

UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

Evaluation of Low-Cost GNSS-RTK Solution for the Future Truth Position
Sensors Supporting Radar and Autonomous Platforms Tests

A THESIS
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for the
Degree of
MASTER OF SCIENCE

By
Brandon Mansur
Norman, Oklahoma
2024

Evaluation of Low-Cost GNSS-RTK Solution for the Future Truth Position
Sensors Supporting Radar and Autonomous Platforms Tests

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Yan Zhang, Chair

Dr. Yaser M. Banad

Dr. Hernan Suarez

©Copyright by Brandon Mansur 2024
All Rights Reserved.

Acknowledgments

As with any large undertaking, this thesis would have been impossible to complete on my own. I would like to first and foremost thank God for granting me the skills, abilities, and perseverance necessary to complete this degree. I would also like to thank him for putting the many loved ones, friends, colleagues, and mentors in my life who have helped and directed me along the way.

I would like to thank my wife and best friend Kylan for her endless patience and support; I look forward to doing life with you and I hope this helps us on our journey, I promise to finally get to those projects. I am thankful to my committee members and professors, specifically Dr. Rockee, Dr. Dyer, and Dr. Davis, whose guidance, support, and mentorship were integral in getting me across the finish line. I want to thank my brother, Tyler Mansur, and friends, Jon Keegan, Richard Perry, and Olga Umanskaya for their support and encouragement along with countless hours of shenanigans, which offered a much needed reprieve from studying. I would like to thank my lab mates Jakob Fusselman and Morgan Lau, along with my colleagues on the ARRC staff and Engineering Team, whose advice and patience were invaluable. Last but not least, I would like to thank my mother and father Mike and Monica Mansur for their abundant and steady source of love and encouragement.

Contents

Acknowledgments	iv
List Of Tables	viii
List Of Figures	ix
Abstract	xii
1 Introduction and Background	1
1.1 Motivation	1
1.2 Backgrounds	2
1.2.1 GPS/GNSS	2
1.2.1.1 Code Phase Measurements	3
1.2.1.2 Carrier Phase Measurements	5
1.2.1.3 Real-Time Kinematics	6
1.2.1.4 GPS/GNSS Error Sources	6
1.3 Requirements for Precise sUAS Navigation	8
1.3.1 GPS/GNSS Augmentation for Small and Medium UAS Operations	9
2 Design of Experiments and Testing Hardware	10
2.1 Experimental Design	10
2.1.1 ISO 17123-8 Overview	10
2.1.1.1 Simple Test	11
2.1.1.2 Full Test	13
2.1.2 Modified ISO 17123-8 for Dynamic Systems	15
2.1.2.1 Modified Simple Test	15
2.1.2.2 Modified Full Test	17
2.2 Hardware Setup	19
3 Experimental Procedures and Data Processing	25
3.1 Base Station Survey	25

3.2	Base and Rover Configuration	26
3.2.1	Ashtech Configuration	26
3.2.1.1	Ashtech Base	27
3.2.1.2	Ashtech Rover	27
3.2.2	ZED-F9P Configuration	28
3.2.2.1	ZED-F9P Base	30
3.2.2.2	ZED-F9P Rover	32
3.2.3	Raspberry Pi Data Logger	32
3.3	Data Conversion and Post Processing	33
3.3.1	Data Conversion	33
3.3.1.1	Ashtech Data Conversion	33
3.3.1.2	ZED-F9P Data Conversion	37
3.3.2	Kinematic Post Processing	39
3.3.2.1	RTKLIB Kinematic Post Processing	39
3.3.2.2	Emlid Kinematic Post Processing	43
3.3.3	RTK Data Extraction	43
4	Results and Discussions	44
4.1	Results	44
4.1.1	Tests	44
4.1.1.1	Test 1	46
4.1.1.2	Test 2	54
4.1.1.3	Test 3	62
4.1.1.4	Test 4	66
4.1.1.5	Test 5	74
4.1.1.6	Test 6	78
4.1.2	Ground Truth Error Summary	86
4.1.3	Modified ISO 17123-8 Results Summary	86
4.1.3.1	Individual Test Deviations	86
4.1.3.2	Cumulative Deviations	88
4.2	Discussion	89
5	Conclusion and Future Work	92
5.1	Conclusion	92
5.2	Future Work	92
	Reference List	93
	Appendix A - List Of Symbols	96
	Appendix B - List Of Acronyms and Abbreviations	100
	Appendix C - Ashtech Quick Reference	103

Appendix D - RPi Data Logger	104
D.1 RPi Wiring Diagram and Pinout	104
D.2 Data Logger Script	105
Appendix E - Matlab Data Analysis	108
E.1 Individual Test Analysis and Graphing	108
E.2 Total Error Analysis	120
Appendix F - Python NMEA/UBX Reader	123
Appendix G - PPP Survey Results	125

List Of Tables

2.1	ISO 17123-8 Series Structure	12
3.1	Base Station PPP Results	26
3.2	Base Station Standard Deviation	26
3.3	ZED-F9P Base Output Messages	31
3.4	ZED-F9P Rover Output Messages	32
4.1	Ground Truth Error Summary	86
4.2	Series Standard Deviation	88
4.3	Cumulative Standard Deviation	89
4.4	Error Budget	89

List Of Figures

1.1	RTK Example	6
2.1	ISO 17123-8 Test Setup	11
2.2	Modified ISO 17123-8 Rover Example	16
2.3	Modified ISO 17123-8 Residuals Example	18
2.4	Data Flow Diagram	19
2.5	Base Station Connection Diagram	21
2.6	Rover Connection Diagram	22
2.7	Base Station Setup	23
2.8	Rover Setup	24
3.1	Ashtech Configuration Diagram	28
3.2	ZED-F9P UART Port Config.	29
3.3	ZED-F9P Sample Rate Config.	29
3.4	ZED-F9P GNSS Config.	30
3.5	ZED-F9P DGNSS Config.	30
3.6	ZED-F9P TMODE 3 Config.	31
3.7	Base Station Data Logger	33
3.8	PCMCIA Reader	34
3.9	Ashtech Download	34
3.10	GraphNav-Convert Tools Location	35
3.11	GraphNav - Raw GPS to GPB Conversion	36
3.12	GraphNav-GPB to RINEX Conversion	37
3.13	RTKCONV Options	38
3.14	RTKCONV UBX to RINEX Conversion	39
3.15	RTKPOST Setting1 Configuration	40
3.16	RTKPOST Positions Configuration	41
3.17	RTKPOST Files Configuration	41
3.18	RTKPOST	42
3.19	RTKPLOT	42
3.20	Emlid Post Processor	43
4.1	Ionospheric Conditions	45
4.2	Test 1 RTKLIB Ashtech	46
4.3	Test 1 RTKLIB ZED-F9P	47

4.4	Test 1 Emlid Ashtech	48
4.5	Test 1 Emlid ZED-F9P	49
4.6	Test 1 Combined	50
4.7	Test 1 Ground Truth	51
4.8	Test 1 RTK	52
4.9	Test 1 GT and RTK	53
4.10	Test 2 RTKLIB Ashtech	54
4.11	Test 2 RTKLIB ZED-F9P	55
4.12	Test 2 Emlid Ashtech	56
4.13	Test 2 Emlid ZED-F9P	57
4.14	Test 2 Combined	58
4.15	Test 2 Ground Truth	59
4.16	Test 2 RTK	60
4.17	Test 2 GT and RTK	61
4.18	Test 3 RTKLIB Ashtech	62
4.19	Test 3 RTKLIB ZED-F9P	63
4.20	Test 3 Emlid Ashtech	64
4.21	Test 3 Emlid ZED-F9P	65
4.22	Test 4 RTKLIB Ashtech	66
4.23	Test 4 RTKLIB ZED-F9P	67
4.24	Test 4 Emlid Ashtech	68
4.25	Test 4 Emlid ZED-F9P	69
4.26	Test 4 Combined	70
4.27	Test 4 Ground Truth	71
4.28	Test 4 RTK	72
4.29	Test 4 GT and RTK	73
4.30	Test 5 RTKLIB Ashtech	74
4.31	Test 5 RTKLIB ZED-F9P	75
4.32	Test 5 Emlid Ashtech	76
4.33	Test 5 Emlid ZED-F9P	77
4.34	Test 6 RTKLIB Ashtech	78
4.35	Test 6 RTKLIB ZED-F9P	79
4.36	Test 6 Emlid Ashtech	80
4.37	Test 6 Emlid ZED-F9P	81
4.38	Test 6 Combined	82
4.39	Test 6 Ground Truth	83
4.40	Test 6 RTK	84
4.41	Test 6 GT and RTK	85
4.42	Test 1 Distributions of Residuals	87
4.43	Test 2 Distributions of Residuals	87
4.44	Test 4 Distributions of Residuals	87
4.45	Test 6 Distributions of Residuals	88
4.46	Cumulative distributions of Residuals	88

C.0.1 Ashtech Quick Reference	103
D.1.1 RPi Connection Diagram	104

Abstract

High-accuracy GNSS-based localization solutions have historically been very cost-prohibitive, typically ranging into the tens of thousands of dollars. However, companies such as U-Blox have recently begun creating high-accuracy RTK solutions for just under \$200 USD. The goal of this research is to assess the precision, accuracy, and reliability of the low-cost ZED-F9P receiver in a dynamic state, with a focus on altitude, for potential use in a portable GNSS-augmentation system that supports Unmanned Aerial System (UAS) operations. The ZED-F9P family of receivers claims a horizontal position accuracy of $0.01 \text{ m} + 1 \text{ ppm}$ Circular Error Probability (CEP), which means that 50% of the received positions will be within a circle of 1 cm radius, whose center is on the true position, and whose radius increases by 1mm for every Kilometer of distance between the rover and base station receivers. Its horizontal position accuracy and low price point make it attractive for use in advanced autonomous systems with high accuracy requirements. For ground based systems this may be sufficient, however the datasheet does not specify the vertical positional accuracy, which is problematic for aerial based platforms. This study intends to fill the gap by comparing the ZED-F9P system and the more mature and certified Ashtech Differential GPS (DGPS) system. The Basic theories and principles of using ZED-F9P as an RTK solution, the specific example system deployment,

and the performance of a supporting UAS-based truth positioning system are tested and presented.

Chapter 1

Introduction and Background

1.1 Motivation

In the context of navigation systems, the word augmentation refers to an additional system that provides increased accuracy and other similar improvements to the navigation and timing of positioning systems. As an example of augmentation, a Ground Based Augmentation System (GBAS) ground facility works in conjunction with GPS satellites to provide corrections to the landing approach of aircraft via a Very High Frequency (VHF) radio link. The ground facility comprises at least 3 GPS antennas at known locations, a central processor, and the VHF radio. Each reference antenna estimates the distance between itself and a GPS satellite by measuring the transmission time between the two stations. The estimated distances from each reference antenna and a GPS satellite are then compared with the known distances. After comparison, the error values can be determined. The average of the error values becomes the correction term that is applied to satellite ranges and used by GBAS avionics.

For the operating, testing, and tracking of Unmanned Aerial Vehicles (UAVs), especially the small UAVs (sUAVs), the augmentation challenge is especially

important based on these demands: (1) Precise landing and navigational support is still critical for the execution of the missions, but could no longer use the GBAS/LAAS types of augmentation system due to different operational environments and cost issues. (2) Precise real-time localization control (at cm level) is critical for the planned missions such as the infrastructure inspections and truth position recording for radar calibrations; (3) The Size, Weight, and Power constraints of sUAS platforms restricted the usage of most of the traditional GPS used in larger aircraft.

1.2 Backgrounds

1.2.1 GPS/GNSS

The Global Positioning System (GPS) and the Global Navigation Satellite Systems (GNSS) are satellite-based positioning systems that provide time and geospatial information. GPS, developed by the U.S. Department of Defense, is a subset of GNSS, which encompasses other systems like GLONASS (Russia), Galileo (EU), and BeiDou (China). These systems use a constellation of Medium Earth Orbit (MEO) satellites to transmit microwave signals in a range of frequency bands (eg., L1, L2, L5), that can then be picked up by receivers on Earth. A GNSS receiver determines its location using trilateration by calculating the time delay in the signals sent from at least four satellites. The time delay can be calculated by using either Code Phase Measurements, Carrier Phase Measurements, or a combination thereof. As a note, Bartosz Ciechanowski [1] gives a thorough and intuitive breakdown of the fundamentals of GPS navigation on his website. The code and carrier phase explanations in the following sections can be found in Chapter 5 of the Global Positioning Systems Textbook [2].

1.2.1.1 Code Phase Measurements

The signals sent by GNSS satellites contain a Course Acquisition Code (C/A) which contains the time at which the signal was transmitted from the satellite. When the signal reaches the receiver, the receiver decodes the C/A code. It calculates the distance between the receiver and the satellite by looking at the time the code was sent versus the time the code was received, giving transit time, and then multiplying it by the speed of light in a vacuum. The clocks on the satellite and the receiver keep time independently and are not in sync, which introduces bias to the system. This biased distance, or range, is known as the pseudorange. When determining the pseudorange, the times with respect to the satellite, the receiver, and a common time reference GPS Time (GPST) must be considered. The pseudorange from the satellites to the receiver is calculated by the following equations.

$$\rho(t) = c[t_u(t) - t^s(t - \tau)] \quad (1.1)$$

c is the speed of light in a vacuum $t^s(t - \tau)$ is the emission time as measured by the receivers clock.

t_u is the signals arrival time as measured by the receivers clock

$\rho(t)$ is the pseudorange

t is the arrival time in GPST from the receiver

τ is the transit time from a satellite to the receiver

The time scales of the satellite and receiver can be related to GPST by the following.

$$t_u = t + \delta t_u(t) \quad (1.2)$$

and

$$t^s(t - \tau) = (t - \tau) + \delta t^s(t - \tau) \quad (1.3)$$

Here,

δt_u is the receivers clock bias

δt^s is the satellites clock bias

Accounting for clock biases, the pseudo-range becomes,

$$\begin{aligned} \rho(t) &= c[t + \delta t_u(t) - (t - \tau + \delta t^s(t - \tau))] + \varepsilon_\rho(t) \\ &= c\tau + c[\delta t_u(t)] - \delta t^s(t - \tau) + \varepsilon_\rho(t) \end{aligned} \quad (1.4)$$

The transit time multiplied by the speed of light in a vacuum is

$$c\tau = r(t, t - \tau) + I_\rho(t) + T_\rho(t) \quad (1.5)$$

Where,

$r(t, t - \tau)$ is the geometric, or true, range between the receiver position at time t and the satellite position at time $(t - \tau)$.

I_ρ is the delay associated with the signal traveling through the ionosphere.

T_ρ is the delay associated with the signal traveling through the troposphere.

ε_ρ represents the modeled effects, modeling errors, and measurement errors in the code phase scenario.

For simplicity, the explicit reference to the measurement epoch t is dropped and the pseudorange model is rewritten as,

$$\rho = r + c[\delta t_u - \delta t^s] + I_\rho + T_\rho + \varepsilon_\rho \quad (1.6)$$

1.2.1.2 Carrier Phase Measurements

The carrier phase measurement is much more precise than the previously mentioned code phase measurement. It is measured in terms of the number of cycles generated or received since the starting point of an interval. The receiver acquires a satellite signal and measures the initial phase difference between the satellite's signal and the receiver's generated signal. Since the receiver initially has no more information than the difference between signal phases, and the whole number of signal cycles is unknown, this is referred to as "Integer Ambiguity". The receiver then acquires a lock with the satellite signal and compares it to the receiver-generated signal. Any change in measurement can then be tracked by the receiver, counting full cycles and keeping track of fractional cycles at each measurement epoch. The carrier phase measurement, units of cycles, can be summarized by (1.7), where λ is the carrier wavelength, r is the geometric range, and N is the integer ambiguity. ε_ϕ represents the modeled effects, modeling errors, and measurement errors in the carrier phase scenario. For a more detailed derivation of (1.7), see chapter 5 of the Global Positioning Systems Textbook [2].

$$\phi = \lambda^{-1}[r + I_\phi + T_\phi] + \frac{c}{\lambda}(\delta t_u - \delta t^s) + N + \varepsilon_\phi \quad (1.7)$$

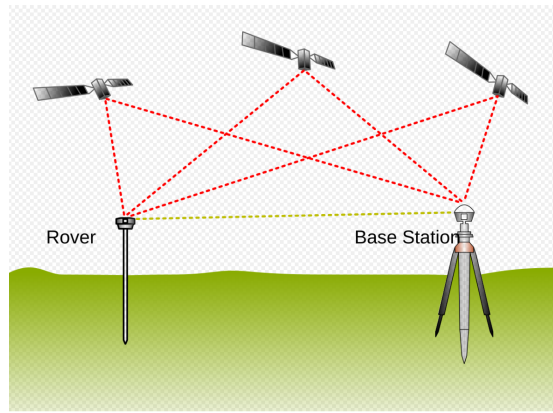


Figure 1.1. *RTK Example*[[3]]

1.2.1.3 Real-Time Kinematics

Real-time kinematics (RTK) is a high-precision satellite navigation technique used to enhance the accuracy of GPS positioning in real-time applications. It is widely used in surveying, agriculture, construction, and other fields where centimeter-level accuracy is crucial. RTK leverages the previously mentioned carrier-phase measurement, along with differential GPS (DGPS), to achieve centimeter-level accuracy. This method takes advantage of the fact that the errors associated with the satellite clock, ephemeris, and atmospheric propagation are similar for receivers separated by small to moderate distances. A base station with a known location compares its own real-time measurements to the known location and generates errors. That error is then transmitted to the rover receiver in real time so the rover can then remove the error from its own measurements.

1.2.1.4 GPS/GNSS Error Sources

Many error sources influence GPS/GNSS position accuracy, impacting code-based and carrier-phase measurements. Though most of these errors can be

largely mitigated so as not to have a large detrimental impact on system accuracy [4], they are still essential to understand, especially in the context of high accuracy systems.

Satellite-Related Errors

Satellite noise refers to the random fluctuations and inaccuracies introduced during the transmission of satellite signals. These noise components can come from imperfections in the satellite's hardware or temperature fluctuations, including clock noise and signal generation errors. While minimized by atomic clocks, satellite clock noise can still introduce small timing discrepancies that propagate into range measurements, leading to positioning inaccuracies. Additionally, signal distortion can occur due to the satellite's transmission hardware, resulting in what is known as satellite hardware bias [5]. These biases can vary between satellites and introduce subtle deviations in the measured pseudorange and carrier phase data. Although these noise sources are generally small, their cumulative effect can impact the precision of GPS/GNSS positioning, especially for applications requiring high accuracy, such as geodetic surveying or precision agriculture.

Atmospheric Delays

Atmospheric delay is caused primarily by the ionosphere and troposphere as the GPS signal travels from the satellite to the receiver. The ionospheric delay results from the signal's interaction with free electrons in the ionosphere, which varies with solar activity, time of day, and the signal frequency. This delay is dispersive, meaning it affects different GPS/GNSS frequencies differently, allowing dual-frequency receivers to correct ionospheric errors by comparing

the delays on two frequencies. Tropospheric delay is caused by the atmosphere's water vapor and dry gases, and unlike the ionosphere, it affects all frequencies equally.

Multipath Error

Multipath occurs when satellite signals reflect off surfaces like buildings, water, or terrain before reaching the receiver. These reflected signals interfere with the direct line-of-sight signals, causing delays that result in incorrect distance measurements and, thus, inaccurate position estimates. Multipath primarily affects the code-based measurements but can also influence the carrier phase, especially at low elevation angles where the signal path is more likely to encounter obstructions.

Receiver Noise

Receiver noise primarily comes from the inherent thermal noise and electronic imperfections within the receiver hardware. This noise presents itself as random fluctuations in the measured pseudorange and carrier phase, reducing the signal-to-noise ratio (SNR) and thus degrading positional accuracy. Receivers also suffer from error due to clock discrepancies, in much the same manner as satellites.

1.3 Requirements for Precise sUAS Navigation

As the sUAS become more affordable, they lend themselves to a wide variety of new markets with a wide array of mission requirements. Small drones are now being utilized in areas such as facility inspection and mapping, truth

observation for radar calibration, and precise delivery services. Each of these applications would benefit greatly from a geospatial solution with high accuracy. There are products on the market today that provide very high accuracy RTK solutions but are in large and cumbersome packages, such as total stations. Furthermore, the cost of traditional RTK solutions is prohibitive, especially when compared to the cost of the sUAS's themselves. High precision, low cost, and compact RTK solutions, such as the ZED-F9P, fill a clear niche for the new frontier of dynamic sUAS missions.

1.3.1 GPS/GNSS Augmentation for Small and Medium UAS Operations

Unlike manned aircraft, sUAS are smaller and more maneuverable, allowing them to be flown in tighter spaces and closer to obstacles where visibility may be compromised. This makes them perfect for unique missions such as infrastructure inspection, radar testing and calibration, and package delivery. While GBAS is historically cheaper than other landing systems [6], it is still prohibitively expensive for smaller entities and hobbyists to implement. While accurate, it is not accurate enough for some advanced autonomous UAS missions. This is where Real-Time Kinematics (RTK) could play a vital role. Current GBAS uses code tracking, which is only capable of meter-level accuracy, while RTK uses carrier-phase tracking, which is capable of sub-centimeter accuracy. RTK is also very inexpensive, costing only a few hundred dollars per module, which makes it achievable for smaller organizations and hobbyists to implement.

Chapter 2

Design of Experiments and Testing Hardware

2.1 Experimental Design

These experiments aim to analyze the dynamic accuracy of the RTK capabilities onboard the Ublox ZED-F9P module. ISO 17123-8 was selected as a starting point from which to design the experiments, as it states that it should be used to determine the best achievable measure of precision for a piece of RTK equipment, and seems to be what manufacturers use to make their accuracy statements [7]. It was then modified to fit a dynamic application similar to what is demonstrated on Arusimple's website [8]. To minimize error and to be as fair to the receivers as possible, all tests were run under good weather conditions, in the open, with clear skies. Satellite health and ionosphere conditions were also considered favorable before testing using the Trimble GNSS Planning website [9].

2.1.1 ISO 17123-8 Overview

Although ISO 17123-8:2015 was referenced directly, further insight was gained from two conference papers [10] [11]. The standard outlines two separate test

procedures: a simple test procedure and a full test procedure. The simple test procedure is helpful in determining gross error, and the full field test procedure is proper for determining the experimental standard deviation for a single position and height measurement. Both tests consist of a base and two rover points that are to be surveyed, with the horizontal and height difference between the two rover points being determined with precision better than 3mm and with methods other than RTK.

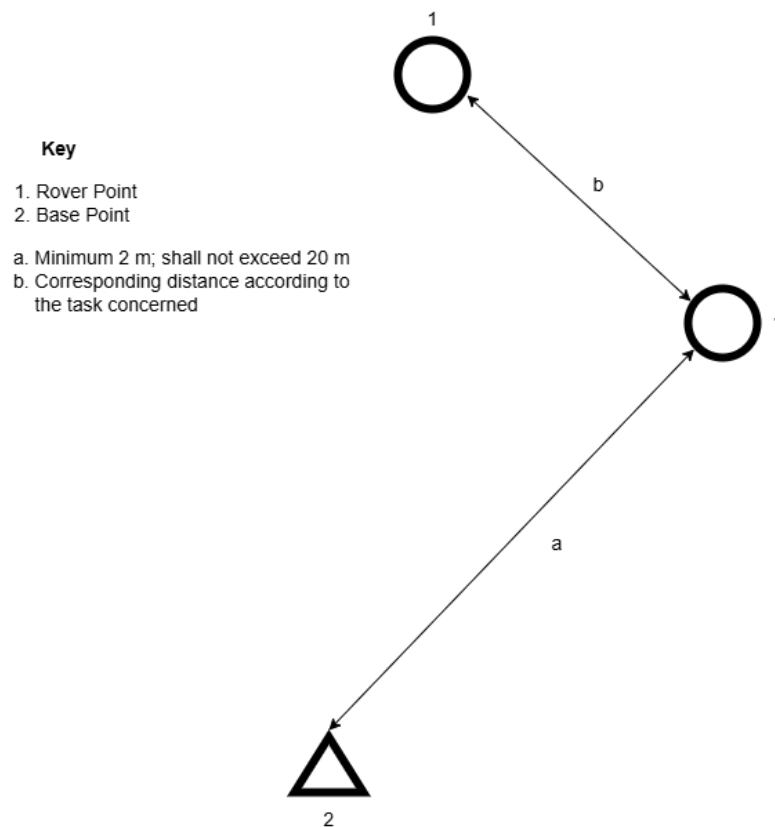


Figure 2.1. *ISO 17123-8 Test Setup.*

2.1.1.1 Simple Test

The simple test is performed by taking ten measurements alternating between the rover points as shown in Table 2.1, with a 5-minute interval between sets.

Seq. No.	Series i	Set j	Rover Point k	Measurement		
				x	y	h
1	1	1	1	$x_{1,1,1}$	$y_{1,1,1}$	$h_{1,1,1}$
2	1	1	2	$x_{1,1,2}$	$y_{1,1,2}$	$h_{1,1,2}$
3	1	2	1	$x_{1,2,1}$	$y_{1,2,1}$	$h_{1,2,1}$
4	1	2	2	$x_{1,2,2}$	$y_{1,2,2}$	$h_{1,2,2}$
5	1	3	1	$x_{1,3,1}$	$y_{1,3,1}$	$h_{1,3,1}$
6	1	3	2	$x_{1,3,2}$	$y_{1,3,2}$	$h_{1,3,2}$
7	1	4	1	$x_{1,4,1}$	$y_{1,4,1}$	$h_{1,4,1}$
8	1	4	2	$x_{1,4,2}$	$y_{1,4,2}$	$h_{1,4,2}$
9	1	5	1	$x_{1,5,1}$	$y_{1,5,1}$	$h_{1,5,1}$
10	1	5	2	$x_{1,5,2}$	$y_{1,5,2}$	$h_{1,5,2}$

Table 2.1. *ISO 17123-8 Single Series Data Points: Alternate between rover points with 5 minutes minimum wait time between sets.*

The horizontal and height distances between the rovers are then calculated for each series using Equations 2.1 and 2.2, and the deviations are calculated using Equations 2.3 and 2.4. Where D^* and h^* are the nominal horizontal distance difference and height difference, respectively.

$$D_{i,j} = \sqrt{(x_{i,j,2} - x_{i,j,1})^2 + (y_{i,j,2} - y_{i,j,1})^2} \quad (2.1)$$

$$\Delta h_{i,j} = h_{i,j,2} - h_{i,j,1} \quad (2.2)$$

$$\varepsilon_{D,i,j} = D_{i,j} - D^* \quad (2.3)$$

$$\varepsilon_{h,i,j} = \Delta h_{i,j} - h^* \quad (2.4)$$

The deviations are then subject to conditions 2.5 and 2.6, where if any deviation value does not satisfy the condition, then the set contains an outlier of gross error and the test should be redone. σ_{xy} and σ_h are the standard deviations determined from either the full test or stated by the manufacturer.

$$|\varepsilon_{D,i,j}| \leq 2.5 \cdot (\sqrt{2}) \cdot \sigma_{xy} \quad (2.5)$$

$$|\varepsilon_{h,i,j}| \leq 2.5 \cdot (\sqrt{2}) \cdot \sigma_h \quad (2.6)$$

2.1.1.2 Full Test

Data collection for the full test is the same as the simple test but with three series of data, each taken 90 minutes apart to allow for changes in the ionosphere, troposphere, and satellite configuration. Once the data is collected, a least squares adjustment is performed on the overall measurements in all series. The mean values $\bar{x}$, $\bar{y}$, and $\bar{h}$ for each rover point ($k = 1, 2$) are calculated via Equations 2.7, 2.8, and 2.9.

$$\bar{x}_k = \frac{1}{15} \sum_{i=1}^3 \sum_{j=1}^5 x_{i,j,k} \quad (2.7)$$

$$\bar{y}_k = \frac{1}{15} \sum_{i=1}^3 \sum_{j=1}^5 y_{i,j,k} \quad (2.8)$$

$$\bar{h}_k = \frac{1}{15} \sum_{i=1}^3 \sum_{j=1}^5 h_{i,j,k} \quad (2.9)$$

The residuals of each x , y , and h for each rover position ($k = 1, 2$) are then calculated with Equations 2.10, then squared and summed using Equations

2.11. Subsequently, the standard deviations can be calculated using Equations 2.13 and Equation 2.12, where m is the number of series, n is the number of sets in each series, p is the number of rover points, and v_x , v_y , and v_h are the degrees of freedom. With these standard deviation values, the experimental standard deviation of a single position ($\sigma_{ISO-GNSSRTK-xy}$) can be found, and the experimental standard deviation of a single height ($\sigma_{ISO-GNSS-h}$) is known, as shown in Equations 2.14.

$$r_{xi,jk} = \bar{x}_k - x_{i,jk} \quad , \quad r_{yi,jk} = \bar{y}_k - y_{i,jk} \quad , \quad r_{hi,jk} = \bar{h}_k - h_{i,jk} \quad (2.10)$$

$$\sum r_x^2 = \sum_{i=1}^3 \sum_{j=1}^5 \sum_{k=1}^2 r_{xi,jk}^2 \quad , \quad \sum r_y^2 = \sum_{i=1}^3 \sum_{j=1}^5 \sum_{k=1}^2 r_{yi,jk}^2 \quad , \quad (2.11)$$

$$\sum r_h^2 = \sum_{i=1}^3 \sum_{j=1}^5 \sum_{k=1}^2 r_{hi,jk}^2$$

$$v_x = v_y = v_h = (m \times n - 1) \times p \quad (2.12)$$

$$\sigma_x = \sqrt{\frac{\sum r_x^2}{v_x}} \quad , \quad \sigma_y = \sqrt{\frac{\sum r_y^2}{v_y}} \quad , \quad \sigma_h = \sqrt{\frac{\sum r_h^2}{v_h}} \quad (2.13)$$

$$\sigma_{ISO-GNSSRTK-xy} = \sqrt{\sigma_x^2 + \sigma_y^2} \quad , \quad \sigma_{ISO-GNSS-h} = \sigma_h \quad (2.14)$$

2.1.2 Modified ISO 17123-8 for Dynamic Systems

2.1.2.1 Modified Simple Test

To modify the ISO 17123-8 standard to fit a dynamic application, the rovers are replaced with a “Ground Truth” system that is more accurate than the reported accuracy for the measured RTK system. The standard states that the rover’s position must be known with a precision better than 3 mm, which is easily achievable with a high-quality total station [12] under static or prolonged conditions but difficult for higher-speed kinematic applications. To overcome this, we explored a ground truth system based on work done by Schrader [13] [14], where there was a reported increase in GPS accuracy by averaging real-time data from multiple receivers. Furthermore, the post processors use a Kinematic model, which employs an Extended Kalman Filter (EKF) to calculate the position and the associated standard deviation via the EKF covariance matrix[2]. The EKF approximates the noise characteristics of each receiver as linear and Gaussian via the use of Taylor Series [15]. Thus, it is reasonable to average the results at each epoch. The proposed ground truth in these tests was created by taking measurements with two receivers, the Ashtech and ZED-F9P, and then processing them using two separate post-processors. The four kinematic post-processed results were then averaged to account for errors in the receiver hardware and the post-processing software. Initial calculations performed on the first test showed promise with a horizontal standard deviation (σ_{xy}) of .003 mm and a vertical standard deviation (σ_h) of .005 mm for a 3D standard deviation (σ_{3D}) of .006 mm. While this is still above the ISO standard precision, it is far better than the ZED-RTK modules’ stated accuracy and is considered acceptable for these tests.

The simple test to check for gross error in ISO 17123-8 outlines a series of tests on two stationary rovers where the distance between sets of consecutive RTK measurements is compared against the known distance between the rovers. In this experiment, each point taken by the RTK system is compared against a point taken by the ground truth system as a new known rover point. So Instead of measuring rover 1, then at rover 2, then starting back to rover 1, it progresses to rover 3, then to rover 4, and so on, as shown in Figure 2.2. Much like the ISO standard, the distances are calculated between points and analyzed for gross error, but for progressive rover positions instead of static ones. The simple test equations are modified to fit a dynamic application as shown in Equations 2.16 - 2.18, where if a test fails to satisfy the conditions shown in Equations 2.19 and 2.20, that rover point gets removed from the dataset.

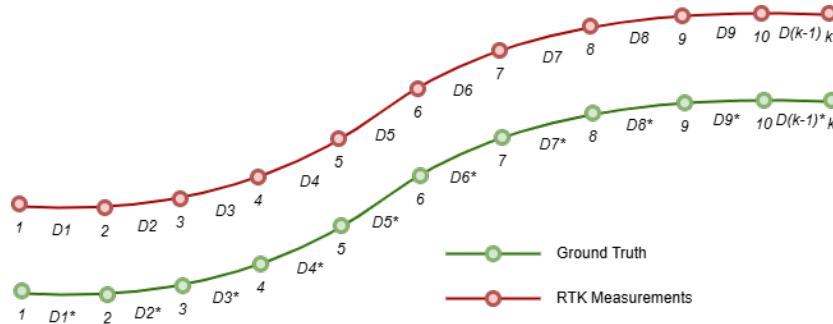


Figure 2.2. ISO 17123-8 Rover Modification: Instead of 2 rovers in a static setting, the rovers will be each point along the "Ground Truth" system.

$$D_{i,k} = \sqrt{(x_{i,k+1} - x_{i,k})^2 + (y_{i,k+1} - y_{i,k})^2} \quad (2.15)$$

$$\Delta h_{i,k} = h_{i,k+1} - h_{i,k} \quad (2.16)$$

$$\varepsilon_{D,i,k} = D_{i,k} - D_{i,k}^* \quad (2.17)$$

$$\varepsilon_{h,i,k} = \Delta h_{i,k} - h_{i,k}^* \quad (2.18)$$

$$|\varepsilon_{D,i,k}| \leq 2.5 \cdot (\sqrt{2}) \cdot \sigma_{xy} \quad (2.19)$$

$$|\varepsilon_{h,i,k}| \leq 2.5 \cdot (\sqrt{2}) \cdot \sigma_h \quad (2.20)$$

2.1.2.2 Modified Full Test

The complete tests are modified in much the same way. Instead of calculating the average and then finding the residuals in relation to that average, the residuals are found in relation to each new rover point as shown in Figure 2.3. This throws out Equations 2.7, 2.8, and 2.9, and changes Equations 2.10, 2.11, and 2.12 to what is shown in Equations 2.21 - 2.23. Where n is the number of data points or rover points, and i is the series number or number of tests run. The entire test procedure can be visualized from data acquisition to the results in Figure 2.4.

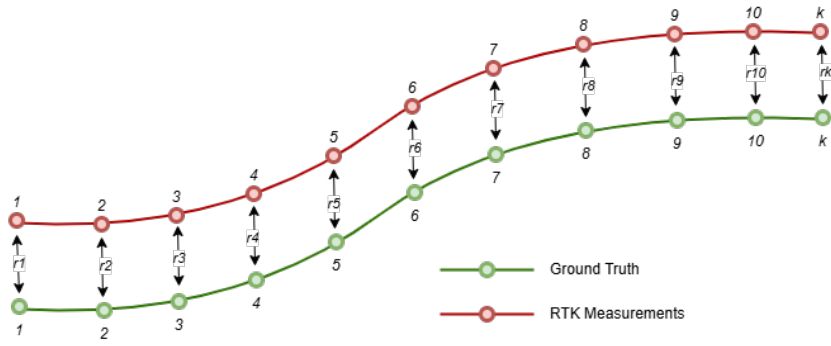


Figure 2.3. ISO 17123-8 Full Test Residuals: Instead of calculating the averages then taking the residuals from the mean, the residuals are taken from each point along the ground truth.

$$r_{xi,k} = x_{i,k} - x_{i,k}^* \quad , \quad r_{yi,k} = y_{i,k} - y_{i,k}^* \quad , \quad r_{hi,k} = h_{i,k} - h_{i,k}^* \quad (2.21)$$

$$\sum r_x^2 = \sum_{k=1}^n r_{xi,k}^2 \quad , \quad \sum r_y^2 = \sum_{k=1}^n r_{yi,k}^2 \quad , \quad (2.22)$$

$$\sum r_h^2 = \sum_{k=1}^n r_{hi,k}^2$$

$$v_x = v_y = v_h = (n-1)_i + (n-1)_{i+1} + (n-1)_{i+2} + \dots \quad (2.23)$$

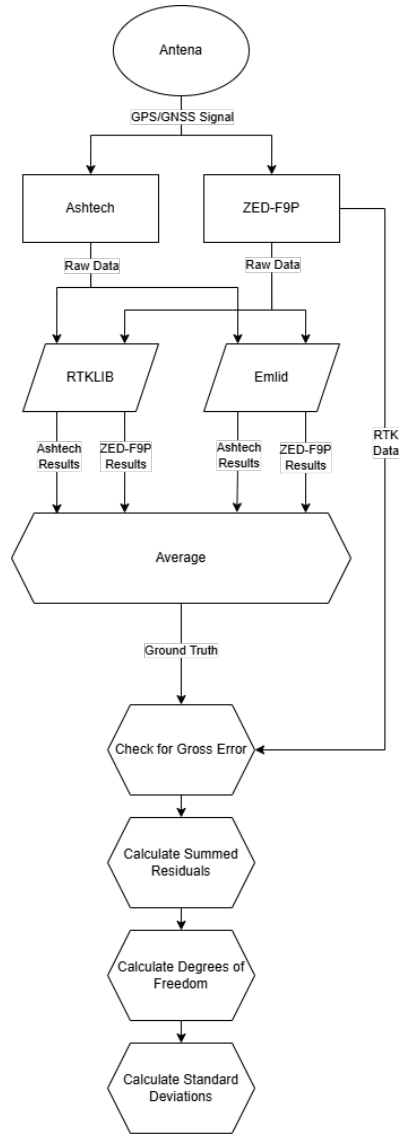


Figure 2.4. Data Flow Diagram: Flow of data from the antenna to the analysis between the RTK data and ground truth.

2.2 Hardware Setup

The Base Station shown in figure 2.5 is comprised of a JCA228f survey-grade antenna [16], splitter, ZED-F9P receiver [17], an Ashtech Z-Xtreme receiver [18], X-Bee radio [19] mounted to the ZED-F9P, and a Raspberry Pi data logger. The

Rover shown in figure 2.6 is identical, except the Raspberry Pi data logger is replaced with a laptop computer, and an external battery is used to supply power to the splitter.

The JCA228F GNSS survey-grade antenna can receive signals from the GPS, GLONASS, COMPASS, and Galileo constellations. This gave the ZED-F9P full access to the satellites it could view. Also, NOAA has conveniently logged the JCA228F in their publicly available antenna calibration files, which allows for compensation of the phase center offset during post-processing.

The Ashtech Z-Xtreme is a high-precision, dual-frequency GPS receiver developed primarily for geodetic and surveying applications. The datasheet [18] states that it has a horizontal accuracy of $1\text{cm} + 1\text{ppm}$ and a vertical accuracy of $2\text{cm} + 1\text{ppm}$, with a typical horizontal accuracy estimated to 2 drms, or with 95% probability. This makes it a great candidate for comparison against the ZED-F9P, whose datasheet also states a horizontal position accuracy of $1\text{cm} + 1\text{ppm}$ but is only estimated at CEP or 50% probability. Using the CEP to 2DRMS conversion [20] 2.4, the ZED-F9P's horizontal accuracy becomes 2.4 cm.

The computer and Raspberry Pi were used to log data. During experiments, the U-center was run on the laptop, and a data logging script was programmed to run on the Raspberry Pi's startup, as shown in Appendix D.2.

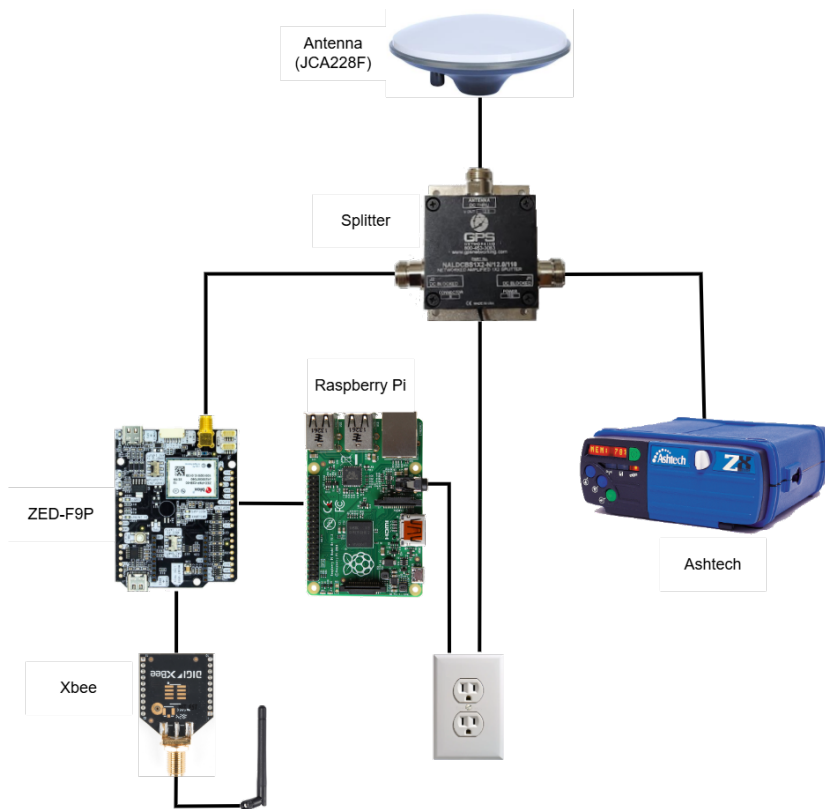


Figure 2.5. Base Station Connection Diagram: Both the Ashtech and ZED-F9P receivers are connected to the same antenna via the splitter. The Xbee radio and Raspberry Pi are connected to the Zed-F9P. The antenna and Raspberry Pi are plugged into a standard 120VAC wall outlet.

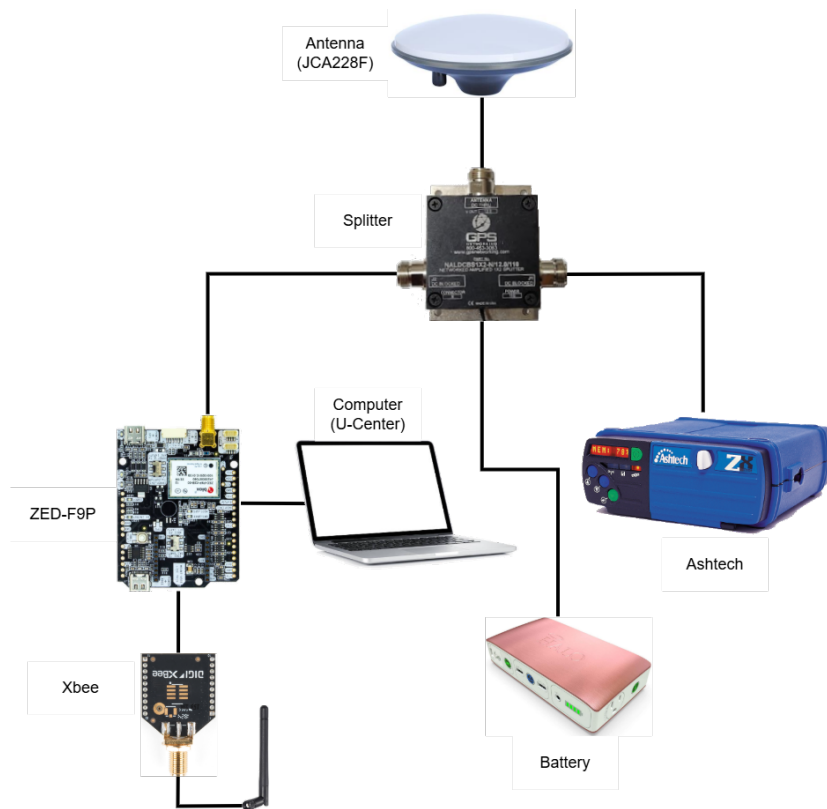


Figure 2.6. Rover Connection Diagram: Both the Ashtech and ZED-F9P receivers are connected to the same antenna via the splitter. The Xbee radio and Computer are connected to the Zed-F9P. The external battery provides power to the splitter, which provides power to the antenna.

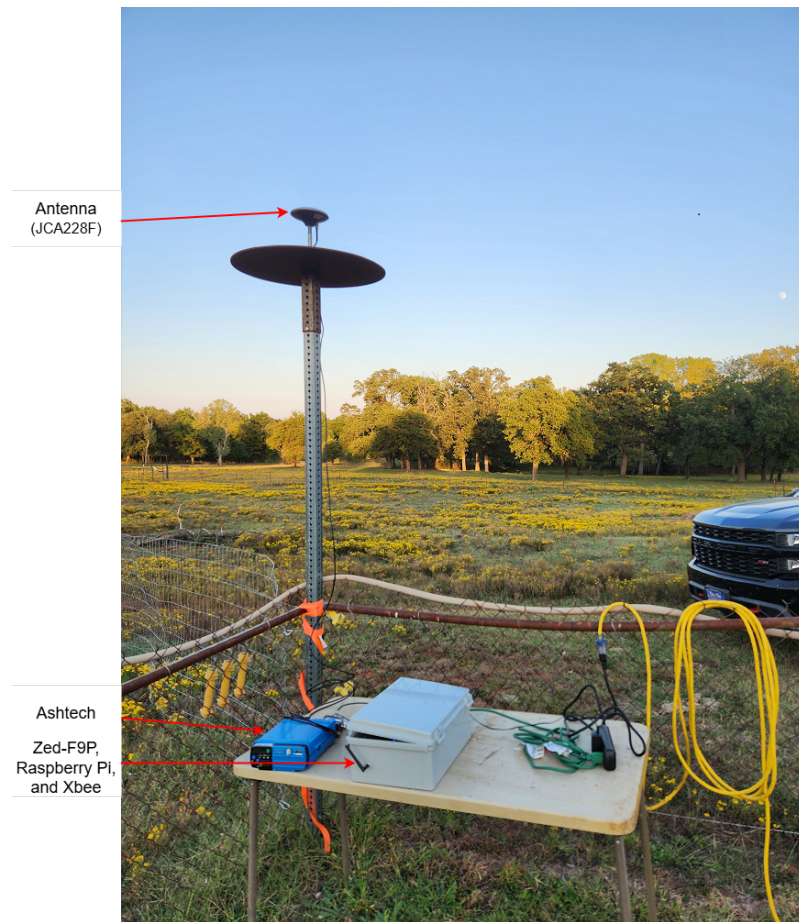


Figure 2.7. *Photo of the Base Station Setup, showing the antenna, Ashtech, and box containing the ZED-F9P, Raspberry Pi data logger, and Xbee radio.*



Figure 2.8. *Photo of the Rover Station Setup, showing the antenna, Ashtech, ZED-F9P, Computer, and Xbee radio*

Chapter 3

Experimental Procedures and Data Processing

3.1 Base Station Survey

Surveying in the base station is critical for the optimal accuracy of RTK since the rover gets corrections directly from the base. To mitigate environmental and systematic errors, data was collected over one 24-hour period and one 12-hour period at 30-second intervals on both the Ashtech and the ZED-F9P operating in static mode. Each data set was processed through the Canadian Spatial Reference System Precise Point Positioning (CSRS-PPP), and the results were averaged to yield the location of the base station. The full reports for those surveys can be found in the appendix but are summarized in Table 3.1. This base station location was used as the reference for kinematic post-processing and in the ZED-F9P modules for RTK.

Sample	Latitude	Longitude	Elevation(m)
Ashtech-24hr	35.31007031	-97.03346104	310.118
ZED-F9P-24hr	35.31007034	-97.03346108	310.171
Ashtech-12hr	35.31007046	-97.03346149	310.113
ZED-F9P-12hr	35.31007049	-97.0334615	310.131
Average	35.3100704	-97.03346128	310.13325

Table 3.1. Results from CSRS-PPP in decimal degrees format and height above ellipsoid along with their averages.

Base Station Error					
	$\sigma_e(m)$	$\sigma_n(m)$	$\sigma_{xy}(m)$	CEP	$\sigma_u(m)$
Total	.0229	.0098	.0249	.0207	.0263

Table 3.2. Base Station Standard Deviations: Error present in the base station.

3.2 Base and Rover Configuration

To take measurements, one set of Ashtech and ZED-F9P receivers needs to be configured to operate as base stations, and the other set needs to be configured as rovers. Data then needs to be logged from all devices. The following sections will cover the configuration of all devices and setting up the Raspberry Pi and U-Center computer interface for data logging.

3.2.1 Ashtech Configuration

The PCMCIA drives that the Ashtech receivers use to log data may periodically become corrupt, so it is necessary to reformat them before configuration and periodically afterward. The entire procedure is shown in the appendix using a Yet Another Terminal (YAT) as the interface between the computer and the Ashtech. The appendix contains the Ashtech Z-Xtreme quick reference guide, which may

be a good reference for base and rover configuration. Additional information about the Ashtech Z-Xtreme receivers can be found in the operations[21] and technical reference[22] manuals.

3.2.1.1 Ashtech Base

To set the Ashtech unit as a base station and start recording data, set the parameters according to Figure 3.1. Under "SurvConf", set "Mode" to "Static", ""ELEV MASK" to "15°", and "REC INT" to desired recording interval. In the case of kinematic tests, "REC INT" was set to "000.1 S", and for Surveying in "REC INT" was set to "030.0 S". Though not necessary, it is useful to set the site name under "Survey: mode" , which should read "Survey:Static" after previous step. To start a session, under "Session" go to "Stop Session", as the unit automatically starts recording data once it has enough satellites in view. Once the previous session is stopped, create a new session under "New Session", then start a new session under "Start Session". The Ashtech should now be configured as the base station and collecting data.

3.2.1.2 Ashtech Rover

The Ashtech rover's settings are similar to those of the base station. The only difference is that under "SurvConf," set the "Mode" to " Kinematic" and rename the test site under "Survey:mode, "SITE.

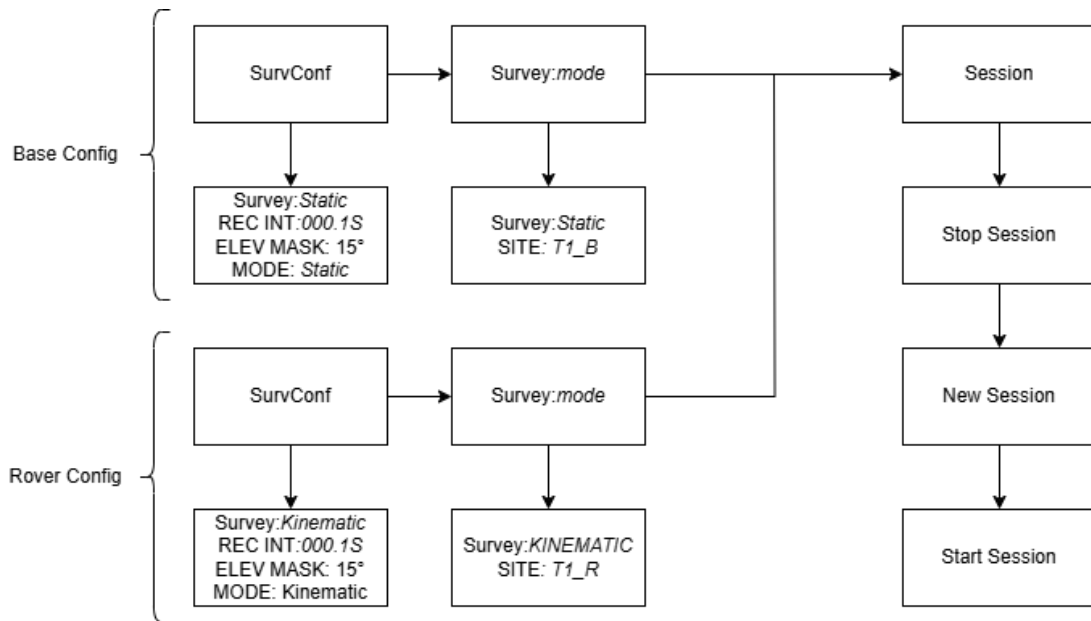


Figure 3.1. Ashtech Configuration Diagram

3.2.2 ZED-F9P Configuration

The ZED-F9P modules need to be configured to operate as regular receivers, collecting raw data that can then be post-processed, and as RTK receivers, passing correction data back and forth between the rover and base receivers. The ZED-F9P receivers do not have a way to store data on board, so data was stored externally via a Raspberry Pi for the base and a laptop for the rover. Xbee radio modules were used to communicate between the two receivers. Further information about the implementation and communication protocol for the ZED-F9P receivers can be found in the Integration Manual[23] and the Interface Description [24]. All configurations are done in the u-blox u-center 22.07 software, with the following configuration settings being shared by both Zed-F9P modules.

- UBX-CFG-PRT (Port Configuration) - Figure 3.2
- UBX-CFG-RATE (Acquisition Rate) - Figure 3.3
- UBX-CFG-GNSS (GNSS Configuration) - Figure 3.4
- UBX-CFG-DGNSS (Differential GNSS Configuration) - Figure 3.5

UBX - CFG (Config) - PRT (Ports)	
Target	1 - UART1
Protocol in	0+1+5 - UBX+NMEA+RTCM3
Protocol out	0+1+5 - UBX+NMEA+RTCM3
Baudrate	115200
Databits	8
Stopbits	1
Parity	None
Bit Order	LSB First

UBX - CFG (Config) - PRT (Ports)	
Target	2 - UART2
Protocol in	0+1+5 - UBX+NMEA+RTCM3
Protocol out	0+1+5 - UBX+NMEA+RTCM3
Baudrate	115200
Databits	8
Stopbits	1
Parity	None
Bit Order	LSB First

Figure 3.2. ZED-F9P UART Port Configuration: The USB, UART1, and UART2 ports were set to 0+1+5-UBX+NMEA+RTCM3 for both Protocol in and Protocol out. The Baudrate, Databits, Stopbits, Parity, and Bit Order were set the same for UART1 and UART2.

UBX - CFG (Config) - RATE (Rates)	
Time Source	1 - GPS time
Measurement Period	100 [ms]
Measurement Frequency	10.00 [Hz]
Navigation Rate	1 [cyc]
Navigation Frequency	10.00 [Hz]

Figure 3.3. ZED-F9P Sample Rate Settings: The acquisition rates were set to 1 sample every .1 seconds (10HZ) for the kinematic tests, and 1 sample every 30 seconds (.033Hz) for surveying in the base station.

ID	GNSS	Configure	Enable	min	max	Signals
0	GPS	☒	☒	8	16	☒ L1C/A
1	SBAS	☒	☒	3	3	☒ L1C/A
2	Galileo	☒	☒	10	18	☒ E1
3	BeiDou	☒	☒	2	5	☒ B1
4	IMES	☐	☐	0	0	☐ L1C/A
5	QZSS	☒	☒	0	4	☒ L1C/A ☐ L1S
6	GLONASS	☒	☒	8	12	☒ L10F
7	NAVIC					

Number of channels: 60
Number of channels to use: 60 ☐ Auto set

Figure 3.4. ZED-F9P GNSS Configuration: All GNSS constellations available to the modules were used to get the best solution available.

UBX - CFG (Config) - DGNSS (Differential GNSS configuration)

Differential: 3 = RTK fixed: Ambiguities are fixed whenever possible.

Figure 3.5. ZED-F9P DGNSS Configuration: This setting is imperative to ensure the RTK fixes ambiguities whenever possible.

3.2.2.1 ZED-F9P Base

The ZED-F9P must be set at a known location for a base station. The average location was calculated from the PPP surveys and input into the Fixed Mode under UBX-CFG-TMODE3 as seen in Figure 3.6. The base must also have the output messages configured under the UBX-CFG-MSG window, listed in Table 3.3. The messages being sent to the USB port are messages that can be read on the laptop via U-center, the messages being sent to the UART1 port are being logged to the Raspberry Pi, and the messages being sent to the UART2 port are being broadcast to the rover receiver via the XBee module. As a note, setting the NAV-PVT Message to USB is critical to ensure that messages will be sent to the rover receiver.

UBX - CFG [Config] - TMODE3 [Time Mode 3] 3774

Mode: 2 - Fixed Mode

Survey-In

Minimum Observation Time: 0 [s]

Required Position Accuracy: 0.0000 [m]

Survey-in ends if BOTH conditions are

Fixed Position

Lat: 35.310070400 [deg]

Long: -97.033461280 [deg]

Alt: 310.1332 [m]

Accuracy: 0.0070 [m]

☒ Use Lat / Lon / Alt Position

Figure 3.6. ZED-F9P TMODE 3 Configuration: Base Station Location.

Message	USB	UART1	UART2
RXM-RAWX	✓	✓	
RXM-SFRBX	✓	✓	
RTCM3.3 1005			✓
RTCM3.3 1077			✓
RTCM3.3 1087			✓
RTCM3.3 1097			✓
RTCM3.3 1127			✓
RTCM3.3 1230			✓
NAV-POSLLH			✓
NAV-PVT	✓	✓	✓
NAV-RELPOSNED			✓
NAV-SIG			✓
NAV-STATUS			✓
NAV-SVIN			✓

Table 3.3. Active output messages on the ZED-F9P Base. **IMPORTANT:** NAV-PVT must be active on the USB port or messages will not send.

3.2.2.2 ZED-F9P Rover

The rover must also have output messages configured under the UBX-CFG-MSG window, listed in Table 3.4. The messages sent to the USB port can be read on the laptop via U-center and saved in a data file, and the messages sent to the UART2 port are broadcast to the rover receiver via the Xbee module. As a note, setting the NAV-PVT Message to USB is still very important, or no messages will be sent to the base receiver.

Message	USB	UART1	UART2
RXM-RAWX	✓		
RXM-SFRBX	✓		
NMEA GxGGA	✓		
NMEA GxGLL	✓		
NMEA GxGSA	✓		
NMEA GxGST	✓		
NMEA GxGNS	✓		
NAV-DOP	✓		✓
NAV-HPPOSECEF	✓		✓
NAV-HPPOSLLH	✓		✓
NAV-POSECEF			✓
NAV-POSLLH			✓
NAV-PVT	✓		✓
NAV-RELPOSNED	✓		✓
NAV-SAT			✓
NAV-SIG			✓
NAV-STATUS			✓
NAV-SVIN			✓

Table 3.4. Active output messages on the ZED-F9P Rover. **IMPORTANT:** NAV-PVT must be active on the USB port or messages will not send.

3.2.3 Raspberry Pi Data Logger

A Raspberry Pi 3B+ was used to collect data from the ZED-F9P base station as it was a cheap alternative to a second laptop. Appendix D.2 shows the wiring diagram, pinout, and associated code with the data logger. It uses the pyubx2

library to read and parse the raw data from the ZED-F9P rover receiver and write it to a local file on the Raspberry Pi. The program runs upon powering up the Raspberry Pi and starts recording data when the rocker switch is pressed. An LED also lights up when the program writes data to indicate that the logger is functioning correctly.

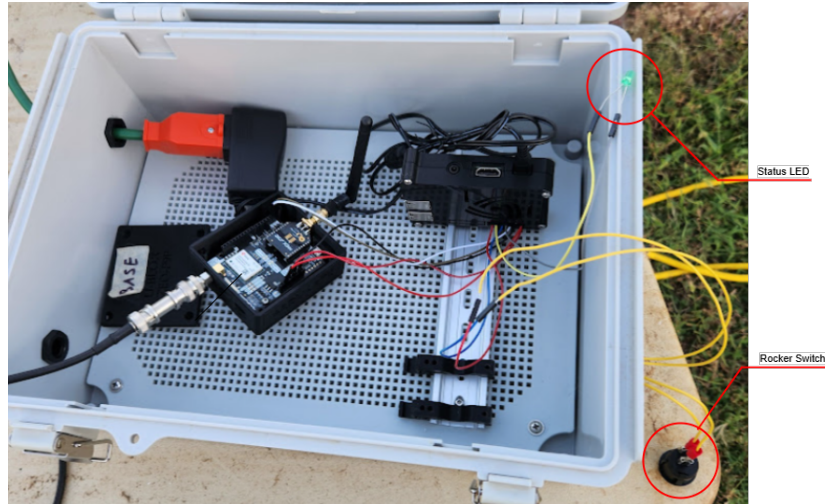


Figure 3.7. Base Station Data Logger: Base station box with ZED-F9P receiver connected to the Raspberry Pi data logger.

3.3 Data Conversion and Post Processing

3.3.1 Data Conversion

3.3.1.1 Ashtech Data Conversion

The Ashtech receivers write data onto PCMCIA cards, so to access the files it is necessary to use the USB card reader shown in Figure 3.8. The files are stored in a proprietary compressed MICRO_Z.BIN format that requires a proprietary program called "Ashtech Download", Figure 3.9, in order to decompress the files.



Figure 3.8. PCMCIA Card Reader: Used to interface between the PCMCIA Cards and Computer.

To decompress the files, the external drive containing the PCMCIA card needs to be selected from the drop-down in the left window, E: in this case, and the drive containing the desired destination needs to be selected from the drop-down in the right, C: local disk in this case. The desired folder can then be navigated to in the right window and the necessary files can be drug from the left window to the right. Figure 3.9 shows file b1_e05.77 being decompressed into the four files in the right window.

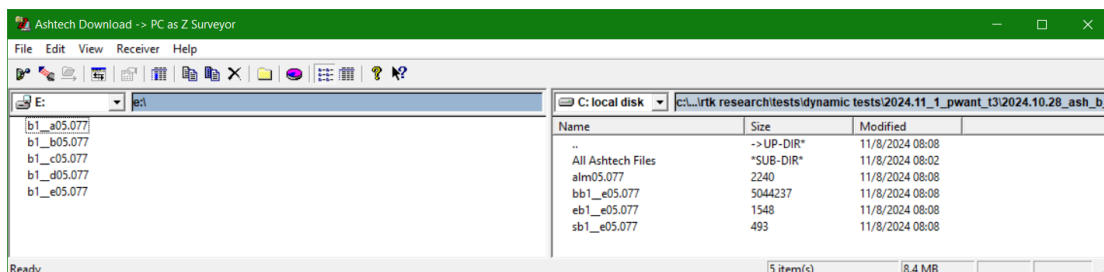


Figure 3.9. Ashtech Download: Used to decompress proprietary Ashtech files.

To post-process the raw Ashtech data, the data must be in observation (.obs) and navigation (.nav) files using RINEX format. Another proprietary software, GrafNav [18] converts the raw GPS data to a proprietary GPB format and then from GPB to RINEX format using two separate tools. To open the raw GPS to GPB converter, select File → Convert → Raw GPS to GPB as shown in Figure 3.10.

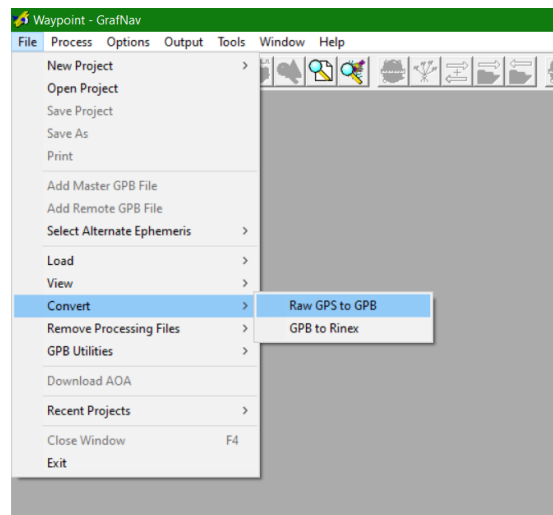


Figure 3.10. *GraphNav: Convert Tools Location.*

Once in the GPS to GPB convert tool, select "Get Folder" to browse to the folder containing the raw GPS files that need to be converted. They will show up under the "Source Files" window. Select "Auto Add All" to move the appropriate files to the "Convert Files" Window. Then press "Convert," and a message window will pop up showing when the process has been completed and how many files have been successfully converted.

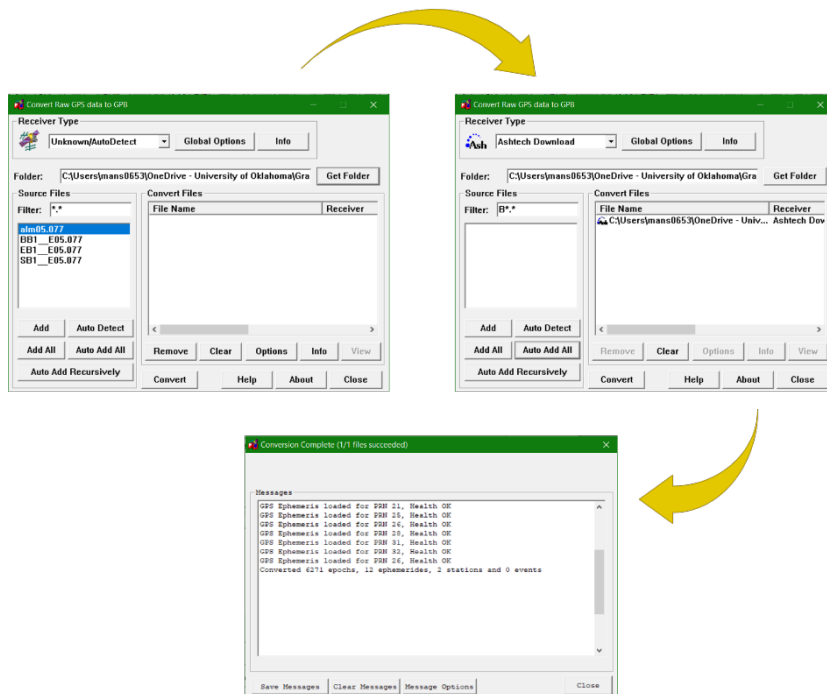


Figure 3.11. GraphNav Raw GPS to GPB Conversion: Select the folder where the files are located using "Get Folder". Push "Auto Add All" to add the appropriate files. Then, push "Convert" to convert them. The message screen should pop up and show that one file has been converted.

The Ashtech data is now available in GPB files but needs to be converted to observation and navigation files in RINEX format. To convert from GPB to RINEX, select File → Convert → GPB to Rinex as seen in figure 3.10. Select "Add" and browse the GPB file that needs to be converted, then select "Convert". The "Message" window will show when the conversion has been completed. Observation and navigation files should now be in the folder along with the GPB file.

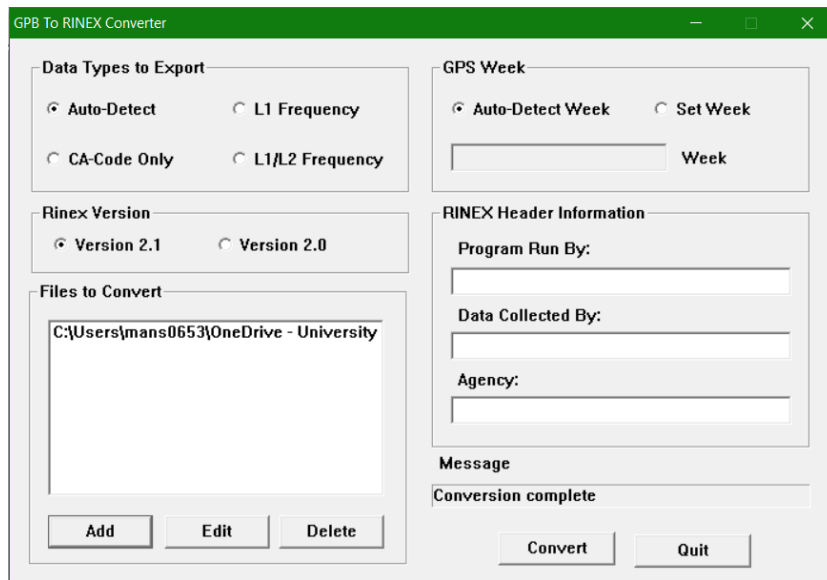


Figure 3.12. GraphNav GPB to RINEX Conversion: Select "Add" and browse to the appropriate GPB file, then press "Convert". The observation and navigation files should then be located in the folder along with the GPB file.

3.3.1.2 ZED-F9P Data Conversion

The ZED-F9P saves raw data in the UBX (.ubx) format. This data also needs to be converted to observation and navigation files in RINEX format for post-processing. The RTKCONV tool in RTKLIB was used for this conversion. Before converting, set the options under the "Options" button to allow all necessary satellite systems and GNSS Signals, and insert the appropriate antenna number as shown in Figure 3.13. Under "RTCM, RCV RAW or RINEX OBS" select the button with the three dots to the right, then browse to the UBX file to be converted. Under "Output Directory," select the button with the three dots to the right and browse to the desired output directory. This step is very important as it will default to a random directory. Press "Convert," and it will display the progress at the bottom and indicate when it is complete. Figure 3.14 shows an example of the RTKCONV window.

Options ✕

RINEX Ver 3.03 ▼ ☐ Sep NAV Station ID 0000 ☐ RINEX2 Name

RunBy/Obsrv/Agency

Comment

Maker Name/#/Type

Rec #/Type/Vers

Ant #/Type

Approx Pos XYZ ☐

Antenna Delta H/E/N

☐ Phase Shift ☐ Half Cyc Corr ☐ Iono Corr ☐ Time Corr ☐ Leap Sec

Satellite Systems

☒ GPS ☒ GLO ☒ GAL ☒ QZS ☒ BDS ☐ NavIC ☐ SBS

Excluded Satellites

Observation Types

GNSS Signals ?

☒ C ☒ L ☒ D ☒ S

☒ L1 ☒ L2/E5b ☐ L5/E5a ☐ L6

Mask...

FCN...

Receiver Options

Time Tolerance (s) 0.005 Debug OFF ▼ OK Cancel

Figure 3.13. RTKCONV Options: Set the appropriate satellite systems, GNSS signals, and Antenna number.

38

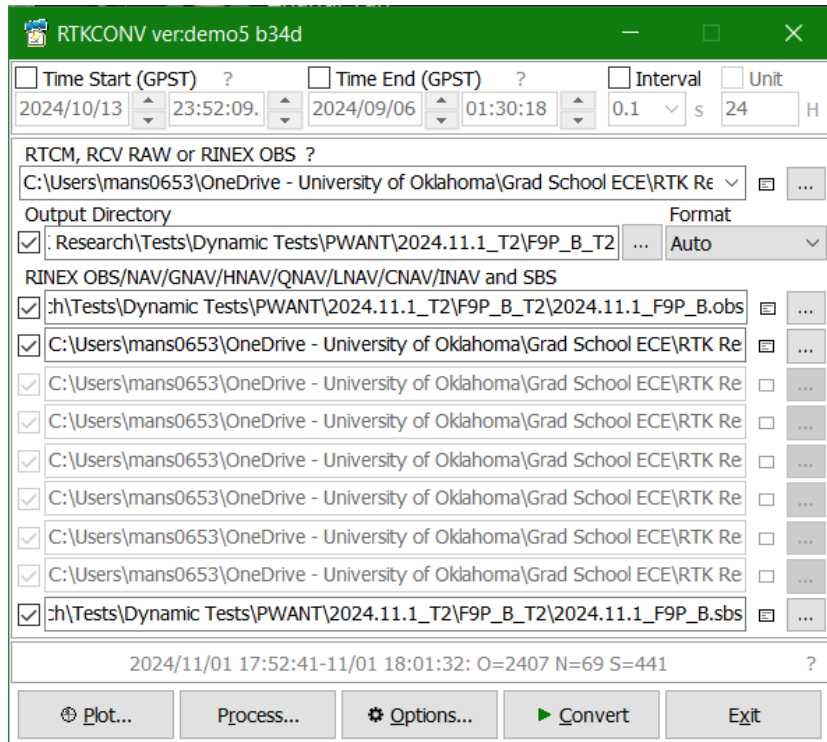


Figure 3.14. RTKCONV UBX to RINEX Conversion: Select the correct file and output directory, then press convert.

3.3.2 Kinematic Post Processing

The Kinematic Post processing was done with two free software suits, RTKLIB and Emlid Studio. RTKLIB was chosen because it is open source, has many documentation [25], and a history of being used in the research community. Emlid is proprietary software but it is free to download and use and offers a necessary comparison to RTKLIB to adjust for any errors inherent to the software.

3.3.2.1 RTKLIB Kinematic Post Processing

The RTKPOST subprogram was used to post process GNSS data in RTKLIB. Before beginning, it is necessary to make modifications to a few of the post

processors settings. Figures 3.15, 3.16, and 3.17, show the current options that have been modified. Under "Setting1", the "Positioning Mode" was switched to "Kinematic" as the tests being ran were kinematic. The "Frequencies" were changed to "L1+L2/E5b" as these are the frequencies at which the JCA228F0001 antenna operates. The "Filter Type" was set to "Combined", which allowed the EKF to provide a forward and backward solution, decreasing the time to a fixed solution. Then all the satellites available to the ZED-F9P were selected. As a note, these same constellations were left on for the Ashtech as it did not seem to make a difference in the solution. Under the "Positions" menu, the position coordinates of the base station were set according to the averages calculated in Table 3.1. Then in the "Files" menu, the antenna calibration file found on the NOAA website was inserted into the "NGS PCV" field.

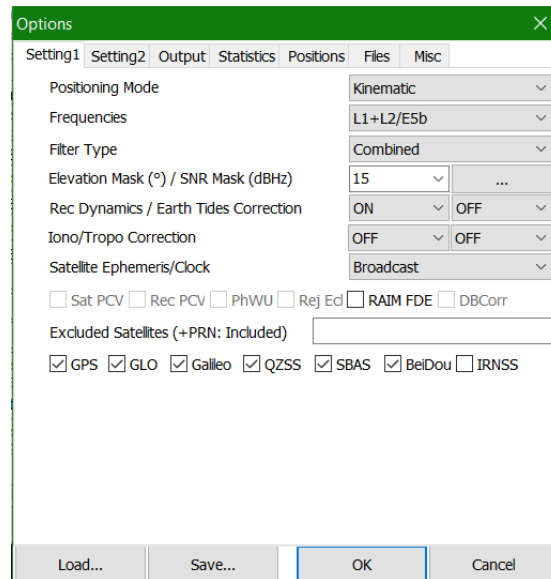
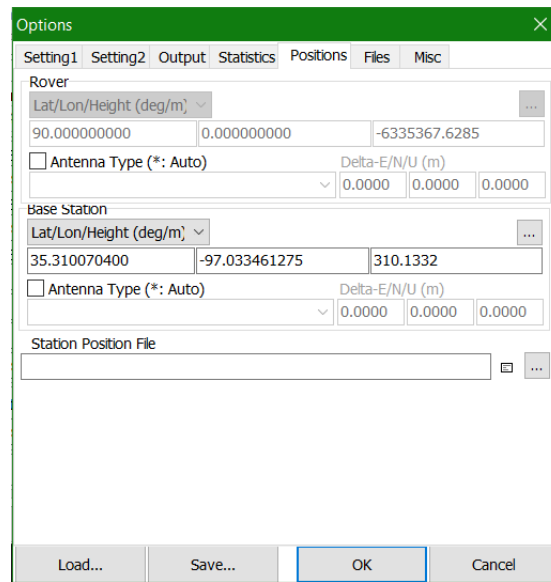
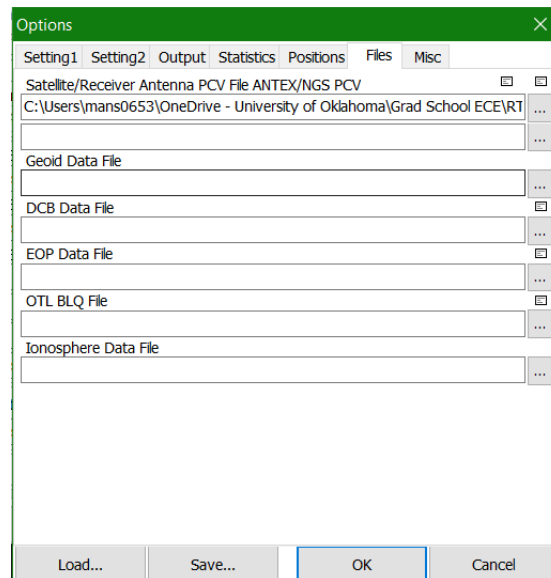


Figure 3.15. RTKPOST Setting1 Configuration: Set Positioning Mode to Kinematic, Frequencies to L1+L2/E5b, Filter Type to Combined, Elevation Mask to 15, and enable the satellite constellations shown.



The 'Options' dialog box has the 'Positions' tab selected. It contains two main sections: 'Rover' and 'Base Station'. Each section has a 'Lat/Lon/Height (deg/m)' dropdown menu, three input fields for coordinates, and a 'Delta-E/N/U (m)' section with three input fields. The 'Rover' section has values: 90.000000000, 0.000000000, -6335367.6285. The 'Base Station' section has values: 35.310070400, -97.033461275, 310.1332. Both sections have an 'Antenna Type (*: Auto)' checkbox and a dropdown menu. At the bottom, there is a 'Station Position File' input field and buttons for 'Load...', 'Save...', 'OK', and 'Cancel'.

Figure 3.16. RTKPOST Positions Configuration: Set the base station location from the PPP averages listed in Table 3.1.



The 'Options' dialog box has the 'Files' tab selected. It contains several input fields for file paths, each with a browse button to its right. The fields are: 'Satellite/Receiver Antenna PCV File ANTEX/NGS PCV' (with a file path: C:\Users\mans0653\OneDrive - University of Oklahoma\Grad School ECE\RT), 'Geoid Data File', 'DCB Data File', 'EOP Data File', 'OTL BLQ File', and 'Ionosphere Data File'. At the bottom, there are buttons for 'Load...', 'Save...', 'OK', and 'Cancel'.

Figure 3.17. RTKPOST Files Configuration: Download the antenna calibration file (ngs20.atx) from NOAA's website, and link to it in the box labeled "NGS PCV".

Once the settings have been modified, exit the Options menu and search for the rover and base station observation files using the buttons to the right

with the three dots. If available, add the base station navigation file and .sbs in the correct locations shown in Figure 3.18, then press the "Execute" button. The post-processor will run and show a notification when complete. The folder should now contain position files containing the solutions. Pressing the "Plot" button in the lower left will automatically generate a plot of the solution data.

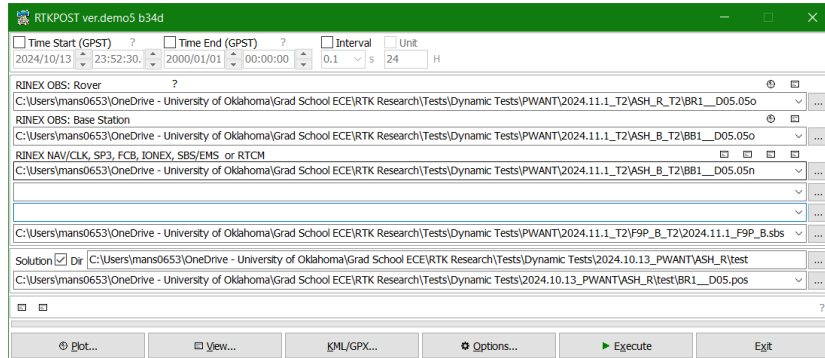


Figure 3.18. RTKPOST: Select the observation file for the rover and the base, the navigation file for the base, and the .sbs file for the base if available. Point to the folder where you would like the solution files placed.

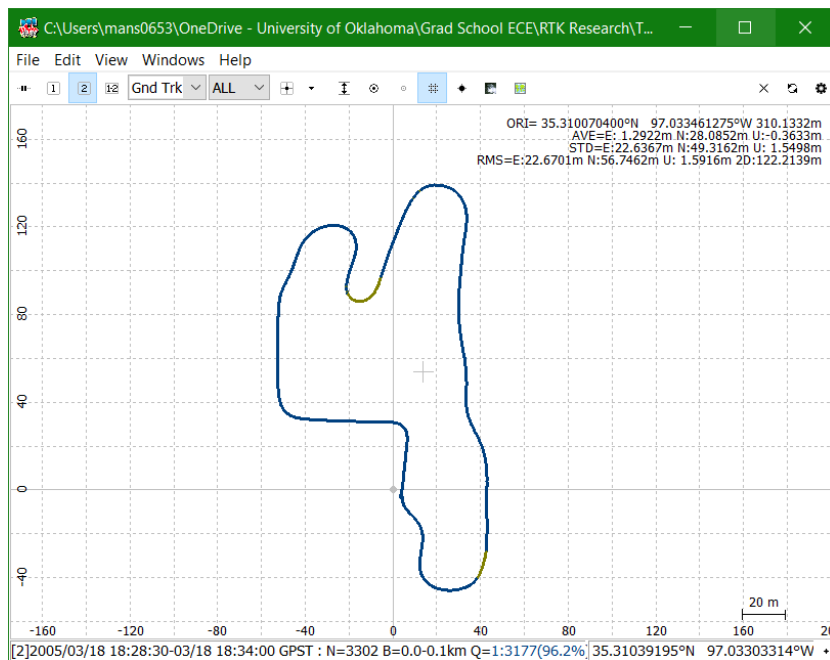


Figure 3.19. RTKPLOTTING: Quick plot generated from the RTKLIB post processed data.

3.3.2.2 Emlid Kinematic Post Processing

Post-processing in Emlid Studio is simple but offers less control over the solution calculation. Select the processing method from the drop down in the upper left, in this case "Kinematic processing", then drag and drop the u-blox or RINEX files into the correct location. Press the gear icon in the bottom left corner and change the "Filter type" to "Combined". Input the location of the base station, in this case the averaged locations from Table 3.1, then press "Process". Figure 3.20 shows an example of the quick plot generated by Emlid Studio.

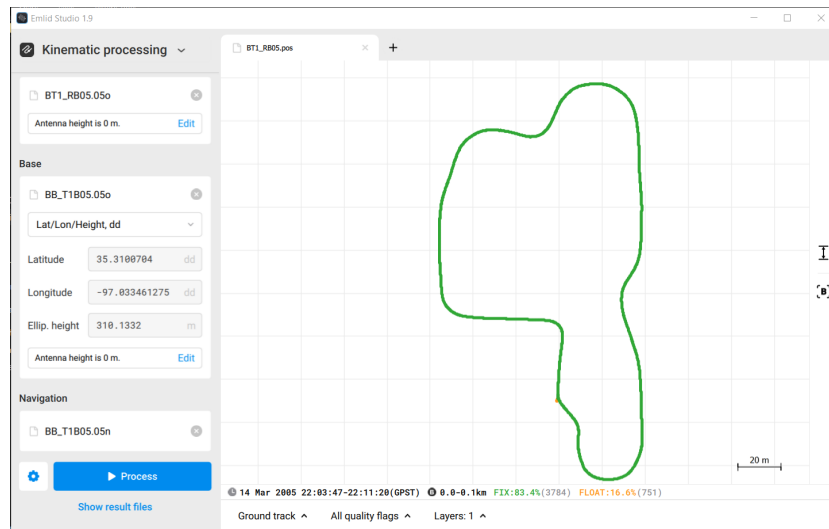


Figure 3.20. *Emlid Post Processor: Quick plot generated from the Emlid post-processed data.*

3.3.3 RTK Data Extraction

The processed RTK data from the rover is stored in NMEA and UBX format on the UBX files. Appendix 5.2 contains the code used to extract and parse through the data using the pyubx2 python library. The ZED-F9P Interface Description [24] contains all of the message identifiers necessary to parse through messages that have been activated in the u-center message configuration window.

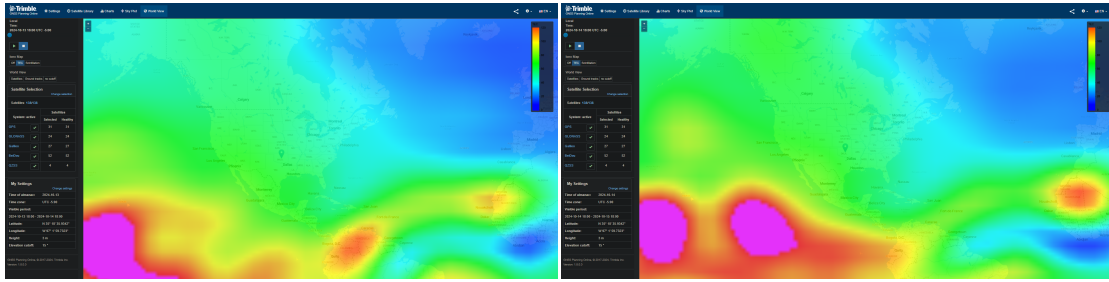
Chapter 4

Results and Discussions

4.1 Results

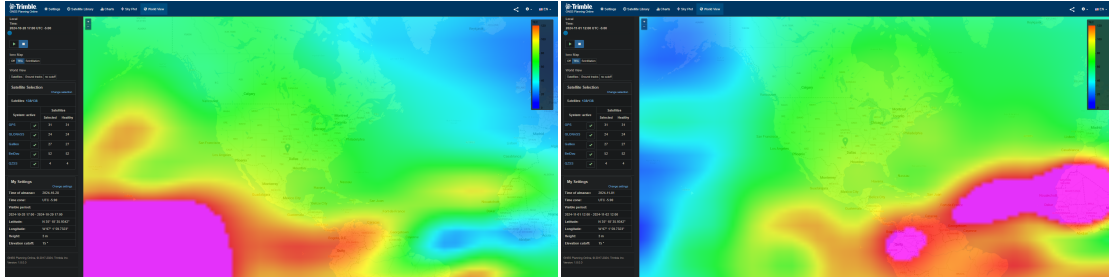
4.1.1 Tests

There were six tests run in total for this set of experiments. The first was on October 13th, the second was on October 14th, the third was on October 28th, and the last three were on November 1st, 2024. The last three tests spanned over an hour and a half, meeting the requirements for the full test parameters in ISO 17123-8. Figure 4.1 shows the ionospheric conditions for the tests. The conditions could have been slightly better, performing tests early in the morning, as the ionospheric disturbance is at a minimum at midnight and maximum during mid-day [26]. All satellites were verified operational, and skies were free of cloud cover, except for Test 3, which will be discussed further. Test 1 was conducted while walking, and the rest were conducted while driving at approximately 4-7 mph in an open field.



(a) 10.13.2024 Test 1

(b) 10.14.2024 Test 2

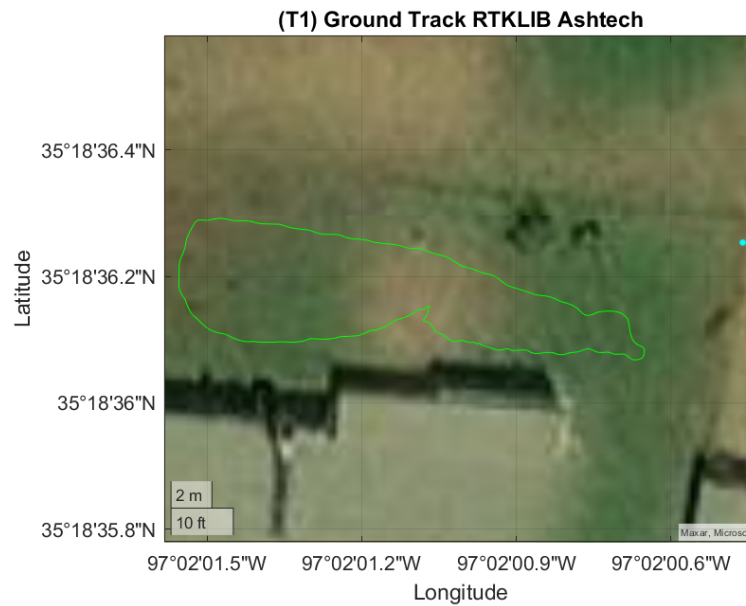


(c) 10.28.2024 Test 3

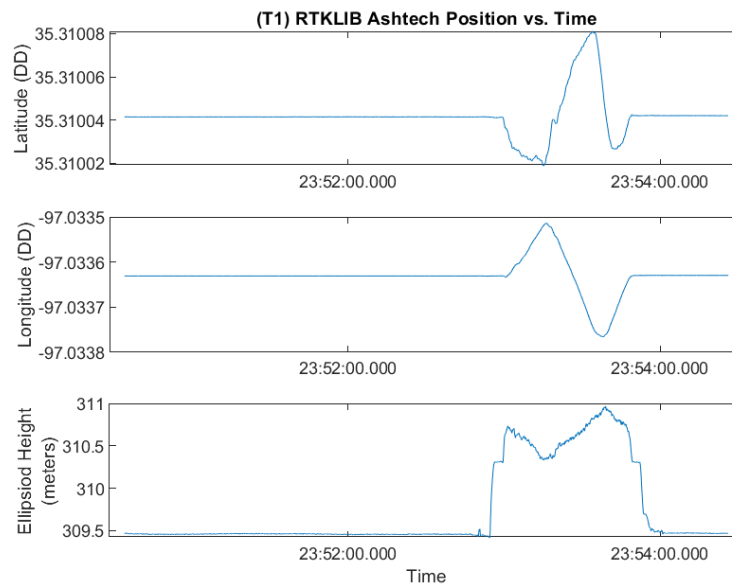
(d) 11.1.2024 Test 4-Test 6

Figure 4.1. Ionospheric Conditions: Ionospheric conditions at the time and day of each test.

4.1.1.1 Test 1

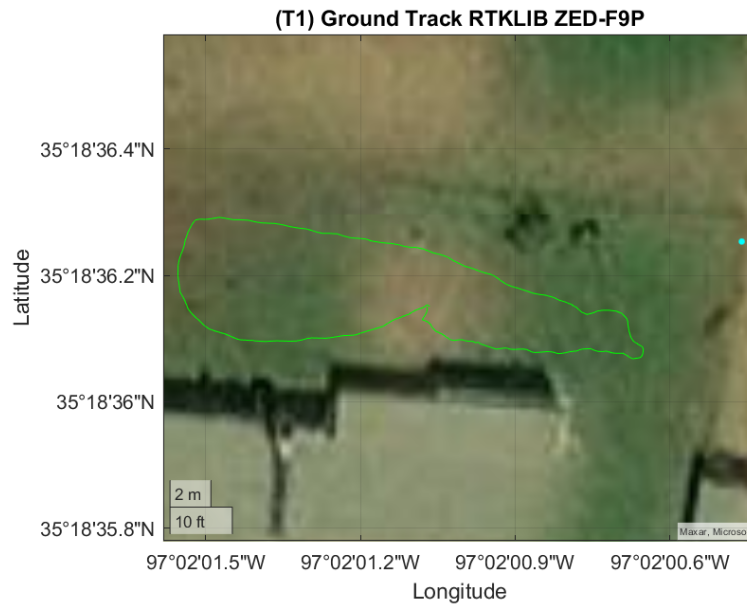


(a) *Test 1, RTKLIB Ashtech, Ground Track*

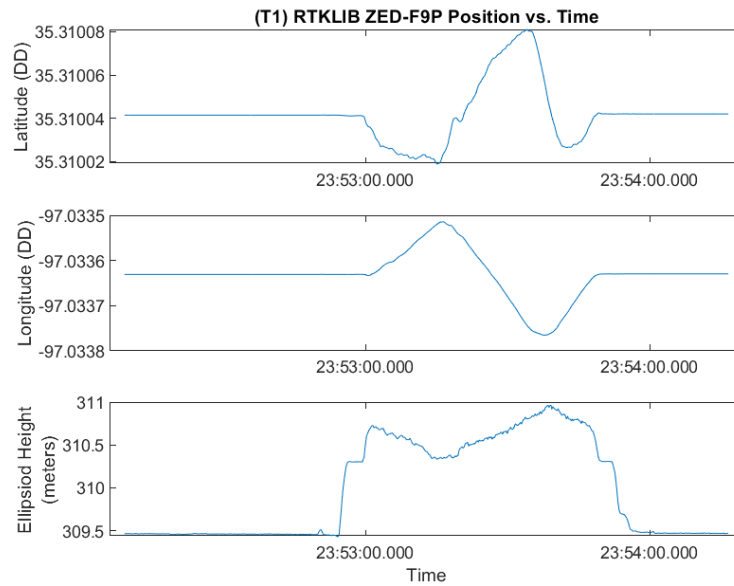


(b) *Test 1, RTKLIB Ashtech, Position*

Figure 4.2. Test 1 RTKLIB Ashtech: RTKLIB Post Processor with the Ashtech Receiver Ground Track and Position view for Test 1.

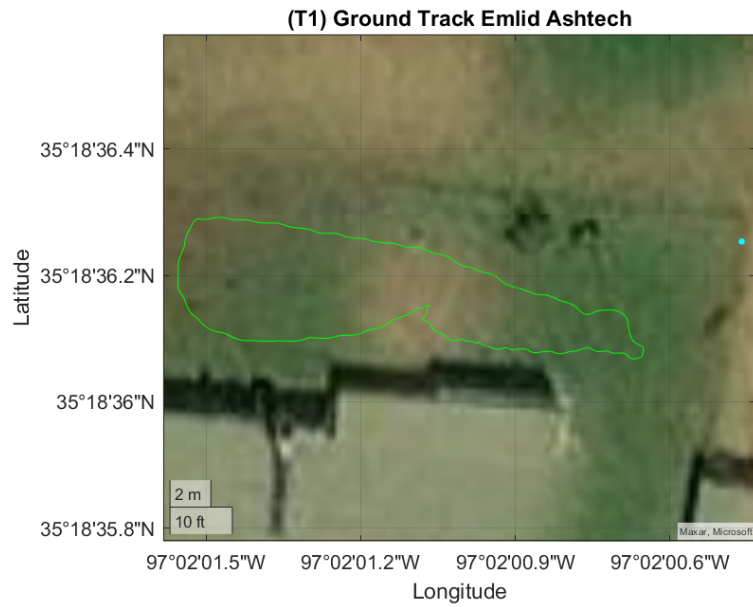


(a) *Test 1, RTKLIB Zed-F9P, Ground Track*

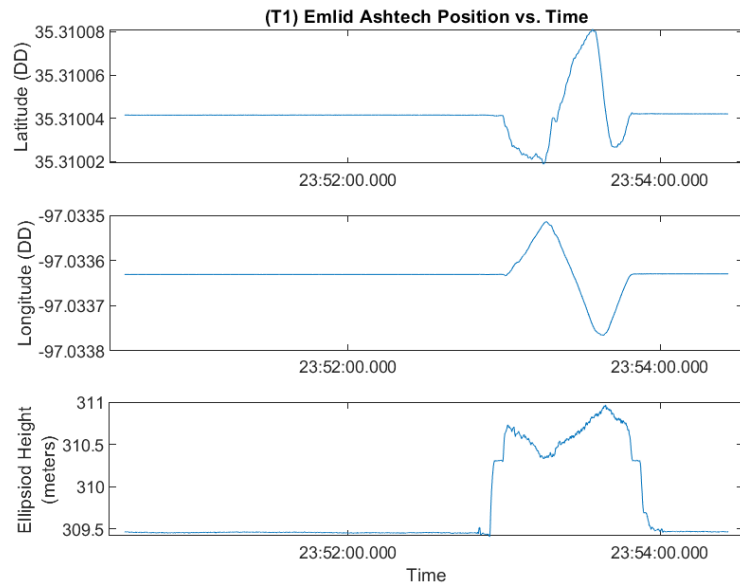


(b) *Test 1, RTKLIB Zed-F9P, Position*

Figure 4.3. *Test 1 RTKLIB ZED-F9P: RTKLIB Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 1.*

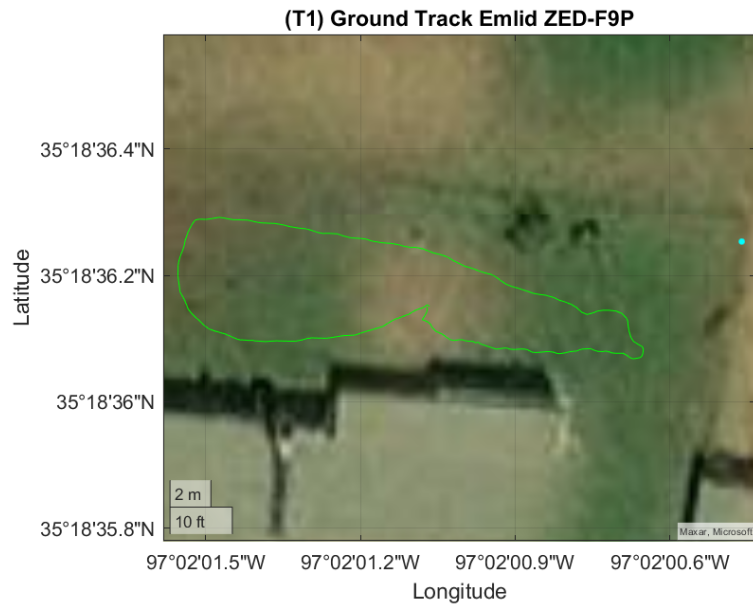


(a) *Test 1, Emlid Ashtech, Ground Track*

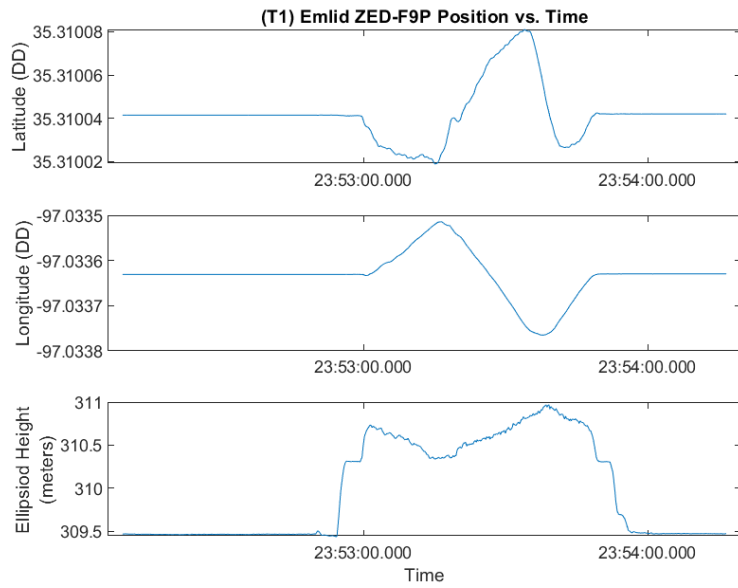


(b) *Test 1, Emlid Ashtech, Position*

Figure 4.4. Test 1 Emlid Ashtech: Emlid Post Processor with the Ashtech Receiver Ground Track and Position view for Test 1.

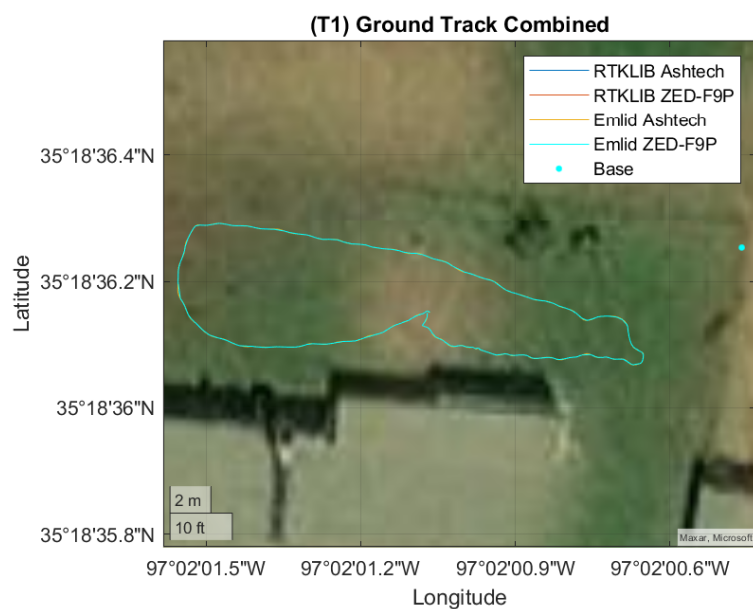


(a) *Test 1, Emlid Zed-F9P, Ground Track*

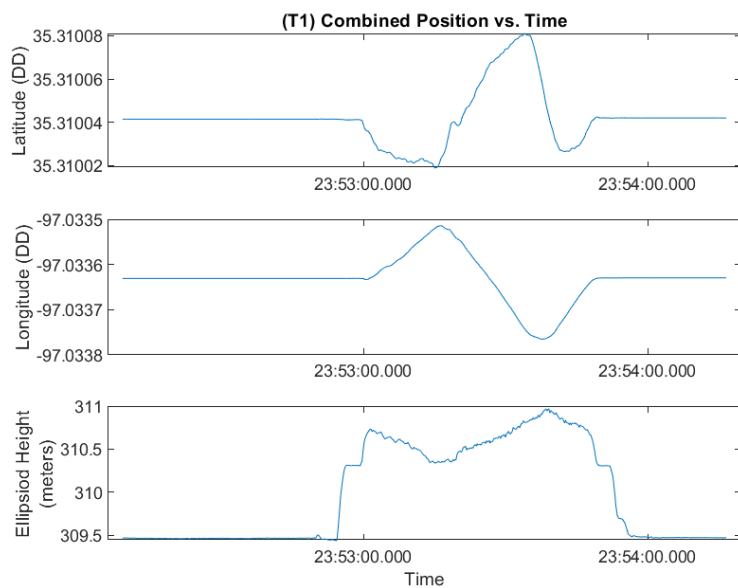


(b) *Test 1, Emlid Zed-F9P, Position*

Figure 4.5. *Test 1 Emlid ZED-F9P: Emlid Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 1.*

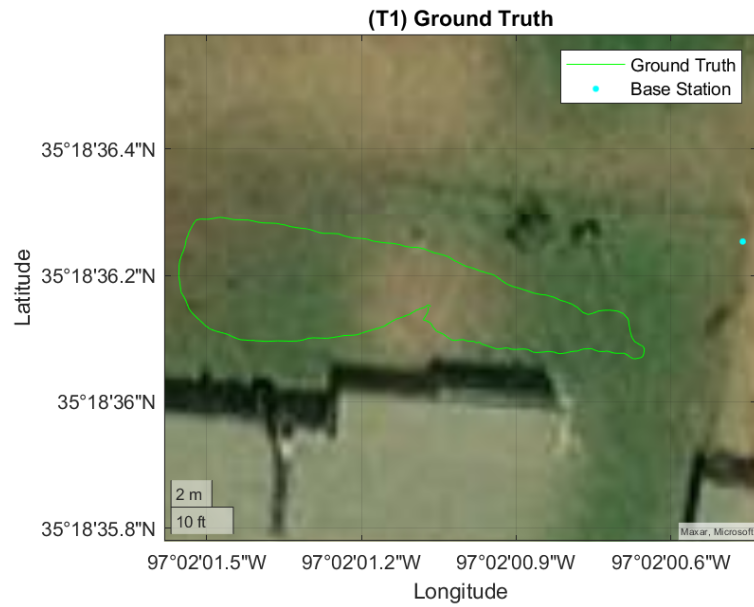


(a) *Test 1, Combined, Ground Track*

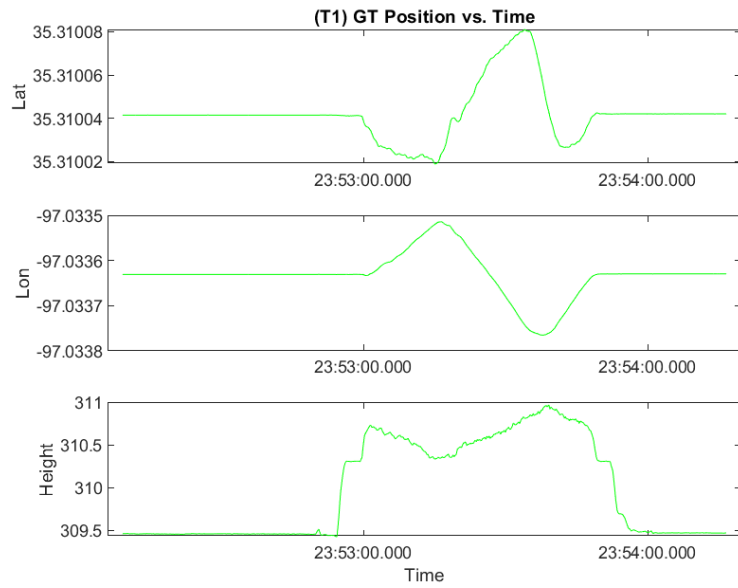


(b) *Test 1, Combined, Position*

Figure 4.6. *Test 1 Combined: Emlid and RTKLIB Post Processors with the ZED-F9P and Ashtech Receivers Ground Track and Position views for Test 1.*

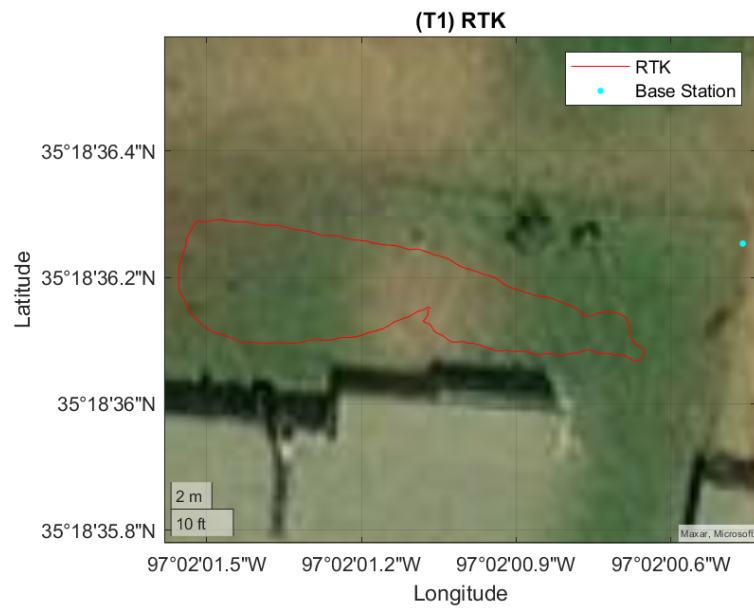


(a) *Test 1, Ground Truth, Ground Track*

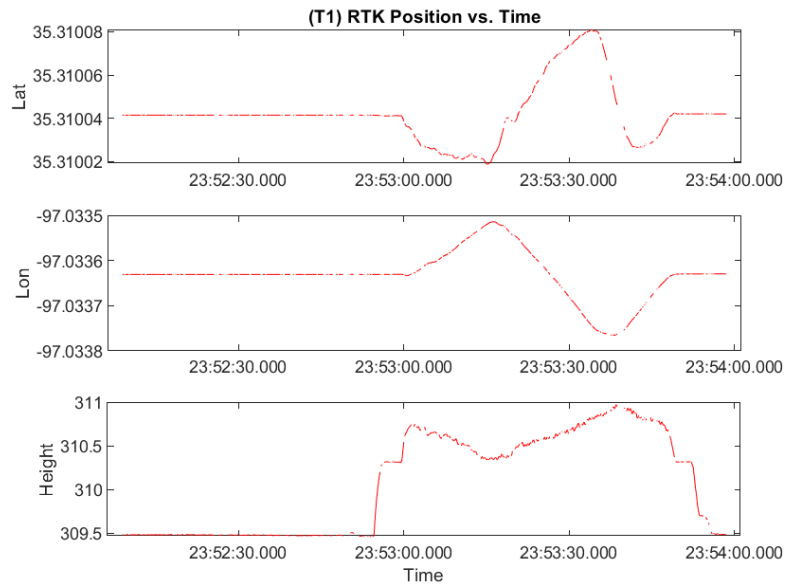


(b) *Test 1, Ground Truth, Position*

Figure 4.7. Test 1 Ground Truth: Ground Truth Ground Track and Position view for Test 1.

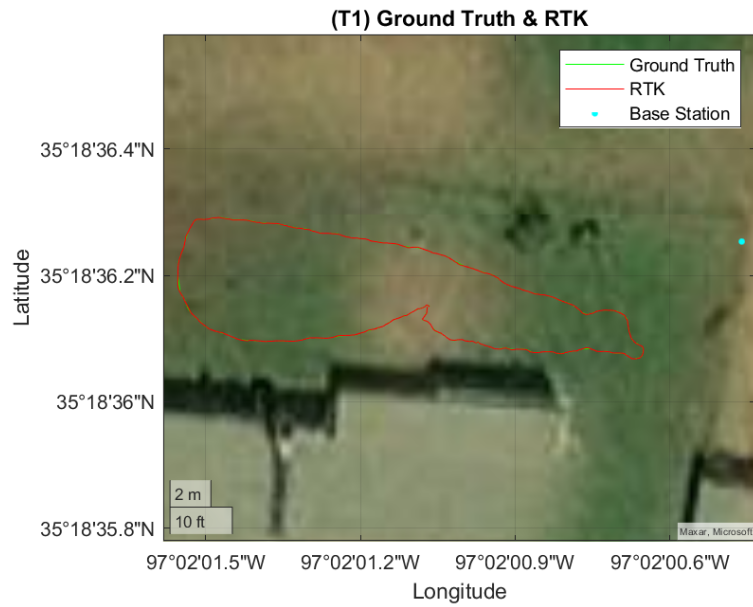


(a) Test 1, RTK, Ground Track

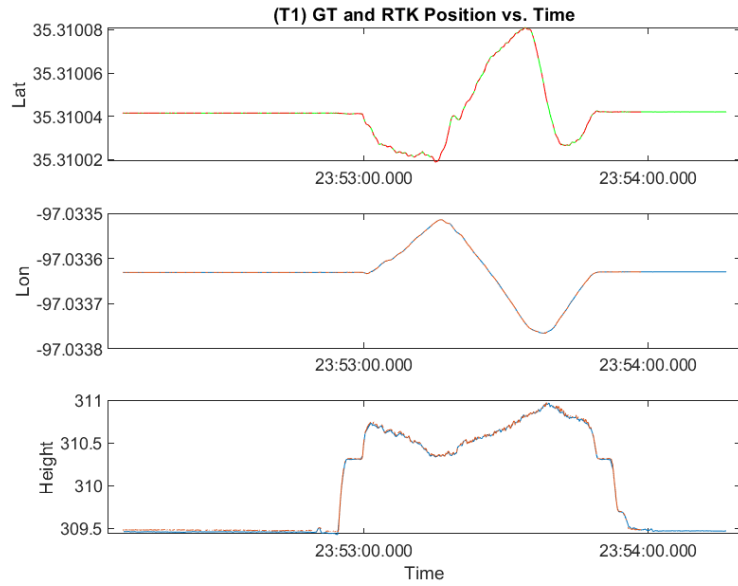


(b) Test 1, RTK, Position

Figure 4.8. Test 1 RTK: RTK Ground Track and Position view for Test 1.



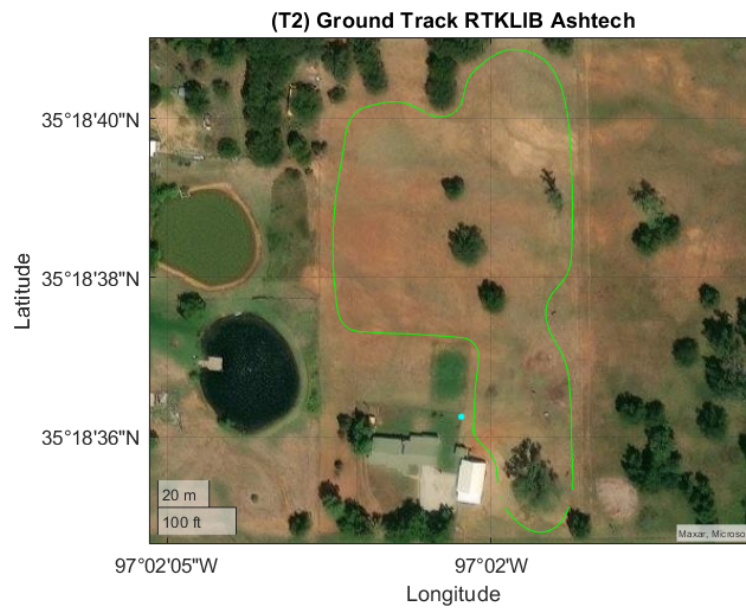
(a) *Test 1, GT and RTK, Ground Track*



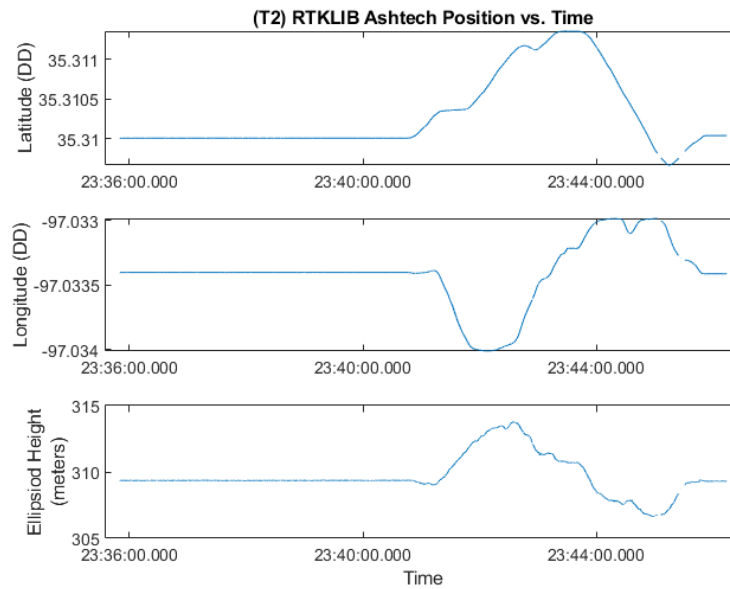
(b) *Test 1, GT and RTK, Position*

Figure 4.9. *Test 1 GT and RTK: Ground Truth and RTK Ground Track and Position view for Test 1.*

4.1.1.2 Test 2

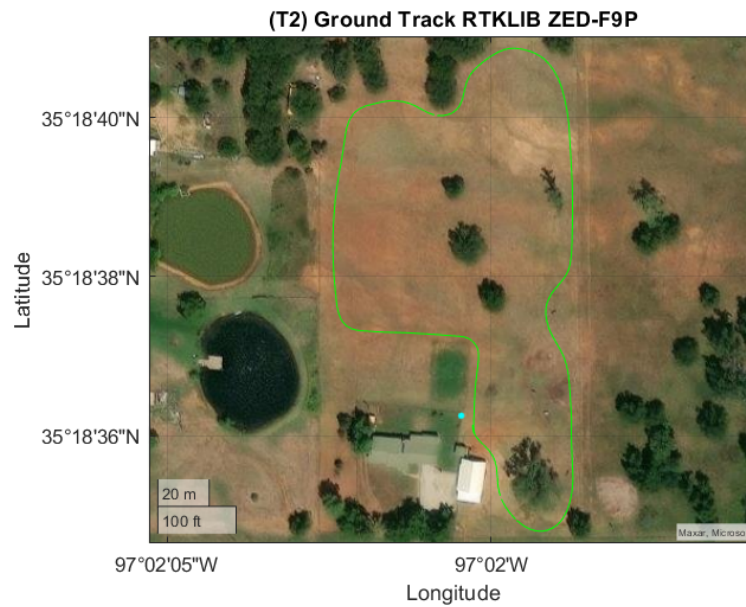


(a) *Test 2, RTKLIB Ashtech, Ground Track*

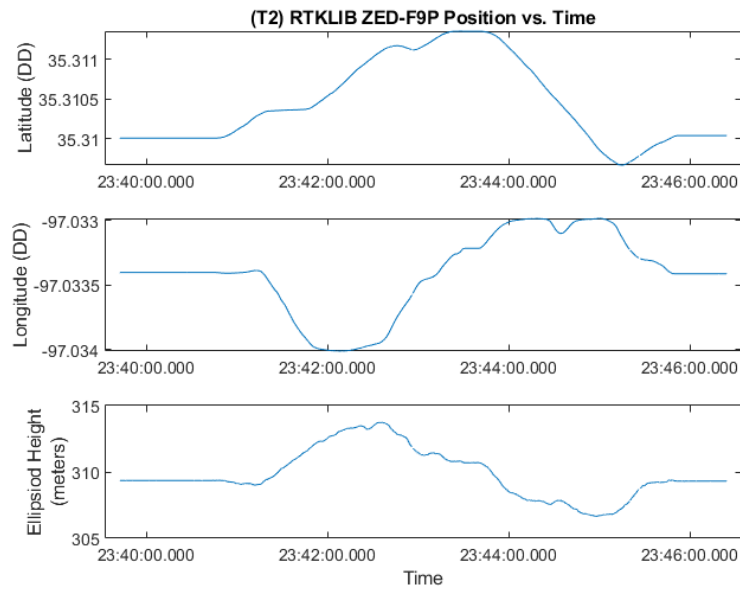


(b) *Test 2, RTKLIB Ashtech, Position*

Figure 4.10. *Test 2 RTKLIB Ashtech: RTKLIB Post Processor with the Ashtech Receiver Ground Track and Position view for Test 2.*



(a) *Test 2, RTKLIB Zed-F9P, Ground Track*

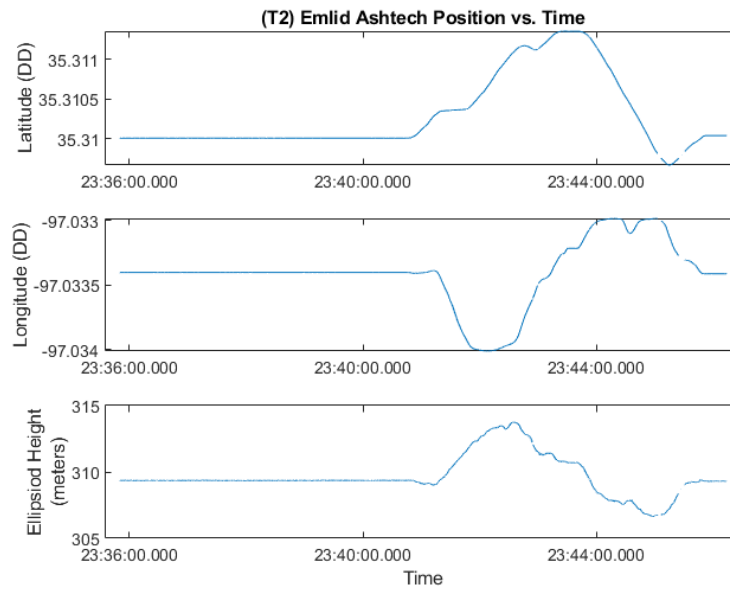


(b) *Test 2, RTKLIB Zed-F9P, Position*

Figure 4.11. *Test 2 RTKLIB ZED-F9P: RTKLIB Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 2.*



(a) *Test 2, Emlid Ashtech, Ground Track*

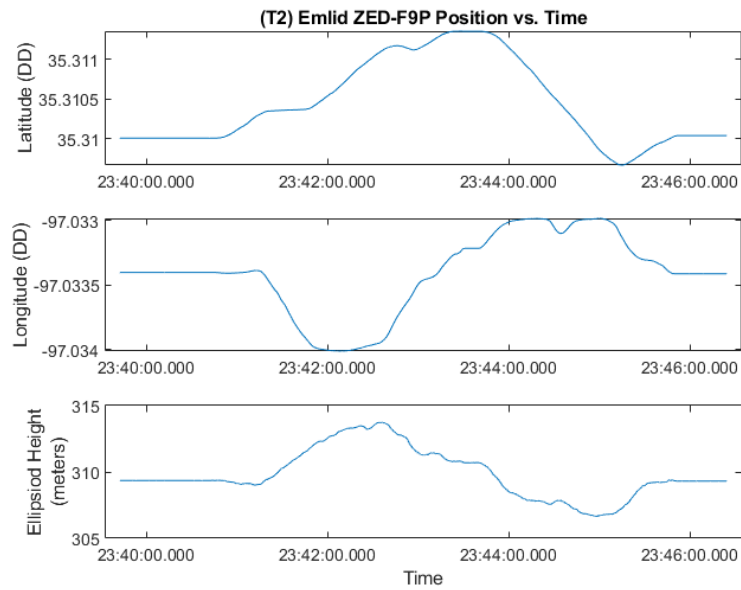


(b) *Test 2, Emlid Ashtech, Position*

Figure 4.12. *Test 2 Emlid Ashtech: Emlid Post Processor with the Ashtech Receiver Ground Track and Position view for Test 2.*

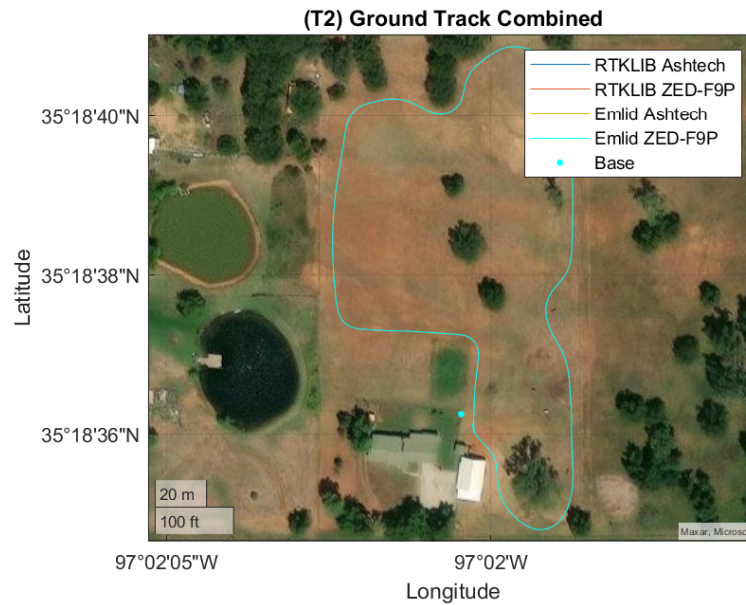


(a) *Test 2, Emlid Zed-F9P, Ground Track*

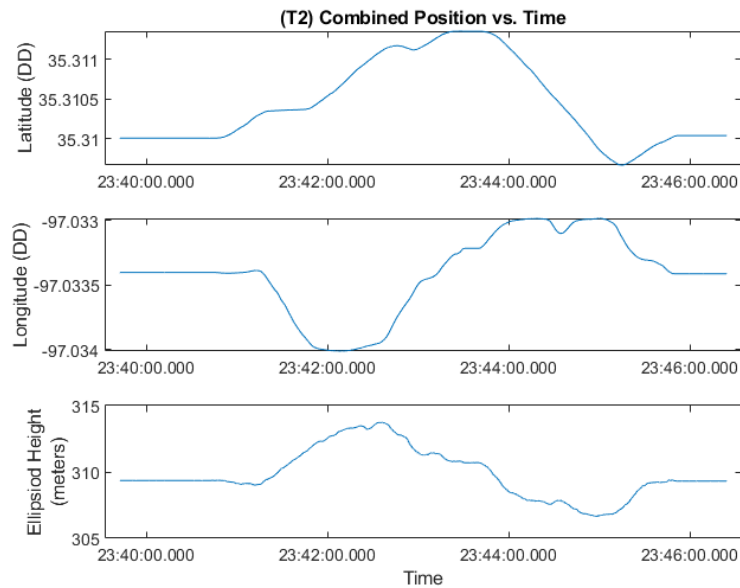


(b) *Test 2, Emlid Zed-F9P, Position*

Figure 4.13. *Test 2 Emlid ZED-F9P: Emlid Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 2.*

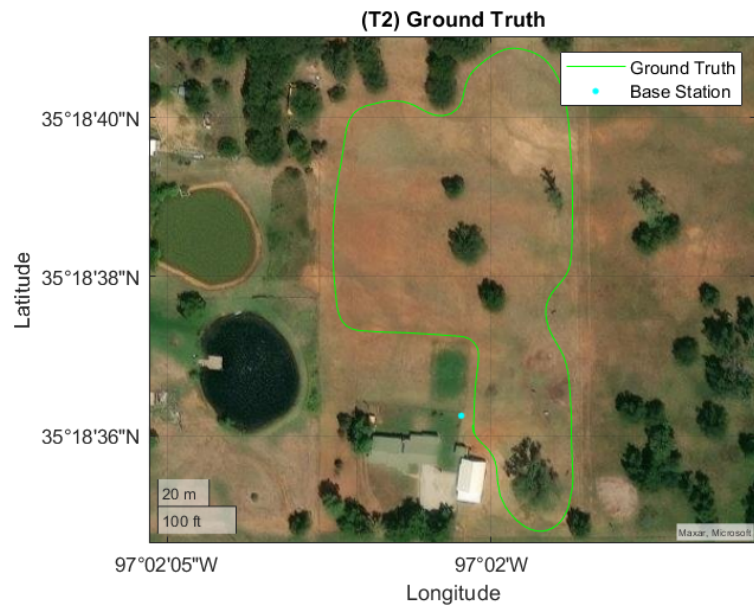


(a) *Test 2, Combined, Ground Track*

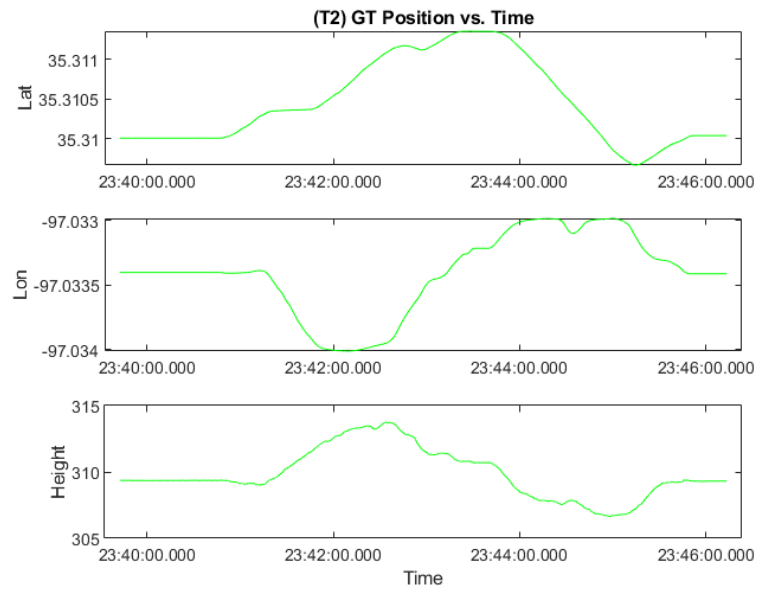


(b) *Test 2, Combined, Position*

Figure 4.14. *Test 2 Combined: Emlid and RTKLIB Post Processors with the ZED-F9P and Ashtech Receivers Ground Track and Position views for Test 2.*

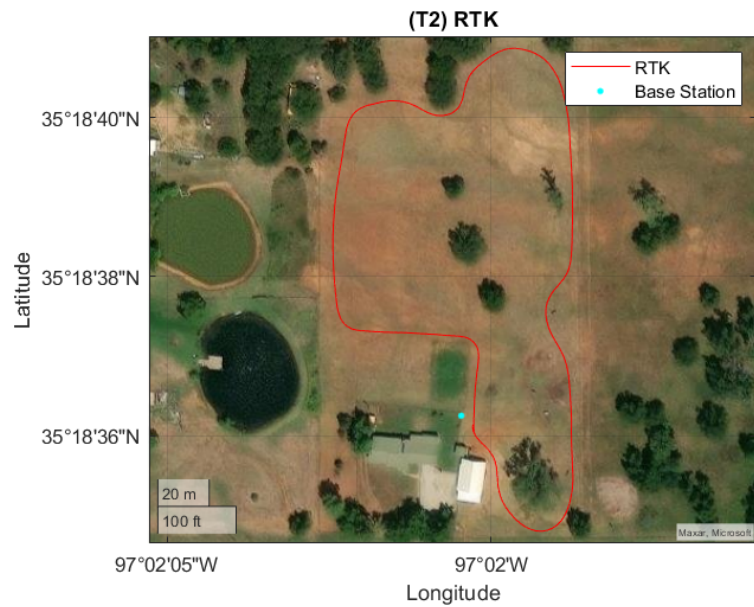


(a) *Test 2, Ground Truth, Ground Track*

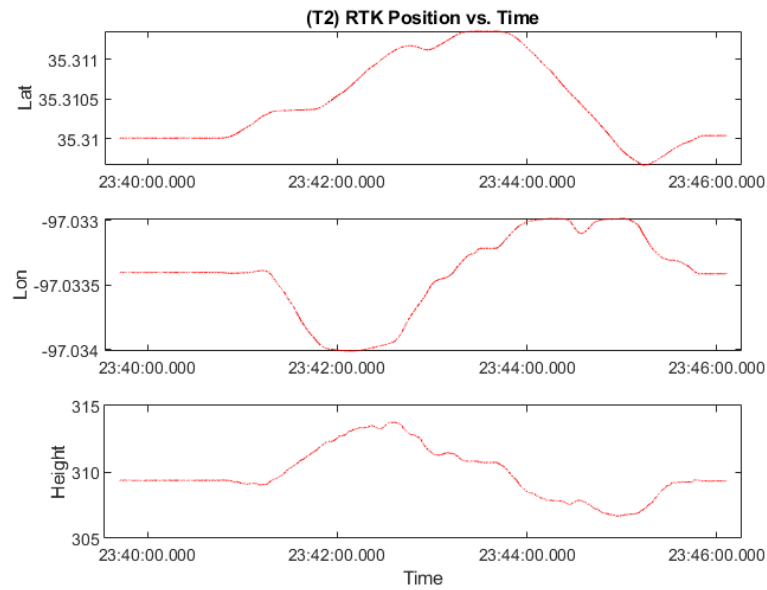


(b) *Test 2, Ground Truth, Position*

Figure 4.15. *Test 2 Ground Truth: Ground Truth Ground Track and Position view for Test 2.*

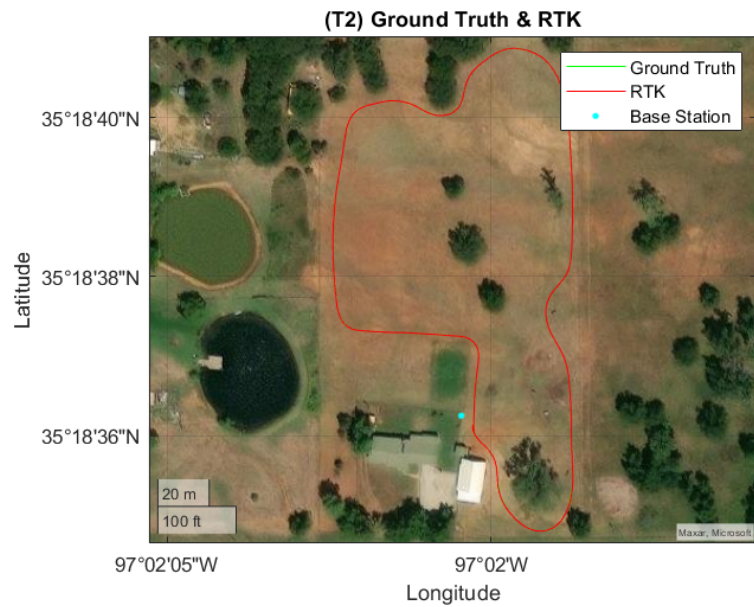


(a) *Test 2, RTK, Ground Track*

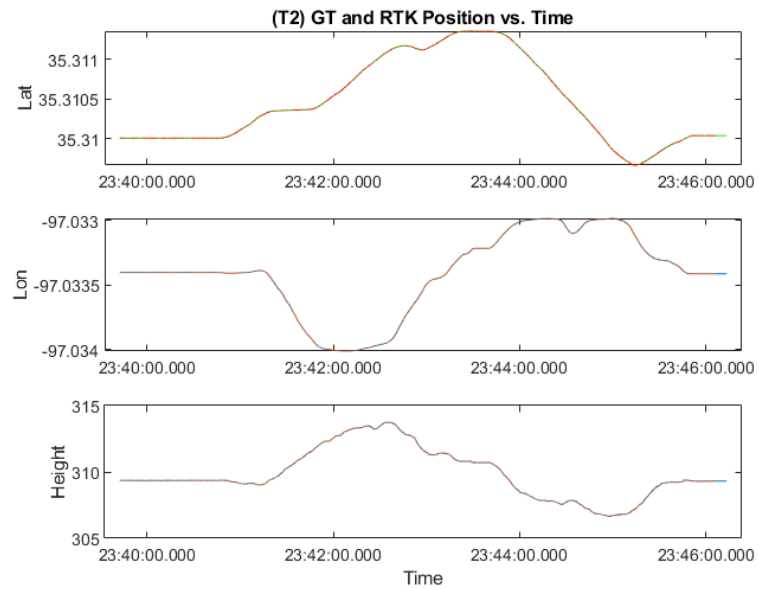


(b) *Test 2, RTK, Position*

Figure 4.16. *Test 2 RTK: RTK Ground Track and Position view for Test 2.*



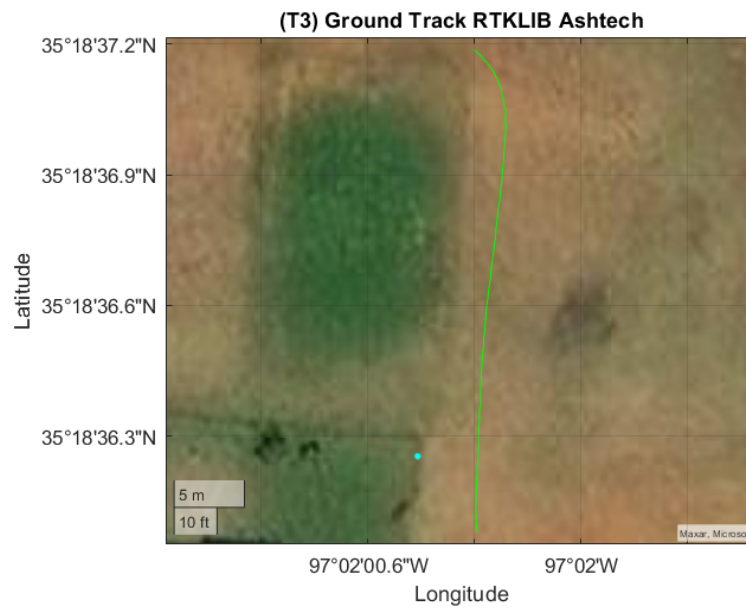
(a) *Test 2, GT and RTK, Ground Track*



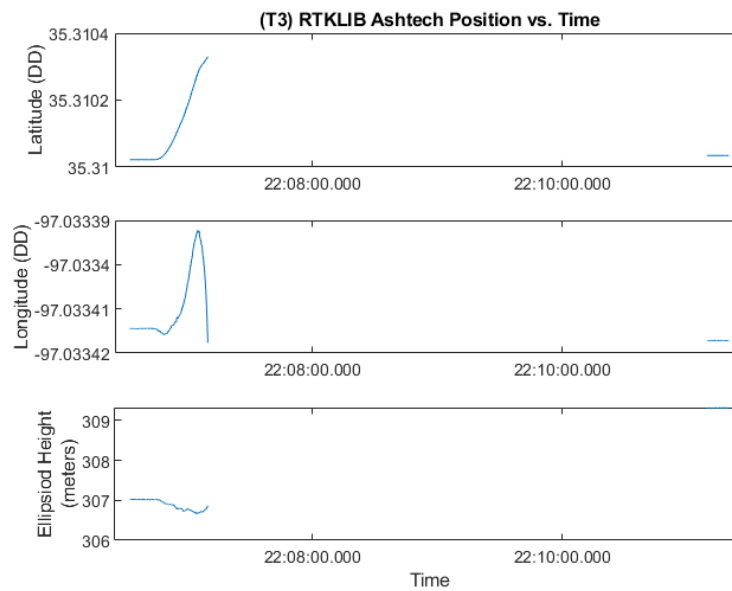
(b) *Test 2, GT and RTK, Position*

Figure 4.17. *Test 2 GT and RTK: Ground Truth and RTK Ground Track and Position view for Test 2.*

4.1.1.3 Test 3

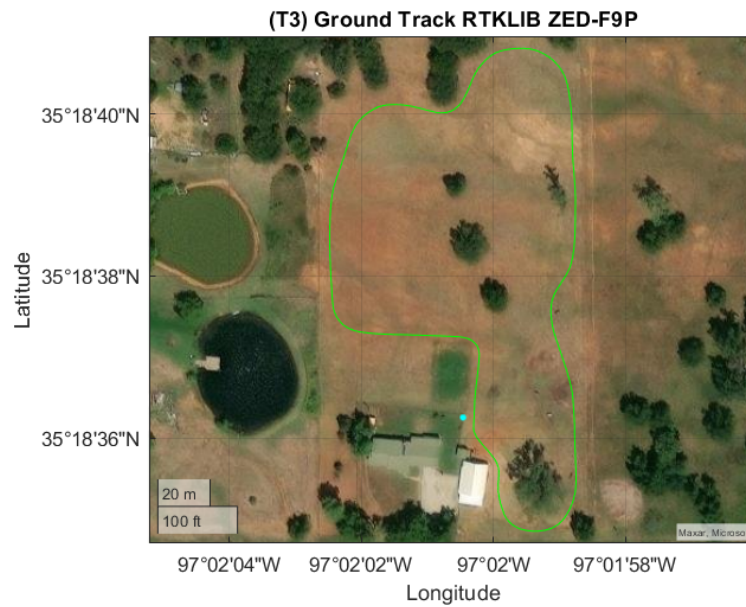


(a) *Test 3, RTKLIB Ashtech, Ground Track*

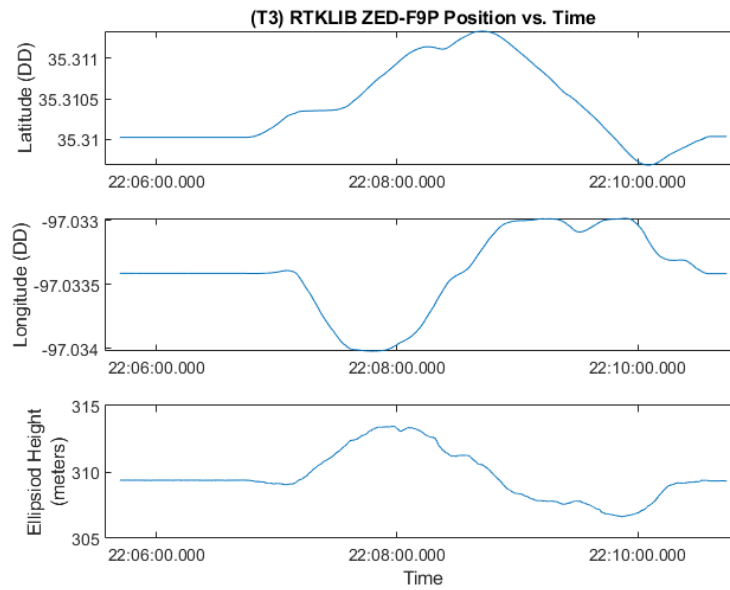


(b) *Test 3, RTKLIB Ashtech, Position*

Figure 4.18. *Test 3 RTKLIB Ashtech: RTKLIB Post Processor with the Ashtech Receiver Ground Track and Position view for Test 3.*



(a) *Test 3, RTKLIB Zed-F9P, Ground Track*

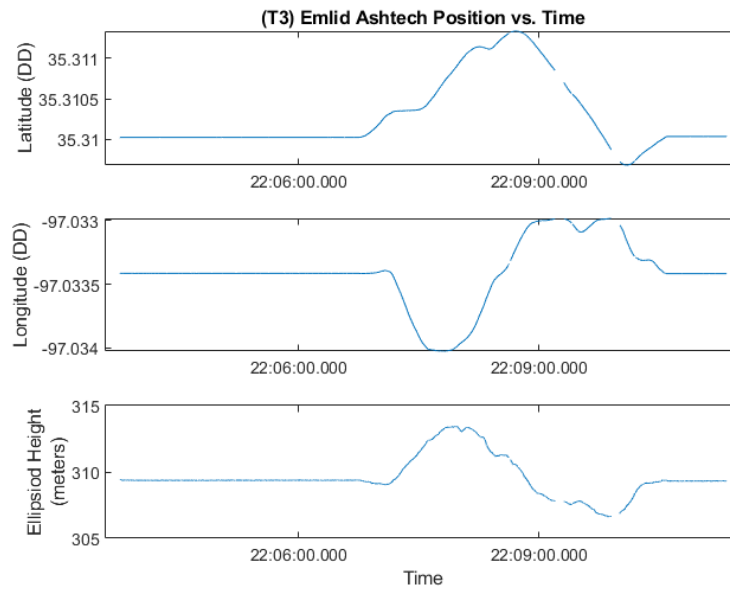


(b) *Test 3, RTKLIB Zed-F9P, Position*

Figure 4.19. *Test 3 RTKLIB ZED-F9P: RTKLIB Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 3.*



(a) *Test 3, Emlid Ashtech, Ground Track*

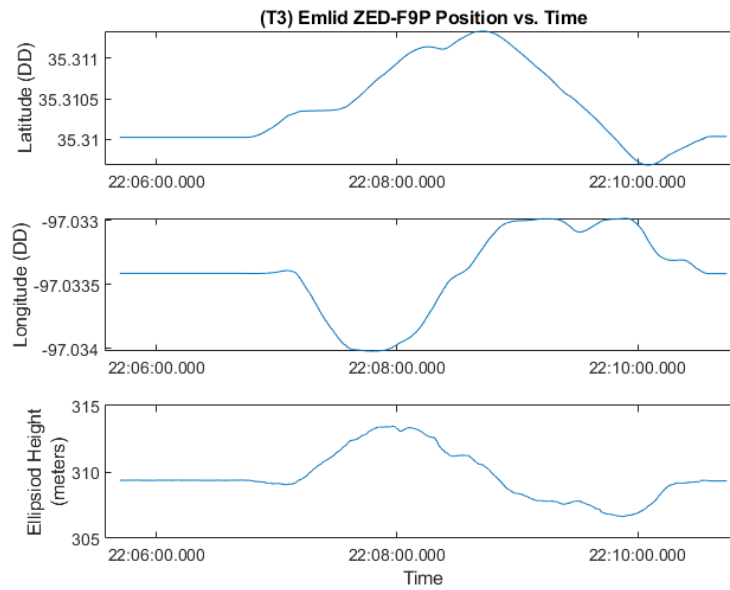


(b) *Test 3, Emlid Ashtech, Position*

Figure 4.20. *Test 3 Emlid Ashtech: Emlid Post Processor with the Ashtech Receiver Ground Track and Position view for Test 3.*



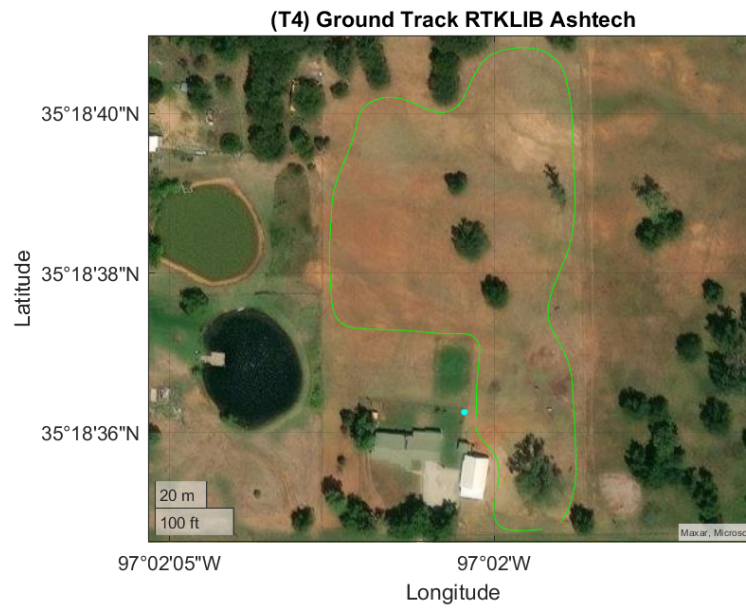
(a) *Test 3, Emlid Zed-F9P, Ground Track*



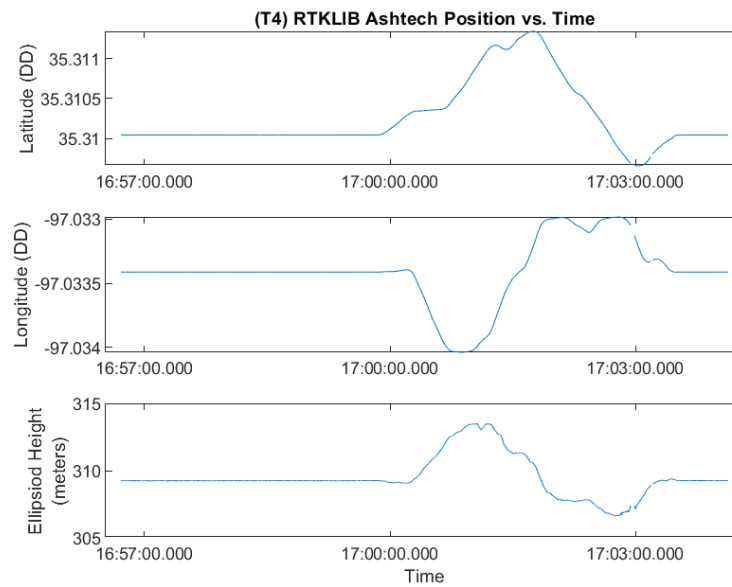
(b) *Test 3, Emlid Zed-F9P, Position*

Figure 4.21. *Test 3 Emlid ZED-F9P: Emlid Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 3.*

4.1.1.4 Test 4



(a) Test 4, RTKLIB Ashtech, Ground Track

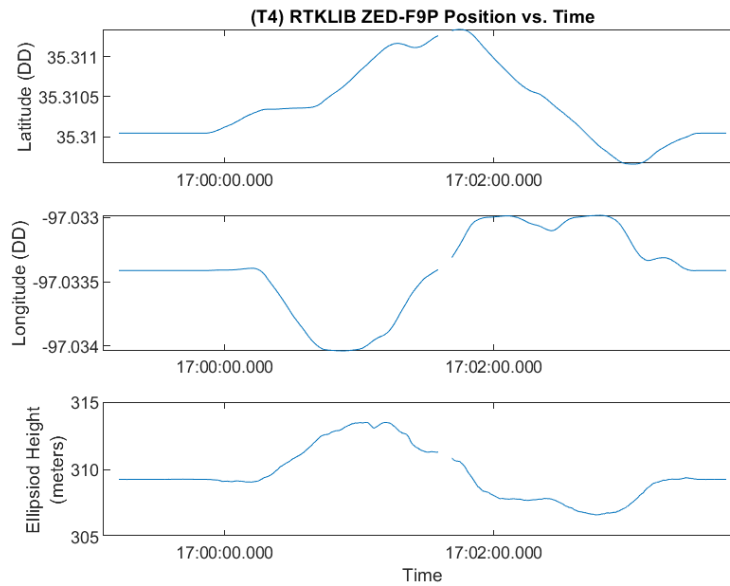


(b) Test 4, RTKLIB Ashtech, Position

Figure 4.22. Test 4 RTKLIB Ashtech: RTKLIB Post Processor with the Ashtech Receiver Ground Track and Position view for Test 4.



(a) *Test 4, RTKLIB Zed-F9P, Ground Track*

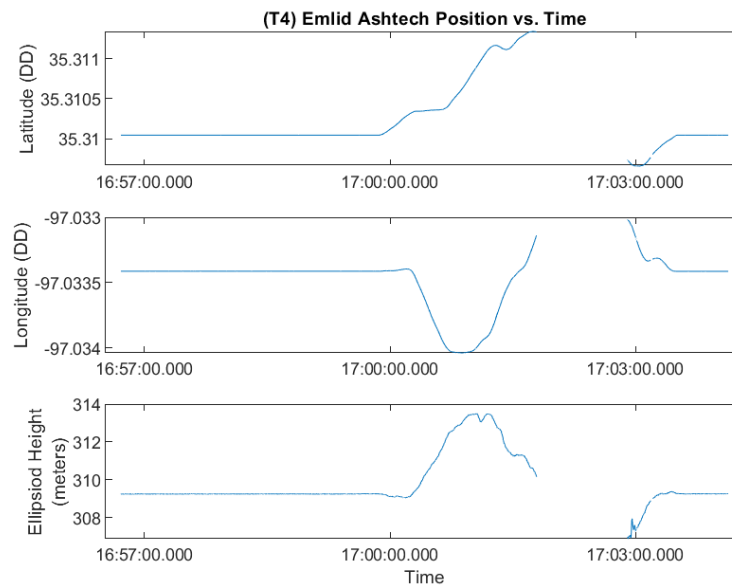


(b) *Test 4, RTKLIB Zed-F9P, Position*

Figure 4.23. *Test 4 RTKLIB ZED-F9P: RTKLIB Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 4.*



(a) *Test 4, Emlid Ashtech, Ground Track*

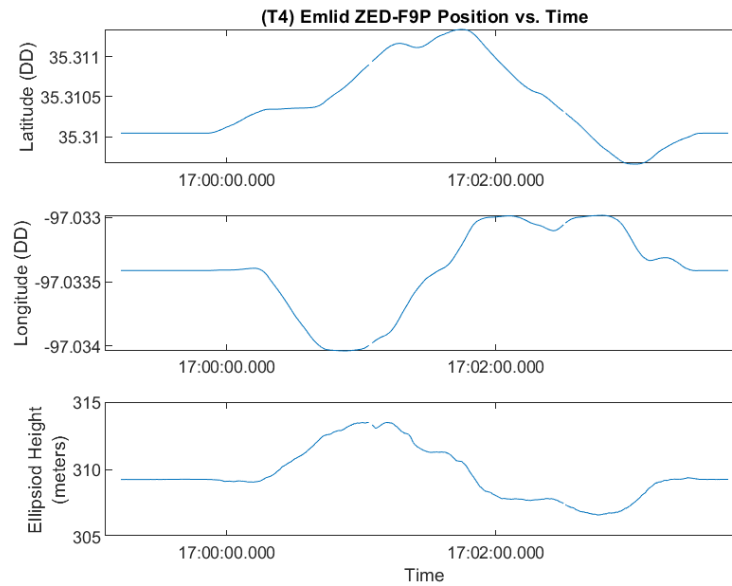


(b) *Test 4, Emlid Ashtech, Position*

Figure 4.24. *Test 4 Emlid Ashtech: Emlid Post Processor with the Ashtech Receiver Ground Track and Position view for Test 4.*

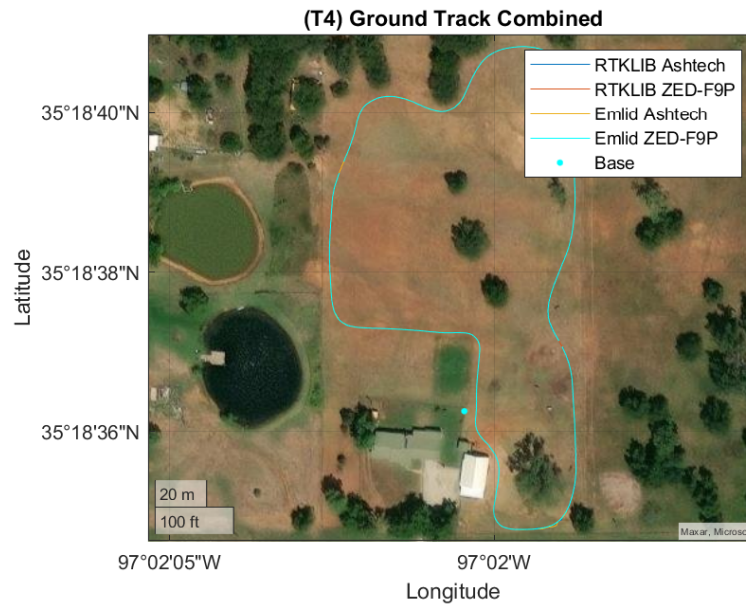


(a) *Test 4, Emlid Zed-F9P, Ground Track*

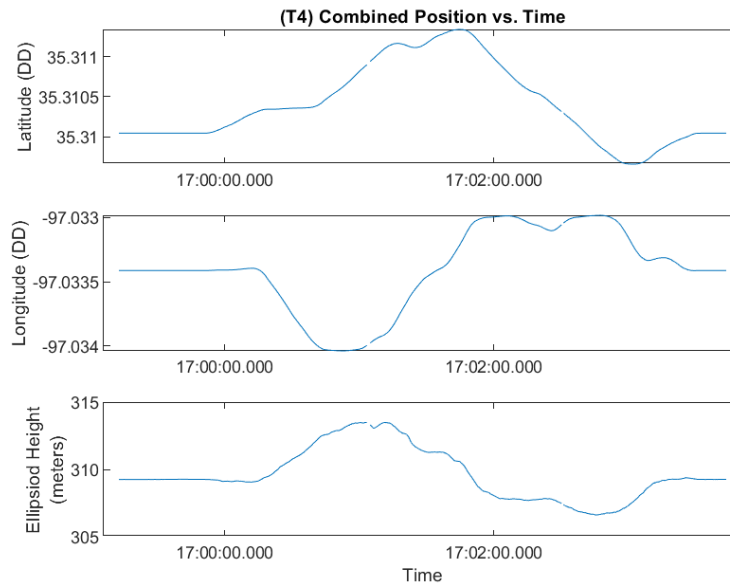


(b) *Test 4, Emlid Zed-F9P, Position*

Figure 4.25. *Test 4 Emlid ZED-F9P: Emlid Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 4.*

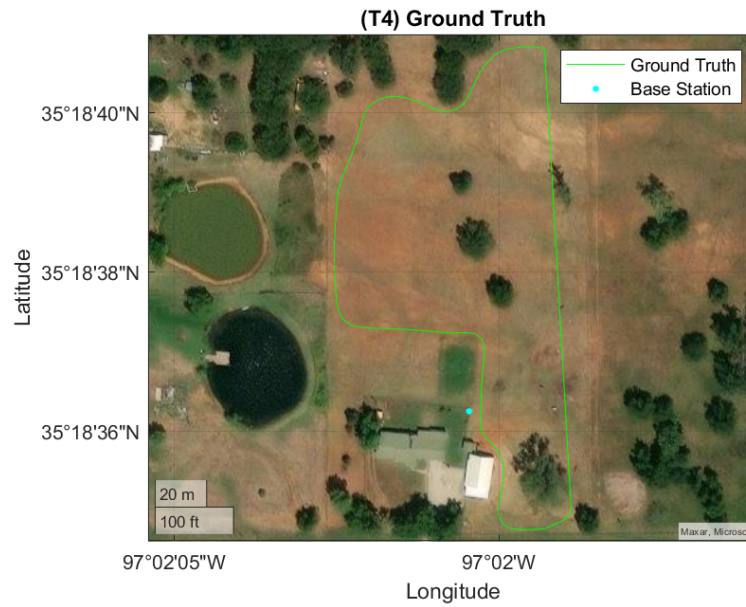


(a) *Test 4, Combined, Ground Track*

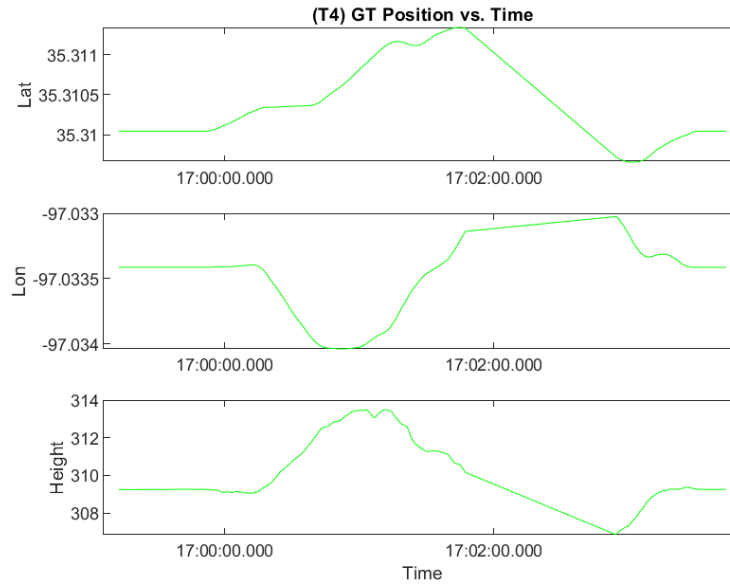


(b) *Test 4, Combined, Position*

Figure 4.26. *Test 4 Combined: Emlid and RTKLIB Post Processors with the ZED-F9P and Ashtech Receivers Ground Track and Position views for Test 4.*

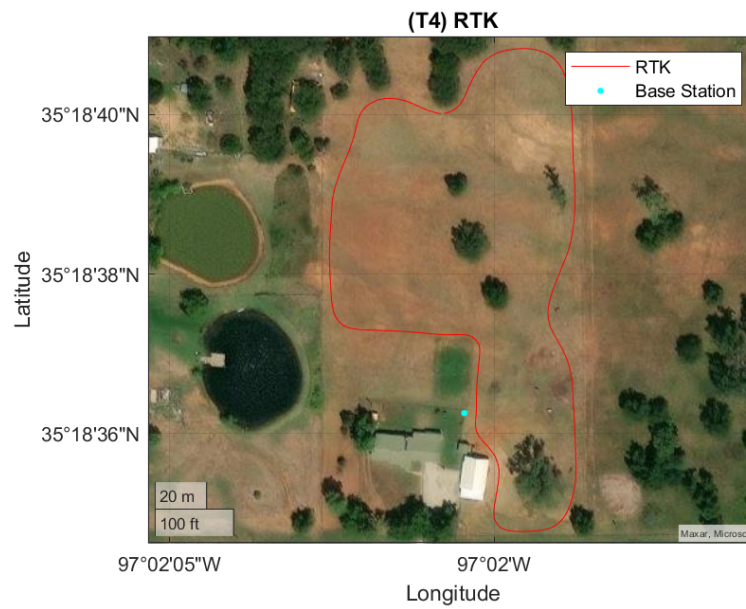


(a) *Test 4, Ground Truth, Ground Track*

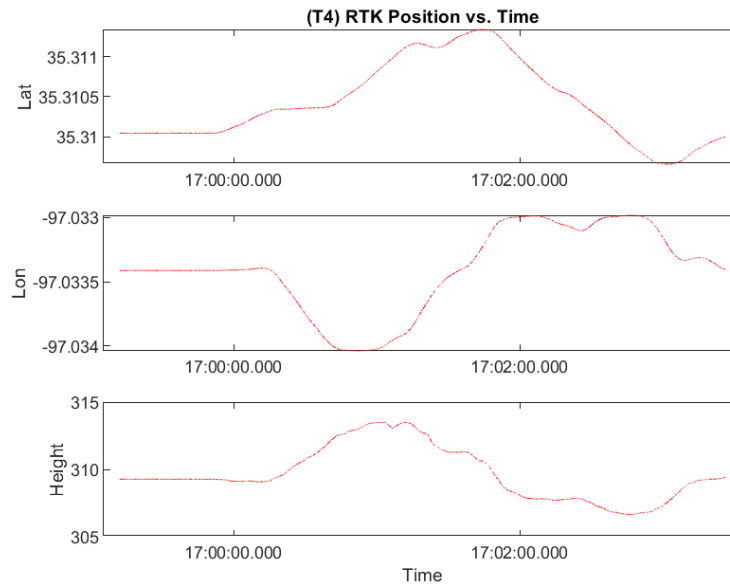


(b) *Test 4, Ground Truth, Position*

Figure 4.27. Test 4 Ground Truth: Ground Truth Ground Track and Position view for Test 4.

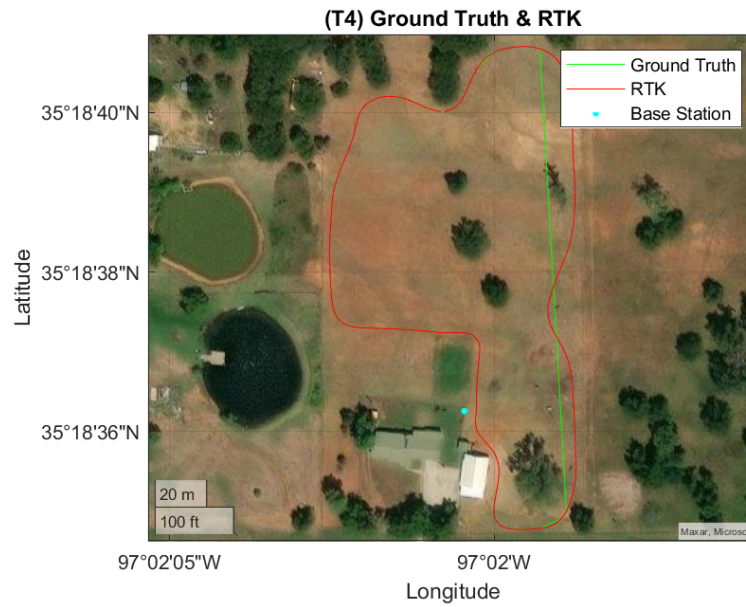


(a) *Test 4, RTK, Ground Track*

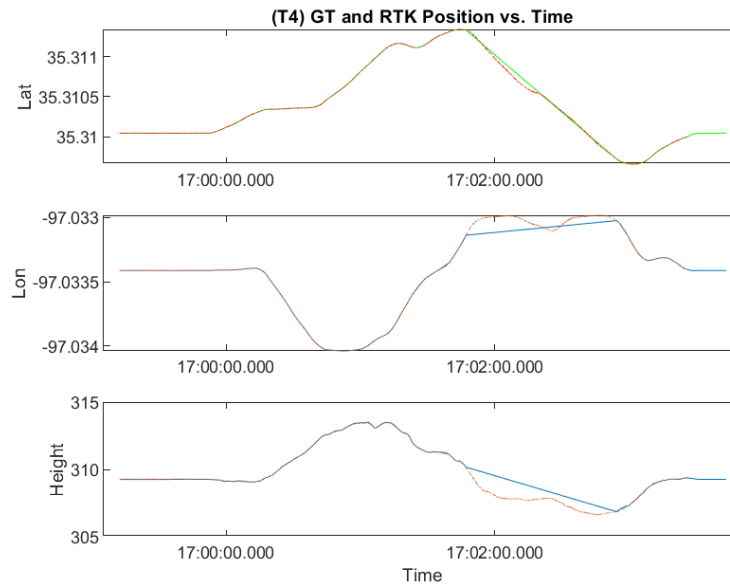


(b) *Test 4, RTK, Position*

Figure 4.28. *Test 4 RTK: RTK Ground Track and Position view for Test 4.*



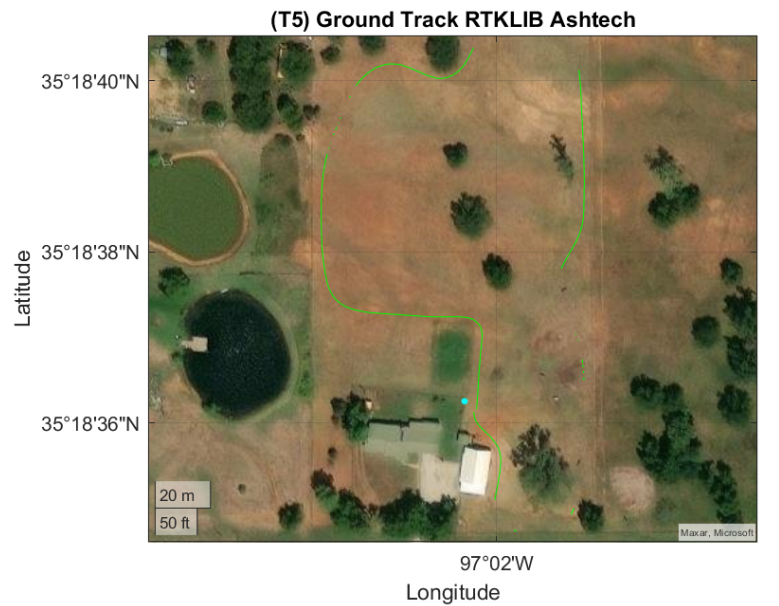
(a) *Test 4, GT and RTK, Ground Track*



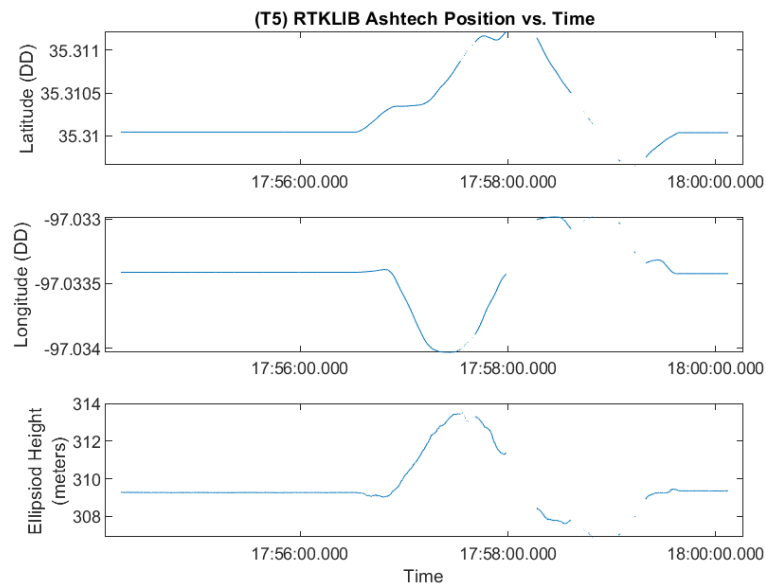
(b) *Test 4, GT and RTK, Position*

Figure 4.29. *Test 4 GT and RTK: Ground Truth and RTK Ground Track and Position view for Test 4.*

4.1.1.5 Test 5



(a) *Test 5, RTKLIB Ashtech, Ground Track*

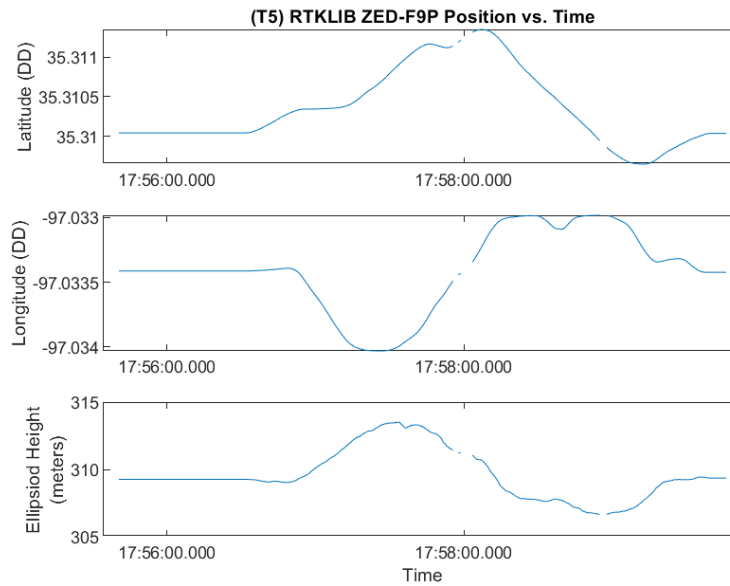


(b) *Test 5, RTKLIB Ashtech, Position*

Figure 4.30. *Test 5 RTKLIB Ashtech: RTKLIB Post Processor with the Ashtech Receiver Ground Track and Position view for Test 5.*



(a) *Test 5, RTKLIB Zed-F9P, Ground Track*

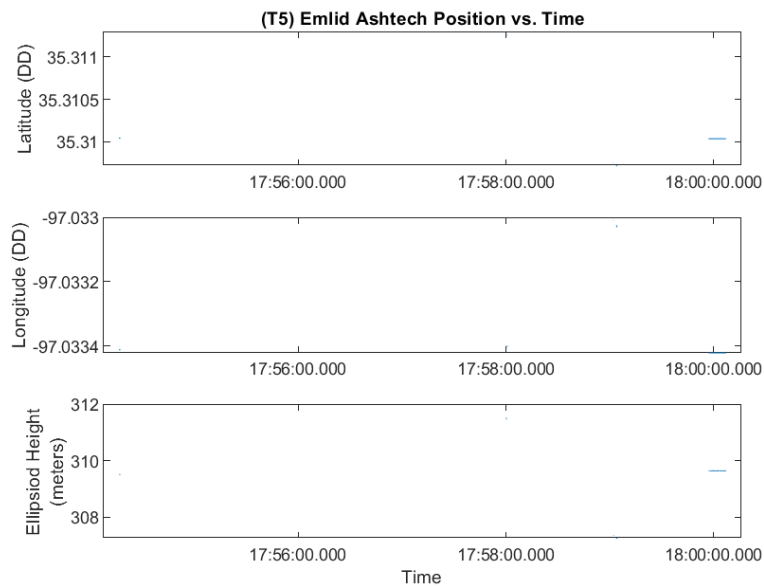


(b) *Test 5, RTKLIB Zed-F9P, Position*

Figure 4.31. *Test 5 RTKLIB ZED-F9P: RTKLIB Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 5.*



(a) *Test 5, Emlid Ashtech, Ground Track*

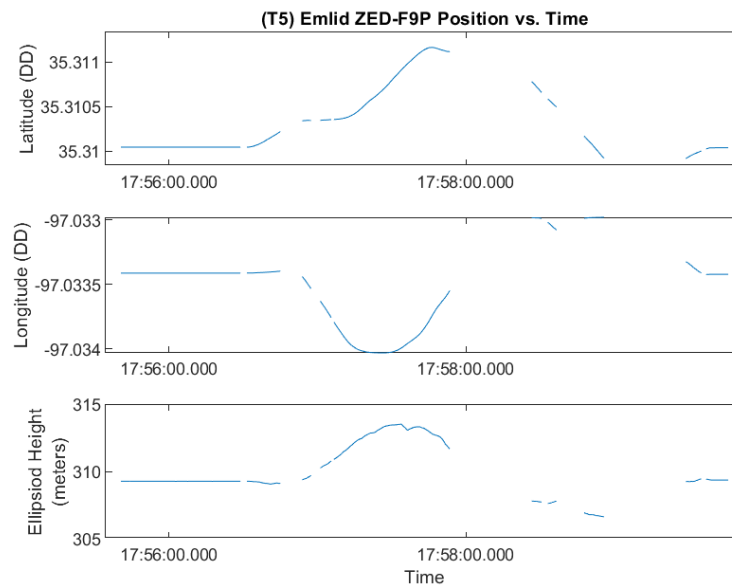


(b) *Test 5, Emlid Ashtech, Position*

Figure 4.32. *Test 5 Emlid Ashtech: Emlid Post Processor with the Ashtech Receiver Ground Track and Position view for Test 5.*



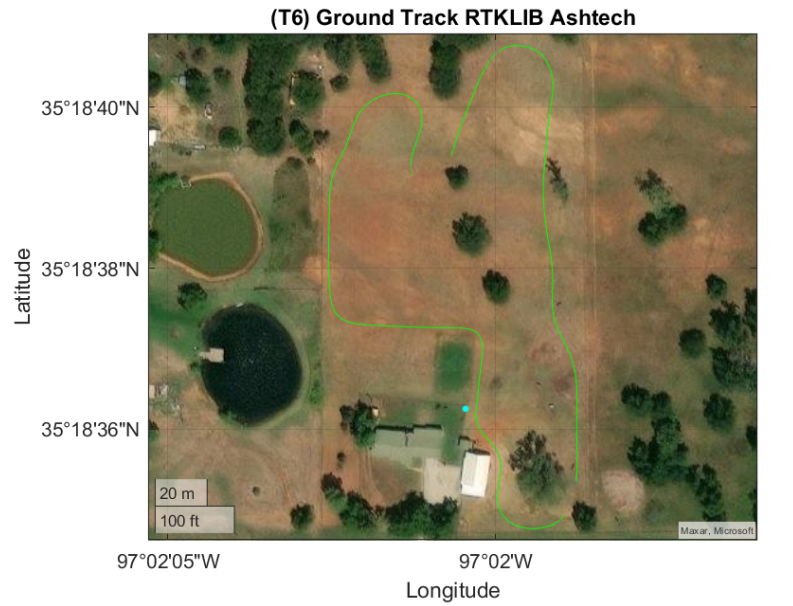
(a) *Test 5, Emlid Zed-F9P, Ground Track*



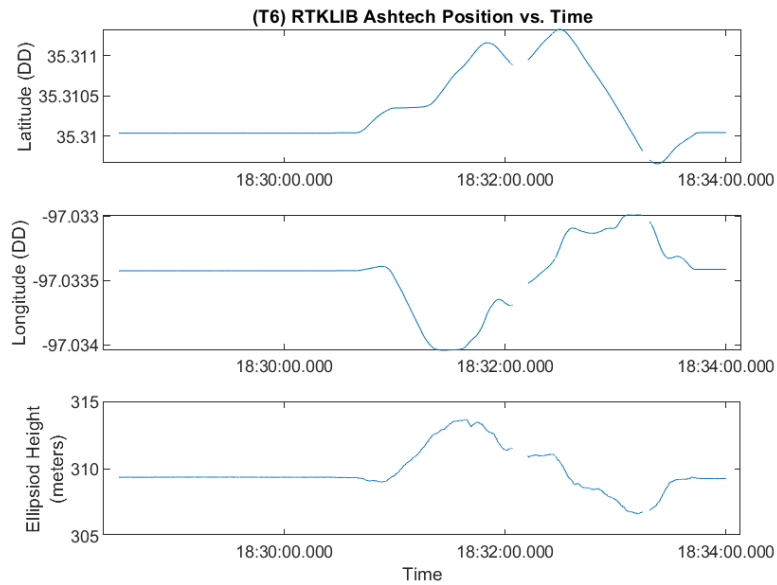
(b) *Test 5, Emlid Zed-F9P, Position*

Figure 4.33. *Test 5 Emlid ZED-F9P: Emlid Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 5.*

4.1.1.6 Test 6

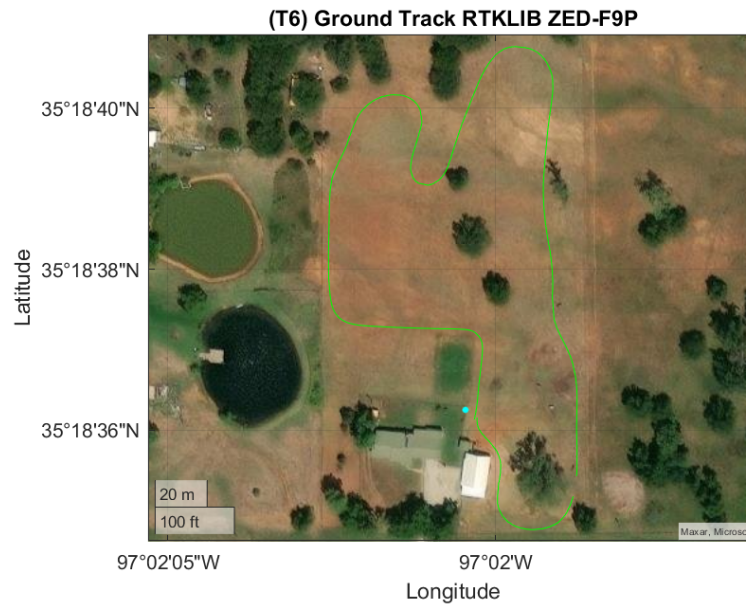


(a) Test 6, RTKLIB Ashtech, Ground Track

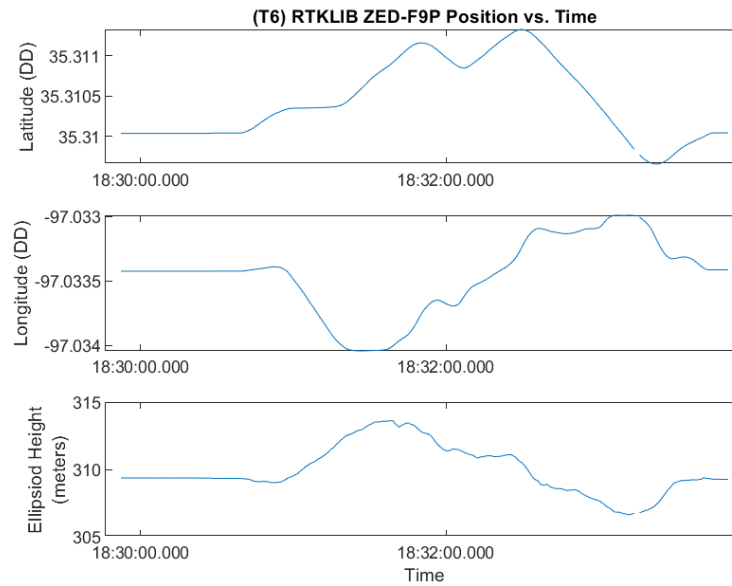


(b) Test 6, RTKLIB Ashtech, Position

Figure 4.34. Test 6 RTKLIB Ashtech: RTKLIB Post Processor with the Ashtech Receiver Ground Track and Position view for Test 6.



(a) *Test 6, RTKLIB Zed-F9P, Ground Track*

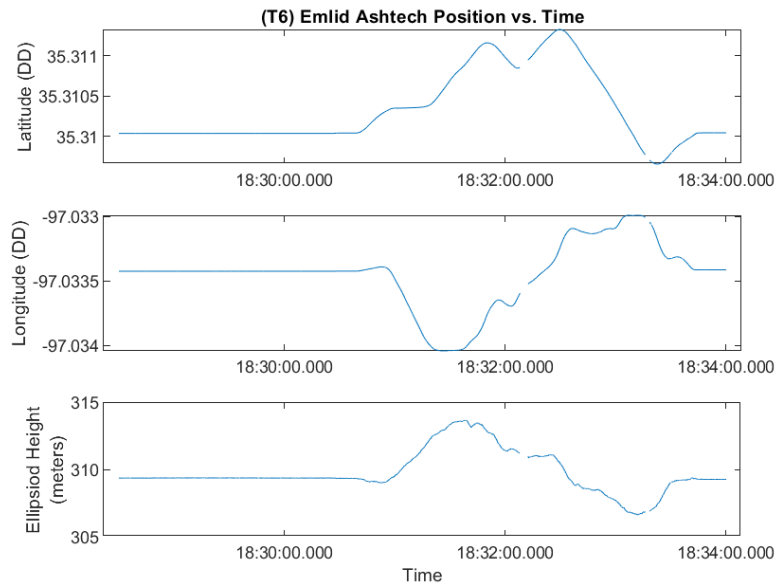


(b) *Test 6, RTKLIB Zed-F9P, Position*

Figure 4.35. *Test 6 RTKLIB ZED-F9P: RTKLIB Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 6.*

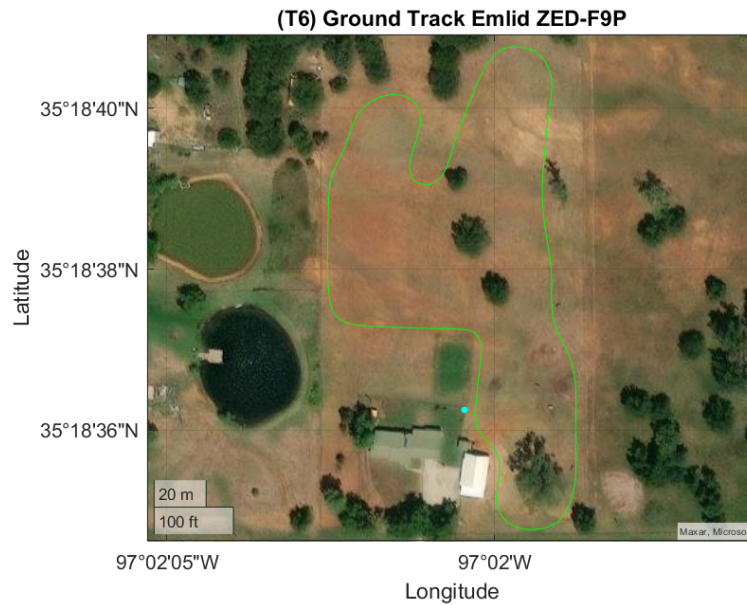


(a) *Test 6, Emlid Ashtech, Ground Track*

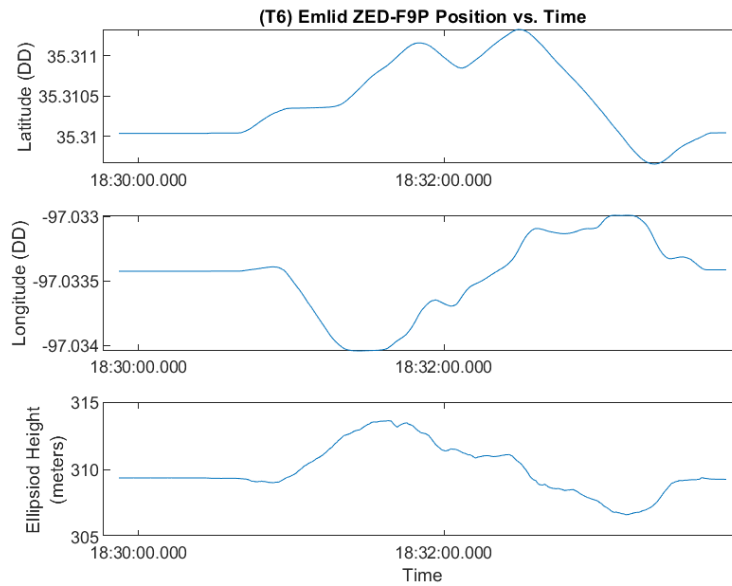


(b) *Test 6, Emlid Ashtech, Position*

Figure 4.36. *Test 6 Emlid Ashtech: Emlid Post Processor with the Ashtech Receiver Ground Track and Position view for Test 6.*

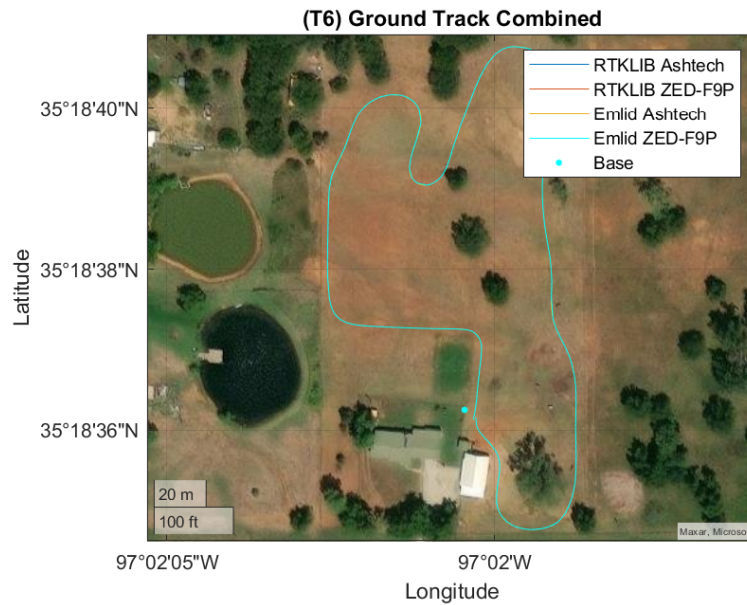


(a) *Test 6, Emlid Zed-F9P, Ground Track*

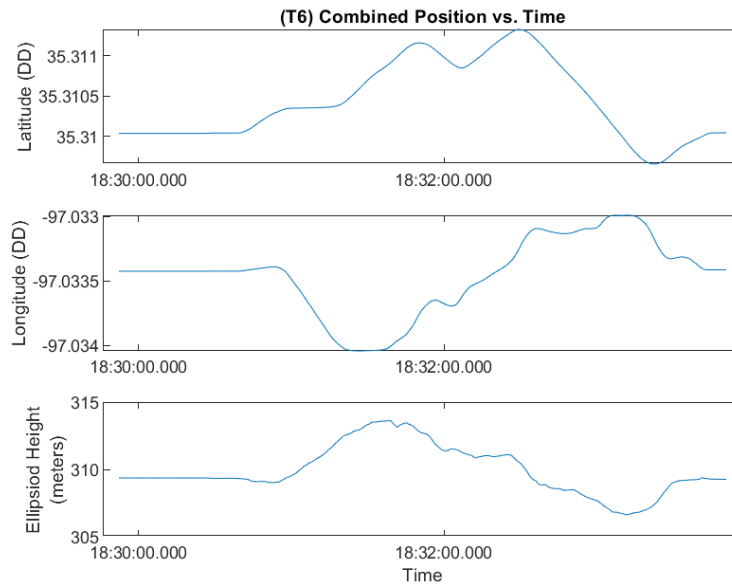


(b) *Test 6, Emlid Zed-F9P, Position*

Figure 4.37. *Test 6 Emlid ZED-F9P: Emlid Post Processor with the ZED-F9P Receiver Ground Track and Position view for Test 6.*

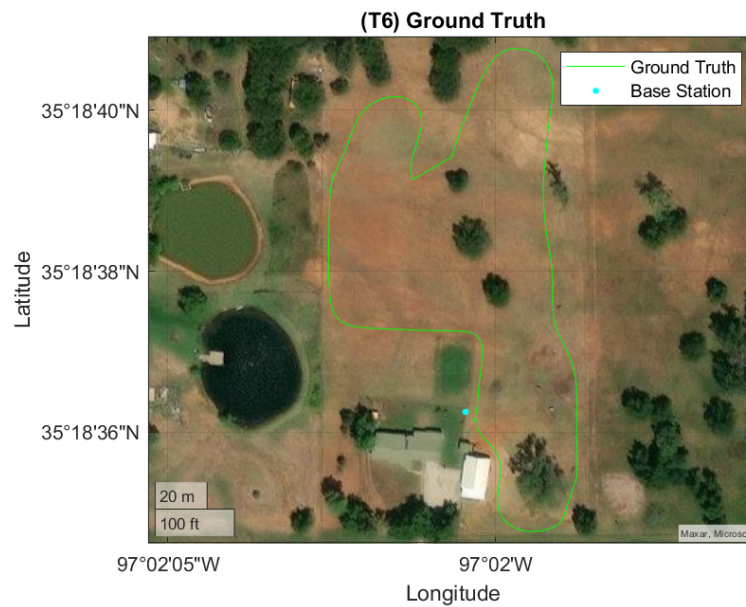


(a) *Test 6, Combined, Ground Track*

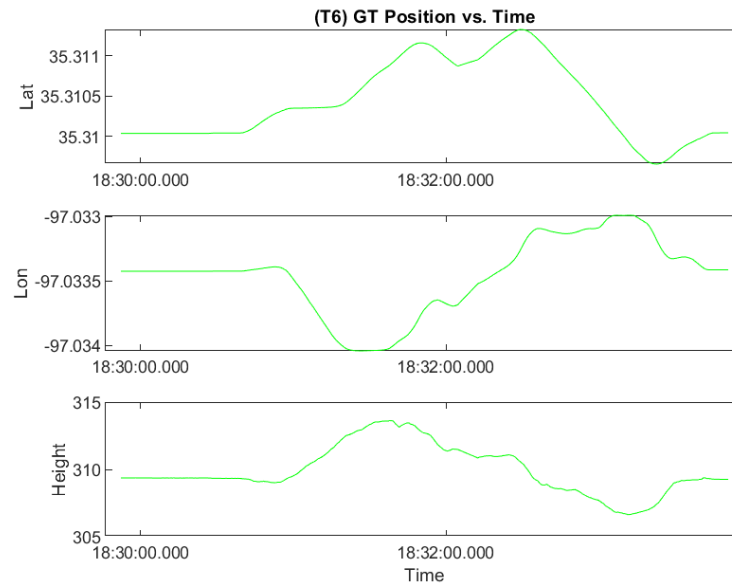


(b) *Test 6, Combined, Position*

Figure 4.38. *Test 6 Combined: Emlid and RTKLIB Post Processors with the ZED-F9P and Ashtech Receivers Ground Track and Position views for Test 6.*

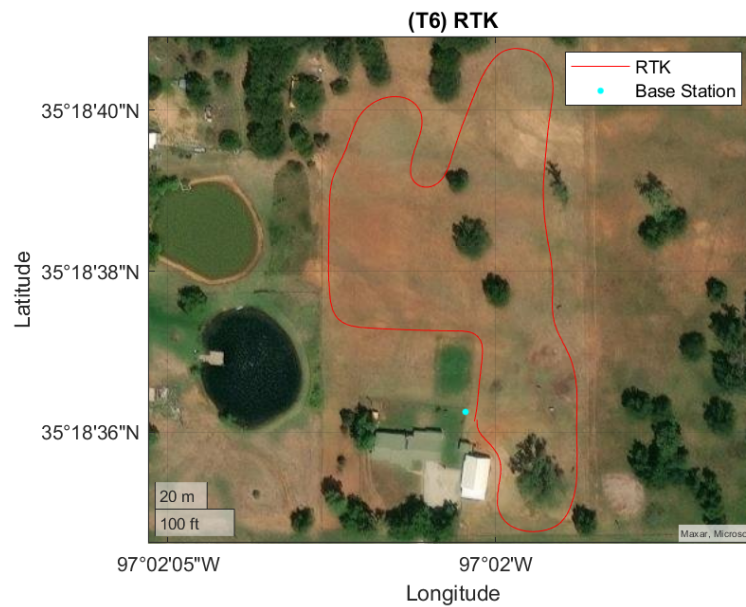


(a) *Test 6, Ground Truth, Ground Track*

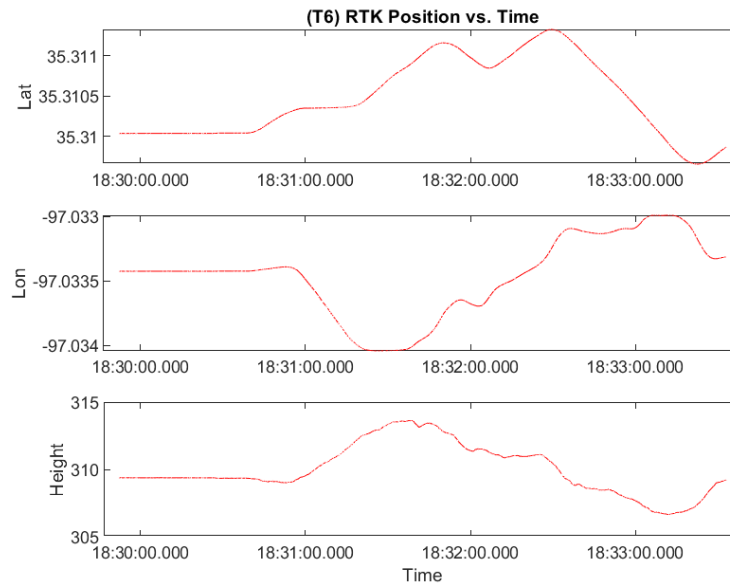


(b) *Test 6, Ground Truth, Position*

Figure 4.39. Test 6 Ground Truth: *Ground Truth Ground Track and Position view for Test 6.*

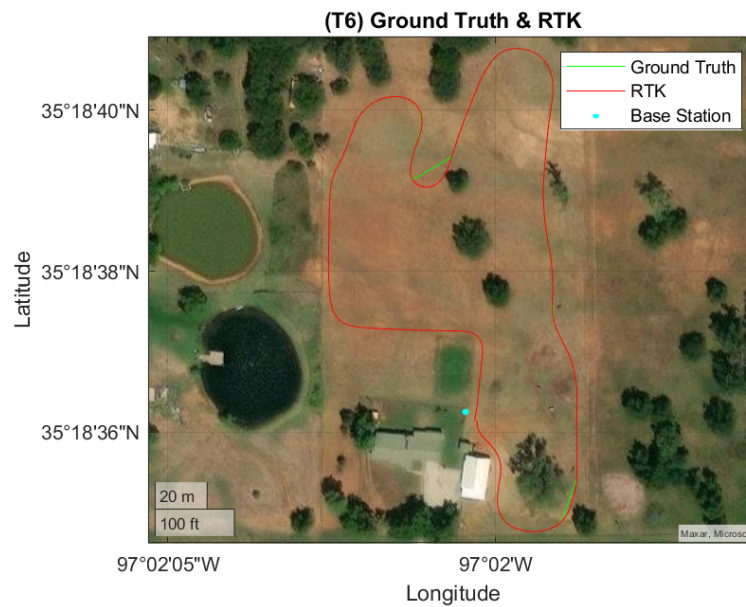


(a) *Test 6, RTK, Ground Track*

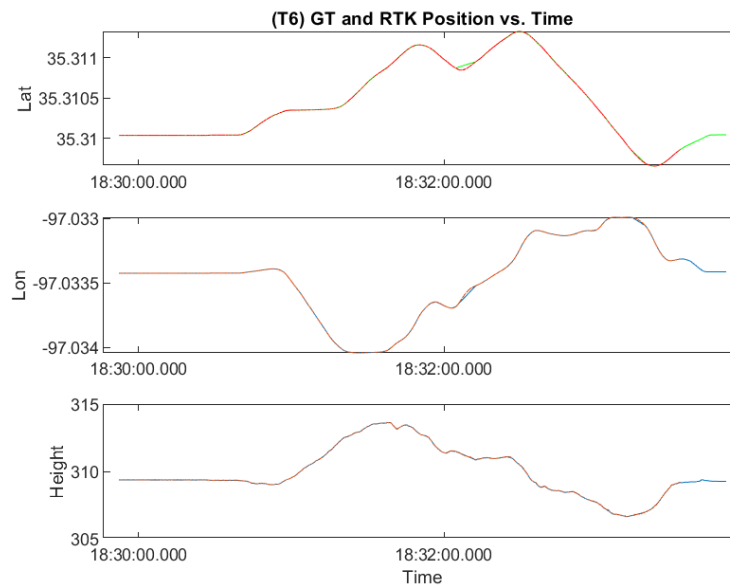


(b) *Test 6, RTK, Position*

Figure 4.40. *Test 6 RTK: RTK Ground Track and Position view for Test 6.*



(a) *Test 6, GT and RTK, Ground Track*



(b) *Test 6, GT and RTK, Position*

Figure 4.41. *Test 6 GT and RTK: Ground Truth and RTK Ground Track and Position view for Test 6.*

4.1.2 Ground Truth Error Summary

Table 4.1 shows the average standard deviation for each of the four datasets in each test with respect to Ground Truth. This gives an idea of the amount of error in each test with respect to Ground Truth at each data point. While the dynamic Ground Truth doesn't quite meet the static 3 mm accuracy requirements, they hover right around double, which is very good for a dynamic application. This could be further improved by getting more 24-hour PPP data points on the base station for a more accurate survey in position or getting the base station professionally surveyed with higher-grade equipment. The error in the base station propagates throughout the system, so making it more accurate should increase the accuracy of the Ground Truth system.

Test	σ_{xy} (m)	σ_h (m)	σ_{3D} (m)
Test 1	.0029	.0044	.0055
Test 2	.0032	.0062	.0072
Test 3	-	-	-
Test 4	.0047	.0062	.0081
Test 5	-	-	-
Test 6	.003	.004	.0052

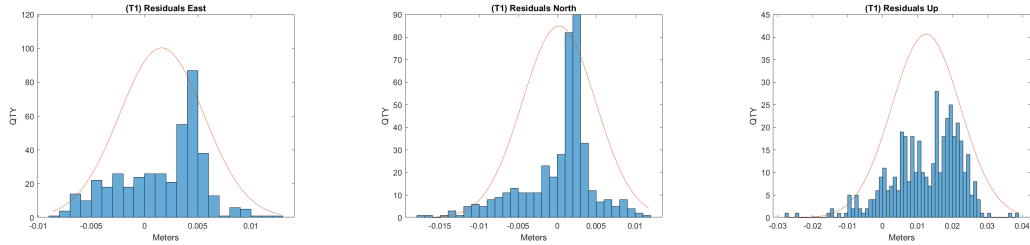
Table 4.1. *Average Standard Deviation from Ground Truth: Average Standard Deviation of each point in each dataset from each test, with respect to Ground Truth.*

4.1.3 Modified ISO 17123-8 Results Summary

4.1.3.1 Individual Test Deviations

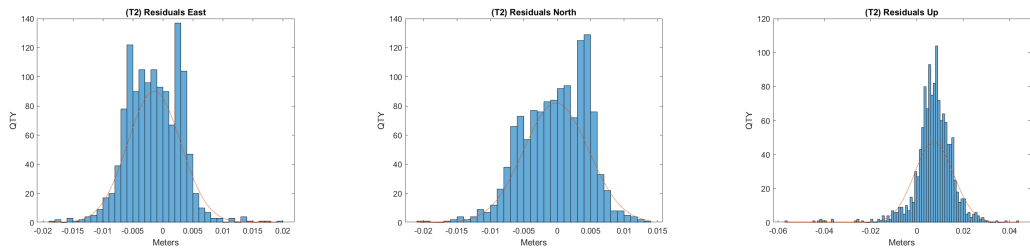
Figures 4.42-4.45 show distributions of the residuals from Tests 1, 2, 4, and 5. The distributions are Gaussian, so the operations outlined in the ISO 17123-8 standard are appropriate. It can also be seen in these figures that some of the

data is not distributed about zero, indicating possible sources of systematic error.



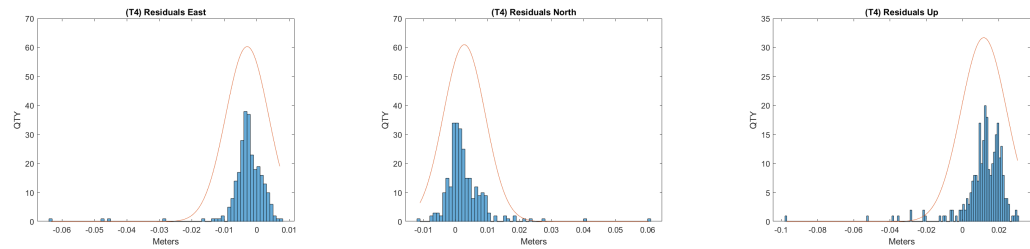
(a) Test 1, Distribution of residuals to the East (b) Test 1, Distribution of residuals to the North (c) Test 1, Distribution of residuals Upwards

Figure 4.42. Test 1 Distributions of Residuals: Displays the residuals for each direction in Test 1 to get a sense of their distribution.



(a) Test 2, Distribution of residuals to the East (b) Test 2, Distribution of residuals to the North (c) Test 2, Distribution of residuals Upwards

Figure 4.43. Test 2 Distributions of Residuals: Displays the residuals for each direction in Test 2 to get a sense of their distribution.



(a) Test 4, Distribution of residuals to the East (b) Test 4, Distribution of residuals to the North (c) Test 4, Distribution of residuals Upwards

Figure 4.44. Test 4 Distributions of Residuals: Displays the residuals for each direction in Test 4 to get a sense of their distribution.

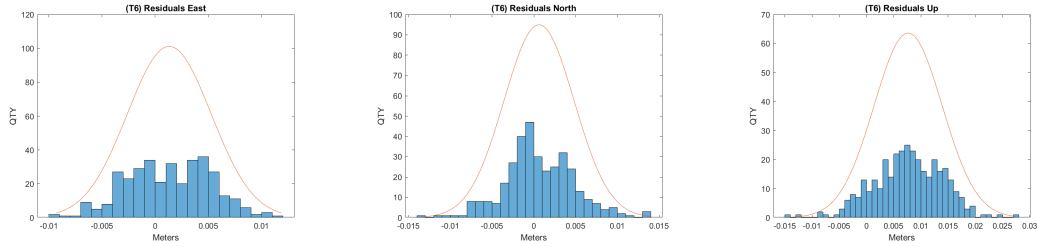


Figure 4.45. Test 6 Distributions of Residuals: Displays the residuals for each direction in Test 6 to get a sense of their distribution.

Test	σ_e (m)	σ_n (m)	σ_{xy} (m)	σ_u (m)
Test 1	.0043	.0047	.0064	.0158
Test 2	.0047	.0049	.0067	.0112
Test 3	-	-	-	-
Test 4	.0073	.0071	.010	.0172
Test 5	-	-	-	-
Test 6	.0042	.0042	.0059	.0099

Table 4.2. Individual Series Standard Deviations: Standard Deviations in each direction for each series of data.

4.1.3.2 Cumulative Deviations

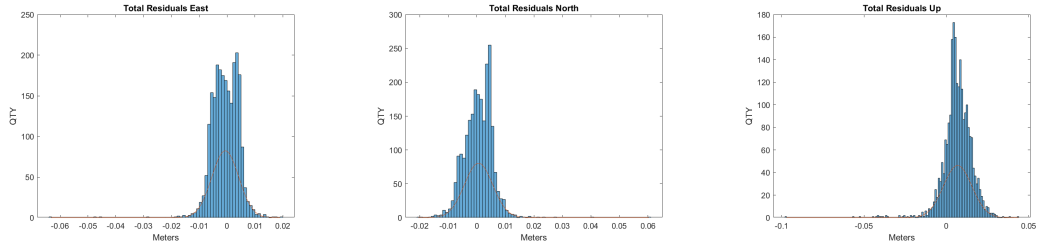


Figure 4.46. Cumulative distributions of Residuals: Displays the residuals for each direction in all Tests to get a sense of their distribution.

Cumulative Standard Deviations					
	$\sigma_e(m)$	$\sigma_n(m)$	$\sigma_{xy}(m)$	CEP	$\sigma_u(m)$
Rover	.0049	.005	.007	.0058	.011

Table 4.3. Cumulative Standard Deviations: Standard Deviations in each direction combined across data series.

Error Budget					
	$\sigma_e(m)$	$\sigma_n(m)$	$\sigma_{xy}(m)$	CEP	$\sigma_u(m)$
Base	.0229	.0098	.0249	.0207	.0263
Rover	.0049	.005	.007	.0058	.011
Total	.0468	.022	.0517	.0431	.057

Table 4.4. Error Budget: Total error present in the system due to both the Base and Rover, calculated using Equation 4.1

$$\sigma_{Total} = 2 \cdot \sqrt{\sigma_{Base}^2 + \sigma_{Rover}^2} \quad (4.1)$$

4.2 Discussion

Upon completion of the field experiments and processing of the data, there were, in total, 4 successful tests. Two tests, Test 3 and Test 4, were thrown out because they failed to yield a FIX status post-processing. In Test 3, data from the Ashtech receiver post-processed in RTKLIB, shown in Figure 4.18, failed to get a good FIX. In Test 5, data from the Ashtech receiver post-processed in Emlid Studio, shown in Figure 4.32 failed to get a good FIX. Both tests were performed under good visibility; however, during Test 3, storm clouds were on the horizon, which most likely caused the issues. Test 5 was curious in that it failed to get a FIX with a combined forward and backward solution but received somewhere around a 95% fix with just the forward solution. It was thrown out in an effort to keep uniformity across tests; however, when the modified ISO tests were run on the data, the results were several orders of magnitude above the results of the other

tests. This seems to validate the testing procedures and the methods of filtering out gross errors. The failure in these two instances was undoubtedly because the Ashtechs are only equipped to detect GPS constellations and are, therefore, limited in the number of solutions they can generate. These failures also validate the importance of averaging errors across receivers and post-processors. In both instances, one post-processor yielded favorable results, while the other did not, demonstrating possible errors inherent to the post-processors.

Experiments 1,2,4, and 6 yielded very promising results when analyzed individually with the modified ISO tests. The residual distributions shown in Figures 4.42 - 4.46 all display very distinct Gaussian Distribution patterns, which shows that the methods used for deriving the standard deviations are appropriate. The deviations for each test shown in Table ?? along with the cumulative test results shown in Table 4.3, match very closely to the residuals found in the post-processed solutions when a FIX is achieved, showing that RTK solution has similar accuracy to the kinematic post-processing.

Recall that the accuracy statement in the ZED-F9P datasheet [23] are in terms of CEP, while the errors calculated in the modified ISO standard are only for one standard deviation. To convert the horizontal standard deviation (σ_{xy}) to CEP [20] a .833 multiplier is used, which yields a CEP of .0058 meters. When creating an error budget, it is common to double the error [7]. This places the calculated CEP of the rover at .0117 meters, almost identical to the .01 meters specified in the ZED-F9P datasheet. The calculated value being slightly higher is expected as the system was in a dynamic state near the base station. This demonstrates that the system is precise and gives confidence that the standard deviation calculation in the "up" (σ_u), or vertical direction, is reliable. If σ_u in

Table 4.3 is doubled, it yields a value of .022 meters, which is about double the horizontal error and typical of higher-end RTK systems.

Though the precision of the system looks favorable, when calculating the total error budget, it is necessary to account for the error associated with the base station. The base station error must be combined with the rover error using Equation 4.1. Table 4.4 shows that when the base station error is taken into account, the system error becomes much larger. This means that the system is precise, but not necessarily accurate.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis aimed to validate the horizontal accuracy and determine the vertical accuracy of the low-cost ZED-F9P receiver under dynamic conditions. Though compounding error from the base station skewed the accuracy estimates, the horizontal precision matched nicely with the manufacturer's stated values, and a reasonable value of vertical precision was found through testing and comparisons with a certified DGPS receiver. It also found that the modified ISO 17123-8 standard may offer a promising general solution for gauging the accuracy of highly dynamic RTK systems.

5.2 Future Work

Although the work in this thesis may compare nicely with the accuracy statements in the ZED-F9P datasheet, the experimental nature of the modified ISO 17123-8 experiment means that it needs further validation as a means to conduct accuracy estimates. The base station needs to be surveyed in with greater accuracy in an effort to minimize error and get better rover accuracy estimates.

To extend this work, comparing these results to a set of tests run as outlined by the original ISO 17123-8 standard would be good. It would also be interesting to perform both the modified ISO test and the original ISO test at distances further away from the base station to see how they compare to one another, along with the +1 ppm stated in the ZED-F9P datasheet. More work could also be done to try to identify the sources of systematic error that caused the residual distributions to shift from zero. Future applications of the RTK technology in GBAS and UAS-based deployments will demand more sophisticated configuration, testing, and real-time data processing.

Bibliography

- [1] Bartosz Ciechanowski, "GPS," Accessed: june 5, 2024. [Online]. Available: {<https://ciechanow.ski/gps/>}
- [2] Pretap Misra, Per Enge, *Global Positioning System: Signals, Measurements, and Performance Revised Second Edition*. Ganga-Jamuna Press, 2012.
- [3] Tran Dinh Trong, Luong Ngoc Dung, "Study on the positioning efficiency of gnss rtk for road profile surveys - case study in vietnam," *Journal of Science and Technology in Civil Engineering*, 2024.
- [4] J. W. Dyer, "A new siting model for reference station placement in a ground based augmentation system," Dissertation, University of Oklahoma, Norman, Oklahoma, 2008.
- [5] R. E. Phelts, D. M. Akos, "Nominal Signal Deformations: Limits on GPS Range Accuracy," Sydney, Australia, December 2004.
- [6] Pedro Oliveira, Jorge Silva, Paulo Soares, "A comparative study between ils and gbas approaches: The case of viseu airfield," *OmniaScience*, 2020.
- [7] International Organization for Standardization, "Optics and optical instruments — Field procedures for testing geodetic and surveying instruments, Part 8: GNSS field measurement systems in real-time kinematic (RTK)," 20015.
- [8] ArduSimple, "Testing simpleRTK2B accuracy in RTK base-rover configuration," Accessed: November 10, 2024. [Online]. Available: {<https://www.ardusimple.com/testing-simplertk2b-accuracy-in-rtk-base-rover-configuration/>}
- [9] Trimble, "GNSS Planning Online," Accessed: October 15, 2024. [Online]. Available: {<https://gnssplanning.com/#/settings>}
- [10] J. Gučević, S. Delčev, V. Ogrizović, S. Miljković, and M. Kuburić, "TERENSKO ISPITIVANJE GNSS PRIJEMNIKA U SKLADU SA PROCEDUROM ISO 17123-8," in *Zbornik GFS*, vol. 34, pp. 735–743, 2018.

- [11] Hans Heister, "The new ISO standard 17123-8 for checking GNSS field measuring systems," Stockholm, Sweden, June 2008.
- [12] S Artese, M Perrelli, "Monitoring a landslide with high accuracy by total station: A dtm-based model to correct for the atmospheric effects," *Geosciences*, 2018.
- [13] D K Schrader, B C Min, E T Matson, and J Eric Dietz, "Real-time averaging of position data from multiple gps receivers," *Measurement*, vol. 90, pp. 329–337, 2016.
- [14] D. K. Schrader, "Combining multiple, inexpensive gps receivers to increase accuracy and reliability," Master's thesis, Purdue, 2013.
- [15] Harveen Singh Chadha, "Extended Kalman Filter: Why do we need an Extended Version?" Accessed: November 2, 2024. [Online]. Available: {<https://towardsdatascience.com/extended-kalman-filter-43e52b16757d>}
- [16] Jinchang Electron, *GNSS Surveying Antenna JCA228F*.
- [17] u-blox, *ZED-F9P u-blox F9 high precision GNSS module Data Sheet*, Switzerland, September 2018.
- [18] NovAtel, "Z Family GPS Receiver Operations Reference Manual," 2008.
- [19] Digi International Inc, *Product Manual v1.xEx - 802.15.4 Protocol*, Minnetonka, MN, 2009.
- [20] NovAtel, "GPS Position Accuracy Measures," 2003.
- [21] Magellan Corporation, "GrafNav/GrafNet User Guide," 2000.
- [22] Thales, "Z Family GPS Receivers Technical Reference Manual," 2002.
- [23] u-blox, "ZED-F9P u-blox F9 high precision GNSS module Integration Manual," 2018.
- [24] —, *u-blox ZED-F9P Interface Description*, Switzerland, July 2019.
- [25] Tomoji Takasu, "RTKLIB Manual: Demo5 version," 2021.
- [26] Jan Van Sickle, "The Ionospheric Effect," Accessed: November 11, 2024. [Online]. Available: {<https://www.e-education.psu.edu/geog862/node/1715#:~:text=It%20is%20usually%20least%20between,than%208%20cm%20per%20minute.>}

Appendix A - List Of Symbols

$\rho(t)$	Pseudorange
$t^s(t - \tau)$	Emission time as measured by the receivers clock
t_u	Signals arrival time as measured by the receivers clock
c	Speed of Light in a vacuum
t	Arrival time in GPST from the receiver
δt_u	Receiver clock bias
δt^n	Satellites clock bias
$r(t, t - \tau)$	Geometric, or true, range between the receiver position at time t and the satellite position at time $(t - \tau)$
I_ρ	Ionosphere Delay
λ	Carrier Wavelength
r	Geometric Range

N	Integer Ambiguity
ϕ	Carrier Phase Measurement
D^*	Nominal Horizontal Distance
h^*	Nominal Vertical Distance
$D_{i,j}$	Horizontal distance between rovers
$\Delta h_{i,j}$	Height distance between rovers
$\varepsilon_{D,i,j}$	Difference in horizontal distance between rovers and nominal distance
$\varepsilon_{h,i,j}$	Difference in vertical distance between rovers and nominal height
σ_{xy}	Horizontal Standard Deviation
σ_{xy}	Vertical Standard Deviation
$\overline{x_k}$	Mean of all x coordinate points
$\overline{y_k}$	Mean of all y coordinate point
$\overline{h_k}$	Mean of all h coordinate point

$r_{xi,j,k}$	Residuals of the x coordinate point
$r_{yi,j,k}$	Residuals of the y coordinate point
$r_{hi,j,k}$	Residuals of the h coordinate point
m	Number of series
n	Number of sets in each series
p	Number of rover points
v_x, v_y, v_h	Degrees of Freedom
σ_x	Standard Deviation of x components
σ_y	Standard Deviation of y components
σ_h	Standard Deviation of h components
$\sigma_{ISO-GNSSRTK-xy}$	Experimental standard deviation of single position
$\sigma_{ISO-GNSS-h}$	Experimental standard deviation of height
σ_{xy}	Standard Deviation of horizontal position

σ_e	Standard Deviation East
σ_e	Standard Deviation West
σ_e	Standard Deviation North
ε_ρ	Represents the modeled effects, modeling errors, and measurement errors in the code phase sce- nario.
ε_ϕ	Represents the modeled effects, modeling errors, and measurement errors in the carrier phase sce- nario.

Appendix B - List of Acronyms and Abbreviations

CSRS-PPP	Canadian Spatial Reference System Precise Point Positioning
CEP	Circular Error Probable
C/A Code	Coarse Acquisition Code
DGPS	Differential GPS
GNSS	Global Navigation Satellite System
ECEF	Earth Centered, Earth Fixed Coordinate System
EKF	Extended Kalman Filter
GBAS	Ground Based Augmentation System
GPS	Global Positioning System
GPST	GPS Time
GT	Ground Truth

GUI	Graphical User Interface
ILS	Instrument Landing System
L1	Link 1 Frequency
L2	Link 2 Frequency
L5	Link 2 Frequency
LAAS	Local Area Augmentation System
LLH	Latitude, Longitude, Height Coordinate System
NOAA	National Oceanic and Atmospheric Administration
ppm	Parts Per Million
P(Y) Code	Precision Acquisition Code
PPP	Precise Point Positioning Service
RPi	Raspberry Pi
RTK	Real Time Kinematics

sUAS	Small Unmanned Aerial System
SNR	Signal-to-Noise Ration
T1	Test 1/Series 1
T2	Test 2/Series 2
T3	Test 3/Series 3
T4	Test 4/Series 4
T5	Test 5/Series 5
T6	Test 6/Series 6
UAS	Unmanned Aerial System
VHF	Very High Frequency
YAT	Yet Another Terminal

Appendix C - Ashtech Quick Reference

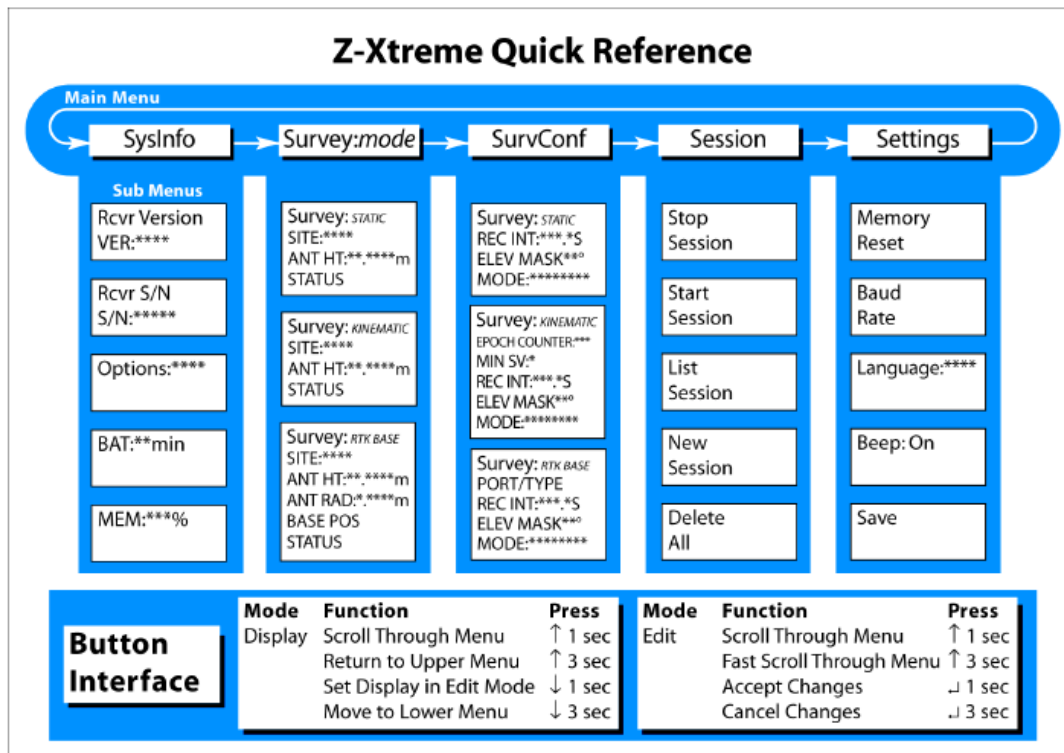


Figure C.0.1. Ashtech Quick Reference

Appendix D - RPi Data Logger

D.1 RPi Wiring Diagram and Pinout

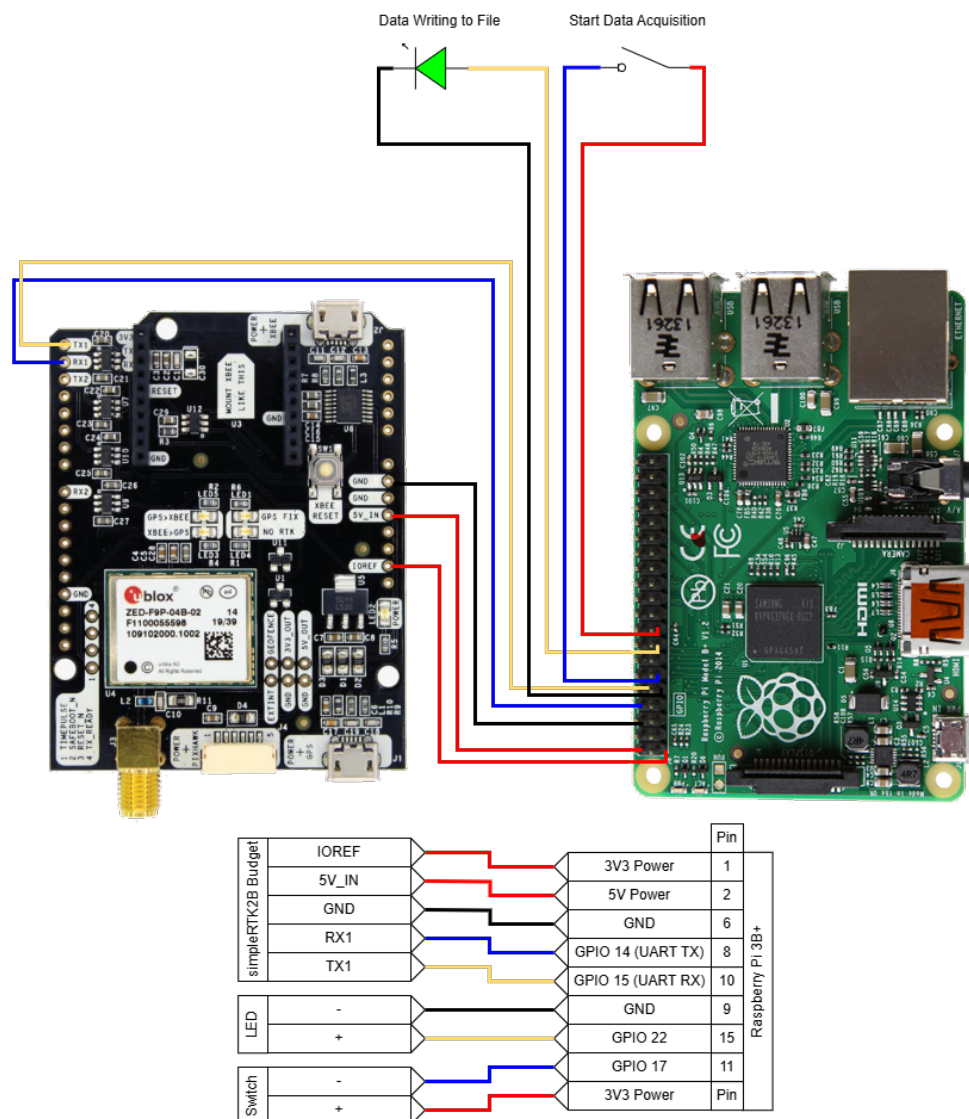


Figure D.1.1. ZED-F9P Connection Diagram

D.2 Data Logger Script

```
#!/usr/bin/env python
import sys
import serial
import datetime
import time
import os
from pyubx2 import UBXReader, NMEA_PROTOCOL, UBX_PROTOCOL
import RPi.GPIO as GPIO

BUTTON_PIN = 11
WRITE_STATUS_LED = 15

GPIO.setmode(GPIO.BOARD)

GPIO.setup(BUTTON_PIN, GPIO.IN, pull_up_down=GPIO.
            PUDDOWN)
GPIO.setup(WRITE_STATUS_LED, GPIO.OUT)

def acquire_data():
    ser=serial.Serial('/dev/serial0', 115200, timeout=3)

    RAW_DATA_FILE_PATH = datetime.datetime.now().strftime
        ("/home/pi/Documents/RTK-Project/Data/RAW-%Y-%m-%
        d_%H%M%S.ubx")
    RTK_DATA_FILE_PATH = datetime.datetime.now().strftime
        ("/home/pi/Documents/RTK-Project/Data/RTK-%Y-%m-%
        d_%H%M%S.txt")
    TIME_STAMP = datetime.datetime.now().strftime("%Y/%m
        /%d_%H%M%S")

    ##      with ser as stream, open(RAW_DATA_FILE_PATH,'ab')
    as raw_file, open(RTK_DATA_FILE_PATH,'w') as rtk_file:
        with ser as stream, open(RAW_DATA_FILE_PATH,'ab') as
        raw_file:

            ubr = UBXReader(stream)
```

```

button_state = poll_button_state()

while (button_state):
    try:
        raw_data, parsed_data = ubr.read()
        button_state = poll_button_state()
        if parsed_data is not None:
            print(f"Message identity: {
                parsed_data.identity}")
            GPIO.output(WRITE_STATUS_LED, GPIO.
                HIGH)
            raw_file.write(raw_data)
            raw_file.flush()
        else:
            print('No data parsed.')
            GPIO.output(WRITE_STATUS_LED, GPIO.
# LOW)

    except Exception as e:
        print(f"Error reading data: {e}")
        GPIO.output(WRITE_STATUS_LED, GPIO.LOW)

#         except KeyboardInterrupt:
#             print('Stopped by user')
#             break

def poll_button_state():
    return GPIO.input(BUTTON_PIN)

while(True):
    try:
        GPIO.output(WRITE_STATUS_LED, GPIO.LOW)
        button_state = poll_button_state()
        print(button_state)
        if button_state == 1:
            acquire_data()
            time.sleep(0.1)
    except KeyboardInterrupt:
#         print('\nExiting')
        GPIO.cleanup()

```

```
GPIO.output(WRITE_STATUS_LED, GPIO.LOW)
ser.close()
sys.exit(0)
```

Appendix E - Matlab Data Analysis

E.1 Individual Test Analysis and Graphing

```
clear all;
close all;

% Read in data to time tables
-----

rtklib_A = readtimetable("RTKLIB_A_T1.txt");
rtklib_F = readtimetable("RTKLIB_F_T1.txt");
RTK = readtimetable("T1.txt");
emlid_A = readtimetable("Emlid_A_T1.txt");
emlid_F = readtimetable("Emlid_F_T1.txt");

% Remove all data that is not FIXED statux
-----

for i=1:height(rtklib_A)
    if rtklib_A.Q(i) > 1
        rtklib_A{i,"latitude_deg_"} = nan;
        rtklib_A{i,"longitude_deg_"} = nan;
        rtklib_A{i,"height_m_"} = nan;
    end
end

for i=1:height(rtklib_F)
    if rtklib_F.Q(i) > 1
        rtklib_F{i,"latitude_deg_"} = nan;
        rtklib_F{i,"longitude_deg_"} = nan;
        rtklib_F{i,"height_m_"} = nan;
    end
end
```

```

end

for i=1:height(emplid_A)
    if emplid_A.Q(i) > 1
        emplid_A{i,"latitude_deg_"} = nan;
        emplid_A{i,"longitude_deg_"} = nan;
        emplid_A{i,"height_m_"} = nan;
    end
end

for i=1:height(emplid_F)
    if emplid_F.Q(i) > 1
        emplid_F{i,"latitude_deg_"} = nan;
        emplid_F{i,"longitude_deg_"} = nan;
        emplid_F{i,"height_m_"} = nan;
    end
end

for i=1:height(RTK)
    if RTK.Var6(i) < 3
        RTK{i,"Var1"} = nan;
        RTK{i,"Var2"} = nan;
        RTK{i,"Var3"} = nan;
    end
end

%---Initial Plots-----

% RTKLIB_A

figure()

geoplot(rtklib_A.latitude_deg_ , rtklib_A.longitude_deg_ , 'g
');
hold on
geoplot(35.3100704, -97.033461275, "c.", "MarkerSize", 10)
title(' (T1) Ground Track RTKLIB Ashtech ')
geobasemap satellite;
saveas(figure(1), 'T1.R.A.GND-TRK.png');

figure()

```

```

    tiledlayout(3,1)
    nexttile
    plot(rtklib_A.GPST,rtklib_A.latitude_deg_);
    title(' (T1) RTKLIB Ashtech Position vs. Time ');
    ylabel('Latitude (DD) ');
    nexttile
    plot(rtklib_A.GPST,rtklib_A.longitude_deg_);
    ylabel('Longitude (DD) ');
    nexttile
    plot(rtklib_A.GPST,rtklib_A.height_m_);
    ylabel({'Ellipsiod Height'; '(meters)'});
    xlabel('Time')
    saveas(figure(2),'T1_R_A_Position.png');

```

%RTKLIB_F

```

figure()

geoplot(rtklib_F.latitude_deg_,rtklib_F.longitude_deg_,'g
');
hold on
geoplot(35.3100704,-97.033461275,"c. ","MarkerSize",10)
title(' (T1) Ground Track RTKLIB ZED-F9P ')
geobasemap satellite;
saveas(figure(3),'T1_R_F_GND_TRK.png');

```

```

figure()

tiledlayout(3,1)
nexttile
plot(rtklib_F.GPST,rtklib_F.latitude_deg_);
title(' (T1) RTKLIB ZED-F9P Position vs. Time ');
ylabel('Latitude (DD) ');
nexttile
plot(rtklib_F.GPST,rtklib_F.longitude_deg_);
ylabel('Longitude (DD) ');
nexttile
plot(rtklib_F.GPST,rtklib_F.height_m_);
ylabel({'Ellipsiod Height'; '(meters)'});
xlabel('Time')

```

```

saveas(figure(4), 'T1_R_F_Position.png');

    % Emlid_A

figure()

geoplot(emlid_A.latitude_deg_, emlid_A.longitude_deg_, 'g')
;
hold on
geoplot(35.3100704, -97.033461275, "c.", "MarkerSize", 10)
title(' (T1) Ground Track Emlid Ashtech ')
geobasemap satellite
saveas(figure(5), 'T1_E_A_GND-TRK.png');

```

```

figure()

tiledlayout(3,1)
nexttile
plot(emlid_A.GPST, emlid_A.latitude_deg_);
title(' (T1) Emlid Ashtech Position vs. Time ');
ylabel('Latitude (DD) ');
nexttile
plot(emlid_A.GPST, emlid_A.longitude_deg_);
ylabel('Longitude (DD) ');
nexttile
plot(emlid_A.GPST, emlid_A.height_m_);
ylabel({ 'Ellipsoid Height'; '(meters)' });
xlabel('Time')
saveas(figure(6), 'T1_E_A_Position.png');

```

```

    %Emlid_F

figure()

geoplot(emlid_F.latitude_deg_, emlid_F.longitude_deg_, 'g')
;
hold on;
geoplot(35.3100704, -97.033461275, "c.", "MarkerSize", 10)
title(' (T1) Ground Track Emlid ZED-F9P ')

```



```
geobasemap satellite
saveas(figure(7), 'T1_E_F_GND-TRK.png');
```

```
figure()
```

```
tiledlayout(3,1)
nexttile
plot(emlid_F.GPST,emlid_F.latitude_deg_);
title(' (T1) -Emlid-ZED-F9P-Position-vs.-Time ');
ylabel('Latitude (DD) ');
nexttile
plot(emlid_F.GPST,emlid_F.longitude_deg_);
ylabel('Longitude (DD) ');
nexttile
plot(emlid_F.GPST,emlid_F.height_m_);
ylabel({ 'Ellipsoid Height'; '(meters)' });
xlabel('Time')
saveas(figure(8), 'T1_E_F_Position.png');
```

```
% Combined
```

```
figure()
```

```
geoplot(rtklib_A.latitude_deg_, rtklib_A.longitude_deg_);
hold on
geoplot(rtklib_F.latitude_deg_, rtklib_F.longitude_deg_);
hold on
geoplot(emlid_A.latitude_deg_, emlid_A.longitude_deg_);
hold on
geoplot(emlid_F.latitude_deg_, emlid_F.longitude_deg_);
hold on;
geoplot(35.3100704, -97.033461275, "c.", "MarkerSize", 10)
title(' (T1) -Ground-Track-Combined ')
geobasemap satellite
legend('RTKLIB-Ashtech', 'RTKLIB-ZED-F9P', 'Emlid-Ashtech',
        'Emlid-ZED-F9P', 'Base')
saveas(figure(9), 'T1_Combined_GND-TRK.png');
```

```
figure()
```

```

tiledlayout(3,1)
nexttile
plot(rtklib_A.GPST,rtklib_A.latitude_deg_);
plot(rtklib_F.GPST,rtklib_F.latitude_deg_);
plot(emlid_A.GPST,emlid_A.latitude_deg_);
plot(emlid_F.GPST,emlid_F.latitude_deg_);
title(' (T1) Combined Position vs. Time')
ylabel('Latitude (DD)');

nexttile
plot(rtklib_A.GPST,rtklib_A.longitude_deg_);
plot(rtklib_F.GPST,rtklib_F.longitude_deg_);
plot(emlid_A.GPST,emlid_A.longitude_deg_);
plot(emlid_F.GPST,emlid_F.longitude_deg_);
ylabel('Longitude (DD)');

nexttile
plot(rtklib_A.GPST,rtklib_A.height_m_);
plot(rtklib_F.GPST,rtklib_F.height_m_);
plot(emlid_A.GPST,emlid_A.height_m_);
plot(emlid_F.GPST,emlid_F.height_m_);
ylabel({'Ellipsoid Height'; '(meters)'});
xlabel('Time')
saveas(figure(10),'T1-Combined-Position.png');

```

```

%Shift RTK to start with the rest of the data
%-----

```

```

RTK2 = retime(RTK,"regular","TimeStep",milliseconds(100))
; % Insert missing timesteps and linearly interpolate
between them
temp = synchronize(rtklib_A,RTK); % Synchronize RTK with
Ashtech to get proper time shift
[N,K] = max(temp.Var1); % Find location maximum value in
RTK data
[M,I] = max(temp.latitude_deg_); % Find location maximum
value in Ashtech data
J = (I-K); % Time difference
RTK2 = lag(RTK2,J); % Shift RTK data to match other data

```

```

% RTK2 = rmmissing(RTK2);

% Rebuild timetables to only include time, lat, lon, and
  height-----

rtklib_F = timetable(rtklib_F.GPST, rtklib_F.
    latitude_deg_, rtklib_F.longitude_deg_, rtklib_F.
    height_m_);
rtklib_A = timetable(rtklib_A.GPST, rtklib_A.
    latitude_deg_, rtklib_A.longitude_deg_, rtklib_A.
    height_m_);
emlid_F = timetable(emlid_F.GPST, emlid_F.latitude_deg_,
    emlid_F.longitude_deg_, emlid_F.height_m_);
emlid_A = timetable(emlid_A.GPST, emlid_A.latitude_deg_,
    emlid_A.longitude_deg_, emlid_A.height_m_);
Combined = synchronize(rtklib_A, rtklib_F, emlid_A,
    emlid_F);
Combined = rmmissing(Combined);

avg_Combined_lat = (Combined.Var1_rtklib_A + Combined.
    Var1_rtklib_F + Combined.Var1_emlid_A +Combined.
    Var1_emlid_F) ./4;
avg_Combined_lon = (Combined.Var2_rtklib_A + Combined.
    Var2_rtklib_F + Combined.Var2_emlid_A +Combined.
    Var2_emlid_F) ./4;
avg_Combined_h = (Combined.Var3_rtklib_A + Combined.
    Var3_rtklib_F + Combined.Var3_emlid_A +Combined.
    Var3_emlid_F) ./4;

GT = timetable(Combined.Time, avg_Combined_lat,
    avg_Combined_lon, avg_Combined_h);

figure()
geoplot(GT.Var1,GT.Var2, "g");
hold on;
geoplot(35.3100704,-97.033461275,"c. ","MarkerSize",10)
legend('Ground Truth','Base Station')
title(' (T1) Ground Truth')
geobasemap satellite
saveas(figure(11),'T1_GT_GND-TRK.png');

```

```

figure()
geoplot(RTK.Var1,RTK.Var2, "r");
hold on;
geoplot(35.3100704,-97.033461275,"c.,""MarkerSize",10)
legend('RTK','Base Station')
title(' (T1) RTK')
geobasemap satellite
saveas(figure(12),'T1.RTK.GND-TRK.png');

```

```

figure()
geoplot(GT.Var1,GT.Var2, "g");
hold on
geoplot(RTK.Var1,RTK.Var2, "r");
hold on
geoplot(35.3100704,-97.033461275,"c.,""MarkerSize",10)
legend('Ground Truth','RTK','Base Station')
title(' (T1) Ground Truth & RTK')
geobasemap satellite
saveas(figure(13),'T1.GT.RTK.GND-TRK.png');

```

```

figure()
tiledlayout(3,1)
nexttile
plot(GT.Time,GT.Var1,'g');
title(' (T1) GT Position vs. Time')
ylabel('Lat');
nexttile
plot(GT.Time,GT.Var2,"g");
ylabel('Lon');
nexttile
plot(GT.Time,GT.Var3,"g");
ylabel('Height');
xlabel('Time');
saveas(figure(14),'T1_GT_Position.png');

```

```

figure()
tiledlayout(3,1)
nexttile
plot(RTK2.Time,RTK2.Var1,'r');
title(' (T1) RTK Position vs. Time')

```

```

ylabel('Lat');
nexttile
plot(RTK2.Time,RTK2.Var2,'r');
ylabel('Lon');
nexttile
plot(RTK2.Time,RTK2.Var3,'r');
ylabel('Height');
xlabel('Time')
saveas(figure(15),'T1-RTK-Position.png');

figure()

tiledlayout(3,1);
nexttile
plot(GT.Time,GT.Var1,"g");
hold on
plot(RTK2.Time,RTK2.Var1,"r");
title('(T1) GT and RTK Position vs. Time')
ylabel('Lat');

nexttile
plot(GT.Time,GT.Var2);
hold on
plot(RTK2.Time,RTK2.Var2);
ylabel('Lon');

nexttile
plot(GT.Time,GT.Var3);
hold on
plot(RTK2.Time,RTK2.Var3);
ylabel('Height');
xlabel('Time');
saveas(figure(16),'T1-GT-RTK-Position.png');

% Average Error across averaged measurements
-----

[de1,dn1,du1] = geodetic2enu(Combined.Var1_rtklib_A ,
    Combined.Var2_rtklib_A , Combined.Var3_rtklib_A , GT.
    Var1 , GT.Var2 , GT.Var3,wgs84Ellipsoid);
[de2,dn2,du2] = geodetic2enu(Combined.Var1_rtklib_F ,
    Combined.Var2_rtklib_F , Combined.Var3_rtklib_F , GT.
    Var1 , GT.Var2 , GT.Var3,wgs84Ellipsoid);

```

```

[de3,dn3,du3] = geodetic2enu(Combined.Var1_emlid_A ,
    Combined.Var2_emlid_A , Combined.Var3_emlid_A , GT.Var1 ,
    GT.Var2 , GT.Var3 , wgs84Ellipsoid);
[de4,dn4,du4] = geodetic2enu(Combined.Var1_emlid_F ,
    Combined.Var2_emlid_F , Combined.Var3_emlid_F , GT.Var1 ,
    GT.Var2 , GT.Var3 , wgs84Ellipsoid);

a_sig_e = sqrt(((de1.^2)+(de2.^2)+(de3.^2)+(de4.^2))./3);
a_sig_n = sqrt(((dn1.^2)+(dn2.^2)+(dn3.^2)+(dn4.^2))./3);
a_sig_u = sqrt(((du1.^2)+(du2.^2)+(du3.^2)+(du4.^2))./3);

a_sig_h = sqrt((a_sig_e.^2)+(a_sig_n.^2)); %horizontal
    error at each point
a_sig = sqrt((a_sig_e.^2)+(a_sig_n.^2)+(a_sig_u.^2)); %
    vertical error at each point

a_xyh = sum(a_sig)/height(a_sig) %average 3d error
a_xy = sum(a_sig_h)/height(a_sig_h) %average horizontal
    error
a_h = sum(a_sig_u)/height(a_sig_u) %average height error

geo = synchronize(GT,RTK2);

geo = rmmissing(geo);

%-----Simplified Test Procedure
%-----

x = zeros(height(geo),1);

d_RTK2_e = x;
d_RTK2_n = x;
d_RTK_u = x;
D_RTK = x;

d_GT_e = x;
d_GT_n = x;
d_GT_u = x;
D_GT = x;

```

```

ED = x;
Eh = x;

flag = 0;
ED_th = 2.5*sqrt(2)*.01; %threshold stated by
    manufacturer
Eh_th = 2.5*sqrt(2)*.02; %threshold estimate

% Raise a flag for any value over the threshold

for i = 1:height(geo)-1

    [d_RTK2_e(i),d_RTK2_n(i),d_RTK_u(i)] = geodetic2enu(
        geo.Var1_RTK2(i+1),geo.Var2_RTK2(i+1),geo.
        Var3_RTK2(i+1),geo.Var1_RTK2(i),geo.Var2_RTK2(i),
        geo.Var3_RTK2(i),wgs84Ellipsoid);
    D_RTK(i) = sqrt((d_RTK2_n(i)^2)+(d_RTK2_e(i)^2));

    [d_GT_e(i),d_GT_n(i),d_GT_u(i)] = geodetic2enu(geo.
        Var1_GT(i+1),geo.Var2_GT(i+1),geo.Var3_GT(i+1),geo.
        Var1_GT(i),geo.Var2_GT(i),geo.Var3_GT(i),
        wgs84Ellipsoid);
    D_GT(i) = sqrt((d_GT_n(i)^2)+(d_GT_e(i)^2));

    ED(i) = D_RTK(i) - D_GT(i);
    Eh(i) = d_RTK_u(i) - d_GT_u(i);

    if abs(ED(i)) >= ED_th | abs(Eh(i)) >= Eh_th
        flag = flag+1;
        geo{i,"Var1_GT"} = nan;

    end

end

geo = rmmissing(geo);
%-----Full Test Procedure
-----

[de,dn,du] = geodetic2enu(geo.Var1_RTK2,geo.Var2_RTK2,geo.
    Var3_RTK2,geo.Var1_GT,geo.Var2_GT,geo.Var3_GT,
    wgs84Ellipsoid);

```

```

% de = (de(de>=-.02 & de<=.02));
% dn = (dn(dn>=-.02 & dn<=.02));
% du = (du(du>=-.05 & du<=.05));

sig_e = sqrt(sum(de.^2)/(height(de)-1)) %final std east
sig_n = sqrt(sum(dn.^2)/(height(dn)-1)) %final std north
sig_xy = sqrt((sig_e^2)+(sig_n^2)) %final std horizontal
sig_u = sqrt(sum(du.^2)/(height(du)-1)) %final std up

```

figure()

```

histogram(de,BinWidth=.001);
hold on;
pd = fitdist(de,'Normal');
x_values = min(de):.0001:max(de);
y = pdf(pd,x_values);
plot(x_values,y);
xlabel('Meters');
ylabel('QTY');
title('(T1)-Standard-Deviation-East');
saveas(figure(17),'T1_ste.png');

```

figure()

```

histogram(dn,BinWidth=.001);
hold on;
pd = fitdist(dn,'Normal');
x_values = min(dn):.0001:max(dn);
y = pdf(pd,x_values);
plot(x_values,y);
xlabel('Meters');
ylabel('QTY');
title('(T1)-Standard-Deviation-North');
saveas(figure(18),'T1_stn.png');

```

figure()

```

histogram(du,BinWidth=.001);
hold on;

```



```

pd = fitdist(du, 'Normal');
x_values = min(du):.0001:max(du);
y = pdf(pd, x_values);
plot(x_values, y);
xlabel('Meters');
ylabel('QTY');
title('(T1) Standard Deviation Up')
saveas(figure(19), 'T1_stu.png');

figure()
plot3(GT.Var2, GT.Var1, GT.Var3);
hold on
plot3(RTK2.Var2, RTK2.Var1, RTK2.Var3);
xlabel('Longitude')
ylabel('Latitude')
zlabel('Height')
title('(T1) 3D Ground Truth vs RTK')
saveas(figure(20), 'T1_3D.png');

```

E.2 Total Error Analysis

```

clear all;
close all;

T1 = load("Analysis\T1.mat");
T2 = load("Analysis\T2.mat");
T3 = load("Analysis\T4.mat");
T4 = load("Analysis\T6.mat");

de = cat(1, T1.de, T2.de);
de = cat(1, de, T3.de);
de = cat(1, de, T4.de);

dn = cat(1, T1.de, T2.dn);
dn = cat(1, dn, T3.dn);
dn = cat(1, dn, T4.dn);

du = cat(1, T1.de, T2.du);
du = cat(1, du, T3.du);

```

```

du = cat(1,du,T4.du);

ve = (height(T1.de)-1)+(height(T2.de)-1)+(height(T3.de)-1)+(height(T4.de)-1);
vn = (height(T1.dn)-1)+(height(T2.dn)-1)+(height(T3.dn)-1)+(height(T4.dn)-1);
vu = (height(T1.du)-1)+(height(T2.du)-1)+(height(T3.du)-1)+(height(T4.du)-1);

sig_e = sqrt(sum(de.^2)/ve) %final std east
sig_n = sqrt(sum(dn.^2)/vn) %final std north
sig_xy = sqrt((sig_e^2)+(sig_n^2)) %final std horizontal
sig_u = sqrt(sum(du.^2)/vu) %final std up

```

figure()

```

histogram(de,BinWidth=.001);
hold on;
pd = fitdist(de,'Normal');
x_values = min(de):.0001:max(de);
y = pdf(pd,x_values);
plot(x_values,y);
xlabel('Meters');
ylabel('QTY');
title('Total Standard Deviation East');
saveas(figure(1),'Total_ste.png');

```

figure()

```

histogram(dn,BinWidth=.001);
hold on;
pd = fitdist(dn,'Normal');
x_values = min(dn):.0001:max(dn);
y = pdf(pd,x_values);
plot(x_values,y);
xlabel('Meters');
ylabel('QTY');
title('Total Standard Deviation North');
saveas(figure(2),'Total_stn.png');

```

figure()

```

histogram(du, BinWidth=.001);
hold on;
pd = fitdist(du, 'Normal');
x_values = min(du):.0001:max(du);
y = pdf(pd, x_values);
plot(x_values, y);
xlabel('Meters');
ylabel('QTY');
title('Total Standard Deviation Up')
saveas(figure(3), 'Total_stu.png');

```

Appendix F - Python NMEA/UBX Reader

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
import math
UBX_DATA_FILE_PATH = r"C:\Users\mans0653\OneDrive\University of Oklahoma\Grad School\ECE\RTK Research\Tests\Python\Filters\2024.11.1_F9P_R_T3\T6.txt"

from pyubx2 import UBXReader, UBX_PROTOCOL, ubxhelpers
with open(r"C:\Users\mans0653\OneDrive\University of Oklahoma\Grad School\ECE\RTK Research\Tests\Python\Filters\2024.11.1_F9P_R_T3\2024_11_1_F9P_R_T3.ubx", 'rb') as stream, open(UBX_DATA_FILE_PATH, 'w') as UBX_file:
    ubr = UBXReader(stream, protfilter=UBX_PROTOCOL)
    for raw_data, parsed_data in ubr:
        # print(parsed_data)
        if parsed_data.identity == 'NAV-PVT':
            print(parsed_data)
            # nano = Decimal((parsed_data.nano)*(10**-9))
            # nano = round(nano, 3)*100
            nano = (parsed_data.nano)*10**-9
            nano = math.trunc(round(nano, 3)*1000)
            utctime = ubxhelpers.itow2utc(parsed_data iTOW)
            print(utctime)
```

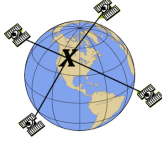
```

#           UBX_file.write(f'{parsed_data.hour}:{
parsed_data.min}:{parsed_data.second}.{nano}      {
parsed_data.lat}      {parsed_data.lon}      {parsed_data.
height}      {parsed_data.hAcc}      {parsed_data.vAcc}
      {parsed_data.fixType} \n')
      UBX_file.write(f'{utctime}''''{parsed_data.lat}
      ''''{parsed_data.lon}''''{parsed_data.height
      *10**-3}''''{parsed_data.hAcc}''''{
      parsed_data.vAcc}''''{parsed_data.fixType}'\
      n')

#           else parsed_data.identity == 'GNGST'
# parsed_data.identity == 'GNGST' or

```

Appendix G - PPP Survey Results



CSRS-PPP 3.54.2 (2022-11-10)



B_Corrected_B05.24o

Data Start	Data End	Duration of Observations
2024-09-02 01:09:00.00	2024-09-02 13:40:30.00	12:31:30
Processing Time		Product Type
15:41:51 UTC 2024/09/03		NRCan Rapid
Observations	Frequency	Mode
Phase and Code	Double	Static
Elevation Cut-Off	Rejected Epochs	Fixed Ambiguities
7.5 degrees	0.00 %	98.55 %
		Estimation Steps
		30.00 sec
Antenna Model	APC to ARP	ARP to Marker
Unknown	Unknown	H:0.000m / E:0.000m / N:0.000m

(APC = antenna phase center; ARP = antenna reference point)

Estimated Position for B_Corrected_B05.24o

	Latitude (+n)	Longitude (+e)	Ell. Height
ITRF20 (2024.7)	35° 18' 36.25365"	-97° 2' 0.46136"	310.113 m
Sigmas(95%)	0.005 m	0.004 m	0.019 m
A priori*	35° 18' 36.27842"	-97° 2' 0.45788"	314.479 m
Estimated – A priori	-0.763 m	-0.088 m	-4.366 m

Orthometric Height
CGVD2013
(CGG2013a)
(2024.7)

337.781 m

(click for height reference information)

95% Error Ellipse (mm)
semi-major: 6 mm
semi-minor: 5 mm
semi-major azimuth: 1° 36' 32.24"

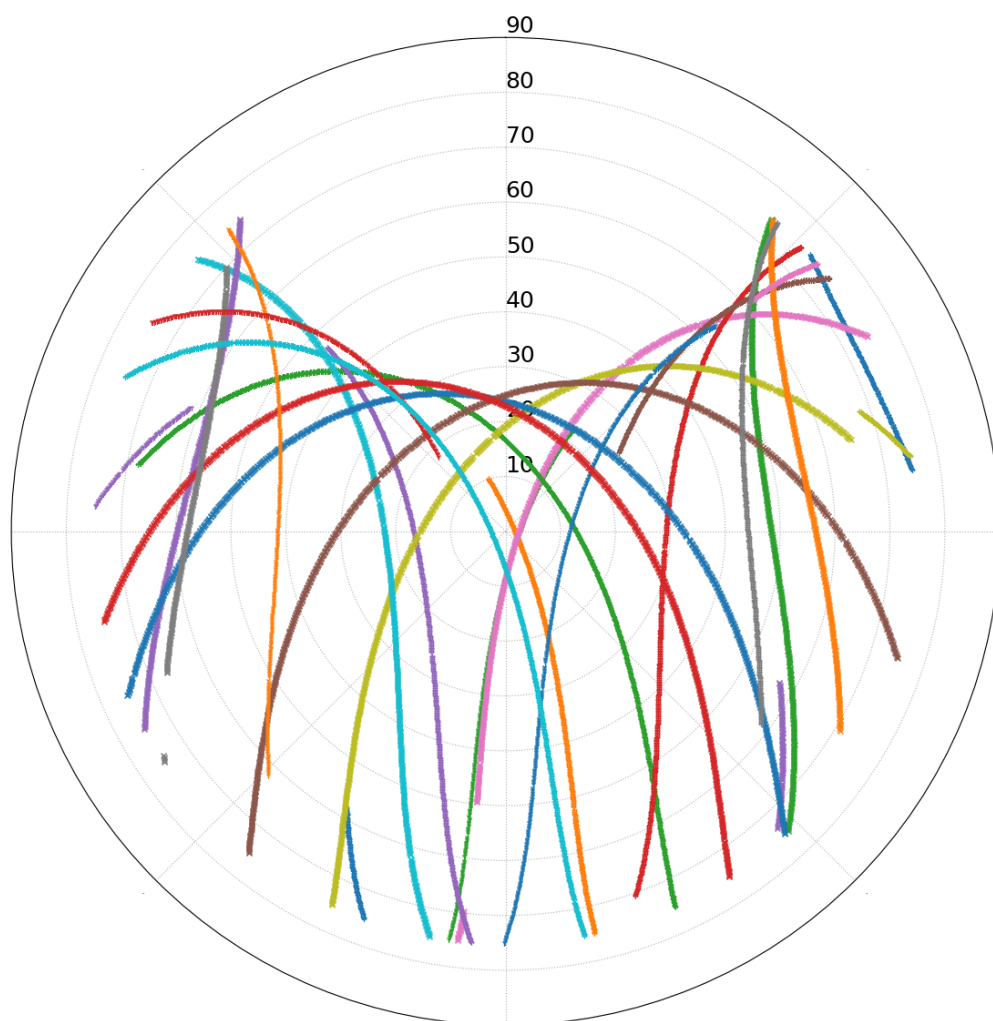
UTM (North)
Zone 14

3909203.373 m (N)
678782.074 m (E)

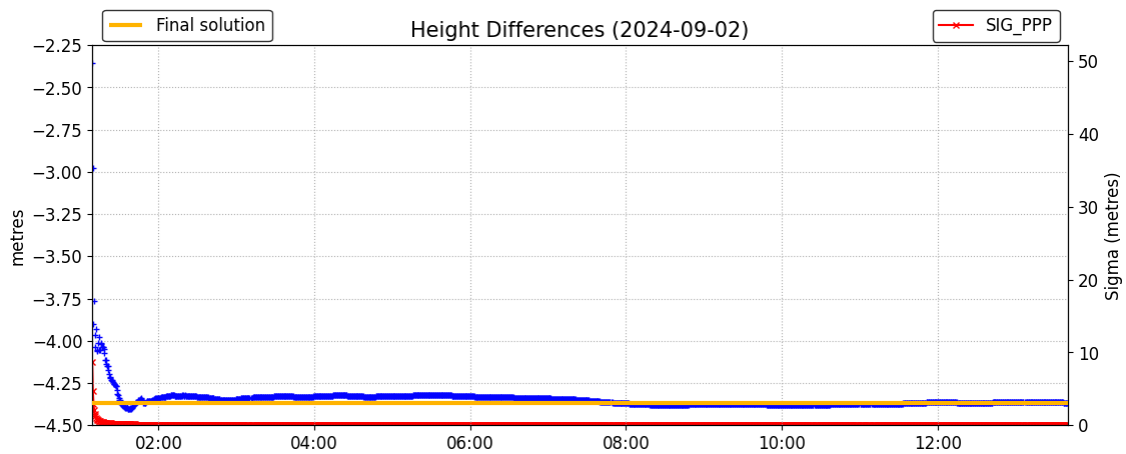
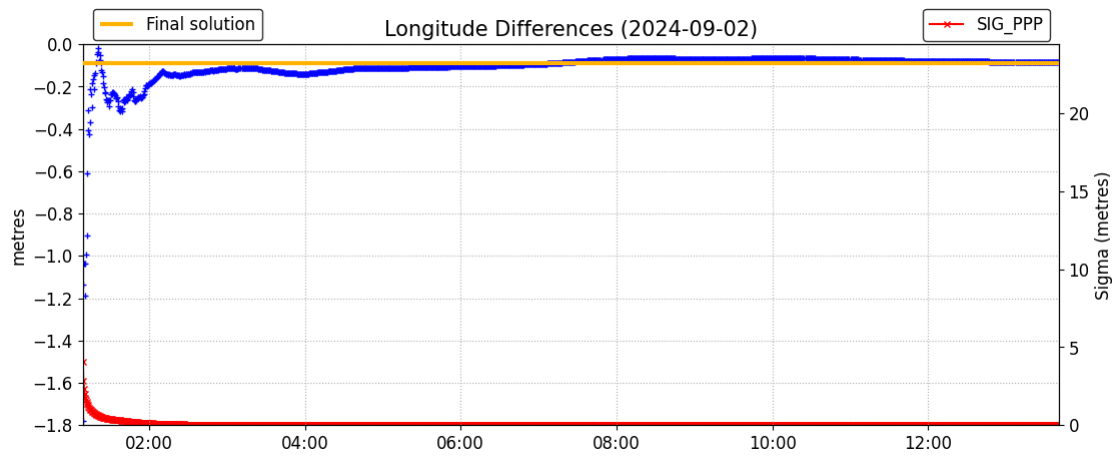
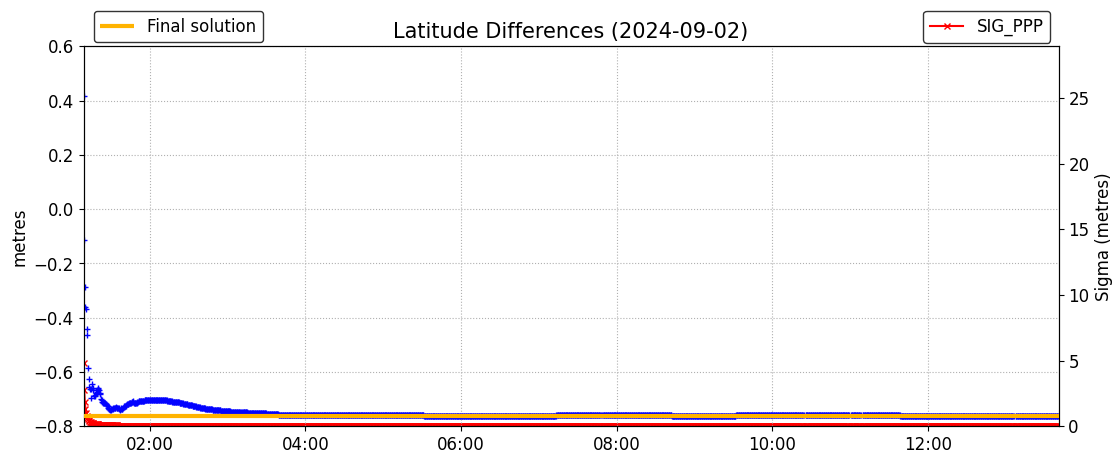
Scale Factors
0.99999392 (point)
0.99994524 (combined)

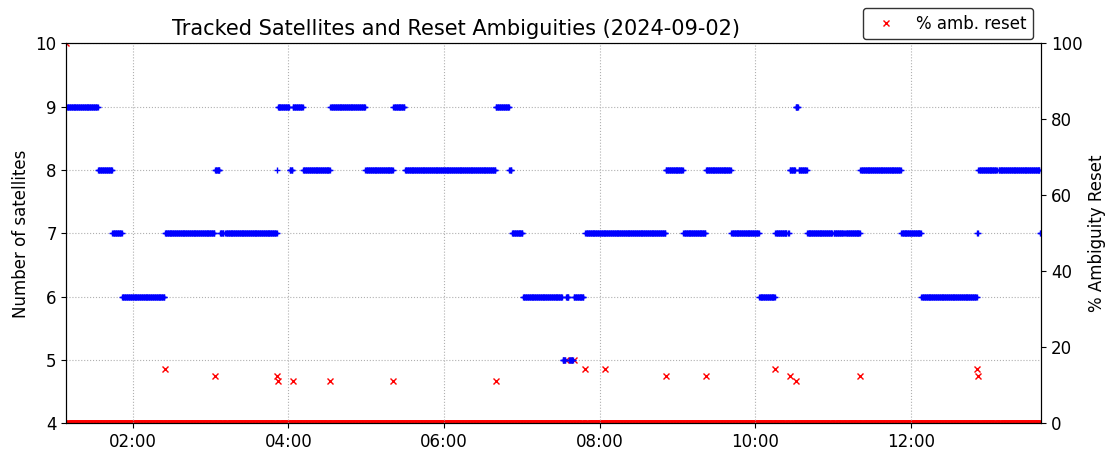
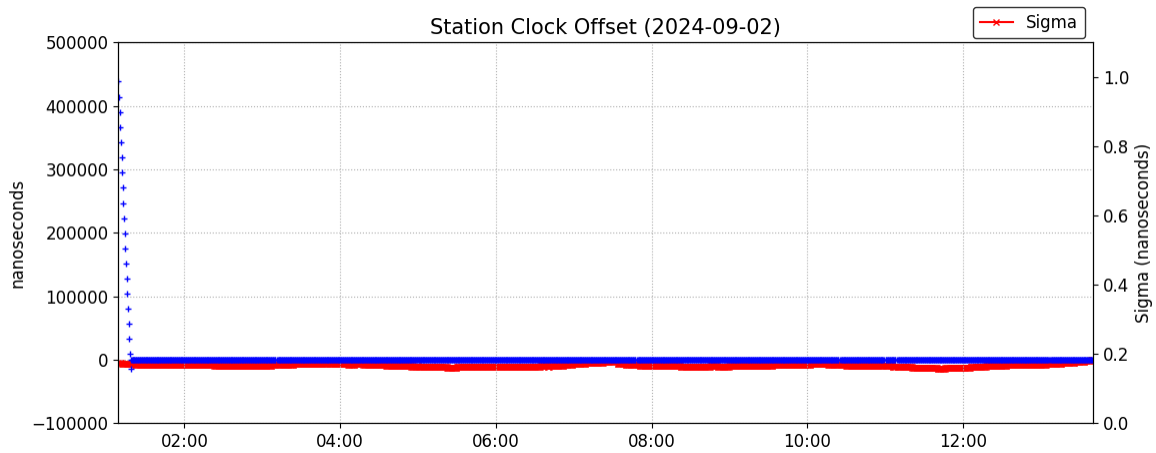
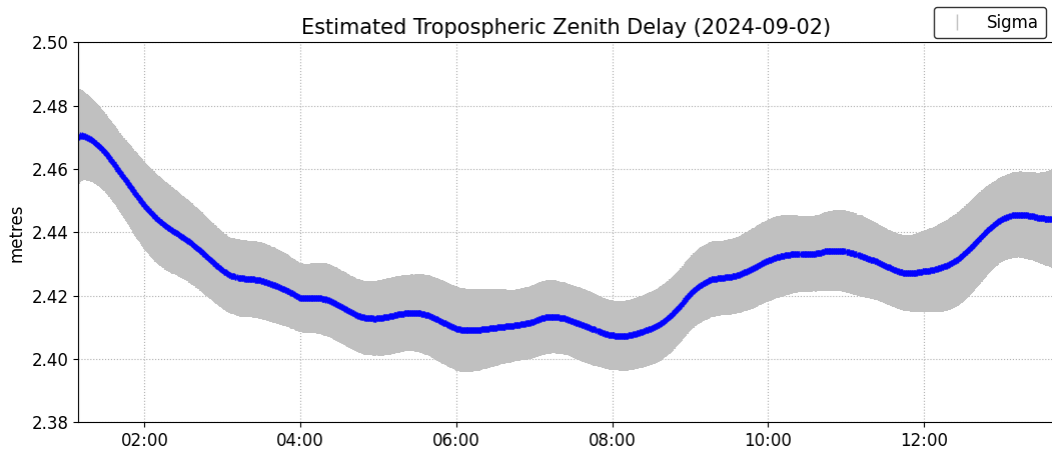
*(Coordinates from RINEX header used as a priori position)

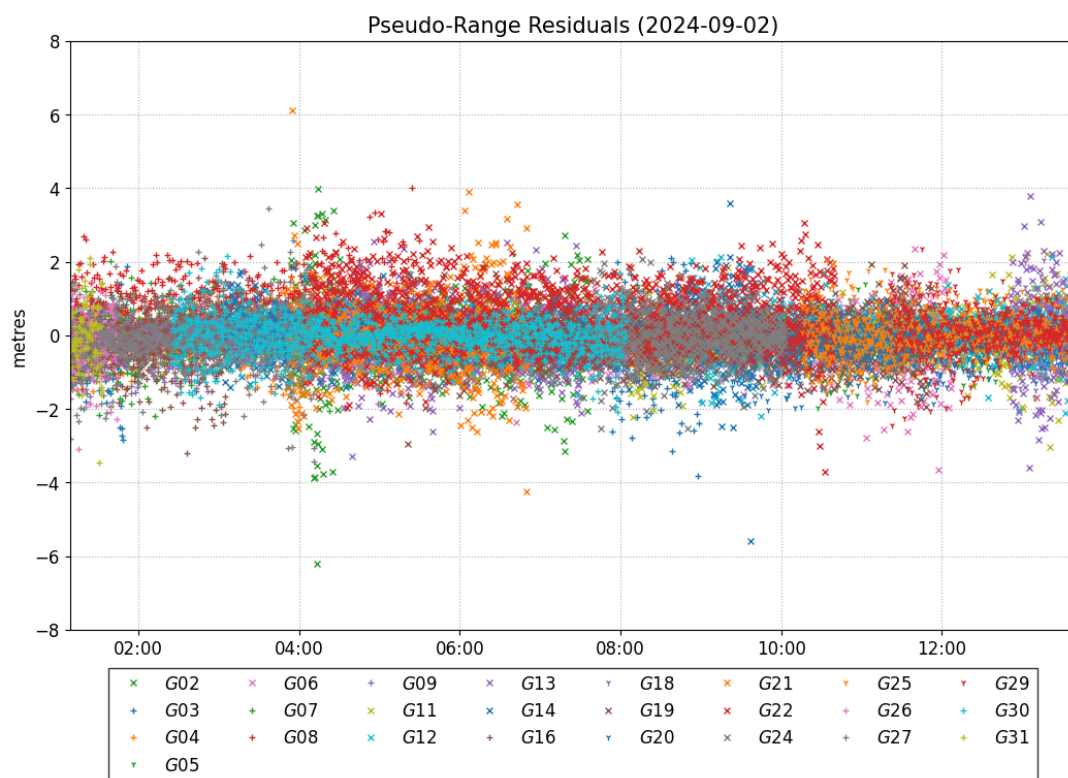
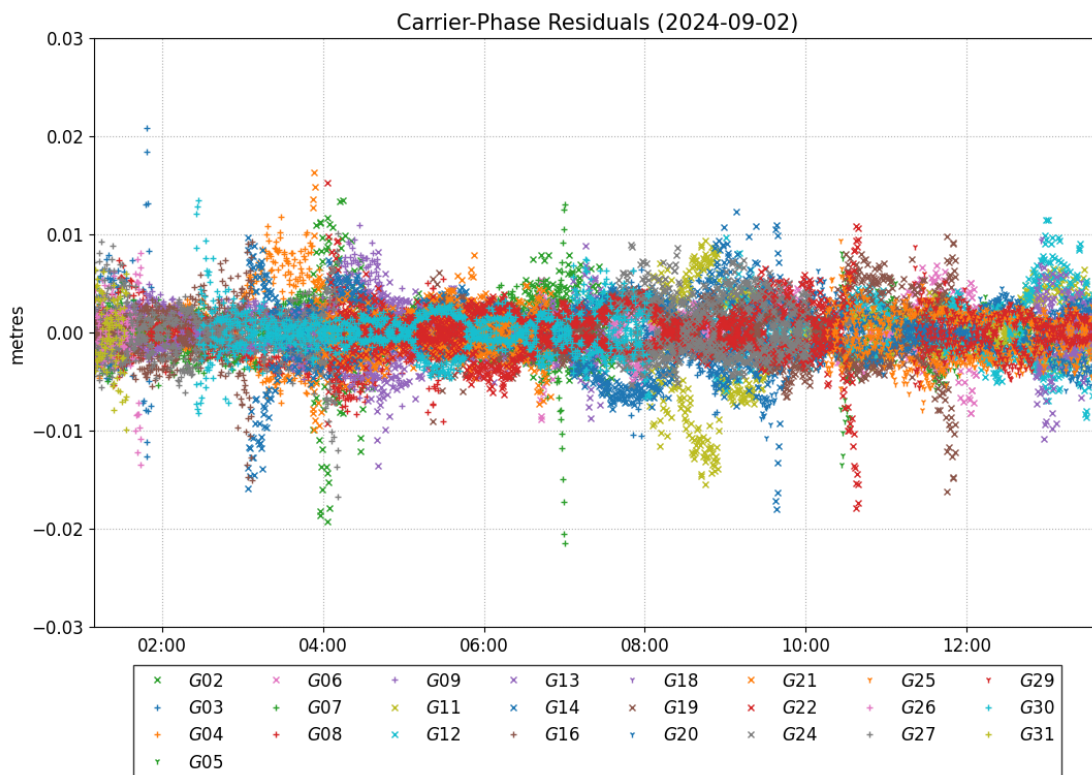
Satellite Sky Distribution



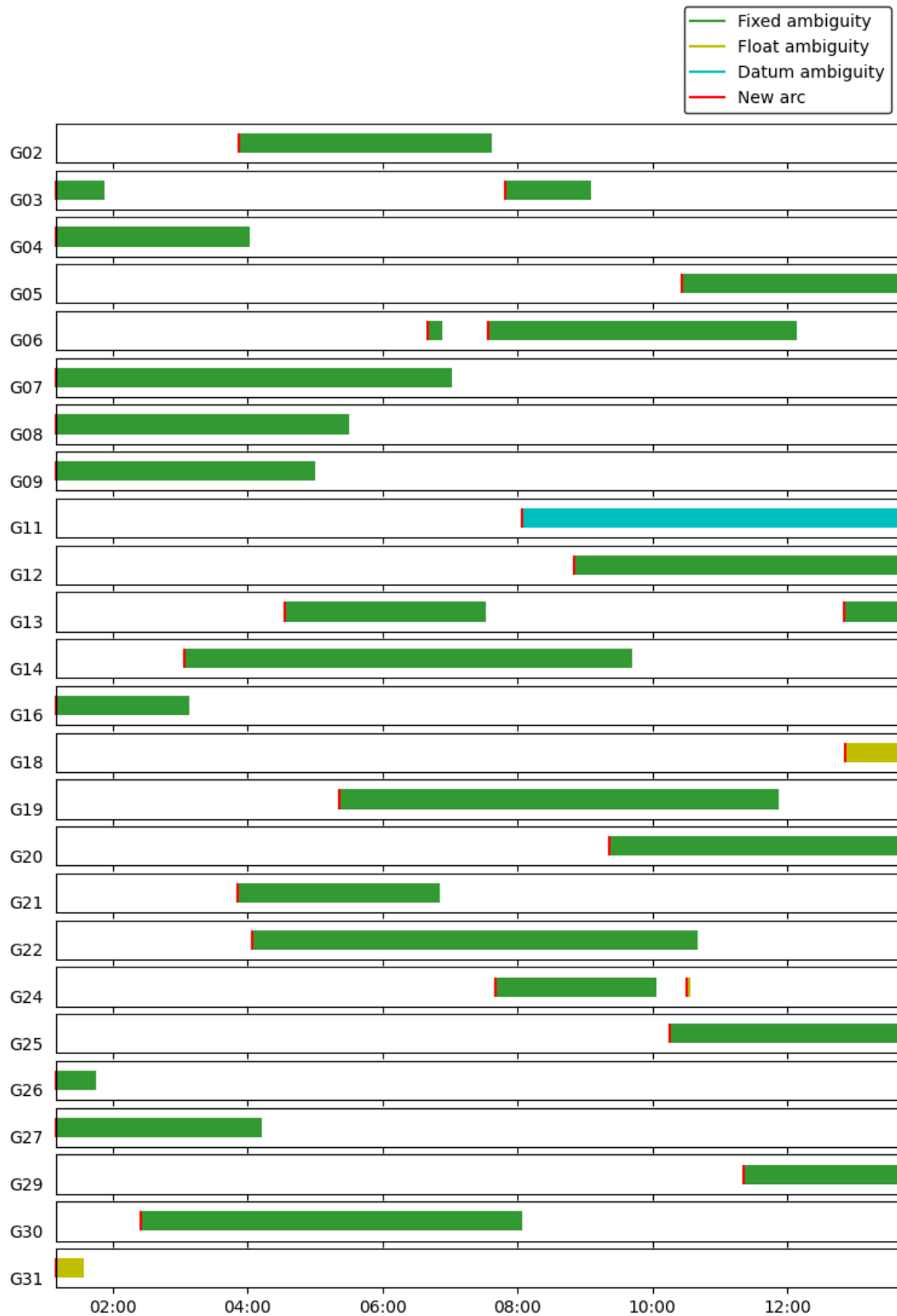
× G02	+ G07	× G12	× G18	× G22	+ G27
+ G03	+ G08	× G13	× G19	× G24	× G29
+ G04	+ G09	× G14	× G20	× G25	+ G30
× G05	× G11	+ G16	× G21	+ G26	+ G31
× G06					

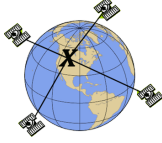






Phase Ambiguity Status (2024-09-02)





CSRS-PPP 3.54.2 (2022-11-10)



B____B05_corrected.obs

Data Start	Data End	Duration of Observations
2024-09-05 01:11:00.00	2024-09-06 03:10:30.00	1 day, 1:59:30
Processing Time		Product Type
12:10:46 UTC 2024/09/25		NRCan/IGS Final
Observations	Frequency	Mode
Phase and Code	Double	Static
Elevation Cut-Off	Rejected Epochs	Fixed Ambiguities
7.5 degrees	0.19 %	98.57 %
		Estimation Steps
		30.00 sec
Antenna Model	APC to ARP	ARP to Marker
Unknown	Unknown	H:0.000m / E:0.000m / N:0.000m

(APC = antenna phase center; ARP = antenna reference point)

Estimated Position for B____B05_corrected.obs

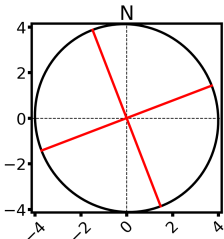
	Latitude (+n)	Longitude (+e)	Ell. Height
ITRF20 (2024.7)	35° 18' 36.25309"	-97° 2' 0.45979"	310.119 m
Sigmas(95%)	0.003 m	0.003 m	0.014 m
A priori*	35° 18' 36.30314"	-97° 2' 0.47469"	328.035 m
Estimated – A priori	-1.542 m	0.376 m	-17.916 m

Orthometric Height
CGVD2013
(CGG2013a)
(2024.7)

337.788 m

(click for height reference information)

95% Error Ellipse (mm)
semi-major: 4 mm
semi-minor: 4 mm
semi-major azimuth: -20° 0' 20.91"



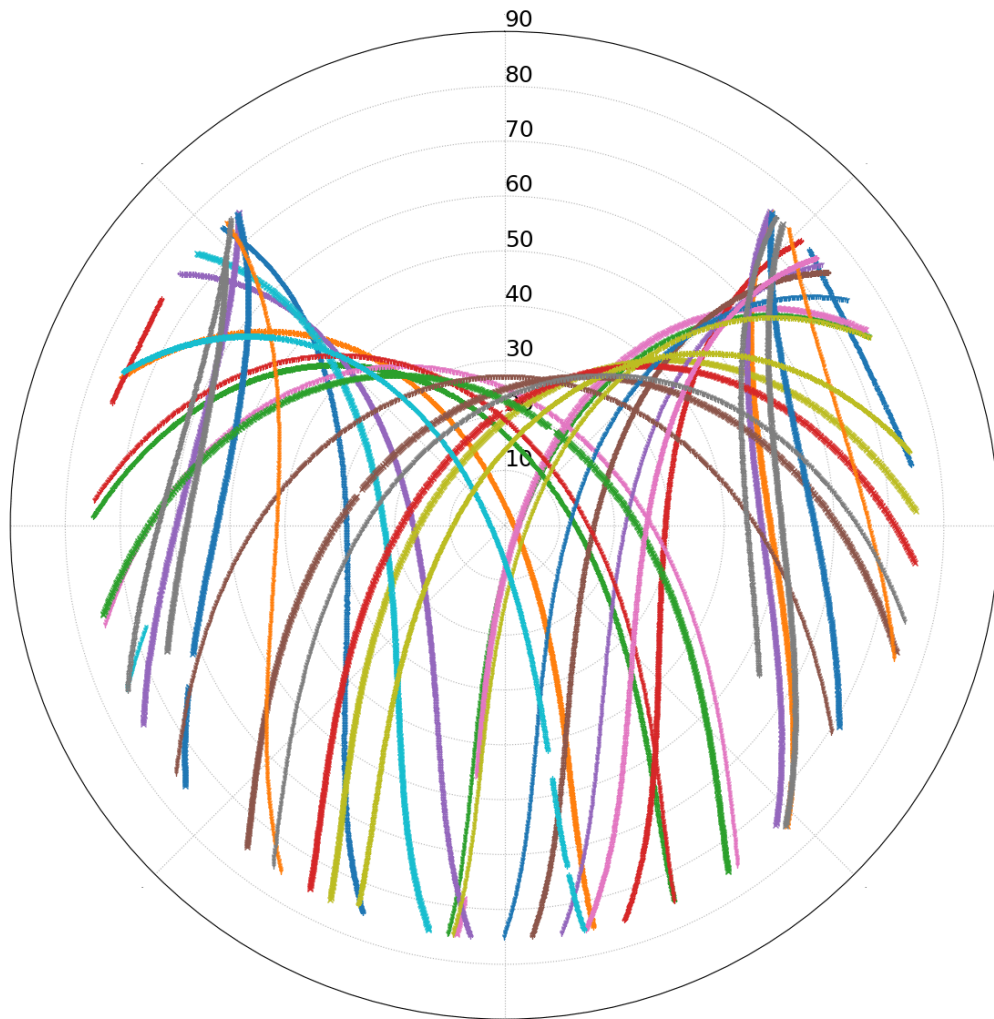
UTM (North)
Zone 14

3909203.356 m (N)
678782.114 m (E)

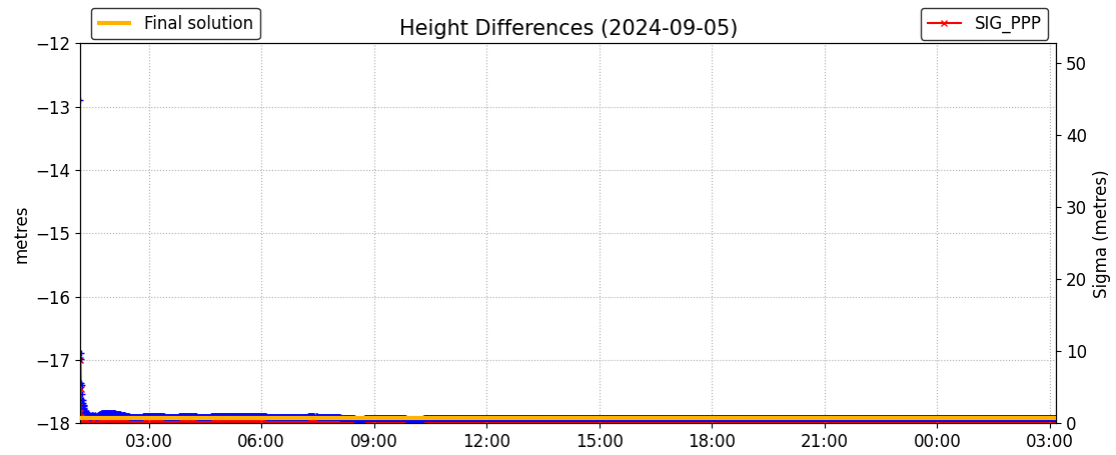
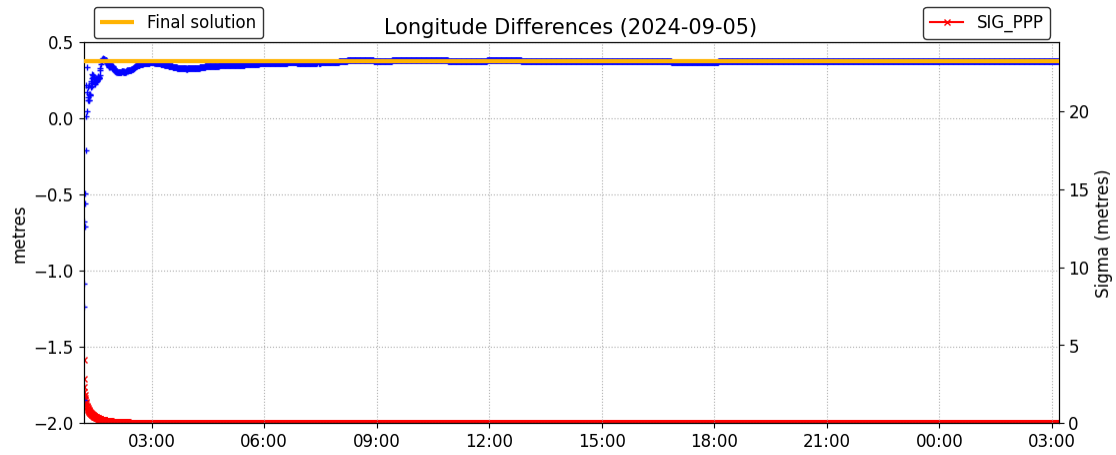
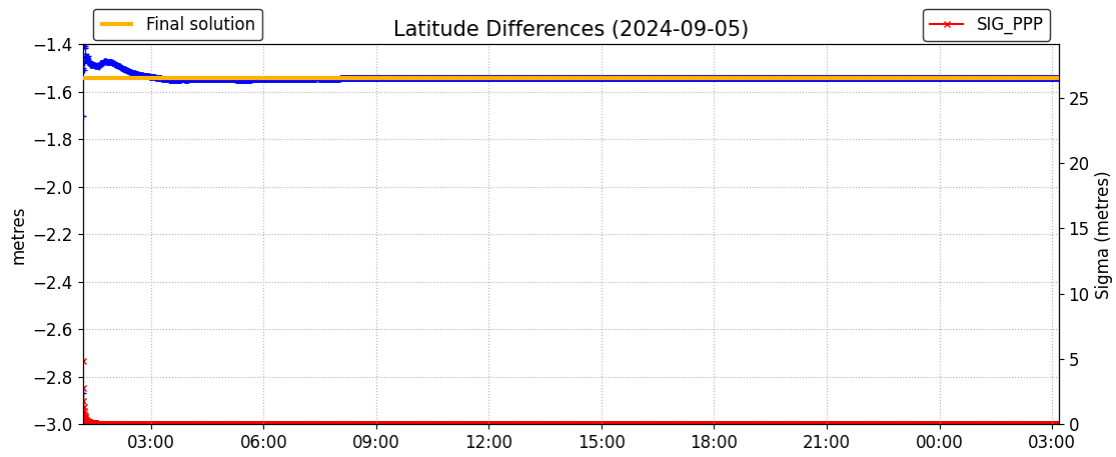
Scale Factors
0.99999392 (point)
0.99994524 (combined)

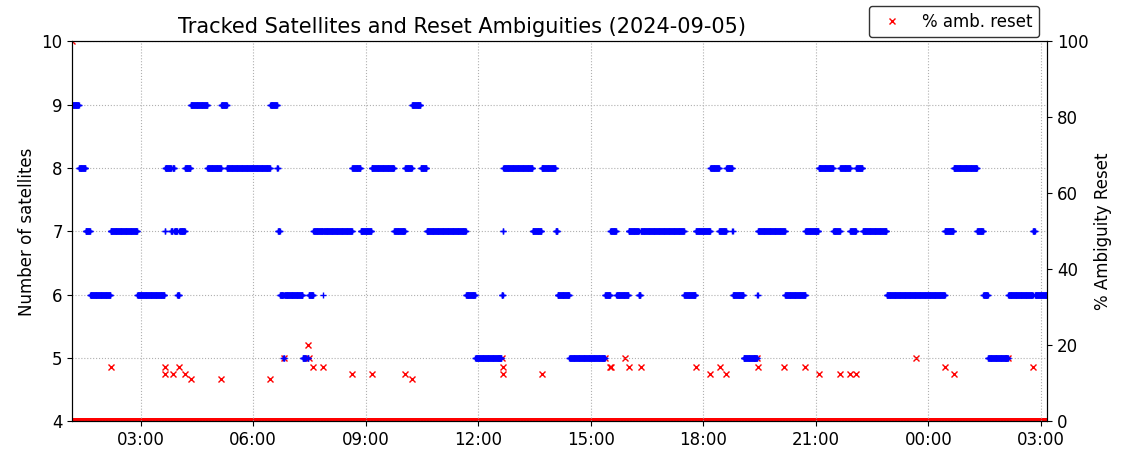
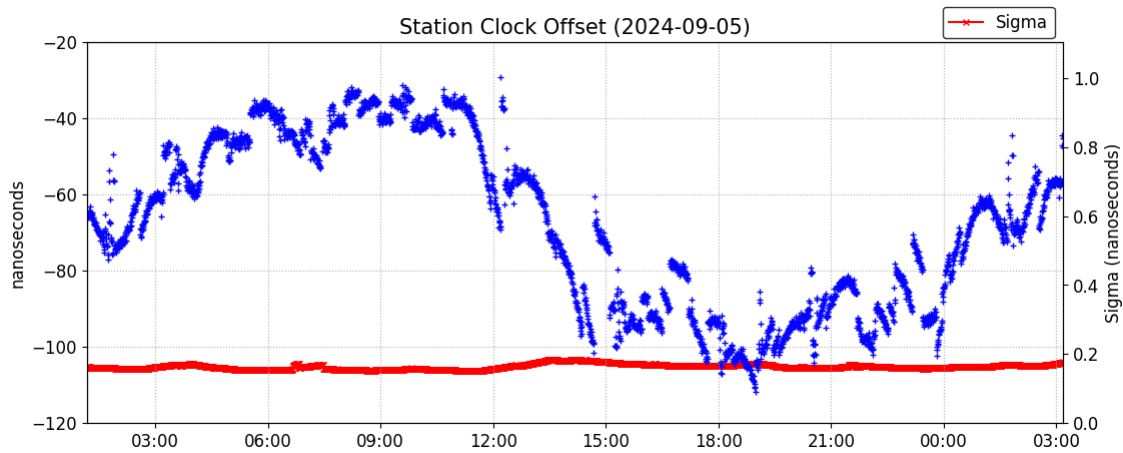
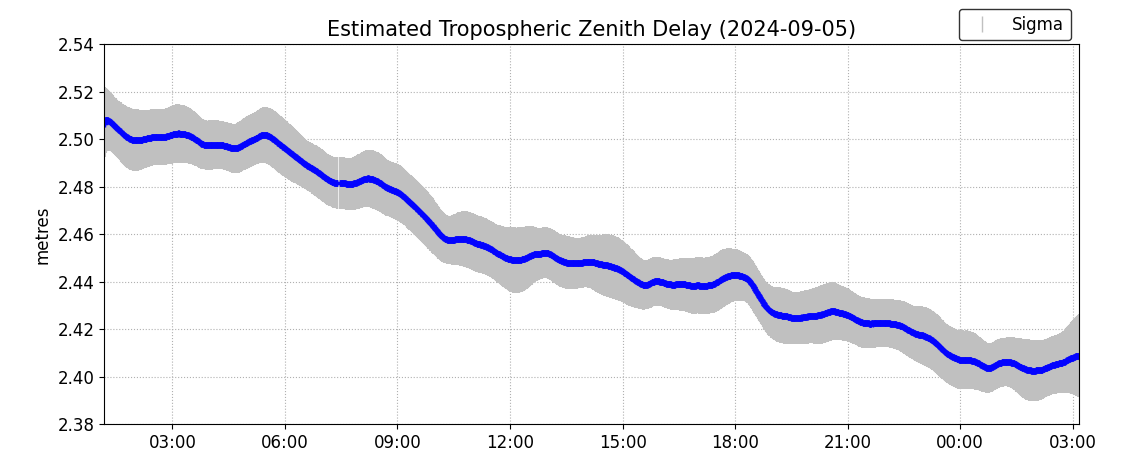
*(Coordinates from RINEX header used as a priori position)

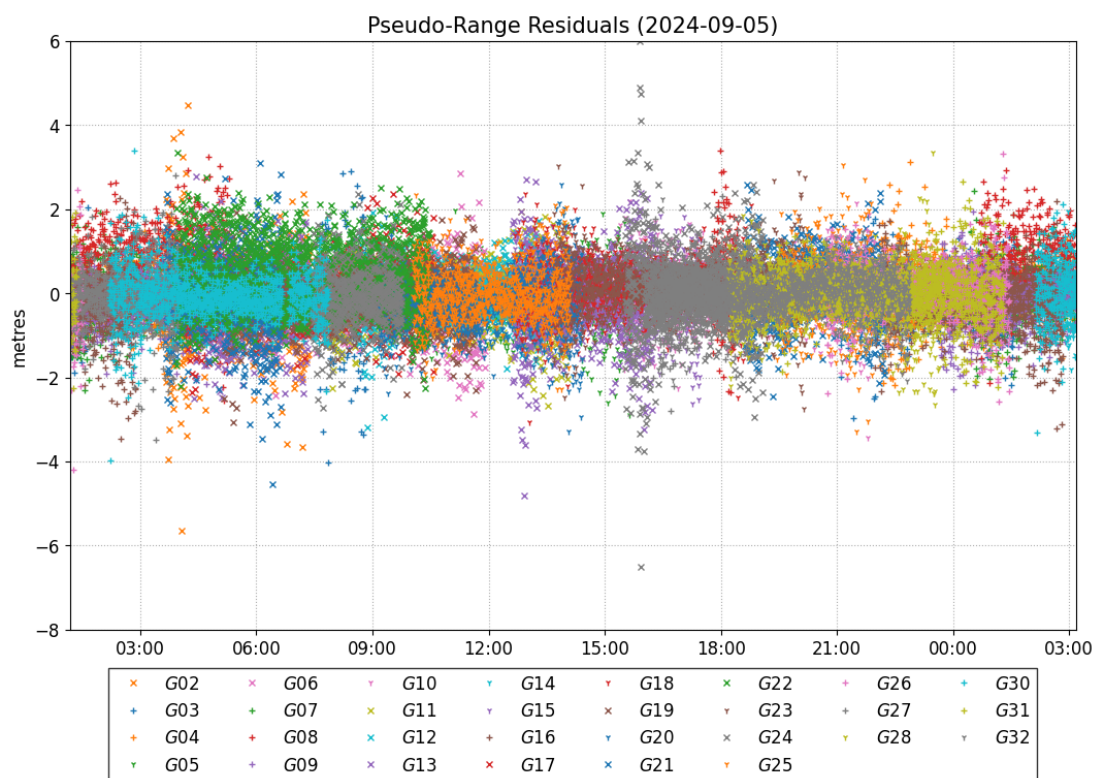
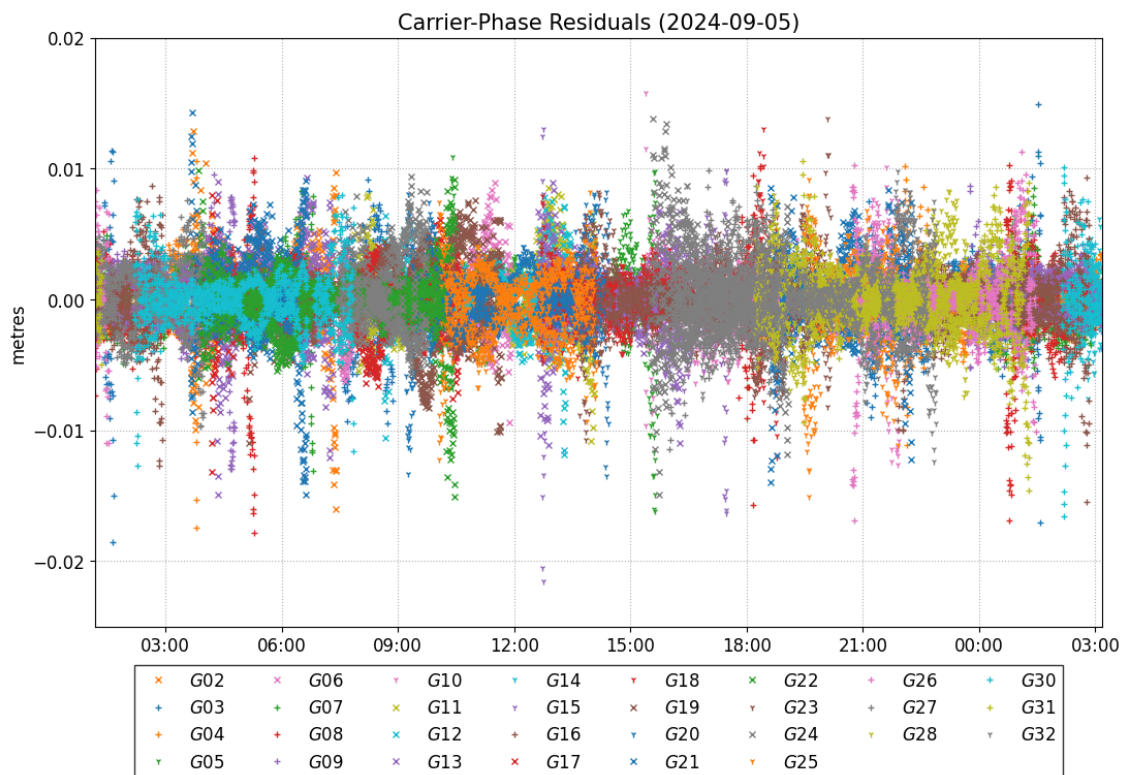
Satellite Sky Distribution



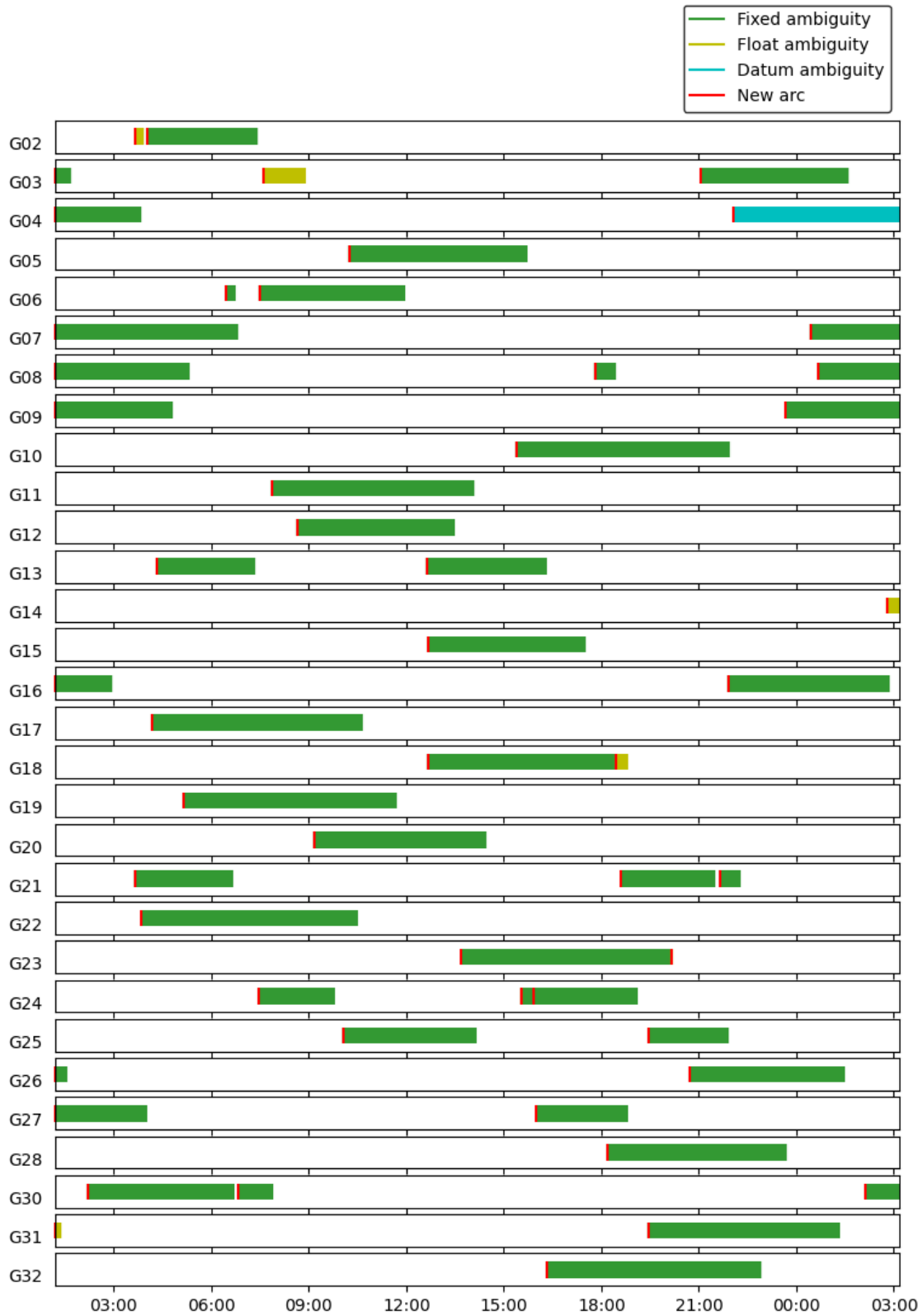
×	G02	+	G07	×	G12	×	G17	×	G22	+	G27
+	G03	+	G08	×	G13	×	G18	×	G23	×	G28
+	G04	+	G09	×	G14	×	G19	×	G24	+	G30
×	G05	×	G10	×	G15	×	G20	×	G25	+	G31
×	G06	×	G11	+	G16	×	G21	×	G26	×	G32

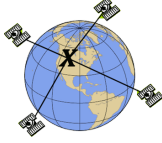






Phase Ambiguity Status (2024-09-05)





CSRS-PPP 3.54.2 (2022-11-10)



2024_9_1_Base_Survey_In.obs

Data Start	Data End	Duration of Observations
2024-09-02 01:32:22.00	2024-09-02 13:43:52.00	12:11:30
Processing Time		Product Type
12:07:43 UTC 2024/09/04		NRCan Rapid
Observations	Frequency	Mode
Phase and Code	Double	Static
Elevation Cut-Off	Rejected Epochs	Fixed Ambiguities
7.5 degrees	0.00 %	90.80 %
		Estimation Steps
		1.00 sec
Antenna Model	APC to ARP	ARP to Marker
Unknown	Unknown	H:0.000m / E:0.000m / N:0.000m

(APC = antenna phase center; ARP = antenna reference point)

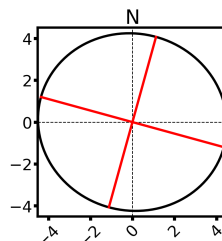
Estimated Position for 2024_9_1_Base_Survey_In.obs

	Latitude (+n)	Longitude (+e)	Ell. Height
ITRF20 (2024.7)	35° 18' 36.25377"	-97° 2' 0.46140"	310.131 m
Sigmas(95%)	0.003 m	0.004 m	0.014 m
A priori*	35° 18' 36.28364"	-97° 2' 0.46341"	311.233 m
Estimated – A priori	-0.921 m	0.051 m	-1.102 m

Orthometric Height
CGVD2013
(CGG2013a)
(2024.7)

337.800 m
(click for height reference information)

95% Error Ellipse (mm)
semi-major: 5 mm
semi-minor: 4 mm
semi-major azimuth: -74° 24' 10.9"

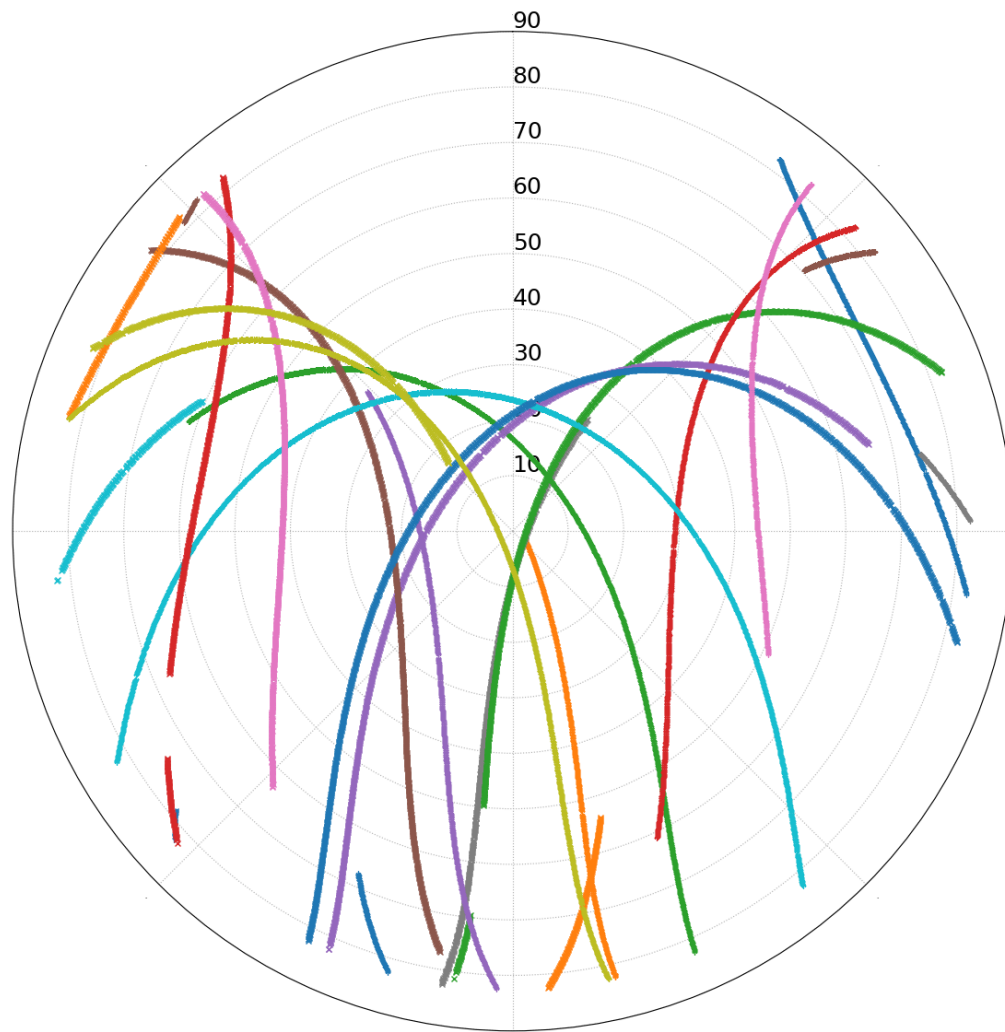


UTM (North)
Zone 14

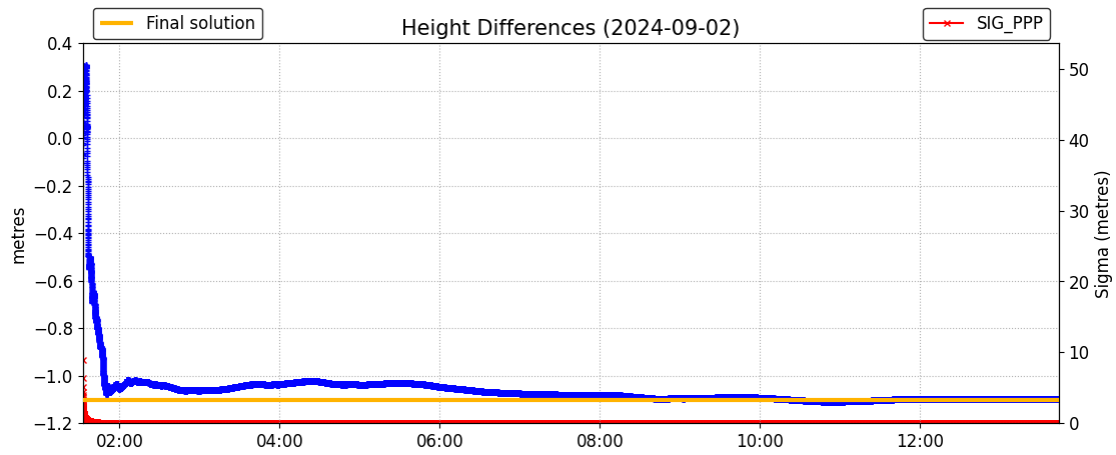
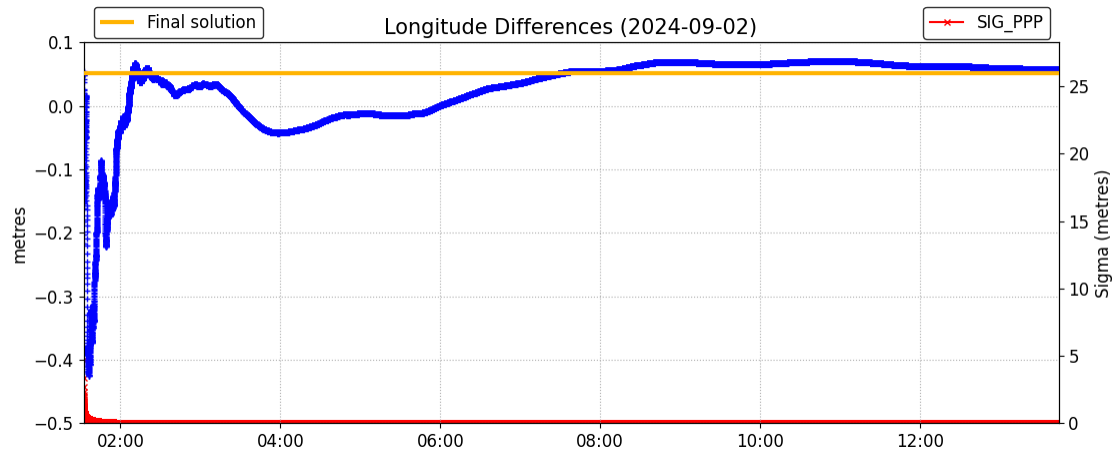
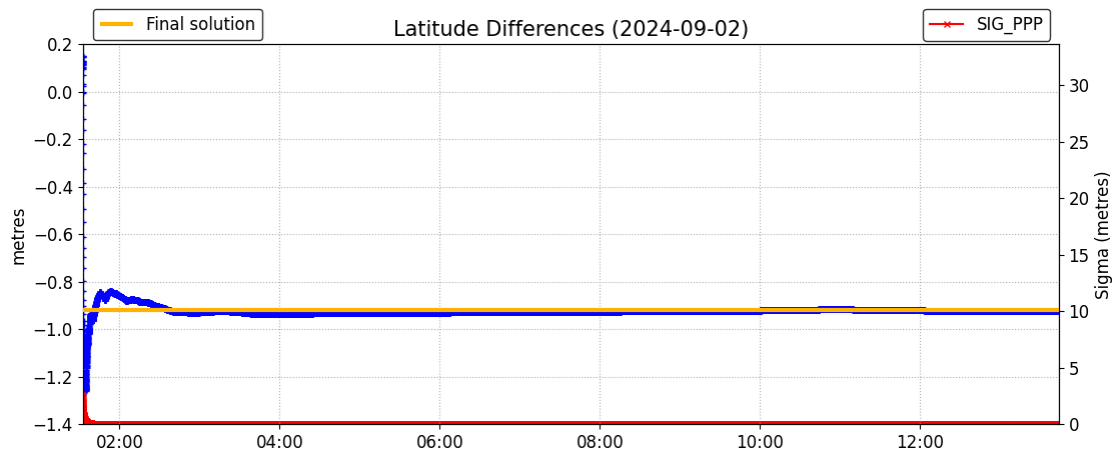
3909203.377 m (N)
678782.073 m (E)
Scale Factors
0.99999392 (point)
0.99994524 (combined)

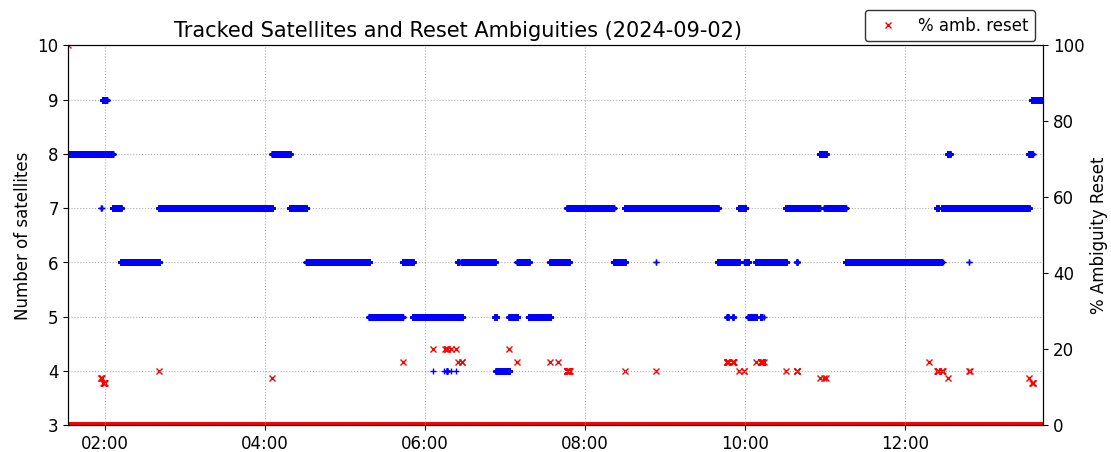
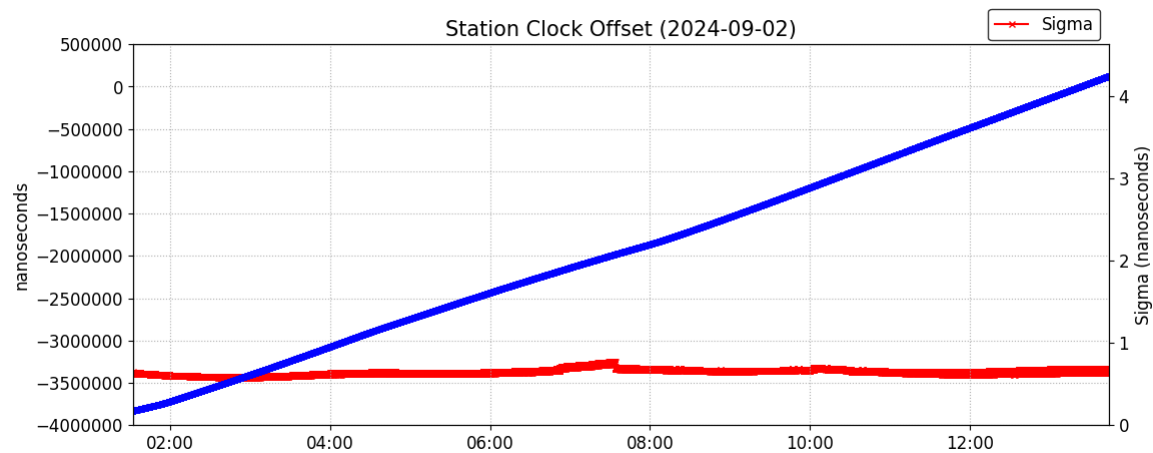
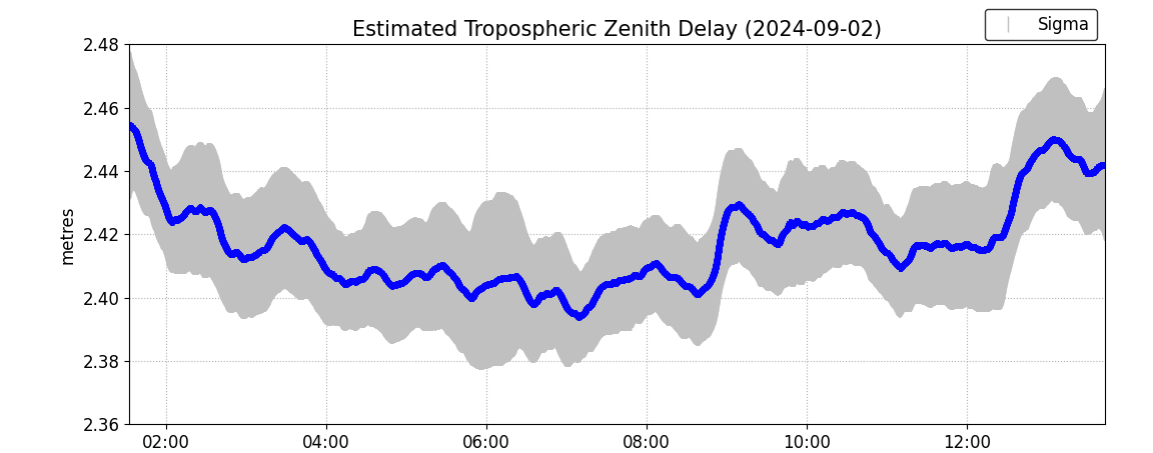
*(Coordinates from a code solution used as a priori position)

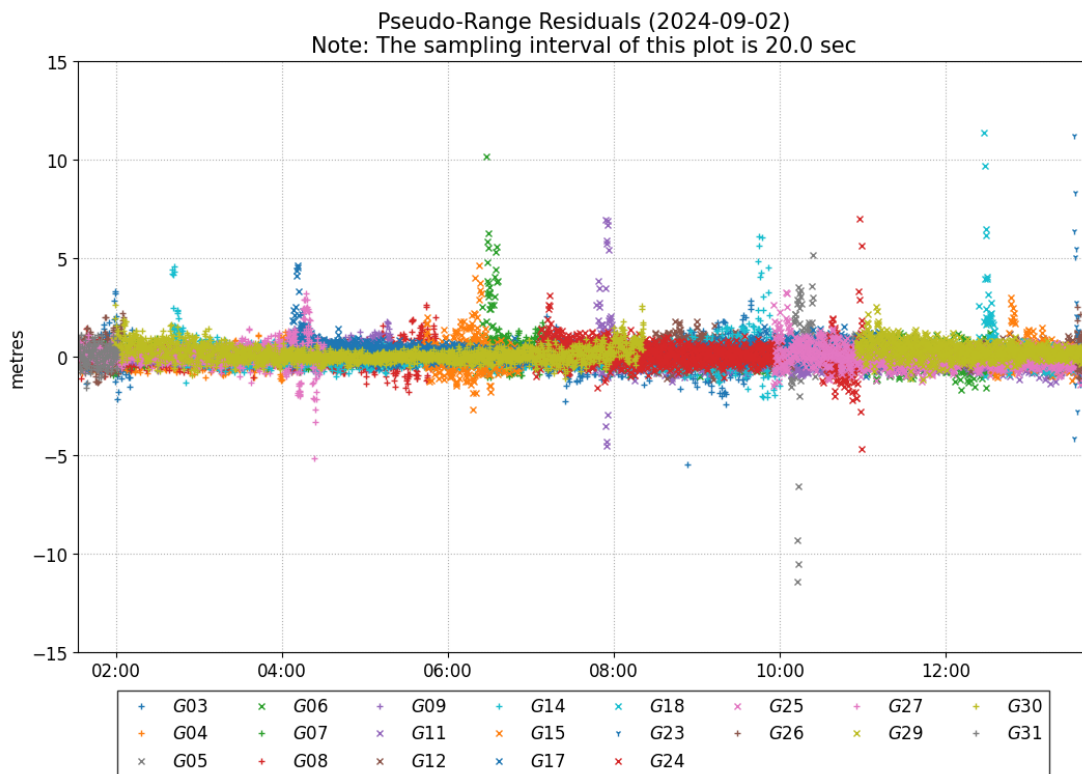
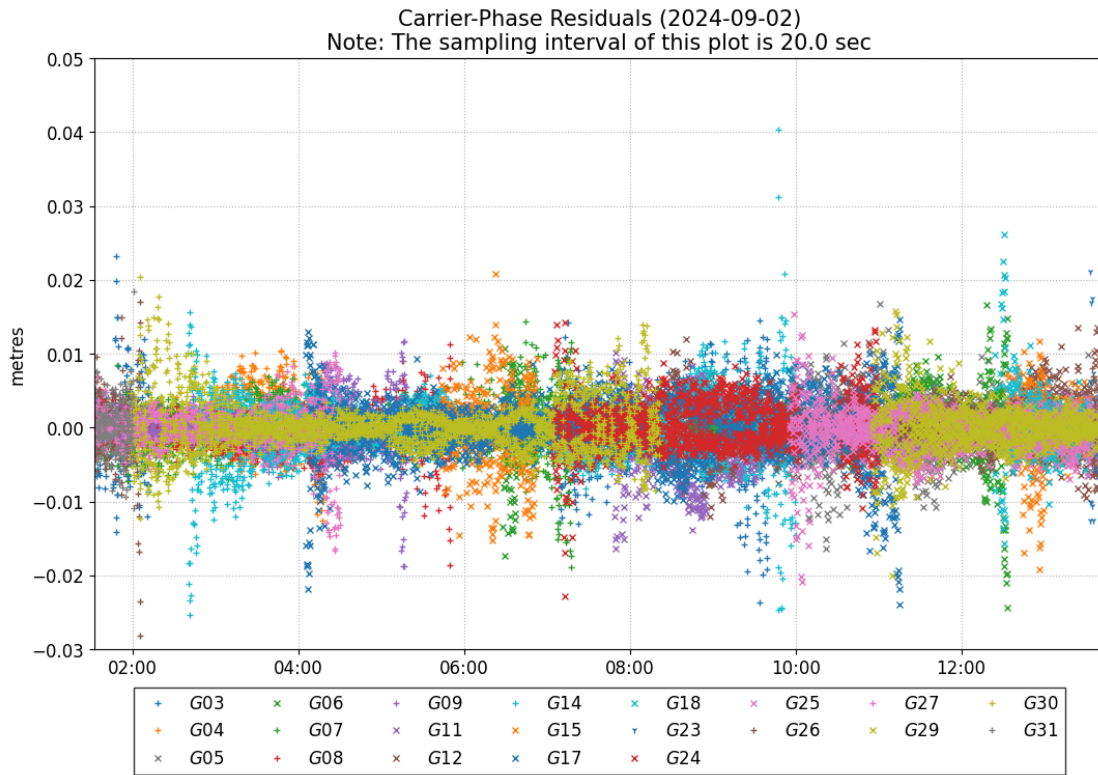
Satellite Sky Distribution
 Note: The sampling interval of this plot is 20.0 sec



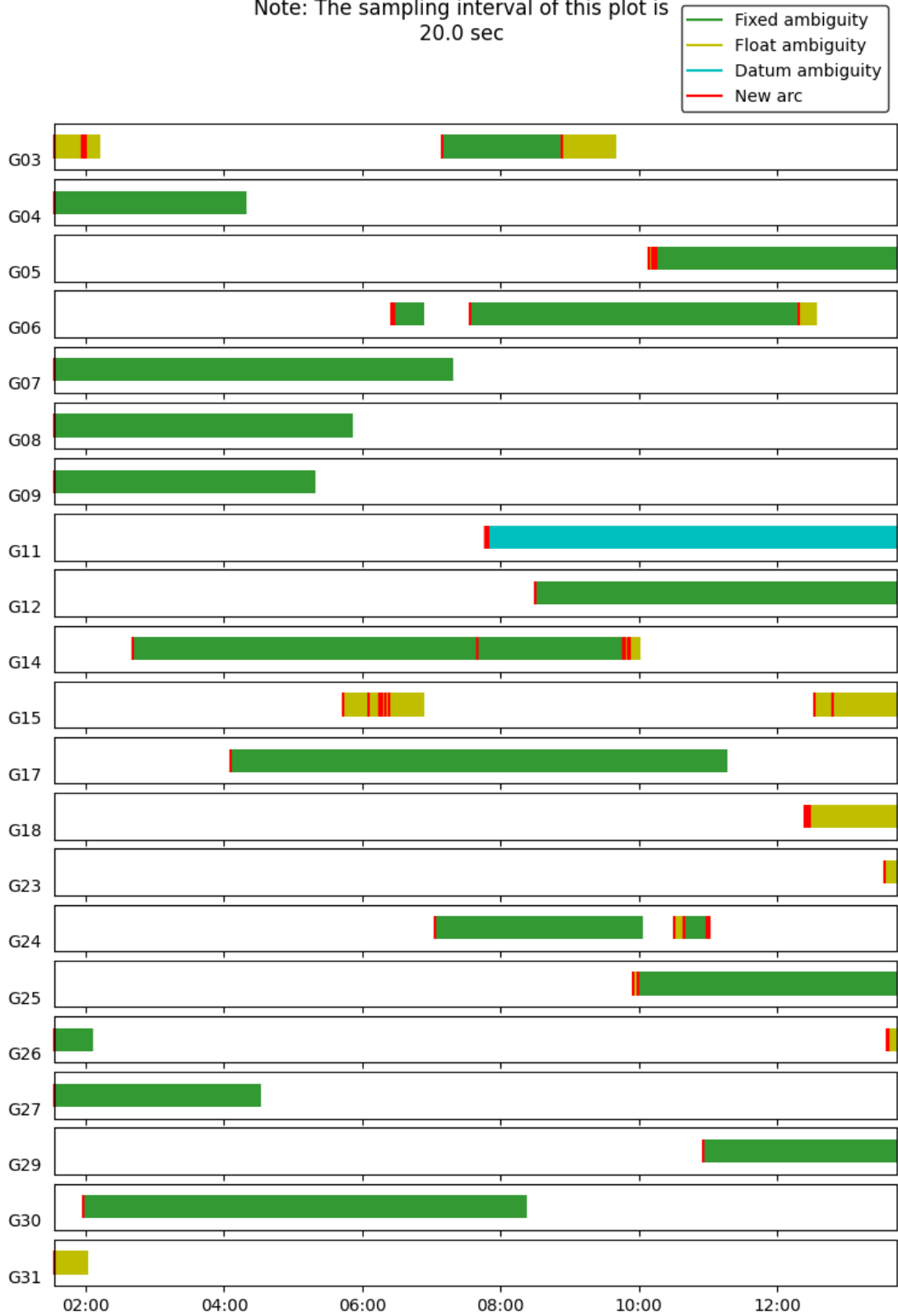
+	G03	+	G07	×	G12	×	G18	×	G25	×	G29
+	G04	+	G08	×	G14	×	G23	+	G26	+	G30
×	G05	+	G09	×	G15	×	G24	+	G27	+	G31
×	G06	×	G11	×	G17						

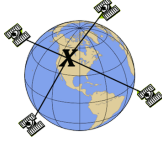






Phase Ambiguity Status (2024-09-02)
 Note: The sampling interval of this plot is 20.0 sec





CSRS-PPP 3.54.2 (2022-11-10)



2024_9_4_Base_Survey_In.obs

Data Start 2024-09-05 06:41:00.00	Data End 2024-09-06 01:30:00.00	Duration of Observations 18:49:00
Processing Time 19:48:59 UTC 2024/09/08		Product Type NRCan Rapid
Observations Phase and Code	Frequency Double	Mode Static
Elevation Cut-Off 7.5 degrees	Rejected Epochs 0.00 %	Fixed Ambiguities 85.59 %
		Estimation Steps 30.00 sec
Antenna Model Unknown	APC to ARP Unknown	ARP to Marker H:0.000m / E:0.000m / N:0.000m

(APC = antenna phase center; ARP = antenna reference point)

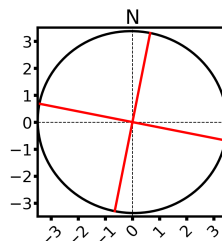
Estimated Position for 2024_9_4_Base_Survey_In.obs

	Latitude (+n)	Longitude (+e)	Ell. Height
ITRF20 (2024.7)	35° 18' 36.25323"	-97° 2' 0.45987"	310.171 m
Sigmas(95%)	0.003 m	0.003 m	0.010 m
A priori*	35° 18' 36.29299"	-97° 2' 0.46835"	308.231 m
Estimated – A priori	-1.225 m	0.214 m	1.940 m

Orthometric Height
CGVD2013
(CGG2013a)
(2024.7)

337.840 m
(click for height reference
information)

95% Error Ellipse (mm)
semi-major: 4 mm
semi-minor: 3 mm
semi-major azimuth: -78° 12' 17.05"

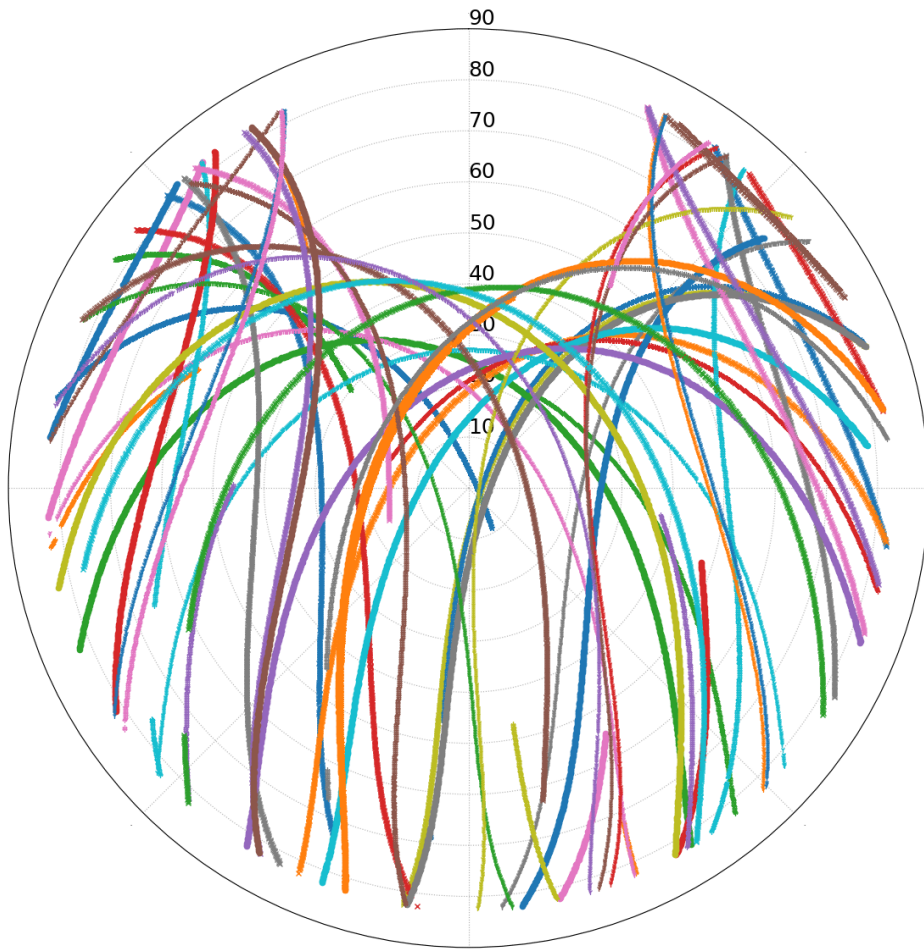


UTM (North)
Zone 14

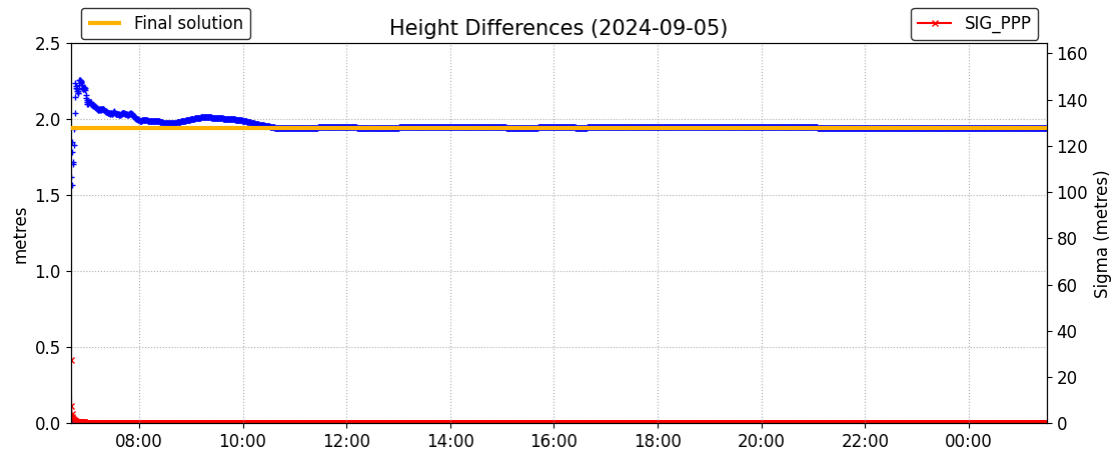
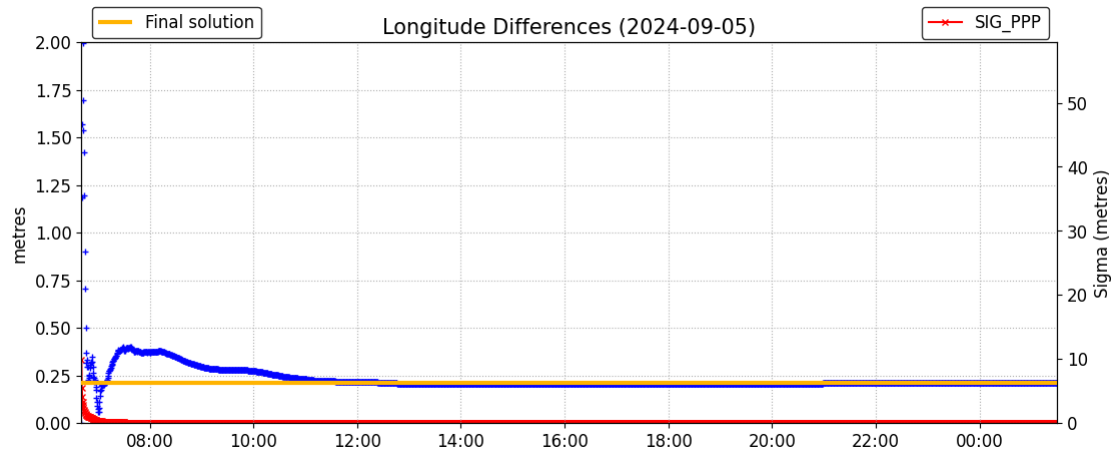
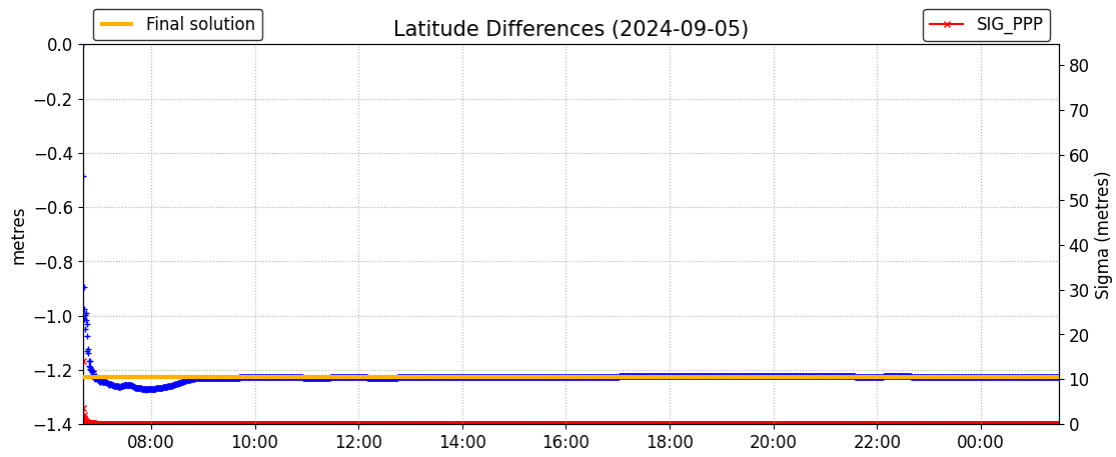
3909203.361 m (N)
678782.112 m (E)
Scale Factors
0.99999392 (point)
0.99994523 (combined)

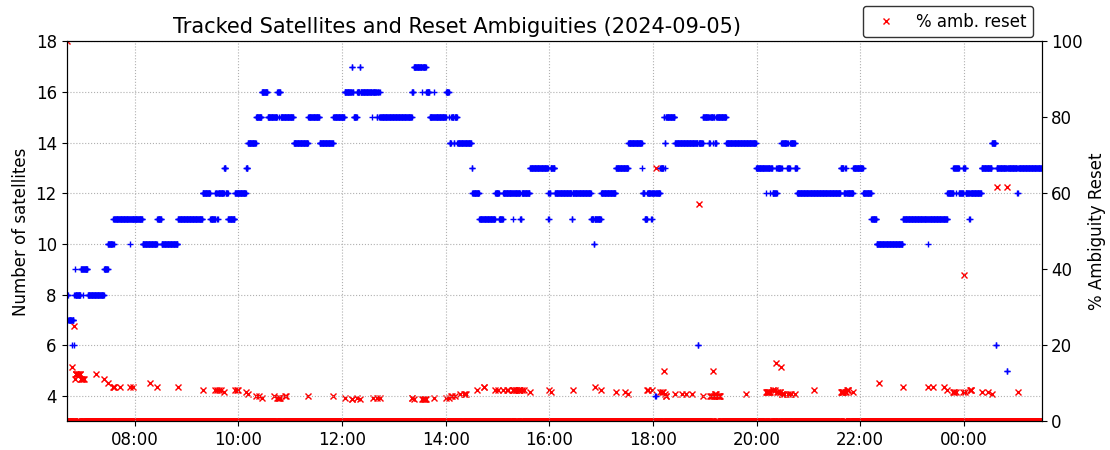
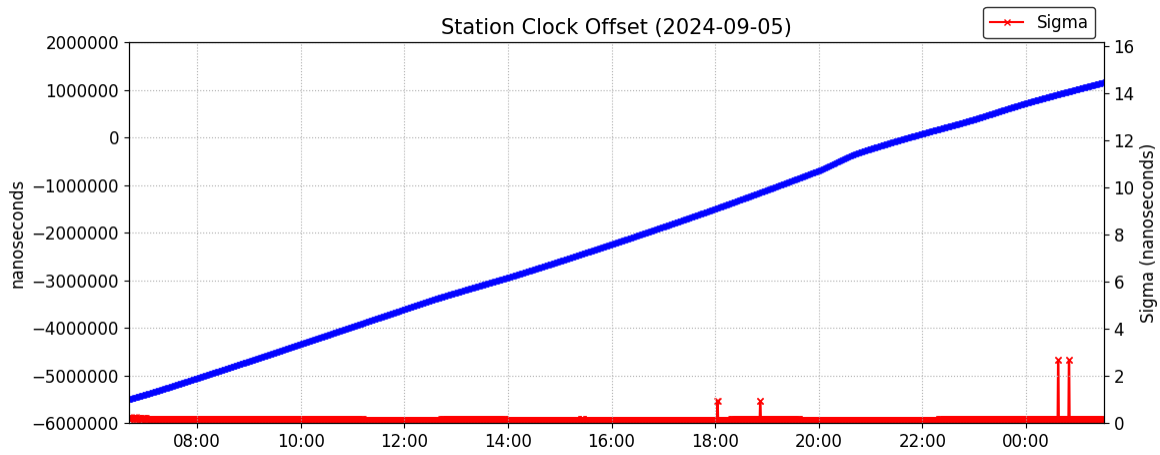
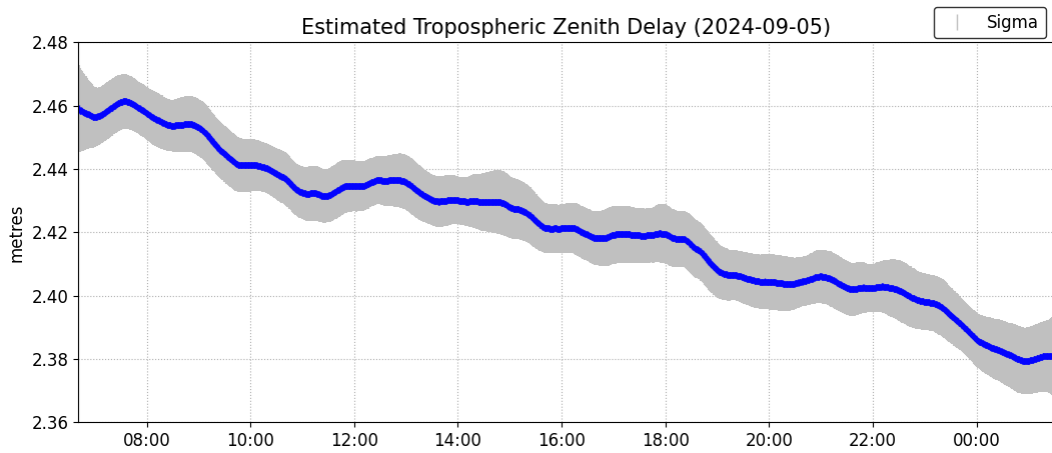
*(Coordinates from a code solution used as a priori position)

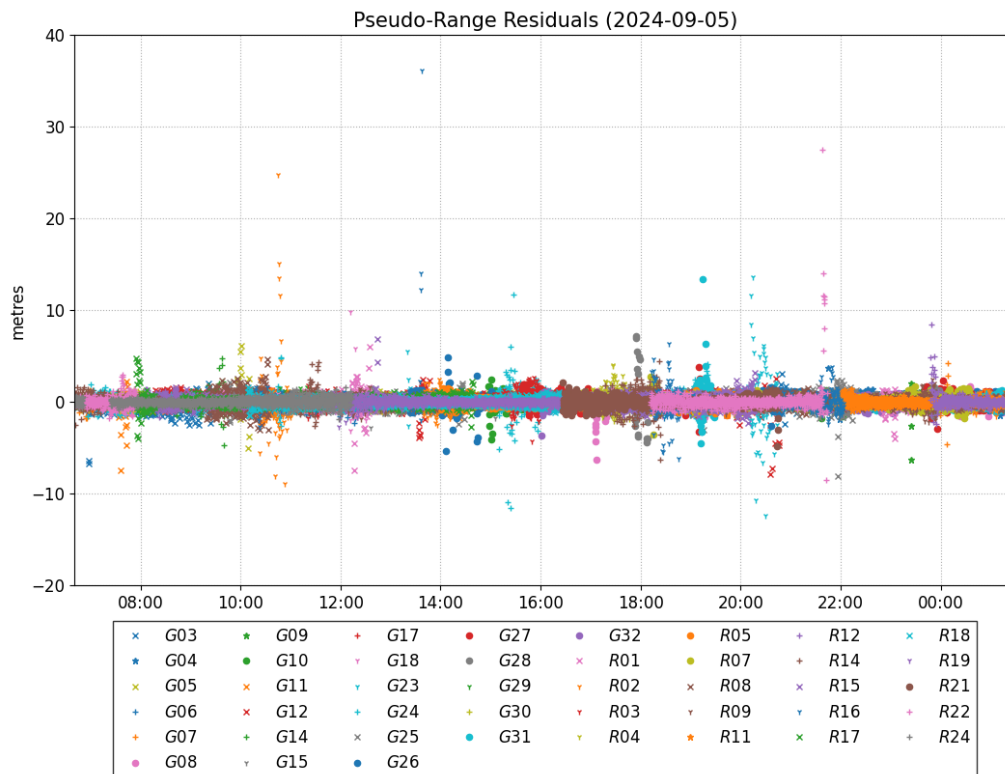
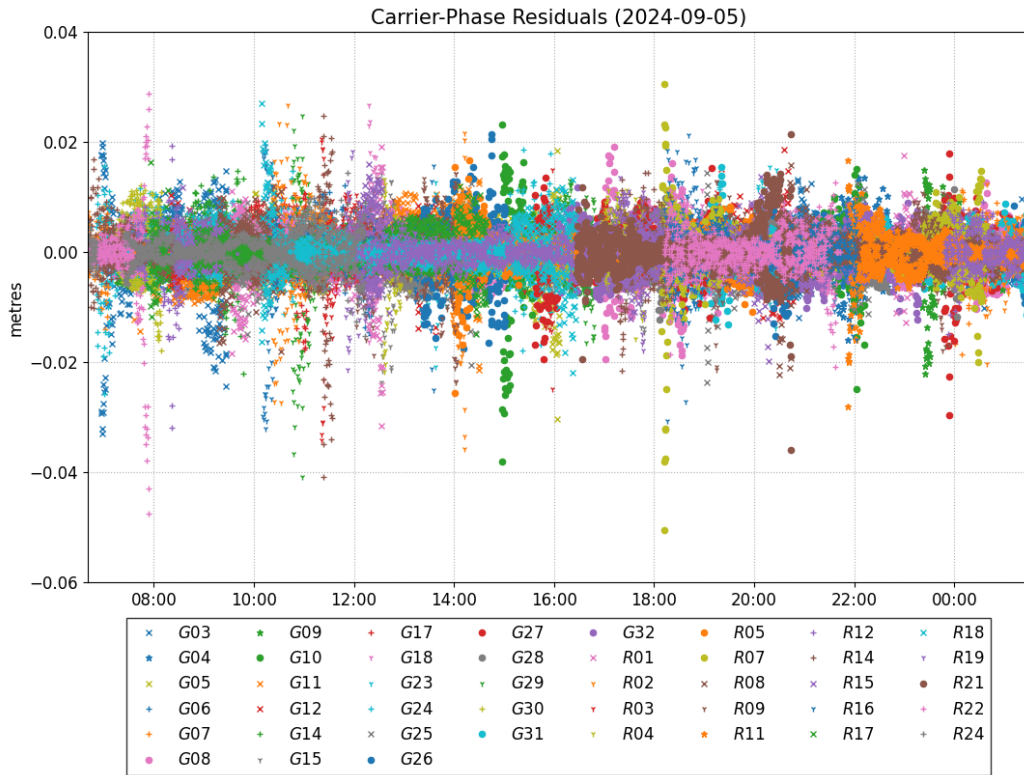
Satellite Sky Distribution



× G03	× G11	+ G24	• G31	• R07	• R16
• G04	× G12	× G25	• G32	× R08	× R17
× G05	+ G14	• G26	× R01	• R09	× R18
+ G06	• G15	• G27	• R02	• R11	• R19
+ G07	+ G17	• G28	• R03	+ R12	• R21
• G08	• G18	• G29	• R04	+ R14	• R22
• G09	• G23	• G30	• R05	× R15	+ R24
• G10					







Phase Ambiguity Status (2024-09-05)

