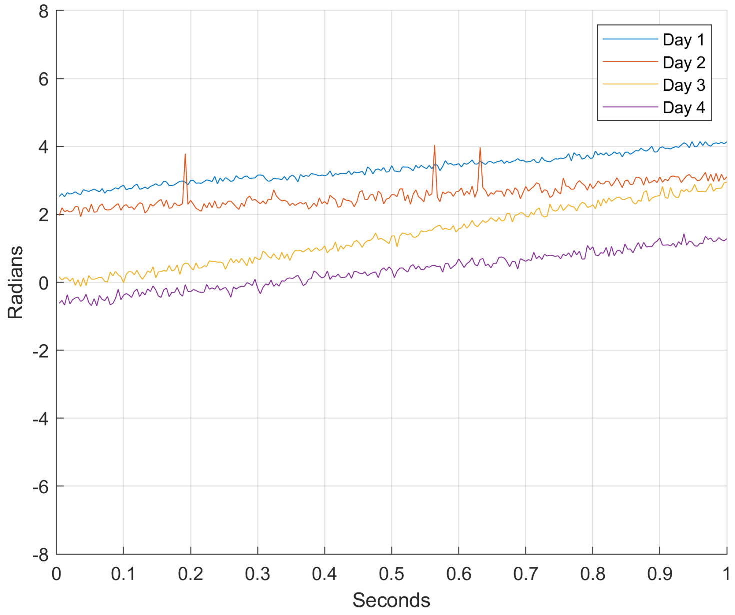

Figure 3.7. Phase Difference over 1 second before oscillator warmup

The above image shows the improved drift over the previous firmware results in section 2.6. There is a slight drift upwards, but nowhere near as dramatic as the previous firmware.

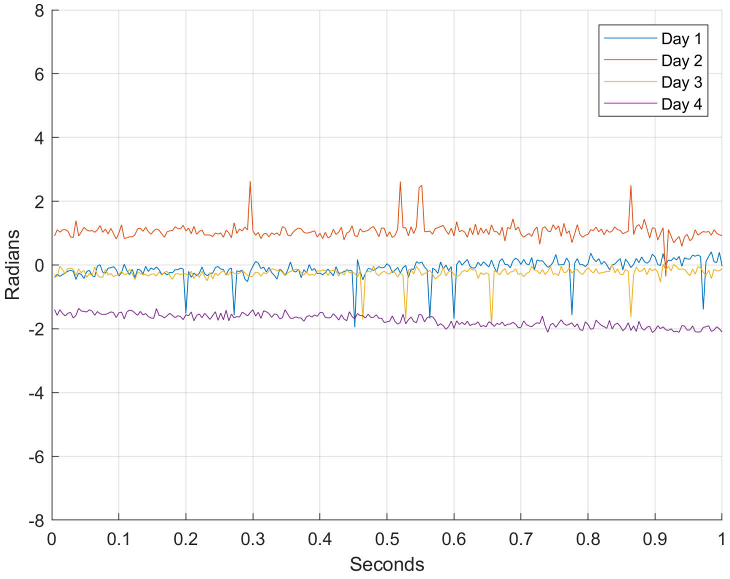

Figure 3.8. Phase difference over one second after oscillator warmup

Furthermore, once the GPSDO was effectively allowed to warm up, we observed near-flat phase across the one second measurement span needed.

The theoretical low coherence value suggests it should be adequate to obtain fairly precise delay measurement needed to compute a high enough accuracy ”inverted GPS-like” solution for orbit determination. Phase synchronization, even after taking measures to control for temperature and software, remains somewhat unreliable.

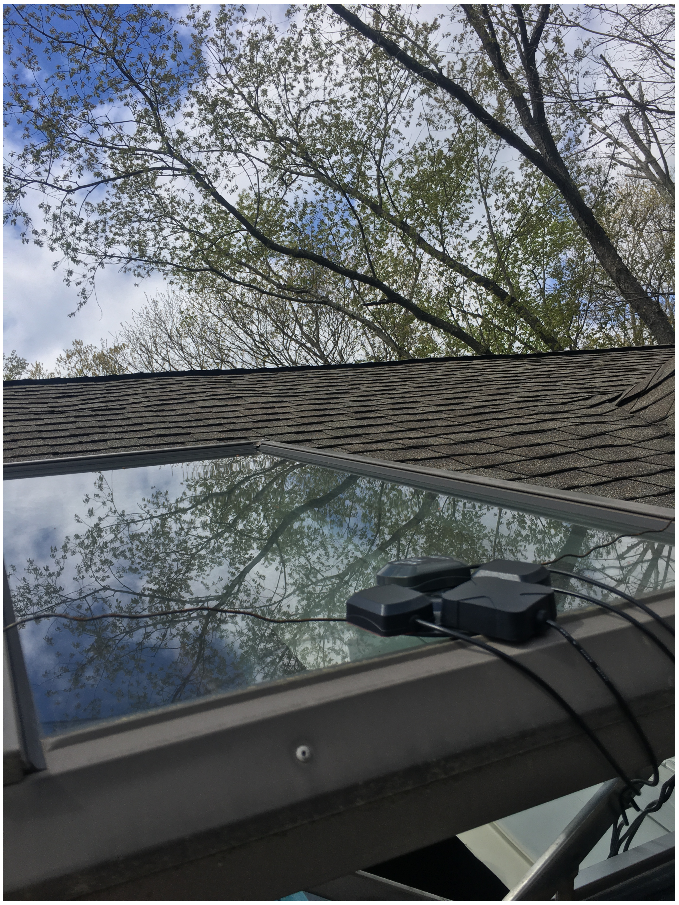

Figure 3.9. Roof Mounted Antennas

In order to achieve the goal of obtaining a single cross-correlation at a point in time, the collection period was limited. Previously, a continuous collection was done (after time synchronization at the start), and the data would be cross-correlated with a nominal integration length of 4 milliseconds, for each 4 ms block of data recorded continuously. A new approach was tried where the collection period was limited to only include enough time to produce a single cross-correlation. Then, time and phase synchronization was re-performed every time before a sample was recorded. This limited the amount of open collection time before phase drift was significant enough to affect solution quality.

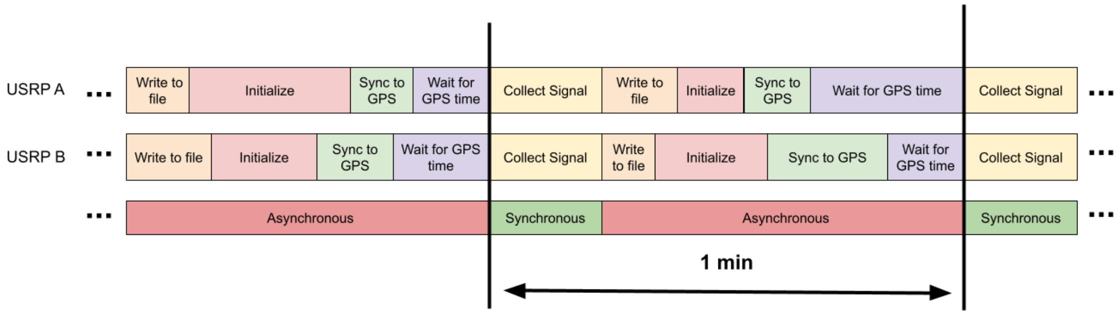

Figure 3.10. Short Signal capture process for Phase synchronization

A downside of this method was that the synchronization process takes anywhere between 10-30 seconds and thus limited the amount of chunks to once every minute. However, this was still frequent enough to produce a sufficiently significant measurement when repeated over a long enough time frame.

4. TIME DELAY OR ARRIVAL VALIDATION

4.1 Path Delay vs. Signal Delay Correlation

First, an experiment was run to verify that the calculated geometric path delay between any pair of receivers to the transmitting satellite is proportional to the delay between when both receivers receive the transmitted signal. While theoretically established in equation 2.1, a variety of factors (further elaborated in chapter 3) could prevent path delay from having a perfectly linear relationship to time delay.

4.1.1 Experiment Setup

Another version of the ”rover” experiment was performed to validate the relationship between path delay and signal delay. Similar to the rover experiment for improving phase drift, a receiver (described in chapter 4) was set up in a stationary location, and another receiver was set up in a car that was moving. The synchronization process described in figure 3.5 was used, however the repetitive synchronization process in figure 3.6 was not applied, as each collection period was started manually once the car reached each location.

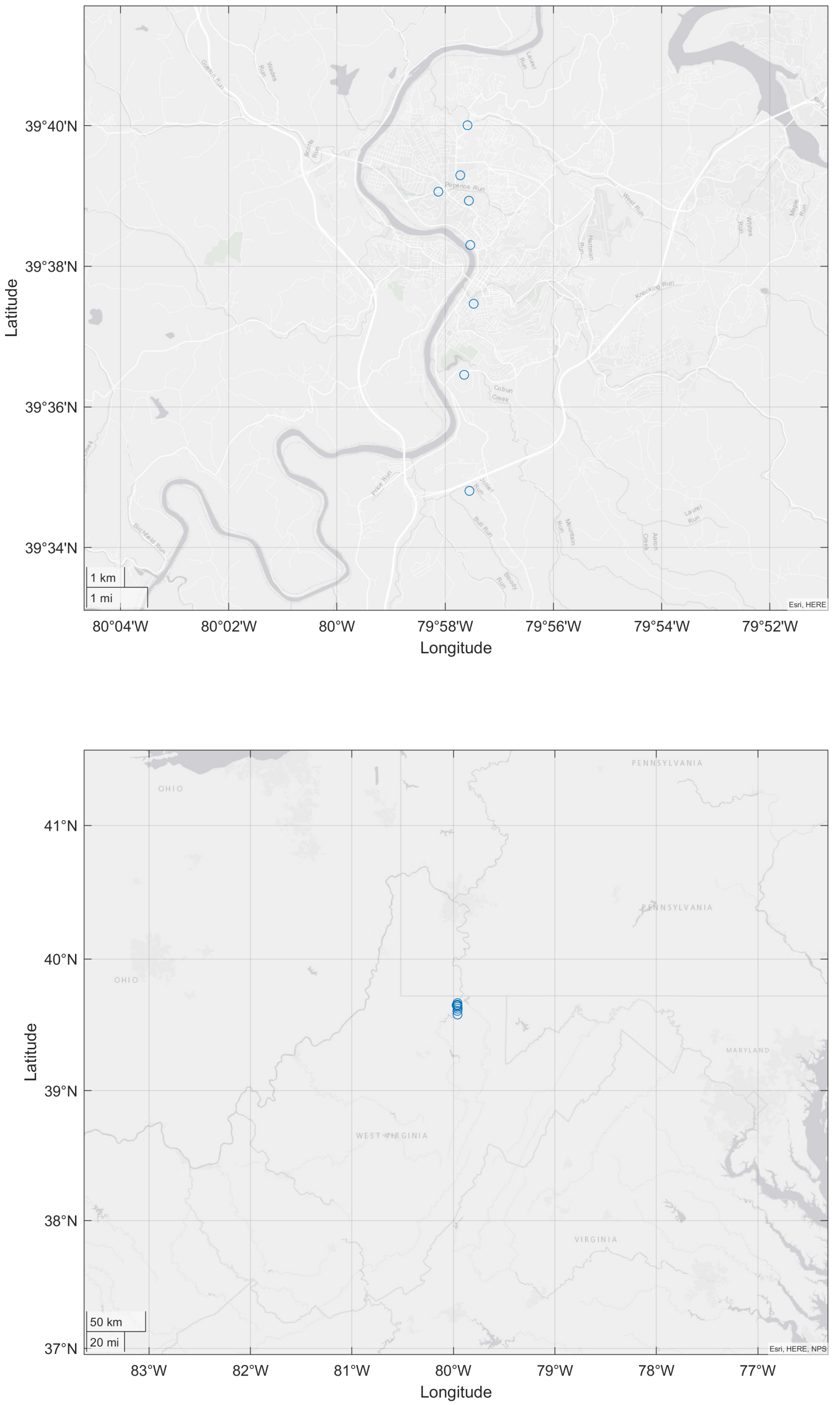

Figure 4.1. Location of collection sites (zoomed in top, zoomed out bottom)

The experiment was carried out around Morgantown, West Virginia (39.629 deg N, 79.9559W). The above image shows the stationary antenna at the northernmost point and seven additional collection sites visited by the car, in addition to a collection at the stationary antenna as well, used as a control. Due to XM's relatively constant location in geostationary orbit, the places of recording were in an approximately North-South direction. Geometrically, this would provide the largest path delay per unit distance moved on the surface of the planet. At each site, the process in figure 3.5 was used to start and synchronize the data recordings. The exact time of recording, as well as the GPS location were recorded at each place.

4.1.2 Path Delay from TLEs

As previously stated, since we are synchronizing signal collection to GPS time, the GPS record time was also captured for each sample. This allows us to compute the actual position of the satellite from which the signal is being recorded in order to compute the distance path delay of the satellite. GPS time of each corresponding data file was first converted to Julian Date. Then, respective signal's satellite's (XM3 for channel 1 and XM4 for channel 2) TLE was acquired from N2YO [20] for the epoch closest to the data collection, and then propagated forward to the exact moment the signal was recorded using an SGP4 [21] propagator. Finally, the earth-centric inertial (ECI) coordinates computed by the SGP4 propagator, were then converted to earth-centric, earth-fixed (ECEF) coordinates. On the ground station side, the GPS information collected at each location also contains NMEA strings that can be parsed at the time of collection. These contain accurate latitude, longitude, and elevation information. This was also converted into the ECEF frame. The ideal path delay can then be computed as:

$$r_{path\ delay} = \|\vec{r}_{base,ECEF} - \vec{r}_{satellite,ECEF}\| - \|\vec{r}_{rover,ECEF} - \vec{r}_{satellite,ECEF}\| \tag{4.1}$$

This generates the following path delay values:

Table 4.1. Calculated Path delay from satellite position in km

	XM-3	XM-4
Home	0	0
1	0.9478	0.8652
2	1.3089	1.4754
3	1.4364	1.1284
4	2.1894	1.7157
5	3.3042	2.5570
6	4.6869	3.8016
7	6.8837	5.4817

4.1.3 Results

Beginning with the cross-correlation results, the below two plots show cross-correlation peaks progressively moving further away from zero with each point further south on the map plot in section 3.6. The control location is the cross-correlation point called ”home” and is centered over zero seconds, where the rover and the stationary antenna were both in the same geographic location.

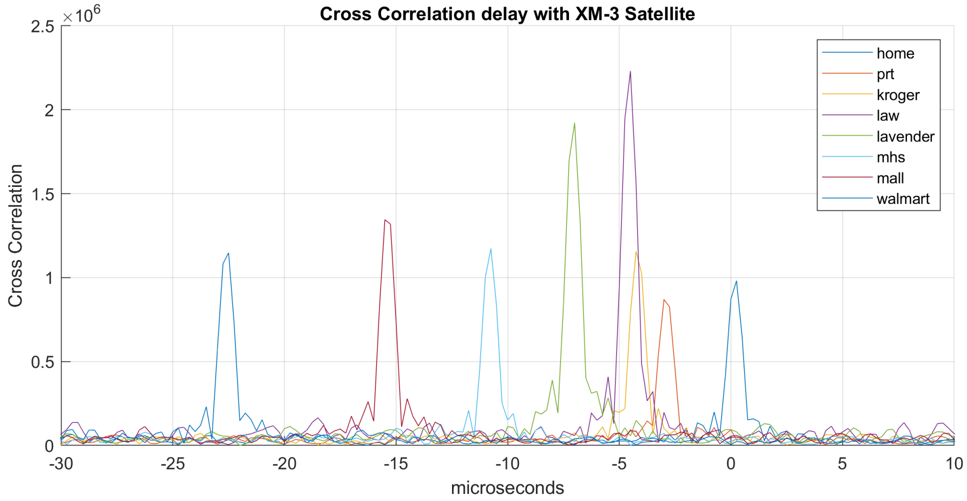

Figure 4.2. Cross-Correlation with XM-3 Satellite

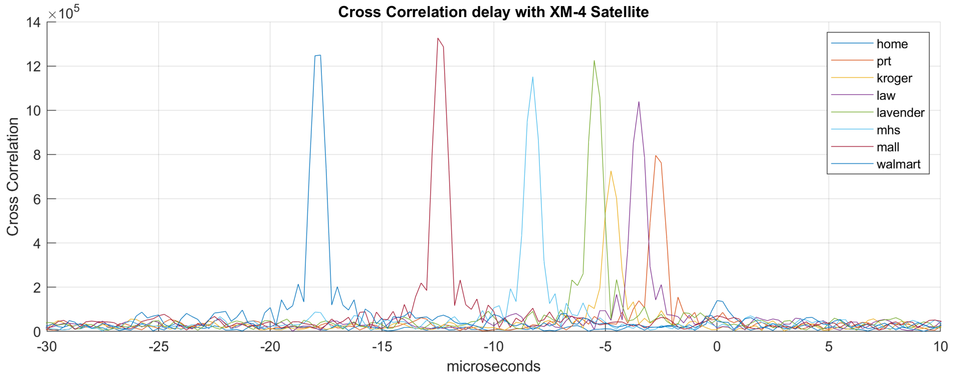

Figure 4.3. Cross-Correlation with XM-4 Satellite

From a coarse first inspection, the data appears to be logically consistent as the geographically distant locations have proportionally offset signal delays.

The following plots show the cross-correlated delay of the two receivers at each location vs. the actual computed path delay distance from the stationary antenna, to the satellite, and then down to the receiver, using the equation 4.1.

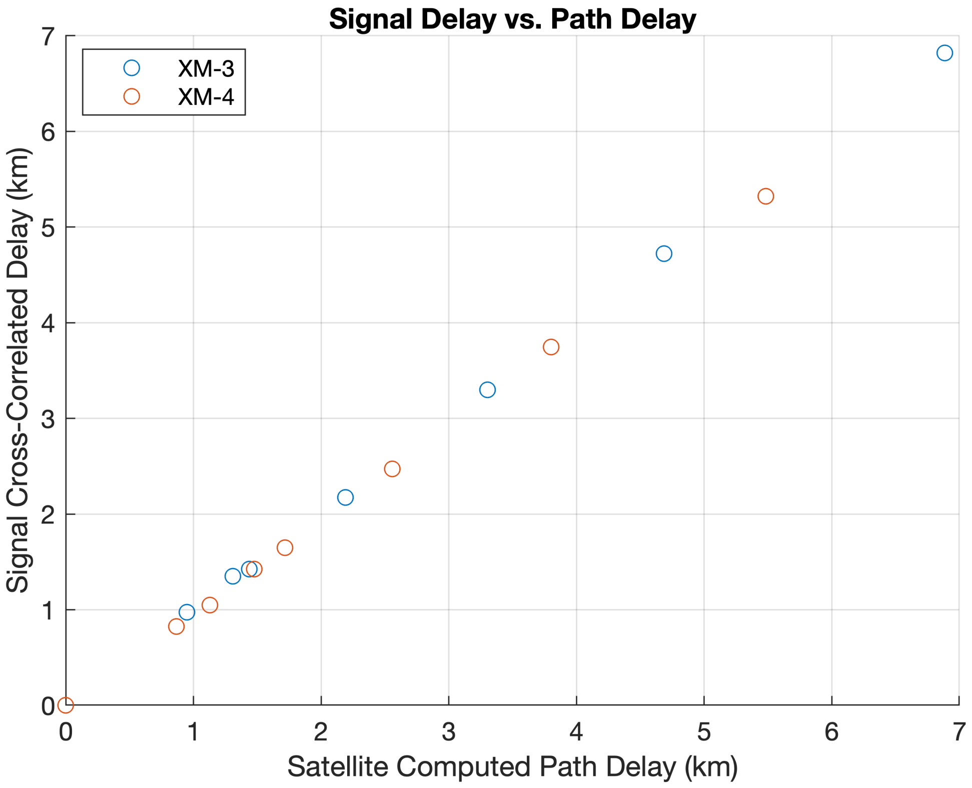

Figure 4.4. Signal Delay v. Path Delay distance for XM-3 and XM-4 Sats

From the above data points, we can observe a clearly linear relationship between the signal delay between two receivers and the satellites respective path delay in kilometers at the exact moment at which the signals were recorded. The correlation coefficient is 0.99970 with a slope of 0.9894, showing nearly a one-to-one relationship between calculating path delay in both ways. **Note** the above data processing only uses a single block of data for one integration period. At all sample sites, exactly one second of data was collected. The cross-correlations use the last four millisecond block of that one second data. Cross-correlations were also performed on **all** 250 blocks of the one second data, and there was little to no shift in the peak of the cross-correlation (at most one index out of 2000). This was made possible by the phase coherence improvements made throughout the semester.

4.2 Path Delay Error Characterization Experiment

Further rover experiments were done in order to verify the findings and better characterize the signal delay to path delay relationship over both time and a much larger quantity of measurements.

4.2.1 Experiment Setup

The antenna was taken on a rover to four locations around Lafayette/West Lafayette.

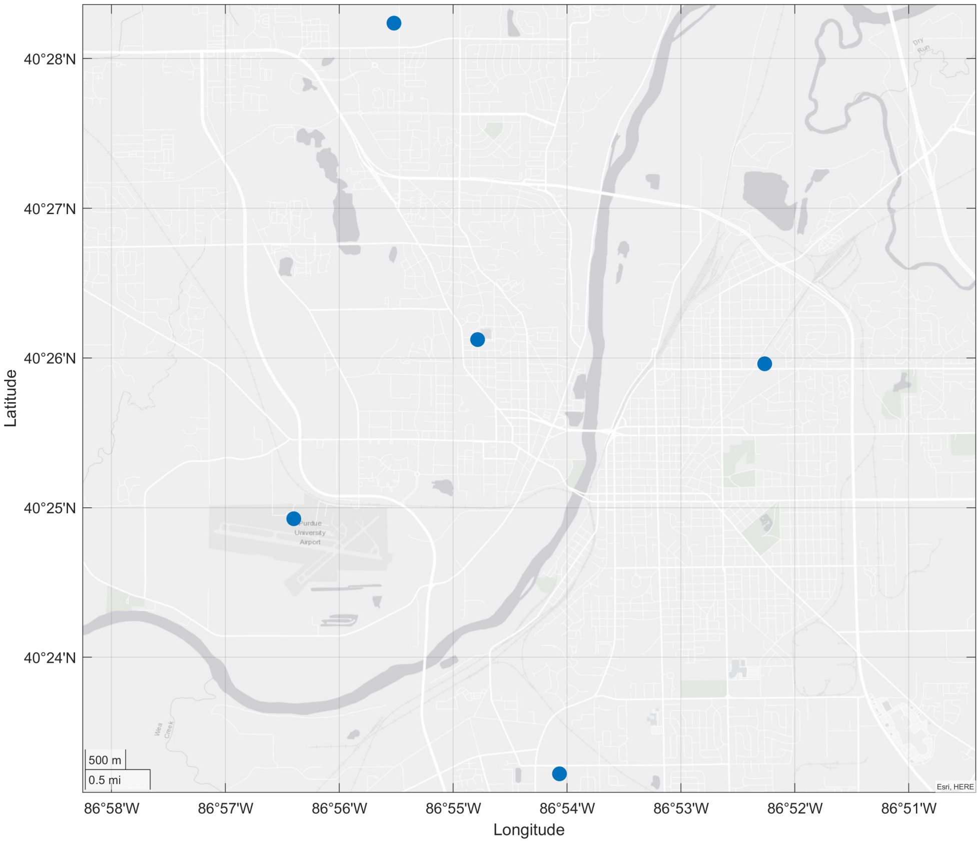

Figure 4.5. 4 outer locations for rover, compared to central antenna

At each location, 25 one-second samples were taken of both XM-3 and XM-4 together, leading to a total of 200 samples.

The data was cross-correlated using the standard four millisecond integration time. The peaks of the cross-correlation were then scaled to obtain the path delay in kilometers. This is plotted against the computed path delay from propagated TLEs, as described in the previous section.

4.2.2 Results

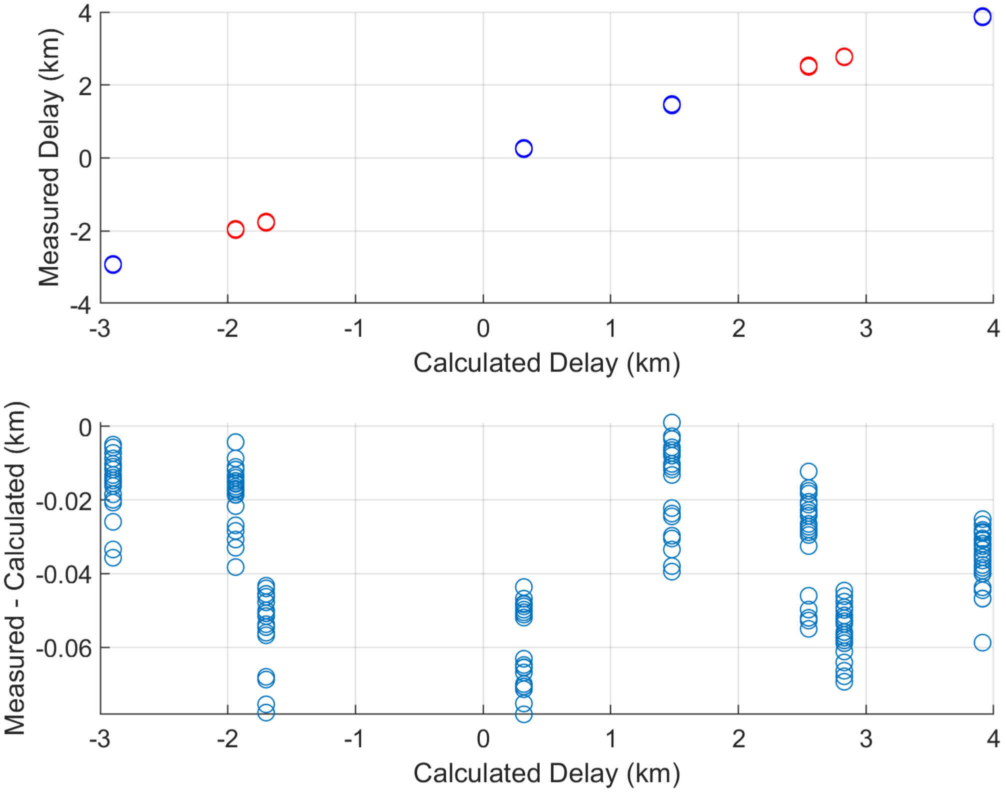

Figure 4.6. XM-3, XM-4 combined correlations and residuals

The correlation coefficient is 0.999921478. The mean error is 30.93 meters and the standard deviation is 29.69 meters, or a variance of 881.4961 meters squared. Furthermore, this value represents the expected variance innate to the TDOA measurement of this system, without the effect of receiver clock bias and ionospheric delay, since the receivers are geographically close to one another.

Since phase synchronization between clocks has been proven over one second, it is possible to extract 250 cross-correlations from each one-second data recording. Currently, only the first four milliseconds are being used. These 250 cross-correlations would also have corresponding GPS times since each is at 4ms multiples after the initial GPS time stamp at which the recording is done. This would yield 50,000 data points instead of the 200 presented above and is a future task.

Furthermore, it is evident from the above plot that at certain locations, the cross-correlation peak shifts between two indices during subsequent recordings. A finer index must be obtained by interpolation of the peak, which is currently not performed but is a future task.

5. DISTRIBUTED ORBIT DERTERMINATION EXPERIMENT

Once the relationship between the signal delay of the measured signals and the path delay scalar was established, an experiment was planned to set up a distributed network of receivers. This network of distributed antennas would provide enough pairs of delay measurements to solve for the position of the satellite being located. Receiver stations were mostly identical, with hardware layout similar to that described in in chapter 3

5.1 Location Selection

Location selection for the experiment was crucial, as the farther apart the receivers are, the better the geometry of the system of equations (improves convergence). However, receiver location is also constrained by space availability as well as the requirement that all receivers be in view of the transmitting satellite. Furthermore, due to hardware limitations, certain components were borrowed from partner schools which thereby became hosts for a receiver stations.

The final list of locations was the following, with the respective number of the station, hence being referred to as station N:

1. Lynwood, Washington (Greater Seattle Area) (47.848910 N, 122.269385 W)
2. Pasadena, California (34.144292 N, 118.113917 W)
3. Boulder, Colorado (40.010433 N, 105.243652 W)
4. West Lafayette, Indiana (Primary Station) (40.416683 N, 86.942610 W)
5. Mississippi State, Mississippi (33.452507 N, 88.787907 W)
6. Rochester, New York (43.304265 N, 77.733740 W)

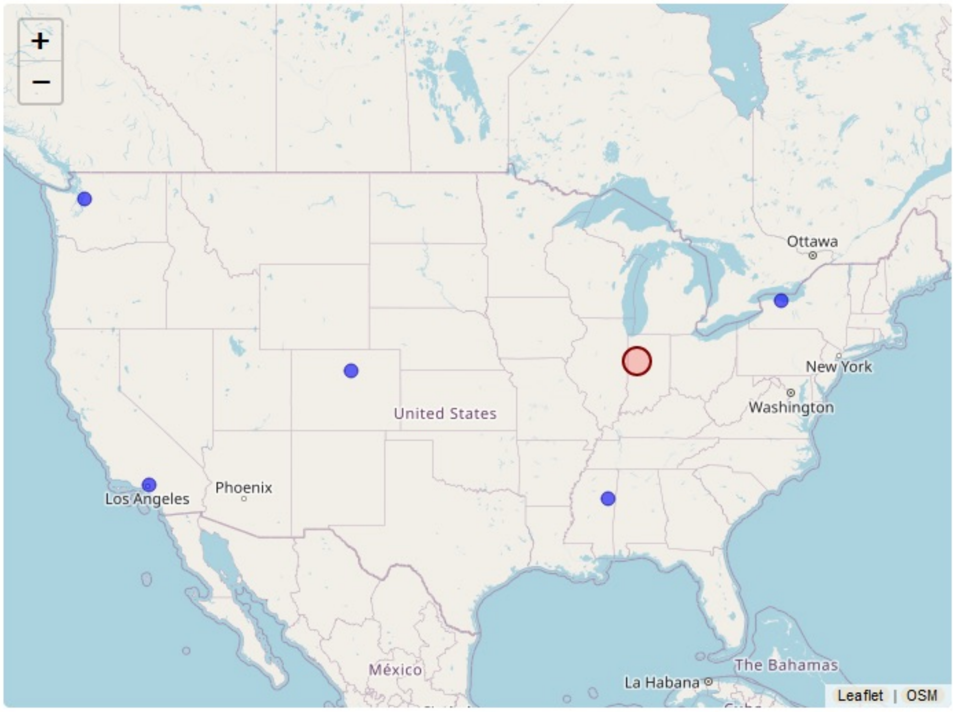

Figure 5.1. Map of Receiver Locations

5.2 Experiment Plan

All six receiver stations were networked through a Wireguard Virtual Private Network (VPN) which was setup through Purdue. This allowed simple and persistent connection from the receiver computers to the Purdue server. The receivers were commanded ahead of time to begin collection at a specified date and time. After this began, the coarse and fine synchronization procedure described in chapter 3 was used.

Three separate collection runs were conducted on the days of April 8th, April 13th, and April 14th. Each run began at Noon, Eastern Standard Time. The first two collections lasted 24 hours to accommodate the full period of a geostationary satellite, with the final run lasting a full week.

Data was collected for exactly 75 milliseconds per minute for the entire collection period.

5.3 Post Processing

A large quantity of raw signal data was collected across all three runs. This data was immediately processed into TDOA observables for ease of handling. Each signal capture from each of the respective six stations was cross-correlated with the others, their peaks interpolated for finer measurement, and then multiplied by the speed of light to compute a path delay. This yielded a total of 36 delay measurements $c\tau_{\mathrm{ij}}$ as shown in the below figure:

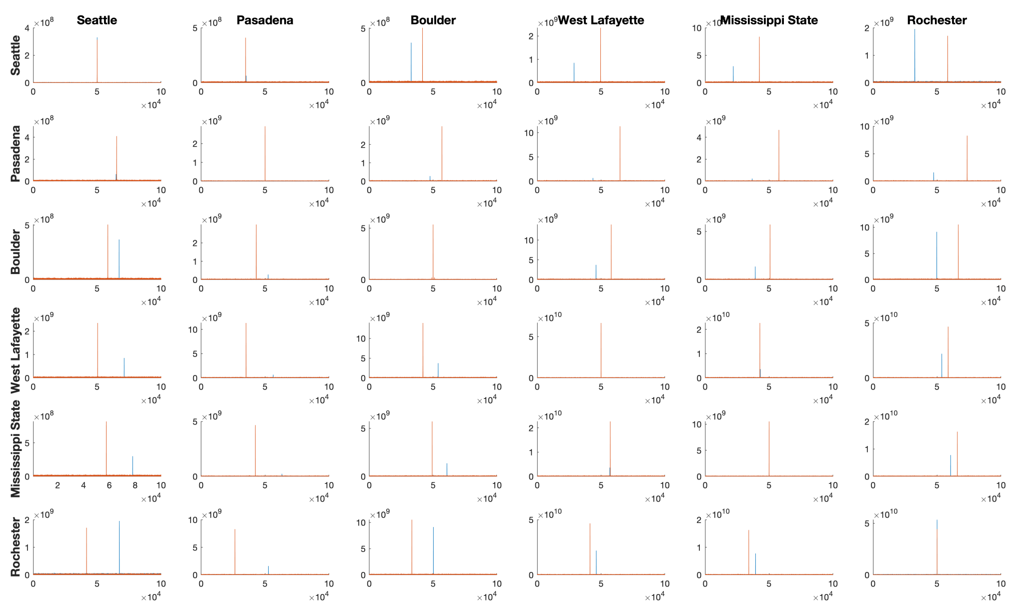

Figure 5.2. Delay Matrix of each station cross-correlated with one another. The main diagonal are autocorrelations. Blue is XM-3, Orange is XM-4

Six of these are auto-correlations of a receiver station (zero time delay). Half of the remaining 30 are merely the opposite cross-correlation of the signal ($c\tau_{\mathrm{ij}} = -c\tau_{\mathrm{ji}}$). This yields 15 unique delay measurements.

5.3.1 Data Validation

A manual visual inspection of these delay observables was performed in order to eliminate outliers and account for unexpected behavior. Below is a plot over time for the 15 unique TDOA measurements.

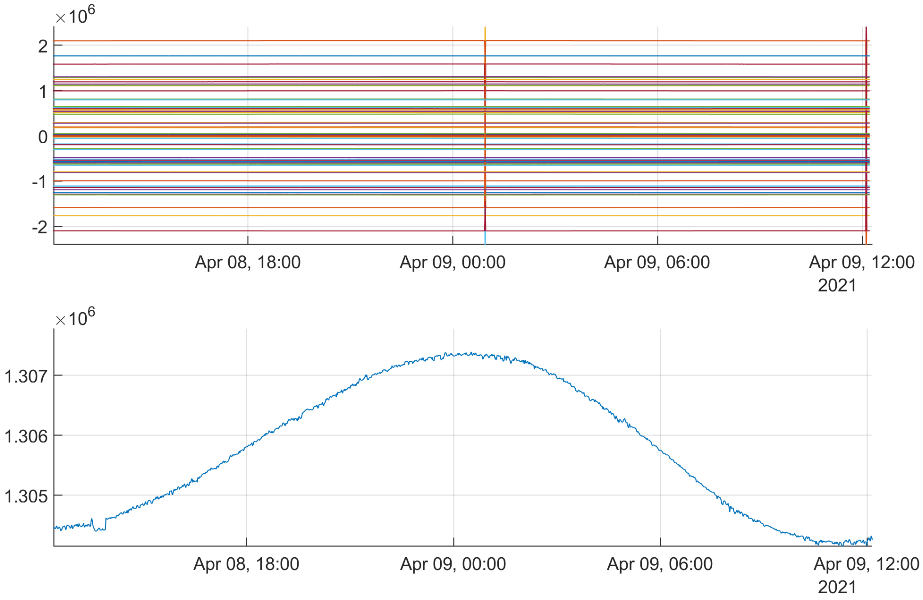

Figure 5.3. April 8th Delay Validation, Delay Measurement in meters, over time

Upon closer inspection (bottom plot), we observe a natural variation of the delay as the geostationary transmitter undergoes motion relative to its fixed location.

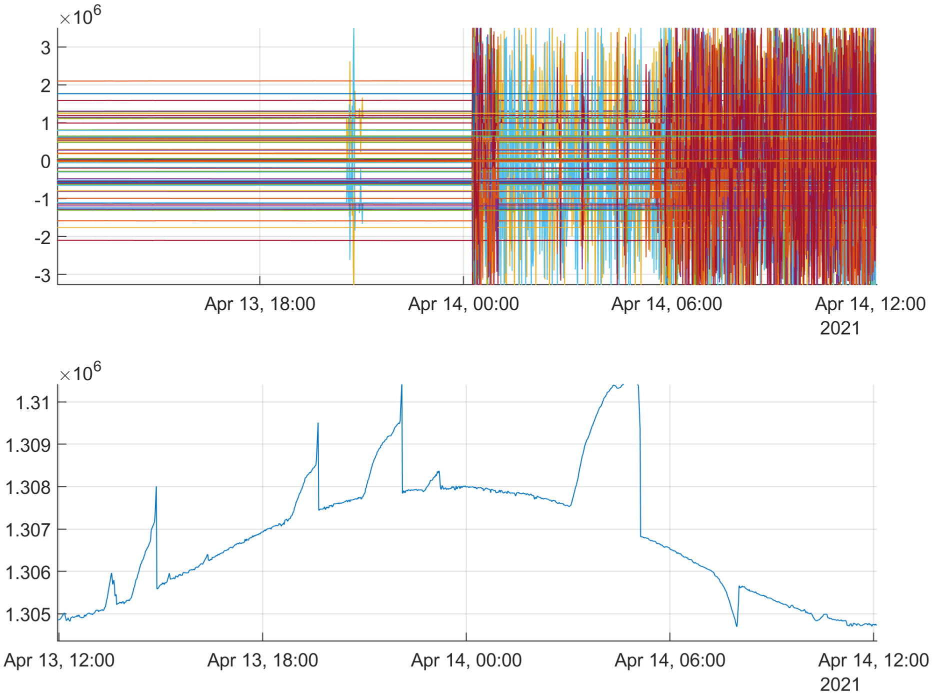

Figure 5.4. April 13th Delay Validation

The April 13th data contains more outliers than the April 8th data. Correlations with receiver 2 experience considerable noise, while correlations with receiver 1 exhibit an unknown drifting behavior periodically. Correlations between receivers 3 through 6 remain intact.

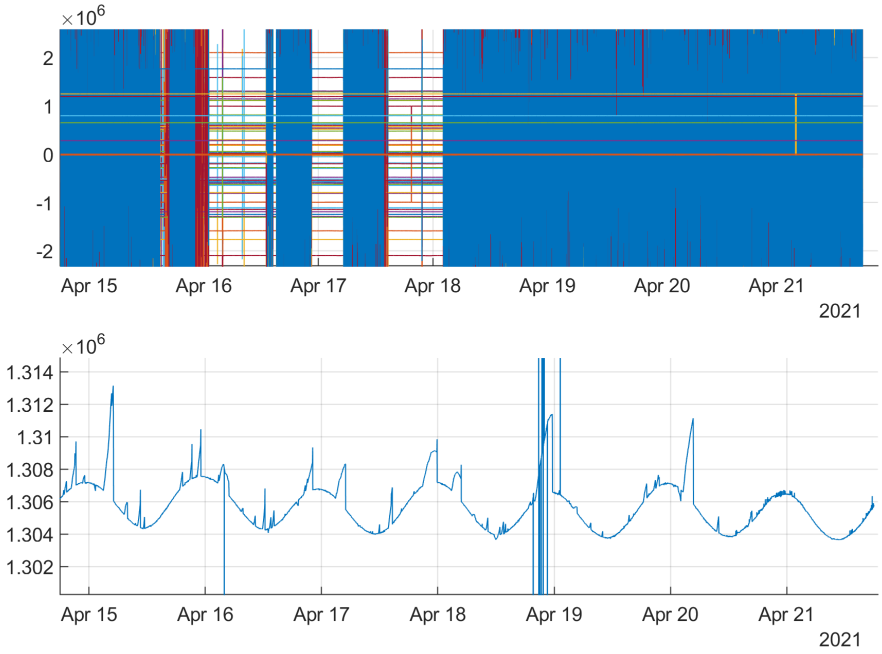

Figure 5.5. April 14th Delay Validation

The April 14th data contains even more anomalies. Receiver 1 continues to exhibit the same unexplained periodic drifting behavior. Receiver 2 is unable to properly acquire the signal and exhibits a signal pattern characteristic with a failure in the front end hardware of the receiver. This is most likely attributed to the fact that the B210 USRP, which serves as receiver 2, is an earlier revision than all the other stations and was known to be unreliable, but used anyway due to hardware constraints. About 48 hours worth of the data from receiver 2 is usable. Again, correlations between receivers 3-6 remain intact.

From the problem definition in Chapter 3, we know that a minimum of three delay measurements are needed for a determined system. From the available datasets, five unique delay measurements are available for the April 8th data, while three are available for the other two datasets.

5.4 Dilution of Precision Comparison

Per the DOP calculations in section 2.3, we can use the initial positions of the receivers to construct a theoretical estimate of the DOP for a given geometry. It is of interest to obtain a theoretical DOP for not only the two XM satellites, but also for various combinations of 4, 5, and 6 receivers, with each station as a different primary station. The theoretical DOP is derived from the H matrix defined in chapter 2.1, with the respective row elements being the vectors of the known receivers. The converged DOP values are obtained from the experimentally determined TDOA values, after they have converged in a least squares solution.

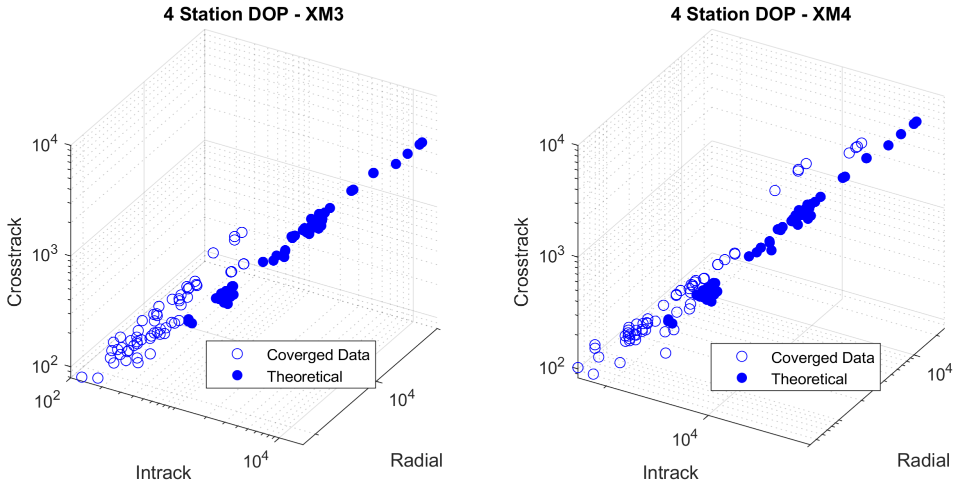

Figure 5.6. 4 Receiver DOP

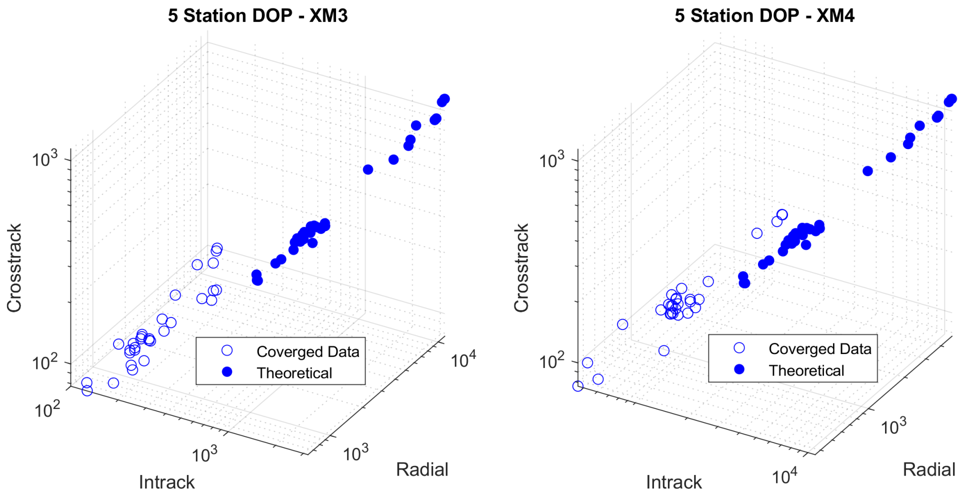

Figure 5.7. 5 Receiver DOP

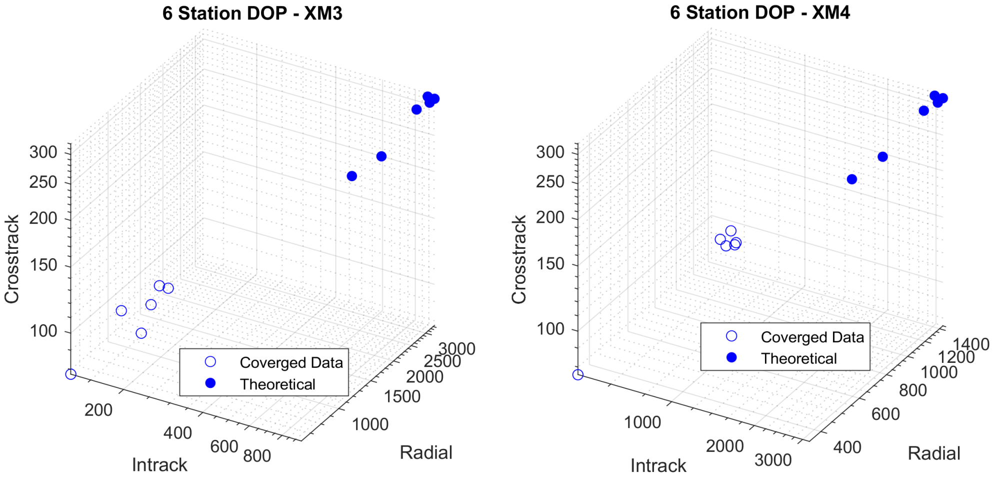

Figure 5.8. 6 Receiver DOP

From the theory, it is expected that there is an overall reduction in component-wise DOP as the number of delay measurements increase. This serves to over-constrain the system and assist in the convergence of the Least squares solution. Furthermore, we observe generally larger DOP values in the radial components relative to cross-track and in-track, in both the converged and theoretical DOP. This corroborates the fact that the problem is not constrained well radially, i.e., all of the receivers are concentrated on the earth facing side of a relative satellite frame.

Table 5.1. Theoretical Predicted DOP for Receiver Configuration

		Radial		In-Track		Cross-Track	
		Mean	Std	Mean	Std	Mean	Std
XM-3	4-Station	12578.17	11699.30	3620.24	3461.05	1181.82	1055.87
	5-Station	5275.51	3087.01	1503.47	885.90	502.89	301.37
	6-Station	2936.89	433.61	831.54	119.72	281.45	41.24
XM-4	4-Station	5507.58	5214.91	11895.51	11050.64	1183.79	1058.55
	5-Station	2298.63	1349.70	4999.88	2923.50	504.64	302.33
	6-Station	1273.99	184.46	2783.71	410.02	282.29	41.19

Table 5.2. Converged DOP from Experiment for Receiver Configuration

		Radial		In-Track		Cross-Track	
		Mean	Std	Mean	Std	Mean	Std
XM-3	4-Station	2209.05	1359.42	453.53	280.18	286.46	221.78
	5-Station	1196.34	405.80	247.56	83.73	147.63	58.52
	6-Station	869.09	126.95	180.41	28.48	105.36	15.40
XM-4	4-Station	2636.07	3761.09	3388.93	4820.34	568.13	853.98
	5-Station	873.83	362.43	1116.16	460.03	171.04	69.03
	6-Station	662.17	139.12	844.88	175.40	130.88	24.86

Tabulating these values, it is apparent that there is a significant reduction in DOP from Theoretical to Converged. The most pronounced DOP reduction is in the in-track direction, which has between a 3.5-8x reduction between XM-3 and XM-4.

Table 5.3. Comparison of Theoretical and Converged DOP

		Difference (Theory-Experimental)			Reduction Factor (Theory/Experimental)		
		Radial	In-Track	Cross-Track	Radial	In-Track	Cross-Track
XM-3	4-Station	10369.12	3166.70	895.35	5.69	7.98	4.13
	5-Station	4079.17	1255.91	355.26	4.41	6.07	3.41
	6-Station	2067.80	651.14	176.09	3.38	4.61	2.67
XM-4	4-Station	2871.52	8506.58	615.66	2.09	3.51	2.08
	5-Station	1424.80	3883.72	333.60	2.63	4.48	2.95
	6-Station	611.83	1938.83	151.41	1.92	3.29	2.16

6. ORBIT DETERMINATION RESULTS

The below image shows a ”bird’s eye” view of the experiment conducted, with a reasonably accurate answer for a strictly kinematic solution. The red dot represents a TLE estimated position for the XM-4 satellite, with the green dot covered underneath showing a close estimate of the satellite’s location. The following ECI image shows that the experimentally determined orbit over a 24 hour period also tracks well with the TLE model.

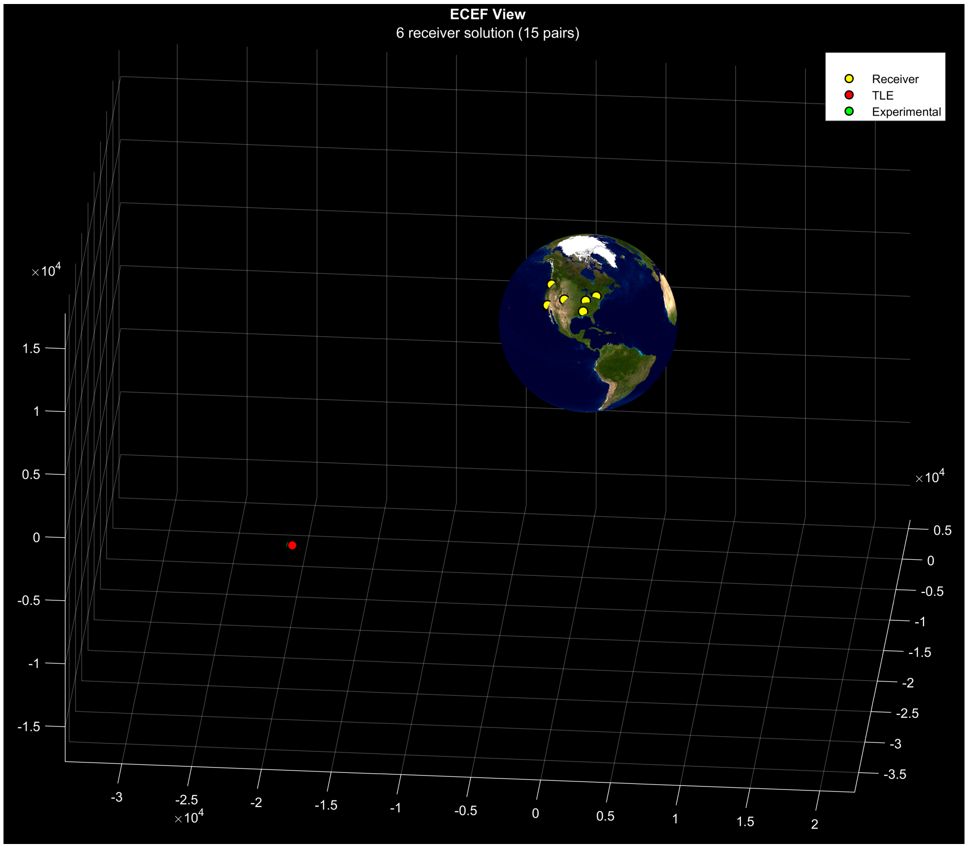

Figure 6.1. ECEF Solution of XM-4

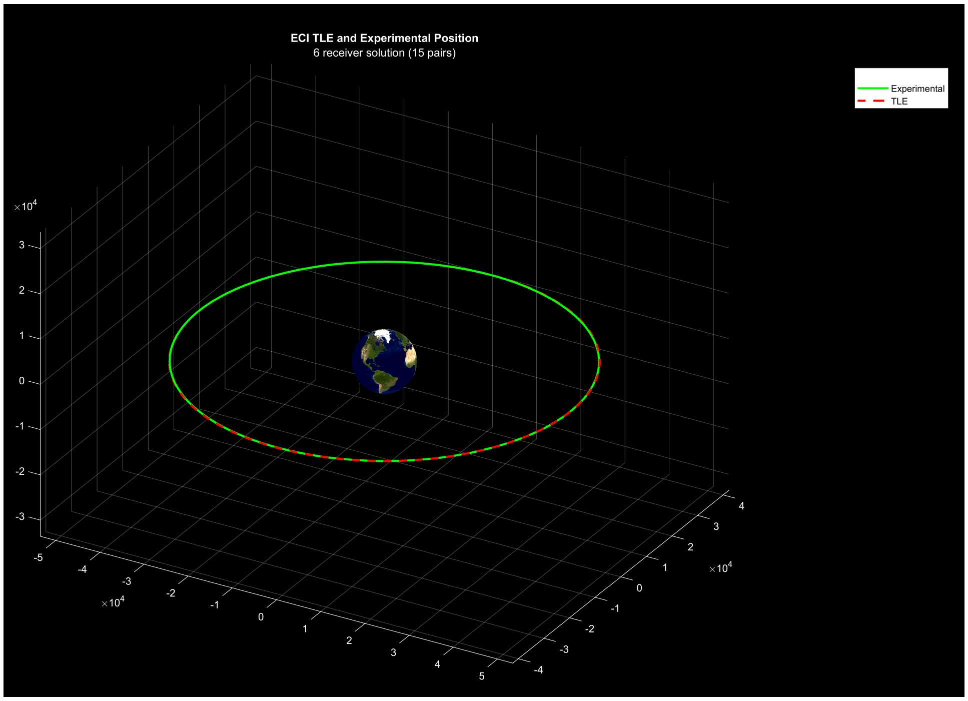

Figure 6.2. ECI Solution of XM-4

6.1 Comparison with TLE Propagation

TLEs provide a relatively convenient data source with which to compare the results of the orbit determination experiment. TLEs were obtained for a given solution period and then propagated with SGP4. This is compared with four- and six-receiver experimental results, which are transformed into a RIC frame given the transmitter's longitude in GEO Ω_G, again assuming 0 inclination:

$$\mathbf{T}_{RIC} = \begin{bmatrix} \cos(\Omega_G) & -\sin(\Omega_G) & 0 \\ \sin(\Omega_G) & \cos(\Omega_G) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

6.1.1 April 8th Dataset

As explained in section 5.3.1, the April 8th collection represents the most compete dataset where all six receivers did not experience anomalies.

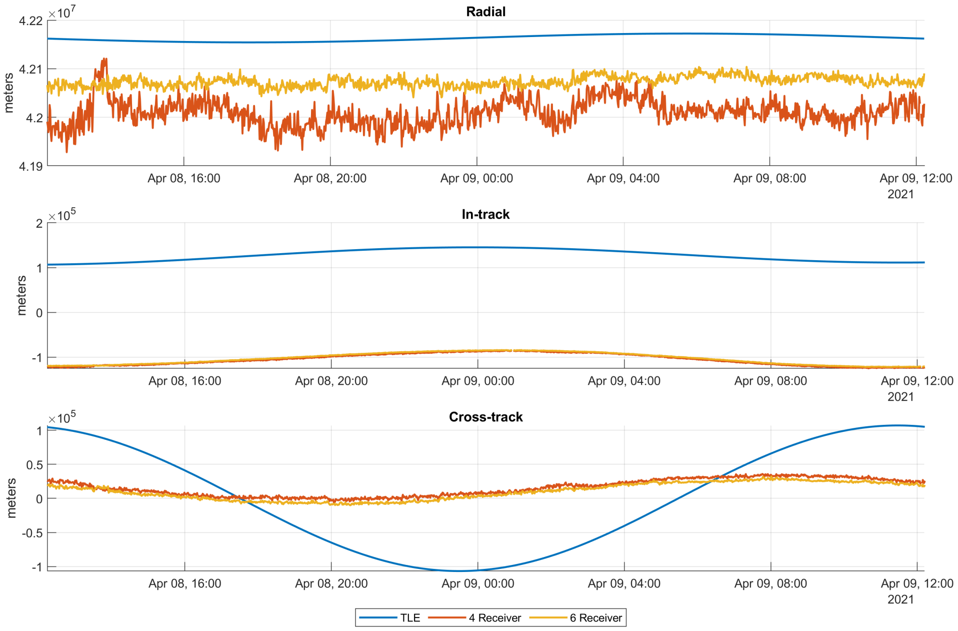

Figure 6.3. XM-3 Comparison of Experimental and TLE - April 8th

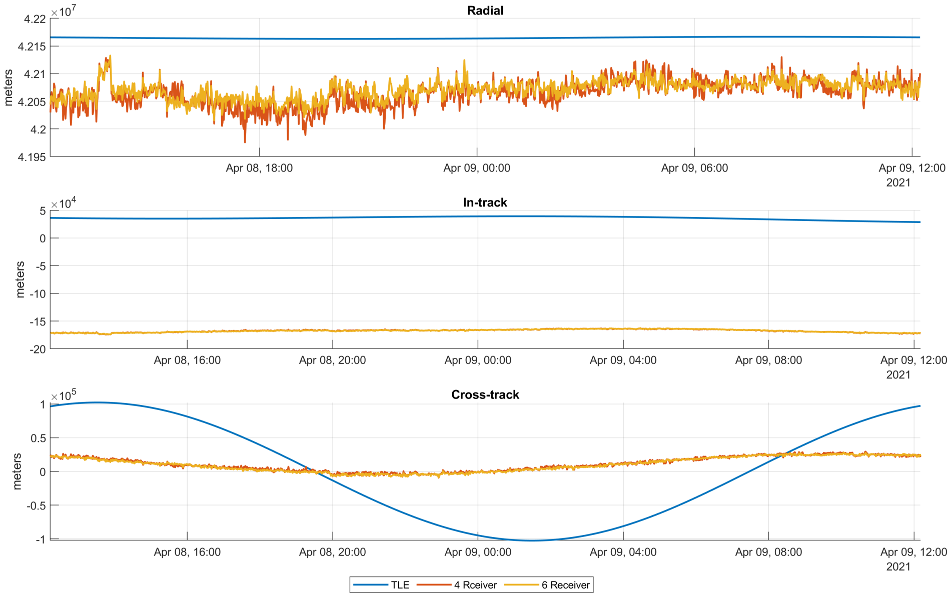

Figure 6.4. XM-4 Comparison of Experimental and TLE - April 8th

We can take note of the relative behavior of the experimental and TLE predictions by observing the error between the two:

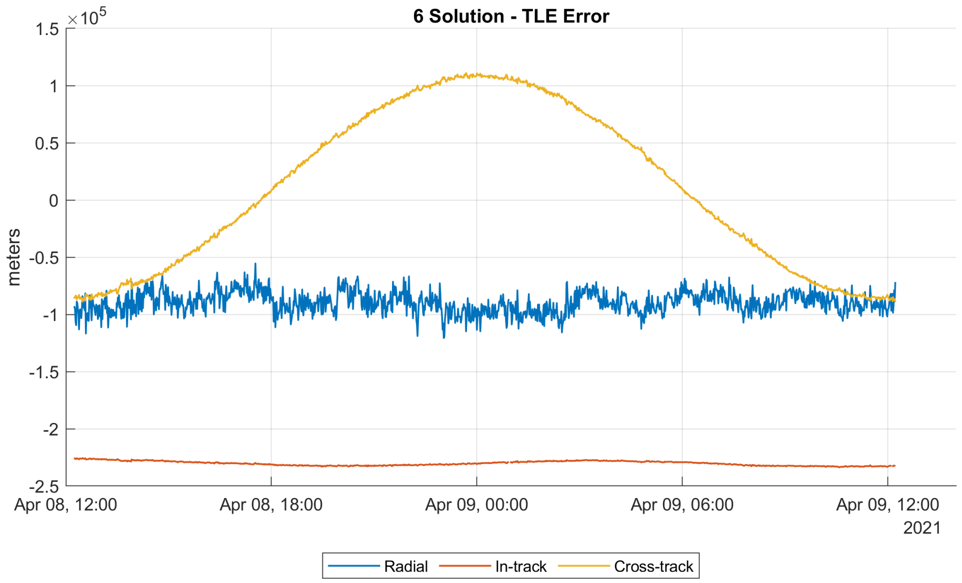

Figure 6.5. XM-3 Error - April 8th

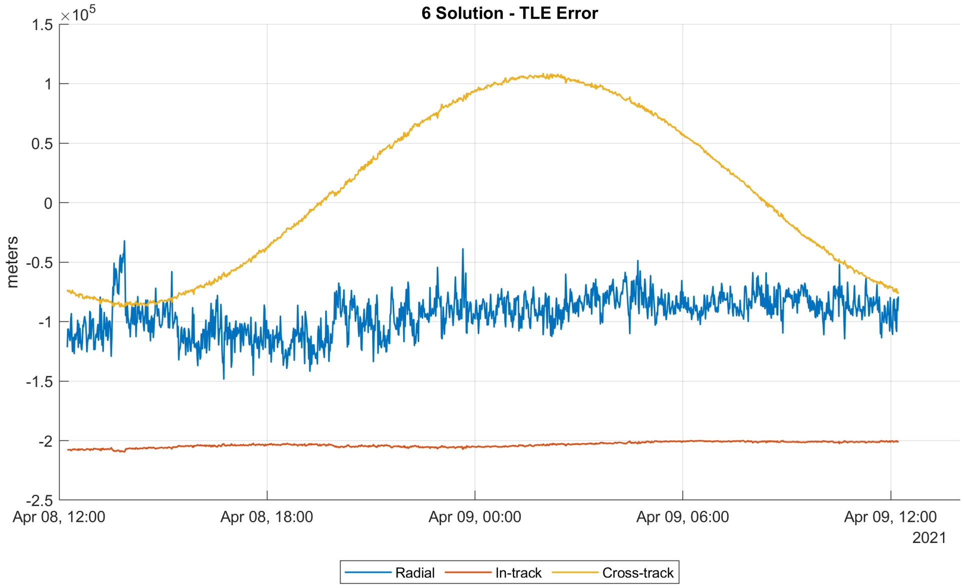

Figure 6.6. XM-4 Error - April 8th

We observe a relatively constant bias between the TLE and experimental results in the radial and in-track directions. In the vertical cross-track axis, there is a periodic error.

Table 6.1. Error Values between 6 receiver solution and TLE

	Radial (meters)		In-Track (meters)		Cross-Track (meters)	
	Mean	Std	Mean	Std	Mean	Std
XM-3	-89863.13	9348.84	-230246.45	1958.80	10205.02	68724.72
XM-4	-93926.80	16532.94	-203467.48	2203.85	10112.00	67433.68

6.1.2 April 14th Dataset

The April 14th dataset is noteworthy for the duration of collection (seven continuous days). Since receivers 1 and 2 suffered from various issues, only the four receiver solution (three unique delay measurements) with receivers 3, 4, 5, and 6 are analyzed.

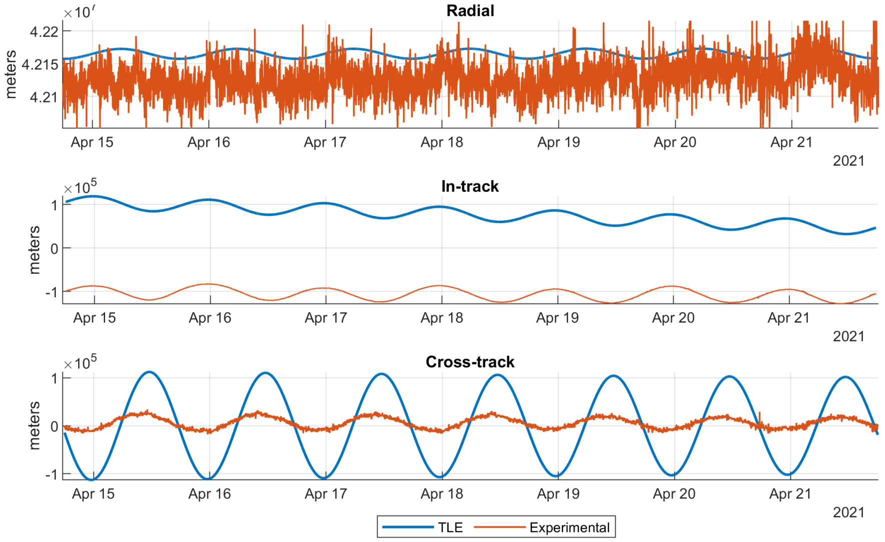

Figure 6.7. XM-3 Comparison of Experimental and TLE - April 14th

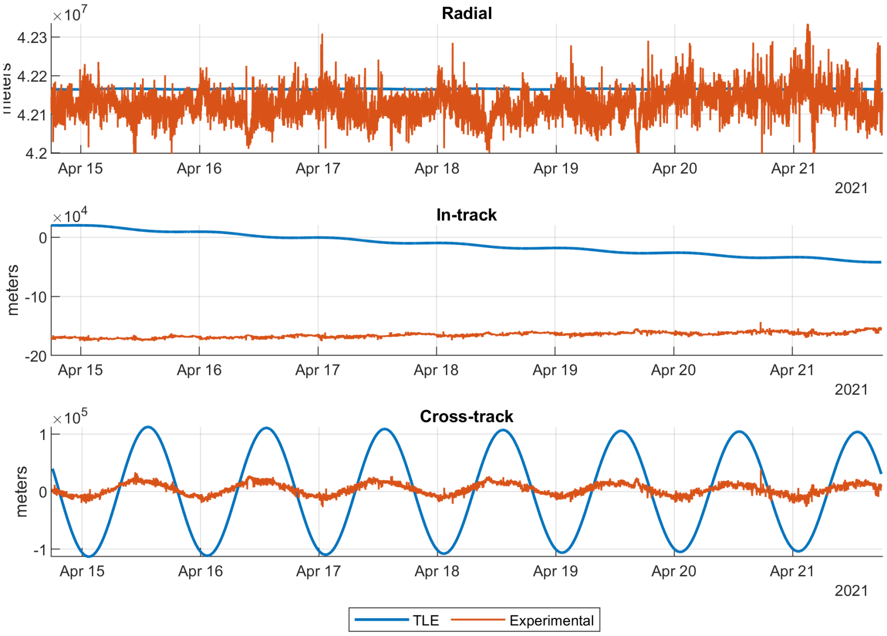

Figure 6.8. XM-4 Comparison of Experimental and TLE - April 14th

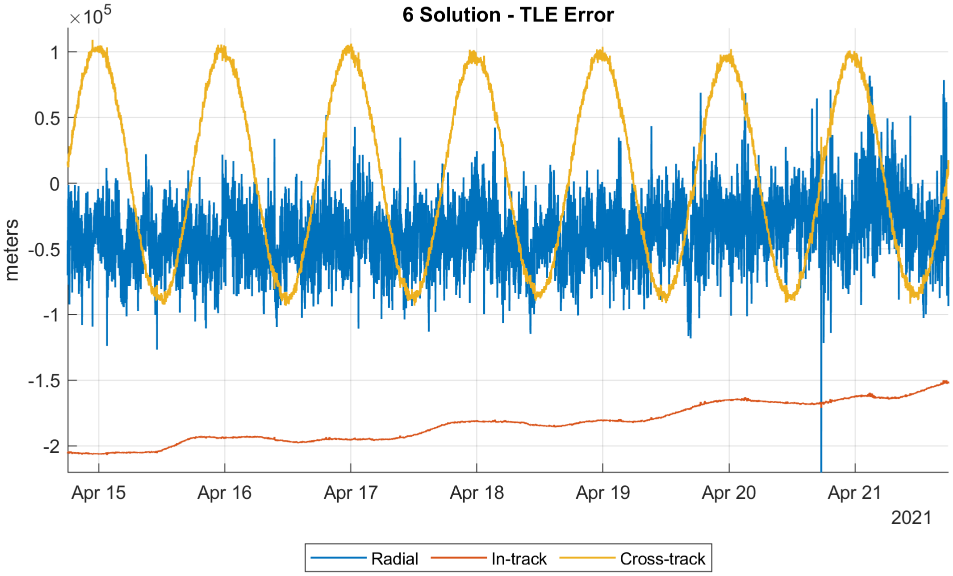

Figure 6.9. XM-3 Error - April 14th

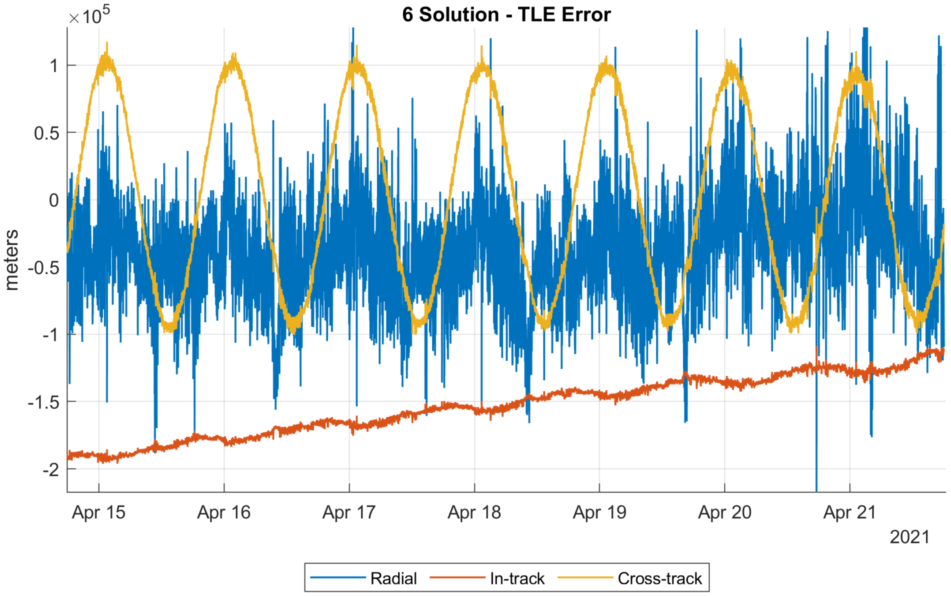

Figure 6.10. XM-4 Error - April 14th

Analyzing over the seven days of data, we observe a few interesting factors. The radial component continues to be the most noisy, most likely due to geometry constraints previously mentioned. The in-track behavior is offset by about 150-180 km from the TLEs, but the periodic motion is similar. In the cross-track component, the same periodic behavior as in the the April 8th dataset is present. While the TLE prediction has an amplitude about five times higher, the phase appears well-aligned.

Table 6.2. Error Values between 6 receiver solution and TLE

	Radial (meters)		In-Track (meters)		Cross-Track (meters)	
	Mean	Std	Mean	Std	Mean	Std
XM-3	-37899.55	24044.79	-182919.63	14530.86	6176.82	65539.10
XM-4	-37955.83	38497.59	-153242.10	21874.10	4117.68	67532.15

6.2 Comparison of Different Primary Stations

It is of interest to note how changing which primary station is used affects the solution itself.

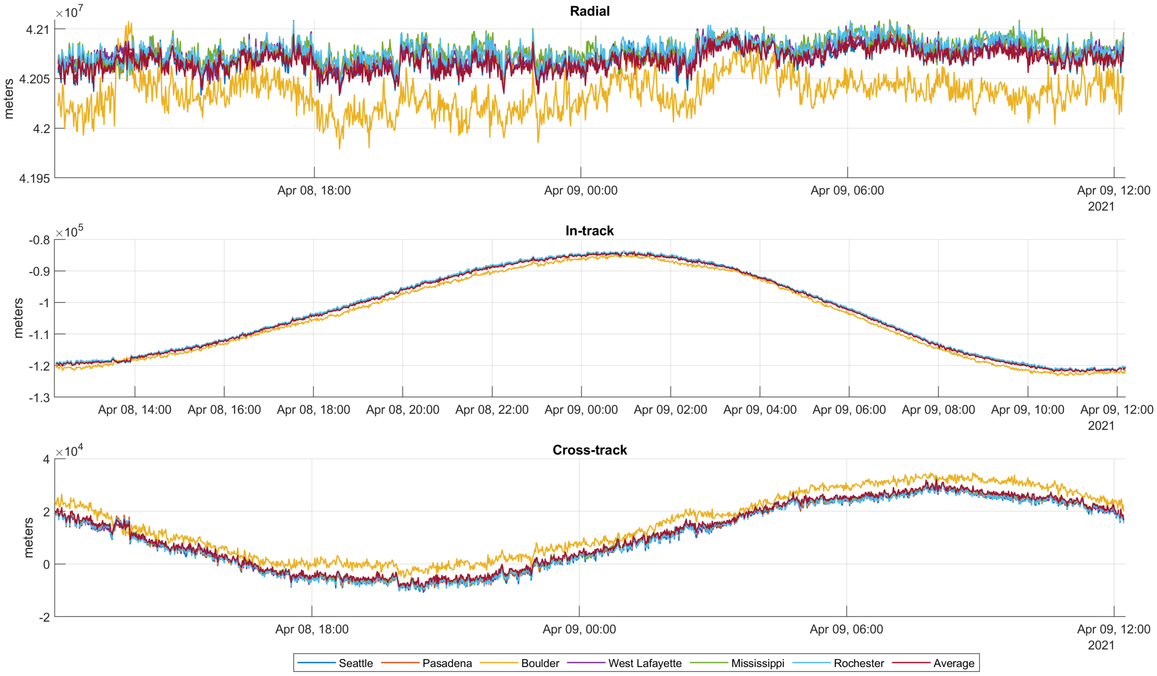

Figure 6.11. Change in solution based on chosen primary station

From the above image, we can see that using Boulder as the primary station produces a solution that remains somewhat separate from the other solutions. It was determined that this was potentially due to the fact that the Boulder station's coordinate value was at a significant offset to the actual location of the S-band antenna. While the S-band antenna was placed in a separate radome, the GPS antenna used to locate the station was actually used from a separate lab feed for the GPS signal. Hence, the location of this receiver was considerably offset as the GPS antenna was not near the S-band antenna. Further work needs to be done to characterize the sensitivity of the kinematic solution to errors in the receiver coordinates.

6.3 Comparison with Optical Observations

On the night of April 20th and April 21st, optical observations were taken of the XM-3 and XM-4 satellites. These observations are performed by an optical telescope run by Dr. Carolin Frueh's Space Information Dynamics group. These observations were done to provide a point of comparison. These observations were converted from topocentric coordinates to the same ECEF, and subsequently, RIC frame.

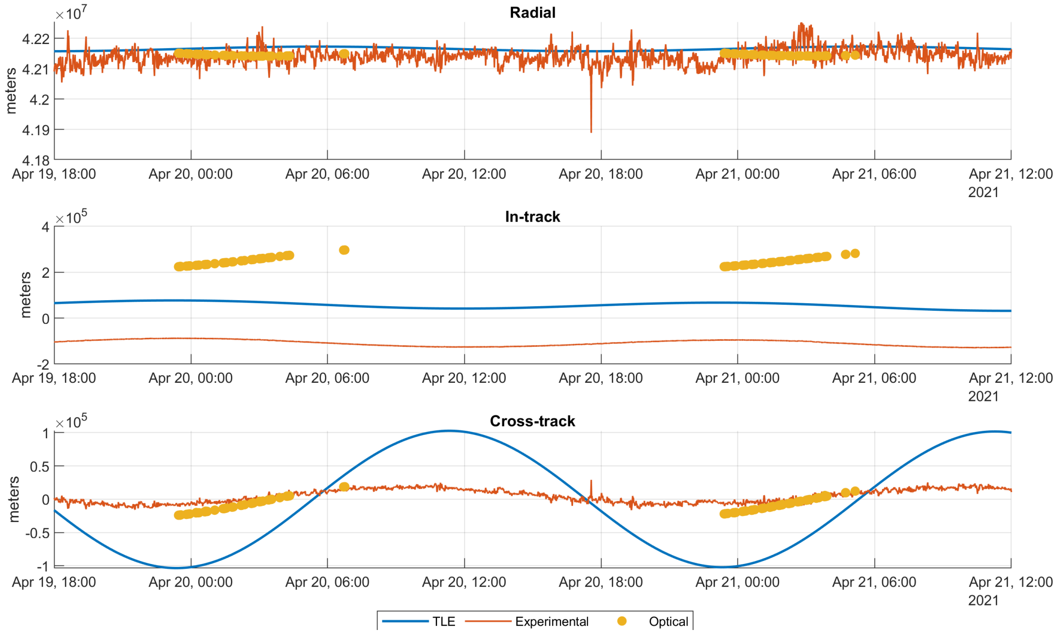

Figure 6.12. Optical Observations for XM-3

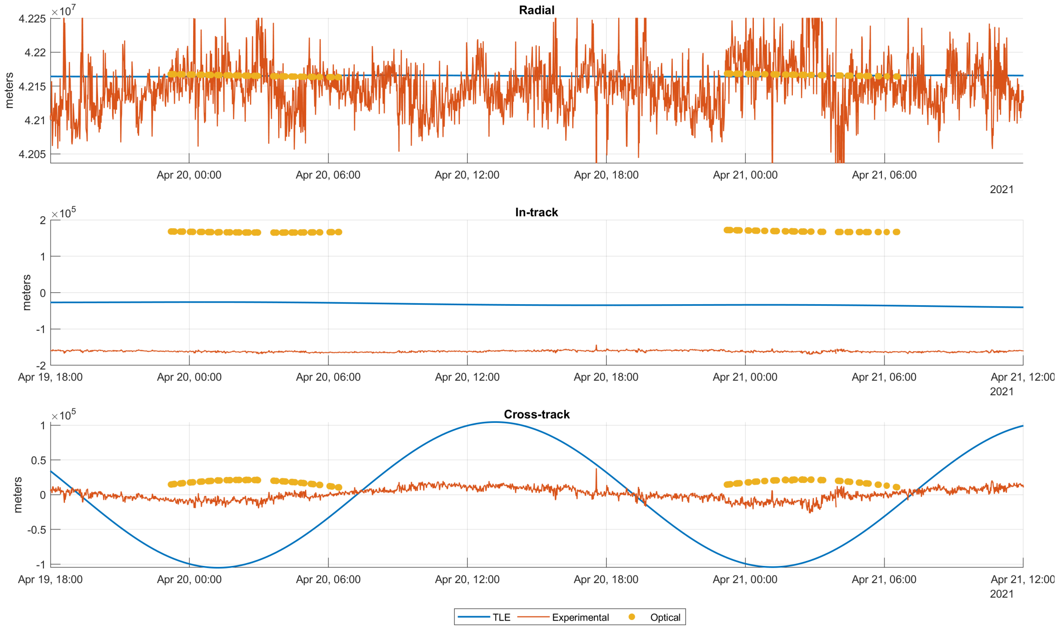

Figure 6.13. Optical Observations for XM-4

Comparison with optical data differs from the TLE data. In the cross-track and in-track components, there is an increased mean error but a decreased standard deviation, as the optical components do not vary nearly as much. However, in the radial component, the average error is significantly less.

Table 6.3. Optical Error compared to TLE Error (std. in meters^2, mean in meters)

		Radial		In-Track		Cross-Track	
		Mean	Std	Mean	Std	Mean	Std
TLE Error	XM-3	-37899.55	24044.79	-182919.63	14530.86	6176.82	65539.10
	XM-4	-37955.83	38497.59	-153242.10	21874.10	4117.68	67532.15
Optical Error	XM-3	6931.47	33709.45	-341392.17	12909.63	7644.04	6716.14
	XM-4	1196.93	43672.08	-330189.91	2555.67	-25857.50	7375.75

6.4 Comparison with Predicted Deviation

The four-receiver DOP computations made in the previous chapter can be used to make a prediction of the expected variance in the least squares results, with respect to a truth:

$$\sigma_{\text{measurement}} \times DOP = \sigma_{\text{sat,predicted}} \tag{6.2}$$

Table 6.4. Comparison of experimental and predicted standard deviation (meters)

		Radial	In-Track	Cross-Track
TLE Std	XM-3	24044.79	14530.86	65539.10
	XM-4	38497.59	21874.10	67532.15
Optical Std	XM-3	33709.45	12909.63	6716.14
	XM-4	43672.08	2555.67	7375.75
Predicted Std	XM-3	65586.77	13465.39	8505.09
	XM-4	78264.86	10617.26	16867.80

The above table looks at the standard deviation of the error between the experimental data, and TLE/optical data as truth comparisons. The final two rows are the predicted standard deviation from the DOP calculations made in chapter 5, with the measurement error characterized of the system in chapter 4. We generally observe that the experimental variance is of the same order of magnitude but generally less than the predicted variance, thus confirming that we are for the most part able to characterize, that for a given TDOA measurement error, we can account for the error in computed the satellite position.

7. CONCLUSION

7.1 Recommendations and Future Work

As this experiment might be repeated in the future, there are some lessons learned and future improvements that that would be necessary. Additionally, since many experimental components were done using at-hand or borrowed hardware, it is clear that some aspects could be improved by acquiring more precise or reliable hardware

7.1.1 Hardware

- The B210 USRP used in this experiment [18] is a software-defined radio platform built around the Analog Devices AD9361 RF ADC component. This component, while providing a cost-effective ADC for the ettus package, is a decade-old design and suffers from signal quality issues. For example, for signal measurements with lower integration times, the AD9361 exhibits artifacting at the center frequency and at every 5 MHz interval above and below the center frequency. This phenomena was the most visible during processing for this experiment and had to be filtered out; however, the data could be be suffering from additional effects that are not as well characterized.

- The front-end antenna used in this experiment was a relatively cheap, low-gain puck antenna intended for commercial purposes. For a transmitter in GEO, a directional antenna with higher gain would have improved signal-to-noise ratio.

- The default internal GPSDO was used, which uses GPS time and an internal oscillator to discipline the sampling of the signal. The results in section 3.3.2 showed that even after using various startup strategies, phase synchronization across long distances was unreliable over more than 100 milliseconds. A more accurate external reference oscillator is needed for

7.1.2 Experiment

- In some locations, an external GPS source was used, which generally came from an antenna that was offset from where the S-band antenna was mounted. For future experiments, the GPS antenna should be no more than one meter away from the receiving antenna.
- The Wireguard VPN system worked well to properly command and control the network of receivers and maintain a robust connection under adverse network situations.
- Additional receivers are necessary to cancel out transmitter clock bias

7.1.3 Software and Post Processing

- The cross-correlation peak can be better detected by fitting a sinc function, but this was not computationally feasible for the dataset size
- The current results do not consider ionosphere and receiver clock bias terms. In post-processing, it would be useful to modify the TDOA equation to add ionospheric delay, as well as solving for clock bias.

7.2 Summary

In this paper, a proof-of-concept experiment was set up to measure the Time Delay of Arrival (TDOA) of S-band signals from the XM constellation. In order to achieve this, experiments were done to validate the TDOA phase measurement and validate that geometric path delay is proportional to this TDOA. A simplified receiver was designed along with a circuit, software-defined radio, and PC. The receiver station hardware and software were designed to be deployed with minimal setup and to operate in an automated fashion.

A total of nine days worth of data was collected of both XM-3 and XM-4. These signals were processed into delay measurements, which were subsequently used to solve for the transmitter's position.

While the transmitter was in GEO, we observed in-track and cross-track errors less than 10 km and 100-200 km in the radial direction with respect to TLEs. Furthermore, comparing with the optical observations, there was significantly higher agreement. Most importantly, we have been able to show that we can account for the transmitter position error and variance as primarily a function of the discrepancies in the receiver TDOA measurement.

In conclusion, TDOA-based orbit determination serves as a viable method of orbit determination. It provides key advantages over other methods when observing a transmitting source, such as the lack of pointing errors and the ability to track multiple satellites simultaneously, as exhibited in this experiment.

REFERENCES

[1] D. Entekhabi, S. Yueh, P. E. O'Neill, *et al.*, "Smap handbook–soil moisture active passive: Mapping soil moisture and freeze/thaw from space," 2014.

[2] E. R. Reisin and J. Scheer, "Evidence of change after 2001 in the seasonal behaviour of the mesopause region from airglow data at el leoncito," *Advances in Space Research*, vol. 44, no. 3, pp. 401–412, 2009.

[3] W. Bertiger, S. D. Desai, A. Dorsey, *et al.*, "Sub-centimeter precision orbit determination with gps for ocean altimetry," *Marine Geodesy*, vol. 33, no. S1, pp. 363–378, 2010.

[4] R. Guo, X. G. Hu, B. Tang, *et al.*, "Precise orbit determination for geostationary satellites with multiple tracking techniques," *Chinese Science Bulletin*, vol. 55, no. 8, pp. 687–692, 2010, ISSN: 10016538. DOI: 10.1007/s11434-010-0074-x.

[5] C. Fruh and T. Schildknecht, "Combination of light curve measurements and orbit determination for space debris identification," in *Proceedings of the 62nd International Astronautical Congress, International Astronautical Federation*, vol. 3, 2011, pp. 1873–1881.

[6] G. J. Bierman and C. L. Thornton, "Numerical comparison of kalman filter algorithms: Orbit determination case study," *Automatica*, vol. 13, no. 1, pp. 23–35, 1977.

[7] S. Schaer, "Société helvétique des sciences naturelles, commission géodésique, mapping and predicting the earth's ionosphere using the global positioning system, vol. 59, institut für geodäsie und photogrammetrie, eidg," *Technische Hochschule Zürich*, 1999.

[8] M. Wang, T. Shan, M. Li, L. Liu, and R. Tao, "Gnss-based orbit determination method and flight performance for geostationary satellites," *Journal of Geodesy*, vol. 95, no. 8, pp. 1–15, 2021.

[9] Y. Huang, X. G. Hu, X. Z. Zhang, *et al.*, "Improvement of orbit determination for geostationary satellites with VLBI tracking," *Chinese Science Bulletin*, vol. 56, no. 26, pp. 2765–2772, 2011, ISSN: 10016538. DOI: 10.1007/s11434-011-4647-0.

[10] B. J. Haines, S. M. Lichten, R. P. Malla, and S.-C. Wu, "A Review of GPS-Based Tracking Techniques for TDRS Orbit Determination," Tech. Rep., Nov. 1993. [Online]. Available: https://ntrs.nasa.gov/search.jsp?R=19940018784.

[11] R. M. Fuster, M. F. Usón, and A. B. Ibars, "Interferometric orbit determination for geostationary satellites," *Science China Information Sciences*, vol. 60, no. 6, Jun. 2017. DOI: 10.1007/s11432-016-9052-y. [Online]. Available: https://doi.org/10.1007/s11432-016-9052-y.

[12] A. Broquetas Ibars, J. Nicolás Álvarez, X. Carreño Megias, O. Fusté i Lara, and A. Aguasca Solé, “Orbit determination for hydroterra mission. an interferometric approach,” in *Remote Sensing Instruments, ARSI’19+ Ka-band EO Radar Mission, KEO’19: proceedings, 11-13 november 2019, ESA-ESTEC, Noordwijk, The Netherlands*, European Space Agency (ESA), 2019, pp. 1–6.

[13] K. Akiyama, K. Bouman, and D. Woody, “First m87 event horizon telescope results. i. the shadow of the supermassive black hole,” *Astrophysical Journal Letters*, vol. 875, Apr. 2019.

[14] K. Akiyama, A. Alberdi, W. Alef, *et al.*, “First m87 event horizon telescope results. ii. array and instrumentation,” *The Astrophysical Journal Letters*, vol. 875, Apr. 2019.

[15] T. Yunck and S. Wu, “Orbit determination of geosynchronous satellites by vlbi and differential vlbi,” in *Astrodynamics Conference*, 1982, p. 1446.

[16] S. DiPierro, R. Akturan, and R. Michalski, “Sirius xm satellite radio system overview and services,” in *2010 5th Advanced Satellite Multimedia Systems Conference and the 11th Signal Processing for Space Communications Workshop*, 2010, pp. 506–511. DOI: 10.1109/ASMS-SPSC.2010.5586866.

[17] R. Michalski, “An overview of the xm satellite radio system,” in *20th AIAA International Communication Satellite Systems Conference and Exhibit*, 2002, p. 1844.

[18] a. N. I. B. Ettus Research, *Usrp b210 usb software defined radio (sdr).* [Online]. Available: https://www.ettus.com/all-products/ub210-kit/.

[19] A. E. E. Rogers and J. M. Moran, “Coherence limits for very-long-baseline interferometry,” eng, *IEEE Transactions on Instrumentation and Measurement*, vol. IM-30, no. 4, pp. 283–286, 1981, ISSN: 0018-9456.

[20] *N2yo.* [Online]. Available: https://www.n2yo.com/.

[21] D. Vallado and P. Crawford, “Sgp4 orbit determination,” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008, p. 6770.

A. POLYNOMIAL INTERPOLATION OF PEAKS

In order to obtain a finer measurement of path delay from the cross-correlation, the peak of the cross-correlation was interpolated with a polynomial. This estimate results in a finer ”sub-index” for the cross-correlation peak, which in turn produces a measurement with higher resolution than just using each discrete index for the highest cross-correlation.

Interpolation is performed with the five immediate points around the cross-correlation peak. The interpolation is done with 2nd through 7th order polynomials.

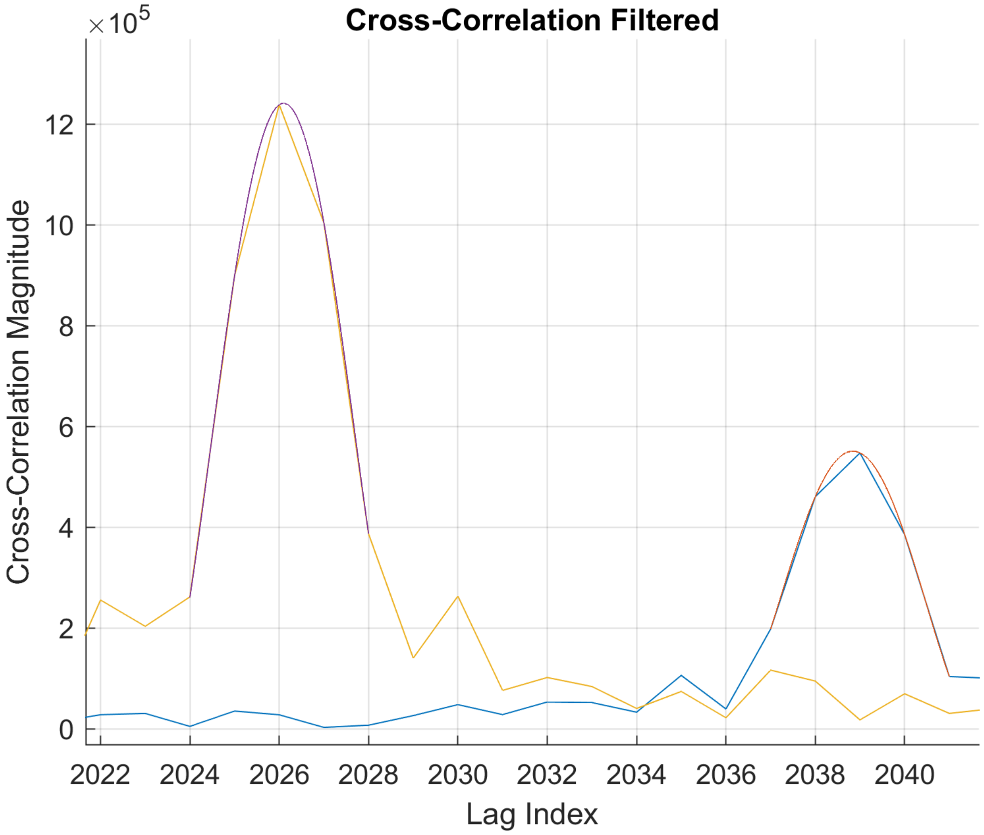

Figure A.1. Sample polynomial fit

The below plots show the new interpolated measured delay points vs. computed delay, plotted in the top half. The bottom half shows the distribution of the error (measured - calculated) across each site. Again, there are four sites; however, since two satellites are collected simultaneously, there are two separate measurements. XM-3 is represented in blue and XM-4 in red.

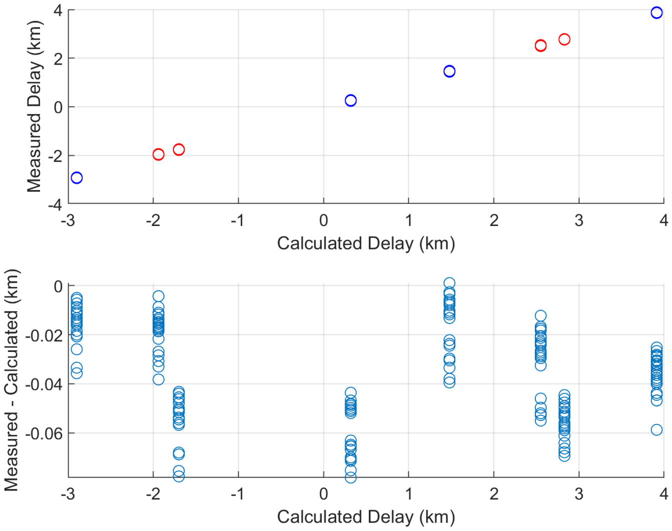

Figure A.2. 2nd Order fit

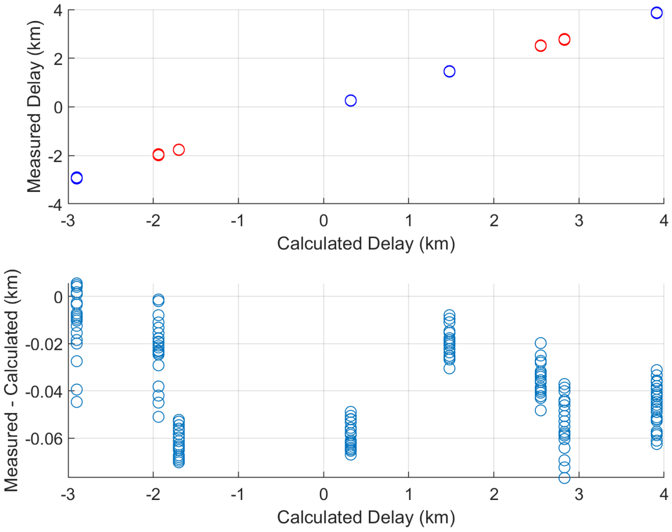

Figure A.3. 3rd Order fit

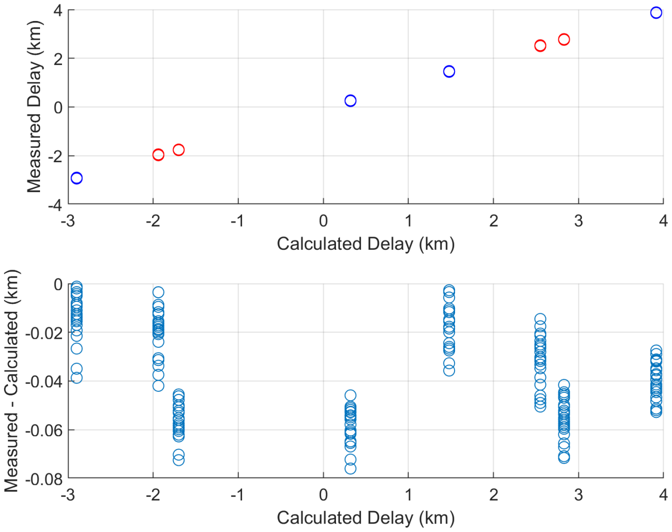

Figure A.4. 4th Order fit

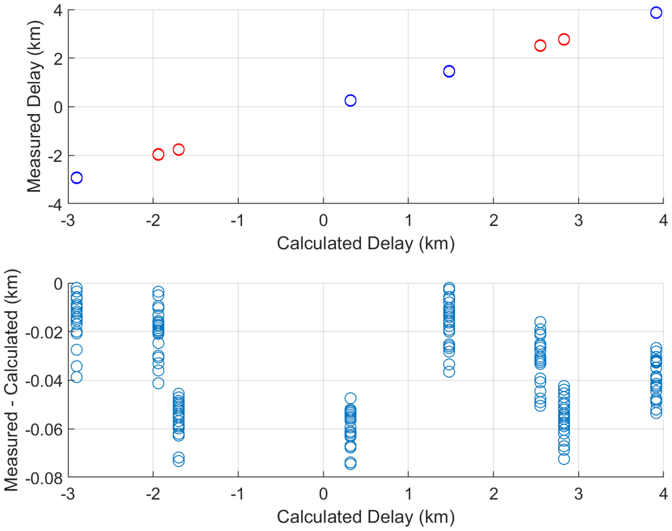

Figure A.5. 5th Order fit

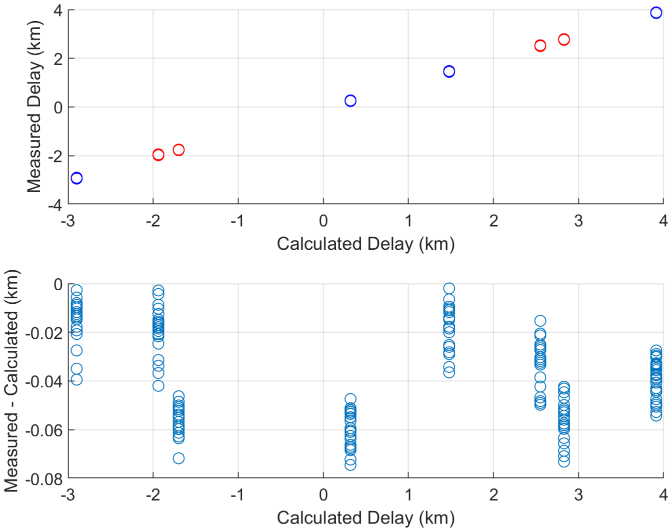

Figure A.6. 6th Order fit

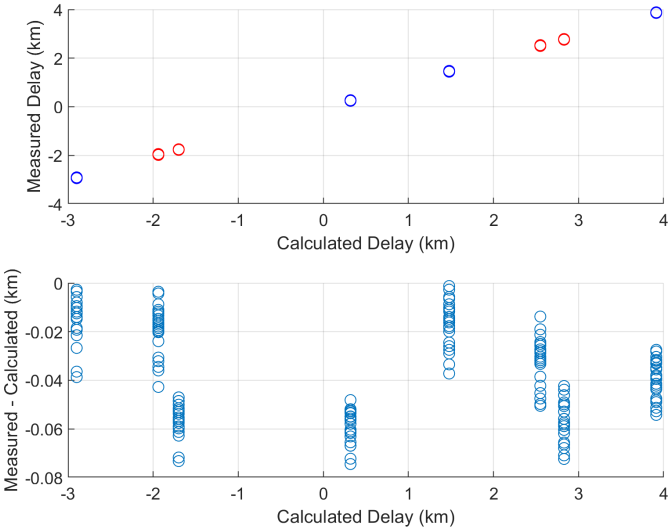

Figure A.7. 7th Order fit

In summary, across all polynomial fits tried, there is about a 10 meter improvement in mean error, but the standard deviation is about six meters greater.

The following table shows the results of the interpolated mean error and standard deviation.

Table A.1. Standard deviation and mean error in meters

	2nd	3rd	4th	5th	6th	7th
R^2	0.9999662	0.9999645	0.9999681	0.9999676	0.9999676	0.9999679
Mean Error	19.9335	20.5373	19.5991	19.7348	19.7288	19.6384
STD	35.0836	39.0060	36.0111	36.3408	36.3520	36.2958

At the end, the 6th and 7th degree were discarded due to over-fitting. The 2nd degree polynomial, while having the least variance, had two groups of solutions, grouped around the discrete cross-correlation indices between which the actual solution fell. Meanwhile, the 3rd, 4th, and 5th degree fits all clustered between the cross-correlation index around a single point, which is where the assumed solution of the actual delay lies.

B. CODE LISTINGS

B.1 Experiment Code

B.1.1 SDR initialization, and coarse synchronization

```
#!/bin/bash

# Compile command for MacOS 11.0 Big Sur (x86)
# export LIBRARY_PATH=$LIBRARY_PATH:/opt/local/lib/
# g++ --std=gnu++11 -I/opt/local/include/ -fvisibility=hidden -fvisibility-inlines-hidden -O2 -g -DNDEBUG -Wsign-compare rover.cpp -luhd -lpthread -lboost_program_options-mt -lboost_filesystem-mt -lboost_thread-mt -lboost_serialization-mt -lboost_system-mt -o ./bin/rover -DHAVE_CONFIG_H -DUHD_IMAGES_DIR=OFF -DUHD_LOG_CONSOLE_COLOR -DUHD_LOG_CONSOLE_LEVEL=2 -DUHD_LOG_FILE_LEVEL=2 -DUHD_LOG_MIN_LEVEL=1

# Compile command for Ubuntu 20.04 Focal Fossa
g++ --std=gnu++11 -I/usr/include/uhd -fvisibility=hidden -fvisibility-inlines-hidden -O2 -g -DNDEBUG  -Wsign-compare record.cpp -luhd -lpthread -lboost_program_options -lboost_filesystem -lboost_thread -lboost_serialization -lboost_system -o record -DHAVE_CONFIG_H -DUHD_IMAGES_DIR=OFF -DUHD_LOG_CONSOLE_COLOR -DUHD_LOG_CONSOLE_LEVEL=2 -DUHD_LOG_FILE_LEVEL=2 -DUHD_LOG_MIN_LEVEL=1


# Settings
freq1=2343125000  # set center frequency
rate=4000000 # set sampling rate 4 MhZ
delaytime=60 # Time to wait
dtype="sc16"
gain1=40
subdev="A:A"
integrationtime=100 #milliseconds

signal1="WestLafayette"
serial1="serial=F61173"
```

```

path="/home/q/Data/"
directory="/home/q/" # set directory
mkdir -p $path

sync; echo 1 > sudo /proc/sys/vm/drop_caches

run_length=600 #seconds
current_epoch=$(date +%s)
target_epoch=$(date -d 'today 16:17' +%s)
sleep_secondsinit=$(( $target_epoch - $current_epoch ))
sleep $sleep_secondsinit
echo $sleep_secondsinit
current_time=$(date +%s)

while [ $(date "+%s") -lt $(( $target_epoch + $run_length)) ]; do

    ./record --args $serial1 --bw $rate --rate $rate --subdev "$subdev" --freq $freq1 --file $path --rfile $rfile --gain $gain1 --wirefmt $dtype --cpufmt $dtype --signal $signal1 --integration $integrationtime --setup $delaytime
    current_time=$(date +%s)
    target_time=$((  $current_time + ($delaytime - ($current_time % $delaytime))))
    echo $target_time
    sleep $(($target_time - $current_time - 20))

done
```

B.1.2 SDR Recording code

```
#include <uhd/types/tune_request.hpp>
#include <uhd/utils/thread.hpp>
#include <uhd/utils/safe_main.hpp>
#include <uhd/usrp/multi_usrp.hpp>
```

```
#include <uhd/exception.hpp>
#include <boost/program_options.hpp>
#include <boost/format.hpp>
#include <boost/thread.hpp>
#include <boost/algorithm/string.hpp>
#include <iostream>
#include <fstream>
#include <csignal>
#include <complex>
#include <math.h>
#include <thread>
#include <chrono>

// =====================================
#include <uhd/convert.hpp>
#include <boost/lexical_cast.hpp>
#include <boost/algorithm/string.hpp>
#include <uhd/utils/paths.hpp>
#include <uhd/transport/usb_control.hpp>
#include <uhd/transport/usb_device_handle.hpp>
#include <uhd/config.hpp>
#include <boost/functional/hash.hpp>
//#include <b200_iface.hpp>
// =====================================

namespace po = boost::program_options;

void tuneLO(uhd::usrp::multi_usrp::sptr &usrp, double &freq, std::string &signal);
void configureUSRP(uhd::usrp::multi_usrp::sptr &usrp, double &gain, double &bw, std::string &subdev, double &rate, std::string &signal);
void packetErrorCode(uhd::rx_metadata_t &md);
int check10MHzLock(uhd::usrp::multi_usrp::sptr &usrp, std::string &signal);
int checkGPSLock(uhd::usrp::multi_usrp::sptr &usrp, std::string &signal);
int setGPSTime(uhd::usrp::multi_usrp::sptr &usrp, std::string &signal);

int UHD_SAFE_MAIN(int argc, char *argv[]){

    uhd::set_thread_priority_safe();

    std::string args, file, rfile, type, ant, subdev, ref, wirefmt, cpufmt, signal;
    size_t total_num_samps, spb, stime, direct_collection_flag, interval;
    double rate, freq, gain, bw, setup_time, integration_time;
    std::ofstream logfile;
    time_t rawtime;

    po::options_description desc("Allowed options");
    desc.add_options()
    ("help", "help message")
    ("args", po::value<std::string>(&args)->default_value(""), "multi uhd device address args")
    ("file", po::value<std::string>(&file)->default_value("usrp_samples.dat"), "name of the file to write binary samples to")
    ("rfile", po::value<std::string>(&rfile)->default_value("usrp_collection_log.txt"), "name of the file to write collection state to")
    ("type", po::value<std::string>(&type)->default_value("short"), "sample type: double, float, or short")
    ("spb", po::value<size_t>(&spb)->default_value(10000), "samples per buffer")
    ("rate", po::value<double>(&rate)->default_value(1e6), "rate of incoming samples")
    ("freq", po::value<double>(&freq)->default_value(0.0), "RF center frequency in Hz")
    ("gain", po::value<double>(&gain)->default_value(10.0), "gain for the RF chain")
    ("signal", po::value<std::string>(&signal)->default_value("XM"), "signal")
    ("nover", po::value<size_t>(&direct_collection_flag)->default_value(0), "Direct collection?")
    ("subdev", po::value<std::string>(&subdev), "daughterboard subdevice specification")
    ("bw", po::value<double>(&bw)->default_value(0.0), "analog frontend filter bandwidth in Hz")
    ("ref", po::value<std::string>(&ref)->default_value("internal"), "reference source (internal, external, mimo)")
    ("wirefmt", po::value<std::string>(&wirefmt)->default_value("sc8"), "wire format (sc8 or sc16)")
    ("cpufmt", po::value<std::string>(&cpufmt)->default_value("sc8"), "cpu format (sc8, sc16, fc32, or fc64)")
    ("setup", po::value<size_t>(&interval)->default_value(30.0), "Collection interval (seconds)")
    ("integration", po::value<double>(&integration_time)->default_value(1.0), "time of buffer length");

    po::variables_map vm;
    po::store(po::parse_command_line(argc, argv, desc), vm);
    po::notify(vm);

    //Create USRP Device object
    std::cout << boost::format("%s: ") %signal << boost::format("Creating the usrp device with: %s") % args << std::endl;
    uhd::usrp::multi_usrp::sptr usrp = uhd::usrp::multi_usrp::make(args);

    configureUSRP(usrp, gain, bw, subdev, rate, signal); //Configure USRP Device
    check10MHzLock(usrp, signal); //Check for 10 MHz Lock
    checkGPSLock(usrp, signal); //Check GPSDO Lock
    tuneLO(usrp, freq, signal); //Tune LO desired center frequency
    setGPSTime(usrp, signal); //Align GPS

    std::string gps_gpgga;
    gps_gpgga = usrp->get_mboard_sensor("gps_gpgga",0).to_pp_string();
    std::cout << boost::format("%s") % gps_gpgga << std::endl;
    std::cout.flush();

```

```cpp
 //Create RX Stream object -> Only for single channel
 uhd::stream_args_t stream_args(cpufmt, wirefmt);
 stream_args.channels = {0};
 uhd::rx_streamer::sptr rx_stream = usrp->get_rx_stream(stream_args);

 //Create Buffer
 std::vector<char> buffer(rate * integration_time/1000.0 * uhd::convert
::get_bytes_per_item(cpufmt));

 //Calculate next interval record time
 double timeNow = usrp->get_time_now().get_real_secs();
 double nextTime = ceil(timeNow / interval) * interval;

 //Create File
 std::ofstream outfile;
 std::string fileName = (boost::format("%s%.0f_%s.%s") % file %
nextTime % signal % wirefmt).str();
 std::cout << boost::format("%s: ") %signal << boost::format("File name
: %s") % fileName << std::endl;

 outfile.open(fileName.c_str(), std::ofstream::binary);

 //Create Stream object every loop
 uhd::stream_cmd_t cmd(uhd::stream_cmd_t::
STREAM_MODE_NUM_SAMPS_AND_DONE);
 cmd.stream_now = false;
 cmd.num_samps = rate * integration_time/1000.0;

 //Issue Stream command to be at 30 second interval
 cmd.time_spec = uhd::time_spec_t(nextTime);
 rx_stream->issue_stream_cmd(cmd);

 //Create metadata object and receive buffer
 uhd::rx_metadata_t md;
 size_t num_rx_samps = rx_stream->recv(&buffer.front(), buffer.size(),
md, interval-0.5);
```

```
    packetErrorCode(md);

    //Create and Issue stop command to streaming (may not actually be needed)
    uhd::stream_cmd_t stop_cmd(uhd::stream_cmd_t::STREAM_MODE_STOP_CONTINUOUS);
    rx_stream->issue_stream_cmd(stop_cmd);

    //write buffer to file
    if (outfile.is_open()) {
        outfile.write((const char*)&buffer.front(), num_rx_samps * sizeof(char));
    }

    //Open for Timing Debugging
    // std::int64_t timeStamp = std::chrono::system_clock::to_time_t(std::chrono::system_clock::now());
    // std::cout << "computer time loop start: " << std::chrono::seconds(timeStamp).count()<< std::endl;

    // std::cout << "USRP time now: " << (boost::format("%0.9f") % timeNow) << std::endl;
    std::cout << "USRP record time: " << (boost::format("%0.9f") % cmd.time_spec.get_real_secs()) << std::endl;

    if (outfile.is_open()){
        outfile.close();
    }

    std::cout.flush();

    return EXIT_SUCCESS;
}

void configureUSRP(uhd::usrp::multi_usrp::sptr &usrp, double &gain, double &bw, std::string &subdev, double &rate, std::string &signal){

    //Sample Rate
    usrp->set_rx_rate(rate);
    //std::cout << boost::format("%d: ") %rate << boost::format("Sample Rate Set") << std::endl;


    boost::this_thread::sleep(boost::posix_time::milliseconds(1100)); // sleep 110ms (~10ms after retune occurs) to allow LO to lock
    usrp->clear_command_time();
    //std::cout << boost::format("%s: ") %signal << boost::format("Center Frequency Tuned") << std::endl;

    //RF Gain
    usrp->set_rx_gain(gain, 0);
    std::cout << boost::format("%d: ") %gain << boost::format("RF Gain Set ") << std::endl;

    //IF filter bandwidth
    usrp->set_rx_bandwidth(bw, 0);
    //std::cout << boost::format("%s: ") %signal << boost::format(" Bandwidth Set") << std::endl;

    //Subdevice Specification
    usrp->set_rx_subdev_spec(subdev);
    //std::cout << boost::format("%s: ") %signal << boost::format(" Subdevice (Channel) Set") << std::endl;

    //Set references to GPSDO
    usrp->set_clock_source("gpsdo", 0);
    usrp->set_time_source("gpsdo", 0);
    //std::cout << boost::format("%s: ") %signal << boost::format(" Reference to GPSDO Set") << std::endl;

}

```

```cpp
void tuneLO(uhd::usrp::multi_usrp::sptr &usrp, double &freq, std::string &signal){

    //Tune LO to Center Frequency
    usrp->clear_command_time();
    usrp->set_command_time(usrp->get_time_now() + uhd::time_spec_t(0.1)); //set cmd time for .1s in the future

    uhd::tune_request_t tune_request(freq);
    usrp->set_rx_freq(tune_request);

}

int check10MHzLock(uhd::usrp::multi_usrp::sptr &usrp, std::string &signal){

    std::vector<std::string> sensor_names = usrp->get_mboard_sensor_names(0);

    if (std::find(sensor_names.begin(), sensor_names.end(), "ref_locked") != sensor_names.end()){
        bool ref_locked = false;
        for (int i = 0; i < 30 && !ref_locked; i++){
            ref_locked = usrp->get_mboard_sensor("ref_locked", 0).to_bool();
            if (!ref_locked){
                std::cout << "." << std::flush;
                boost::this_thread::sleep(boost::posix_time::seconds(1));
            }
        }
        if (ref_locked){
            //std::cout << boost::format("%s: ") %signal << "10 MHz Locked " << std::endl;
        }
        else{
            std::cout << boost::format("%s: ") %signal << "Failed to lock to GPSDO 10 MHz Reference. Exiting." << std::endl;
            exit(EXIT_FAILURE);
        }
    }
    else{
        std::cout << boost::format("ref_locked sensor not present on this board.\n");
        return EXIT_FAILURE;
    }

    return EXIT_SUCCESS;
}

int checkGPSLock(uhd::usrp::multi_usrp::sptr &usrp, std::string &signal){

    std::vector<std::string> sensor_names = usrp->get_mboard_sensor_names(0);

    if (std::find(sensor_names.begin(), sensor_names.end(), "ref_locked") != sensor_names.end()){
        bool gps_locked = true;
        for (int i = 0; i < 30 && !gps_locked; i++){
            gps_locked = usrp->get_mboard_sensor("gps_locked", 0).to_bool();
            if (!gps_locked){
                std::cout << "." << std::flush;
                boost::this_thread::sleep(boost::posix_time::seconds(1));
            }
        }
        if (gps_locked){
            //std::cout << boost::format("%s: ") %signal << "GPS Locked" << std::endl;
        }
        else{
            std::cout << "Failed to lock to GPS Time Reference. Exiting." << std::endl;
            exit(EXIT_FAILURE);
        }
    }
    else
    {
        std::cout << boost::format("gps_locked sensor not present on this board.\n");
        return EXIT_FAILURE;
    }

    return EXIT_SUCCESS;
}

int setGPSTime(uhd::usrp::multi_usrp::sptr &usrp, std::string &signal){

    //Set to GPS time
    uhd::time_spec_t gps_time = uhd::time_spec_t(int64_t(usrp->get_mboard_sensor("gps_time", 0).to_int()));
    usrp->set_time_next_pps(gps_time + 1.0, 0);

    //Wait 2 seconds. Known issue
    boost::this_thread::sleep(boost::posix_time::seconds(2));

    //Check times
    gps_time = uhd::time_spec_t(int64_t(usrp->get_mboard_sensor("gps_time", 0).to_int()));
    uhd::time_spec_t time_last_pps = usrp->get_time_last_pps(0);

    //std::cout << "USRP time: " << (boost::format("%0.9f") % usrp->get_time_last_pps(0).get_real_secs()) << std::endl;
    //std::cout << "GPSDO time: " << (boost::format("%0.9f") % gps_time.get_real_secs()) << std::endl;

    if (gps_time.get_real_secs() == time_last_pps.get_real_secs()){
        std::cout << boost::format("%s: ") %signal << "USRP time synchronized to GPS time" << std::endl;
    }
    else {
        std::cerr << boost::format("%s: ") %signal << "Failed to synchronize USRP time to GPS time" << std::endl;
        return EXIT_FAILURE;
    }

    return EXIT_SUCCESS;
}

void packetErrorCode(uhd::rx_metadata_t &md){

    switch (md.error_code){
        case uhd::rx_metadata_t::ERROR_CODE_NONE:
            break;
        case uhd::rx_metadata_t::ERROR_CODE_OVERFLOW:
            std::cout << boost::format("Overflow\n");
            break;
        case uhd::rx_metadata_t::ERROR_CODE_TIMEOUT:
            std::cout << boost::format("Timeout\n");
            break;
        case uhd::rx_metadata_t::ERROR_CODE_ALIGNMENT:
            std::cout << boost::format("Alignment error\n");
            break;
        case uhd::rx_metadata_t::ERROR_CODE_BAD_PACKET:
            std::cout << boost::format("Bad packet\n");
            break;
        default:
            std::cerr << boost::format("md.error_code   %d") % md.error_code << std::endl;
            break;
    }

}
```

B.2 Post-Processing Code

B.2.1 Generate timing vectors and Propagate TLE

```
%Generate TLE history and receiver position

clc
clear all
warning(’off’)
Date = ’April14’;
load([’Data/’,Date,’/signaldelay.mat’]’)

receiverLLA =   [47.848910, -122.269385, 141.9;
                    34.144292, -118.113917, 241.1;
                    40.010433, -105.243652, 1624.3;
                    40.416683, -86.942610, 187.9;
                    33.452507, -88.787907, 134.5;
                    43.304265, -77.733740, 179.5];

rReceiverECEF = lla2ecef([receiverLLA(:,1),receiverLLA(:,2),receiverLLA
    (:,3)]);

% Epoch TLE for XM3
fid1 = fopen([’Data/’,Date,’/xm3.tle’]);
tline1 = fgets(fid1);
tline2 = fgets(fid1);
[˜, ˜, ˜, satrec3] = twoline2rv(tline1,tline2,’c’,’d’,’i’,721);
jdtle3 = satrec3.jdsatepoch + satrec3.jdsatepochf;
fclose(fid1);

% Epoch TLE for XM4
fid2 = fopen([’Data/’,Date,’/xm4.tle’]);
tline1 = fgets(fid2);
tline2 = fgets(fid2);
[˜, ˜, ˜, satrec4] = twoline2rv(tline1,tline2,’c’,’d’,’i’,721);
jdtle4 = satrec4.jdsatepoch + satrec4.jdsatepochf;
```

```
fclose(fid2);

parfor i = 1:length(signaldelay4)

    jdtime = (unixtimevec(i)/86400) + 2440587.5;
    [year,mon,day,hr,min,sec] = invjday(fix(jdtime),jdtime-fix(jdtime)); % epoch time
    utc = [year mon day hr min sec];

    t3 = (jdtime-jdtle3)*1440;
    t4 = (jdtime-jdtle4)*1440;

    [~, rSat3ECI, ~] = sgp4(satrec3,t3);
    [~, rSat4ECI, ~] = sgp4(satrec4,t4);
    rXM3ECI(i,:) = rSat3ECI*1000;
    rXM4ECI(i,:) = rSat4ECI*1000;

    rSat3ECEF = eci2ecef(utc, rSat3ECI*1000)';
    rSat4ECEF = eci2ecef(utc, rSat4ECI*1000)';
    rXM3ECEF(i,:) = rSat3ECEF;
    rXM4ECEF(i,:) = rSat4ECEF;

    for k = 1:6
        rReceiverECI(i,:,k) = ecef2eci(utc,rReceiverECEF(k,:));
    end

end

t1 = datetime(2021,4,14,18,0,0);
t2 = datetime(2021,4,21,17,59,0);
datevec = t1:minutes(1):t2;

save(['Data/',Date,'/positiondata.mat'],'rReceiverECEF','rReceiverECI','rXM3ECEF','rXM3ECI','rXM4ECEF','rXM4ECI','datevec');
```

B.2.2 Cross-correlation and interpolation between receiver stations

```
%Cross Correlation Script

clc
clear all
close all
fclose(’all’);
warning(’off’);

%Select which date of measurement
Date = ’April13’;
loc1 = dir([’Data/’,Date,’/Seattle’]);
loc2 = dir([’Data/’,Date,’/Pasadena’]);
loc3 = dir([’Data/’,Date,’/Boulder’]);
loc4 = dir([’Data/’,Date,’/WestLafayette’]);
loc5 = dir([’Data/’,Date,’/Mississipi’]);
loc6 = dir([’Data/’,Date,’/Rochester’]);

%Filter Setup
Fs = 4e6; %Sampling Frequency
Ti = 0.025; %Integration Time
samps = Fs*(2*Ti);
points = samps/2;


fd = fdesign.lowpass(’N,Fp,Fst’,500, 1.6e+6/2 ,1.82e+6/2,Fs);
dd = design(fd, ’equiripple’);
fdz = impz(dd, samps/2);
fc = [-1.84/2 1.84/2]*1e+6;
t = (0:(samps/2)-1)/Fs;

%XM-3 filter
fcarr1 = exp(2*pi*1j*fc(1)*t’);
fdf1 = fft(fdz.*fcarr1);
%XM-4 filter
```

```
fcarr2 = exp(2*pi*1j*fc(2)*t’);
fdf2 = fft(fdz.*fcarr2);

fit = 2;

updateWaitbar = waitbarParfor(length(loc1)-2, "Calculation in progress...");
unixtimevec = zeros(length(loc1)-2,1);

%Loop through each time point
parfor i = 3:length(loc1)

    filename = loc1(i).name;
    unixtimevec(i-2) = str2num(filename(1:10));

    fid1 = fopen(strcat([’Data/’,Date,’/Seattle/’],loc1(i).name),’rb’);
    fid2 = fopen(strcat([’Data/’,Date,’/Pasadena/’],loc2(i).name),’rb’);
    fid3 = fopen(strcat([’Data/’,Date,’/Boulder/’],loc3(i).name),’rb’);
    fid4 = fopen(strcat([’Data/’,Date,’/WestLafayette/’],loc4(i).name),’rb’);
    fid5 = fopen(strcat([’Data/’,Date,’/Mississipi/’],loc5(i).name),’rb’);
    fid6 = fopen(strcat([’Data/’,Date,’/Rochester/’],loc6(i).name),’rb’);

    data1 = fread(fid1, samps, ’int16’);
    data2 = fread(fid2, samps, ’int16’);
    data3 = fread(fid3, samps, ’int16’);
    data4 = fread(fid4, samps, ’int16’);
    data5 = fread(fid5, samps, ’int16’);
    data6 = fread(fid6, samps, ’int16’);

    datamain = zeros(6,points);
    datamain(1,:) = (data1(1:2:end) + 1i*data1(2:2:end))’;
    datamain(2,:) = (data2(1:2:end) + 1i*data2(2:2:end))’;
    datamain(3,:) = (data3(1:2:end) + 1i*data3(2:2:end))’;
    datamain(4,:) = (data4(1:2:end) + 1i*data4(2:2:end))’;
    datamain(5,:) = (data5(1:2:end) + 1i*data5(2:2:end))’;
    datamain(6,:) = (data6(1:2:end) + 1i*data6(2:2:end))';

    tile = 1;

    %6x6 cross-correlation
    for p = 1:6

        %Grab XM-3 and XM-4 signal
        Ui1 = fft(datamain(p,:))'.*fdf1;
        Ui2 = fft(datamain(p,:))'.*fdf2;

        for q = 1:6

            %Grab XM-3 signal to cross-correlate with
            Uj1 = fft(datamain(q,:))'.*fdf1;
            crosscorr3 = fftshift(ifft(Ui1.*conj(Uj1))); %corss-correlate

            [~,I] = max(crosscorr3); %Find peak index
            if I > points-2   I < 3
                signaldelay3(p,q,i-2) = I;
            else
                xval = [I-2:0.001:I+2];
                coeffs = polyfit([I-2:I+2], abs(crosscorr3([I-2:I+2])),fit); %fit 2nd order polynomial
                [~,J] = max(polyval(coeffs,xval));
                signaldelay3(p,q,i-2) = (points/2 - xval(J) + 1) * 1e-6/4 * 299792458; %convert delay to kilometers

            end


            %Repeat with XM-4
            Uj2 = fft(datamain(q,:))'.*fdf2;
            crosscorr4 = fftshift(ifft(Ui2.*conj(Uj2)));

            [~,I] = max(crosscorr4);
            if I > points-2   I < 3
                signaldelay4(p,q,i-2) = I;
            else
                xval = [I-2:0.001:I+2];
                coeffs = polyfit([I-2:I+2], abs(crosscorr4([I-2:I+2])),fit);
                [~,J] = max(polyval(coeffs,xval));
                signaldelay4(p,q,i-2) = (points/2 - xval(J) + 1) * 1e-6/4 * 299792458;

            end


            %6x6 Cross-Correlation debugging plot
%             figure(1)
%             subplot(6,6,tile)
%             hold on
%             plot(1:points,abs(crosscorr3))
%             tile = tile+1;
%             plot(1:points,abs(crosscorr4))

        end

    end

    fclose(fid1);
    fclose(fid2);
    fclose(fid3);
    fclose(fid4);
    fclose(fid5);
    fclose(fid6);

    updateWaitbar();

end

```

```
t1 = datetime(2021,4,13,11,50,0);
t2 = datetime(2021,4,14,12,9,0);
datevec = t1:minutes(1):t2;
% t1 = datetime(2021,4,14,18,0,0);
% t2 = datetime(2021,4,21,17,59,0);
% datevec = t1:minutes(1):t2;

%%
figure(1)
hold on
grid on
for i = 1:6
    for j = 1:6

        plot(datevec,squeeze(signaldelay4(i,j,:)))
    end
end

save(['Data/',Date,'/signaldelay.mat'],'signaldelay3','signaldelay4','unixtimevec')
```

B.2.3 Tansform solution, initialize least squares, DOP

```
function [datevec,RTvec,rXMTLE,rReceiverECEF,T,Q] = posHistorySCFrame(Date,sat,locpick)

    aa = ['Data/',Date,'/signaldelay.mat']; %Load Cross-Correlated position delay
    bb = ['Data/',Date,'/positiondata.mat']; %Load TLE Data & Time
    load(aa);
    load(bb);

    if sat == 3 %85 degree rotation for XM-3
        T = angle2dcm(0,0,deg2rad(85),'XYZ');
        signaldelay = signaldelay3;
        rXMTLE = rXM3ECEF;
    else %115 degree rotation for XM-4
        T = angle2dcm(0,0,deg2rad(115),'XYZ');
        signaldelay = signaldelay4;
        rXMTLE = rXM4ECEF;
    end
    Q = zeros(3,3,length(signaldelay));

    parfor i = 1:length(signaldelay)

        RT = rXMTLE(i,:); %change for solution in ECEF or ECI
        Rs = squeeze(signaldelay(:,:,i));

        %Solve for position with 1e-8 tolerance, and 50 iteration cap
        [H,RT] = solvePos(RT,rReceiverECEF,Rs,locpick,1e-8,50);

        %Calculate DOP from outputted H matrix
        Q(:,:,i) = T * inv(H'*H) * T';
        RTvec(i,:) = RT * T;

    end

end
```

B.2.4 Least squares for given choice of TDOA measurements

```
{function [H,RT] = solvePos(RT,Rrec,Rs,choice,tol,looplim)

    %GPS-like solution for selected receivers

    n = length(choice);
    err = 9e12;
    loop = 1;


    while err > tol

        %Create vector of transmitter guessed position
        RTv  = repmat(RT,6,1);
        RNT = RTv - Rrec; %Difference with position of receiver

        y = zeros(n-1,1);
        H = zeros(n-1,3);

        c = 1;
        i = choice(1);
        for j = choice([2:end])

            computedDelay = norm(RNT(i,:)) - norm(RNT(j,:));
            measuredDelay = Rs(i,j);

            %build y and H matrix
            y(c) = measuredDelay - computedDelay;
            H(c,:) = RNT(i,:)/norm(RNT(i,:)) - RNT(j,:)/norm(RNT(j,:));

            c = c+1;
        end

        %compute update to RT
        deltaRT = (H'*H)\H'*y;
        RT = RT + deltaRT';
        err = norm(deltaRT);

        %loop break condition
        if loop > looplim
            break
        end
        loop = loop + 1;
    end

end}
```