Sub-Metre Position Accuracy on a Smartphone

Team E

ENG 4000

Jared Yen
Ryan Nguyen
Melvin Gagarao
Nathan Stachow

Executive Summary

Global Navigation Satellites Systems (GNSS) have transformed positioning and navigation. From land surveying to augmented reality for gaming, GNSS is used on a daily basis and is relied on in many sectors of society. In particular, smartphones have been increasingly used for positioning based on GNSS measurements. However, the current GNSS receivers on smartphones can only determine positional accuracy at the 10s of metres and can be noisy.

In the project, the team focused on improving the positional accuracy of smartphones to the metre level based on the GNSS measurements recorded on the devices. The key objective of the project was to reach metre level positional accuracy. While doing so, the solution had to be able to run on an I5 processing system with a 64 GB memory. The software had to run successfully on MATLAB processing software and comply with Canadian laws regarding controlled goods.

The group developed a relative augmentation positioning algorithm in order to process GNSS results and improve their accuracy. By increasing the number of measurements in the estimation process, adding additional environmental corrections, and removing residual outliers, the objective of determining positions at the metre level of accuracy was achieved. While the estimated positional accuracies were not the decimetre level, the smartphone user would still be able to have superior accuracy to what was previously available. Furthermore, the developed algorithm had the functionality to just use the current GNSS measurements without any additional hardware components or system remodelling. As a result, the algorithm could be used on current smartphones without redeveloping the smartphone hardware configuration.

Increased positional accuracy resulting from the relative augmentation positioning algorithm has great potential. Better positioning estimation can improve navigation estimation and end up saving a smartphone user time. Likewise, higher accuracy positioning based on smartphone data can advance augmented reality in gaming such as Pokemon Go. Furthermore, increased smartphone positioning accuracy can improve navigation for individuals with physical disabilities. An individual in a wheelchair can be navigated more accurately to his/her destination. Therefore, there are many uses for increased smartphone accuracy in society.

As the solution developed was an algorithm, there is a great potential for additions to be made to improve positional estimation. Map matching can be incorporated into the solution to shift points that are off of main features such as a road back to the right location. Likewise, further global GNSS constellation measurements such as BeiDou and GLONASS can be added into the processing to increase the redundancy in the system. Errors could also be improved through modelling phase ambiguities and introducing real-time kinematic (RTK) processing. While the current solution could be improved with additional processing, the relative positioning augmentation algorithm proved to be an effective means of estimating smartphone positions.

1 Preliminary Business Use Case

Although the project was developed with use of smartphone equipment, it has applications that can be applied to any low-grade GNSS receivers. Doing so would allow for better positioning accuracy for products that use smaller receivers. Furthermore, such receivers would be more efficient based on the dollar cost. In the paragraphs below, cases were identified for business uses for high accuracy smartphone positioning in the market.

1.1 Spatial Awareness for Autonomous Driving

By having decimetre level accuracy positioning for smartphones, the autonomous driving sector would be transformed. Users would be able to determine where a vehicle is on the road. Likewise, the smartphone's positions can be used in reference to other vehicles and features on or along roadways. High accuracy GNSS positioning would also enable autonomous vehicles to remain in their lane and accurately back into parking spaces. While there are potential solutions such as LiDAR already in the market, such technology is quite expensive. The majority of drivers already have smartphones. As a result, improving the GNSS positioning capabilities of the smartphones would reduce the cost of an autonomous vehicle as the precision would come from the phone rather than expensive hardware on the car. Since this is the case, improving GNSS positioning accuracy of smartphones would have a very positive impact on the implementation of autonomous vehicles.

1.2 Navigation Guide For Wheelchair Accessibility And Special Needs

Individuals with special needs can sometimes have to rely on technology or other individuals for navigation purposes. Improving smartphone accuracy can help such individuals find their way around a city. A smartphone application can guide users to different locations such as a wheelchair access, washroom locations, and other points of interest on their phone. Meanwhile, individuals with visual impairments can listen to their phones for such directions. Increased accuracy for smartphones would also aid in autonomous driving of motorised wheelchairs. Likewise, individuals with wheelchairs could also have more accurate identification of features of interest.

In general, GNSS signals tend to be noisier in areas surrounded by tall buildings due to signal blockage. As a result, the GNSS signals suffer from multipath and can potentially mislead a user about the location of a point of interest. Nevertheless, a combination of GNSS measurements with Bluetooth and wifi services can help increase the quality of the signal in urban areas. Therefore, improved positioning estimation using GNSS measurements on smartphones can improve the quality of life and experience of individuals with special needs or disabilities.

1.3 Augmented Reality Smartphone Applications

Augmented reality has been extensively used through smartphone applications. Pokemon Go is a popular augmented reality game that served as a pioneer for many more games to come. Outside of gaming, there have been augmented reality experiments for navigation to help with travelling in between tall buildings. Navigation can be improved with the help of greater specificity to small detail, especially for walking traverses. A great example of such a project in use is through a video found on LinkedIn by Dent Reality (https://www.linkedin.com/feed/update/urn:li:activity:6904832744111886336/). In the video, the application is discussed and the importance of the connection between augmented reality and high accuracy positioning is also identified. When raising the phone in camera mode in the application, an overlay for directions is shown over the camera lens. After the phone is lowered in reading mode, a compass to the destination is included. Such additions can be integrated into Google Maps API to have the same delivery for road navigation.

1.4 Greater Detail in the Entertainment Industry

The entertainment market is always expanding year over year with films and games becoming more refined and immersive in detail. Going along such a trajectory, it appears that greater detail will be added into the entertainment industry based on the real world. Companies like Industrial Pixel have helped create many scenes in films such as The Boys, Captain Marvel, The Revenant, and many more (Industrialpixelvfx.com). Many scenes overlay the background of the real-world environment with the greenscreen environment. For example, a wintery forest captured in British Columbia needs to be accurately captured and positioned with respect to a spaceship that enters the scene. The spaceship needs to be matched with the real environment especially if particles will be fired or destroyed. Consequently, the accurate location of where it happened has to be determined based on GNSS positioning.

1.5 Spatial Awareness on the Road

With the development of the internet and smartphones, navigation and location estimation has been transformed. A phrase mentioned in the GIS community is that approximately 80% of data has some sort of spatial component (Esri.com). In this age, data and the internet are co-related and have been developing at an exponential rate. By improving positioning accuracy, data with the connection of the Internet can describe information much better. A vehicle's navigation system can utilise this GIS technology by finding its position with respect to other objects like potholes, pedestrians, and stop signs. Such developments can help with lane change navigation and autonomous driving by providing data to the software of the vehicle to understand road safety. As a result, machine learning can be integrated with GNSS positioning to improve road navigation. With accurate positioning, data and numbers can be transformed into pictures and figures that help describe a situation. Therefore, GNSS positioning can greatly aid in spatial awareness on roadways.

2 Stakeholders

Associated stakeholders that could be interested in this project are smartphone companies, app developers, Google, and phone consumers. After discussions with Professor Bisnath, there was mention that Huawei sponsored their GNSS labs to produce accurate positioning solutions. Although specifics were not mentioned, it would seem that smartphone companies would be interested in greater positioning solutions for their products. Smartphone application developers would be stakeholders in this project as high accuracy positioning could be key to developing augmented or virtual reality video game applications. There needs to be a defined position to connect objects within the virtual and augmented environments. Google would be a confirmed stakeholder as they are the company that created the challenge through their subsidiary company, Kaggle. Phone consumers or anyone who owns a smartphone would be a stakeholder as this project would improve the performance of their smartphone. The improvements to the individual smartphones' GNSS receivers would also lead to applications that require higher accuracy solutions as mentioned previously.

3 Discussion Regarding the Data

All data used in the project was provided by Kaggle. Users placed smartphones on the dashboards of vehicles and drove around Silicon Valley on a number of different days. Different smartphones were used such as Samsung. Afterwards, high accuracy GNSS measurements were taken along the same route at the same positions that the smartphone measurements were observed. The challenge was to reach decimetre level accuracy for the positions of the smartphone locations when comparing the processed results to the reference high accuracy coordinates. Since the data was provided, the team did not collect their own measurements but used the provided observations.

In order to understand the limitations of the project, the data had to be fully identified and understood. Smartphone GNSS chips are of a very low quality in order to ensure that phones can handle the GNSS technology and are at a reasonable price. As a result, the smartphone GNSS chips tend to read rather poor pseudorange measurements. Such a situation meant that there was only so much that the team could do to improve accuracy. Regardless of the rigorous calculation process and steps followed during the processing of the measurements, having poor measurements cannot be fixed easily. If poor measurements are used, the expected product would be a poorly calculated solution. The organizers of the Kaggle competition seemed to be expecting that students could use such measurements to generate a good positioning result that actually requires advanced GNSS processing methods and hardware. Therefore, this concept was the problem of the decimetre challenge. The expectation was that individuals such as the team for the project could make poor measurements generate a high accuracy solution. At this time, this was an almost impossible challenge. While rigorous testing and estimation procedures can be done to get the phone measurements down to the metre level, the sub-metre level is very challenging to reach for all measurements at all times.

Despite the issues with the GNSS receivers, the way in which the data was collected also had an impact on the quality of the data. The smartphones used during the data collection process were placed on the dashboard of a moving car. As a result, all GNSS signals that reached the smartphone from the satellite had to pass through the windshield of the vehicle. This would have had an impact on reducing the quality of the measurements. In the same way, the moving system of the car would also affect the quality of the measurements. When considering a static GNSS system, the stationary position of the receiver allows for outliers to be better identified when they are compared to the original stationary position. In a moving vehicle, this cannot be done. Consequently, the positions maybe would not be as accurate as for a stationary system due to the amount of impact the moving car can have on estimations. Such errors in the data collection independent of the GNSS chip hardware also had to be acknowledged and considered when attempting to drive down the positional error.

The locations in which the GNSS data was collected also had an effect on the accuracy. In the case of the project, the data was gathered in Silicon Valley. Some of the data was gathered around the San Jose city centre while other data was collected in the suburbs and the highways. As a result, the quality of the data depended on the number of vertical obstructions in the area. An example of this was with the data collected on April 22, 2021. In this region, there were a number of apartment buildings and skyscrapers. The path of travel was along roads in between these tall obstructions. As a result, the measurements suffered from a significant amount of multipath when driving through an area surrounded by taller buildings. Meanwhile, measurements on April 28, 2021 were along a roadway with fewer buildings. Furthermore, the roadway had more lanes which also increased the amount of skyview area available to capture better quality signals. Due to these reasons, the system had less multipath and was better able to record better measurements.

As presented in the previous paragraphs, it was nearly impossible to reach sub-metre level accuracy simply because of the poor quality of the measurements. After reflection, the team decided that the metre level solution positioning system would be a reasonable metric for the developed positioning solution. By providing a positioning accuracy between 10 and 0 m, a wide range would be provided in order to ensure that a realistic accuracy target would be reached.

4 General Method of Accounting for Poor Measurements

The team attempted to make the best of having poor measurements through a series of processes. Firstly, an attempt was made to incorporate as many GNSS measurements per epoch as possible. Due to the nature of the least squares estimation process, having more measurements can tend to result in the estimation of the true coordinates being driven closer to the actual values. As more measurements are added to the system, observations that are outliers can be better isolated and identified. Such values would have a smaller likelihood of skewing the estimation of the user's position in the wrong direction. In order to add more measurements, the maximum number of constellation measurements have to be processed. In each epoch for the observations, there generally appeared to be measurements from four global GNSS systems. These were GPS, GLONASS, Galileo, and BeiDou. Therefore, the team would anticipate developing processing software for each particular constellation.

The next major correction method in the system would be to identify and remove outliers from the system. Not all measurements in the system were expected to be good or bad. Depending on the location in which the measurements were taken, some observations would be better than others. By assessing the residuals of the observations, obvious outliers could be expected to be identified. Therefore, such errors could be removed from the system.

Another potential correction attempt would be to model ionospheric errors correctly. While ionospheric errors can be cancelled when there are two frequencies, only having one frequency can present an issue. By modelling the ionospheric error with one frequency and then removing it, the overall error in the system can be reduced.

5 Review and Analysis of the Requirements

Table 1: List of Requirements for the Solution

Requirement Identifier	Requirement Text	Source	Validation Approaches And Timing
Performance 1	The software developed shall be able to compute a location down to at least a metre level resolution in open country.	The Kaggle project definition	The requirement is expected to be validated through evaluating the residuals. The earliest test point will be when the initial design of the program is developed in October while the latest test point will be after additional optimization procedures are added in March.
Performance 2	The software shall be able to determine position using GPS, GLONASS, Galileo, and BeiDou global GNSS systems.	Team agreement stemming from available measurements in RINEX files	The requirement is expected to be validated through testing the program, plotting the results on Google Earth, and comparing the calculated positions against the reference positions. The earliest test point will be on January 31, 2022 when the positioning algorithm is adjusted to apply to global GNSS systems. Meanwhile, the last test point will be on April 1, 2022 when the final verification of the program will be completed.
Performance 3	The calculated smartphone positions shall match the reference positions at the metre level. 1.000 m - 9.999 m	Reference positions provided by Kaggle	The requirement is expected to be validated by finding the differences between the computed smartphone positions and reference smartphone positions. The earliest test point will be after the development of the GPS positional program on November 30, 2021. Meanwhile, the last test point will be on April 1, 2022 during one of the final tests of the positional accuracy of the program.
Performance	Satellites must not be below 10	GNSS Signal Theory	This requirement is expected to be met by thresholding out satellite positions

4	degrees to the horizon.		below an elevation angle threshold. The first test point will be on January 15, 2022 when the enhancements are added to the code. The last test point will be April 1, 2022 during one of the final tests of the positional accuracy of the program.
Performance **5**	There must be enough satellites to ensure a good Dilution of Precision (DOP) of below 7 for a good positioning solution.	Least-squares principles	This requirement is expected to be validated by rejecting solutions with an insufficient DOP or number of satellites. The first test point will be on November 30, 2021 after the first iteration of the test prototype GPS program has been developed. Meanwhile, the last test point will be on April 1, 2022 during the final testing and computation of the program.
Functionality **6**	The software solution shall determine the estimated positions in the geodetic coordinate system.	Kaggle competition requirement	The requirement is expected to be validated by converting the X, Y, and Z components of the estimated positions to latitude, longitude, and elevation. The earliest test point will be November 30, 2021 after the first iteration of the test prototype GPS program has been developed. However, the last test point will be on April 1, 2022 during the final test and computation of the program.
Functionality **7**	The software solution shall be developed within a budget of $200.	Lassonde School of Engineering's ENG4000 budget limit	The requirement is expected to be validated through keeping a budget. The earliest test point will be when the budget is proposed on October 8, 2021. Meanwhile, the latest test point will be in February 2022 when the last project purchases are expected to be made.
Interface **8**	The position optimization program shall be able to run on MATLAB software.	The resources available to complete the project	The requirement is expected to be validated through programming using MATLAB software. The earliest test point will be November 5, 2021 after the first draft of the program is completed. Meanwhile, the latest test point will be on April 10, 2022 for the final testing of the program.

Interface **9**	The program developed shall apply accurate corrections specific to the current GNSS constellation type.	ICD-GPS-200 and GLONASS Control Interface documents	The requirement is expected to be validated by the ICD-GPS-200, GLONASS, and GALILEO Control Interface documents. The earliest test point will be October 23, 2021 during the development of the first test of the program. Afterwards, the latest test point will be on February 28, 2022 after satellites from other GNSS constellations have been incorporated into the program.
Regulatory **10**	The positional accuracy software shall comply with all governmental laws and regulations in Canada and the United States of America.	Canadian legal codes	The requirement is expected to be validated by accessing programming software legally and reviewing privacy laws. The earliest test point will be November 25, 2021 during the testing stage of the software. Lastly, the latest testing point will be on April 10, 2022 during the last testing of the program.
Regulatory **11**	The GNSS positional accuracy software must not include any restricted goods or systems outlined in Canada's Export Control List, 2016 version.	Government of Canada	The requirement is expected to be validated by developing a GNSS system that does not reduce navigational error of systems below those specified for controlled goods in Canada. The first test point will be when the GPS code is finished being made by November 30, 2021. However, the final verification of the program will be on April 1, 2022 with a final test of the program.
Regulatory **12**	The GNSS positional software will not decrypt ranging code designated for the Government of Canada's use.	Government of Canada	The requirement is expected to be validated by only using civilian C/A code rather than P code. The earliest test point will be when the GPS code is developed on November 30, 2021. Meanwhile, the last testing date will be on April 1, 2022 with a final test on the code of the program.
Programmatic **13**	The software program will be able to run on a computer with an	Limits of the Computers Being Used	The requirement is expected to be validated by ensuring that the program can run on a computer with an I5 processing system with a 64 GB

I5 processing system with a 64 GB memory.	memory. The earliest test point will be when the GPS code is first developed on November 30, 2021. On the other hand, the last test date will be on April 1, 2022 during one of the final tests of the positioning code.

5.1 Requirement 1

Based on the final development of the algorithm, the final solution was not in compliance with Requirement 1 as outlined in the Critical Design Review completed in December 2021. In the original requirement, the software was to be able to estimate the position of the smartphone to at least a metre level resolution in open country. Such open country was considered to be a highway with an open skyline and areas without many buildings around. While the positional accuracy of the program was definitely at the metre level, the positional accuracy was not maintained at a metre level resolution at all times in open country. Due to the noisy nature of the measurements, the positional accuracy tended to increase and decrease at times on the highway. There was not a definite way in order to check that the measurements were always at the metrelevel on open highways. Therefore, it can only be concluded that the positioning algorithm did not satisfy Requirement 1 in all cases.

In regards to Requirement 1, the wording of the requirement resulted in the algorithm not meeting the standard. If the team had used the wording “to at least the metre level resolution”, the group would have been compliant if the positional accuracy results were between 1.000 m to 9.999 m. However, the group used the phrase “to at least a metre level resolution” which outlined that the positional accuracy had to be at or less than a metre. This was a clearly defined requirement that the group could not meet. While the positional accuracy was at the metre level, it was not at a metre level resolution. In retrospect, the team learned to be careful about laying out too strict requirements during the designing process. While it is good to be clear with the requirements for a design, there is the danger of the programmers getting hedged in to meet a goal that cannot be attained. Consequently, greater care could have been taken to choose the threshold for a requirement that was not so restrictive.

After reflection, the team decided it would be better to change the requirement to be at the metre level rather than at a metre resolution. However, doing so resulted in Requirement 1 and Requirement 3 being the same. Due to this reason, it was decided that it would be better to disregard Requirement 1 to avoid being repetitive.

5.2 Requirement 2

Requirement 2 as recorded in the Critical Design Review was partially satisfied. The relative positioning augmentation algorithm was able to successfully determine position based on measurements from the GPS and Galileo global GNSS systems. While considerable progress was made with processing GLONASS measurements, the GLONASS program failed the accuracy tests during the test phase. Further details regarding this situation were provided later in the report. In the same way, progress was also made with the BeiDou measurement processing. Nevertheless, the computed positional accuracies were quite poor. The situation was discussed further on in the report. After consultation, the team determined that it would be wiser to focus on poor residual removal and handling programmatic bugs rather than introducing more error into the system with partially completed GLONASS and BeiDou systems. This meant that Requirement 2 was met relating to GPS and Galileo but not for GLONASS and BeiDou.

5.3 Requirement 3

Once the results were tested and verified, Requirement 3 was satisfied. The final relative positioning augmentation algorithm was able to estimate coordinates that related to the reference positions at the metre level. By the metre level, the team meant that the errors between the reference and computed coordinates were within the range of 1 m to 9.999 m. Since the final mean error in the easting was 6.0 m and the mean error in the northing was 7.4 m, the solution was at the metre level. In the same way, the root mean square error (RMSE) calculations were 8.1 m for the easting and 9.3 m for the northing which were also at the metre level. As a result, Requirement 3 was met in the system.

5.4 Requirement 4

After analysing the results, the final solution met Requirement 4. In the code, the elevation angle was calculated based on the current observation and satellite correction information for each epoch. If the elevation angle was found to be less than 10 degrees or 0.1745329252 radians, the observation was removed before the least squares estimation was conducted. Therefore, none of the observations recorded from satellites with an elevation angle below 10 degrees were used.

5.5 Requirement 5

The final positioning algorithm was in compliance with Requirement 5. In each separate iteration of the position estimation for global constellations such as GPS and Galileo, the DOP levels tended to be below 4. The DOP values improved with the combination of global constellations and the removal of measurements with large residuals. For the combination of the GPS and Galileo measurements, the DOP values were below 2. Therefore, the requirement of the DOP being below 7 was adequately met.

5.6 Requirement 6

In the same way, the relative positioning augmentation met Requirement 6. After analysis, the team adjusted the language of the requirement. The original requirement stated that the solution had to determine the positions in the geographic coordinate system. After reflection, the group decided to change the requirement to relate to the geodetic coordinate system to account for the ellipsoidal representation of the WGS84 ellipsoid. The resulting adjustment to the requirement caused the estimated coordinates to more accurately satisfy the requirements for the Kaggle competition. In the algorithm, the latitude, longitude, and height were computed based on the WGS84 datum for the geodetic coordinate system. These were able to be related to the reference latitude and longitude coordinates of the positions provided by Kaggle. Consequently, the algorithm satisfied Requirement 6.

5.7 Requirement 7

Requirement 7 was satisfied in the relative positioning augmentation program. The team maintained their budget goals over the course of the year. Since MATLAB was freely offered by York University, there were no expenses related to the project. The group did not use any other forms of software that required subscriptions. Therefore, the algorithm was developed in compliance with Requirement 7.

5.8 Requirement 8

The relative positioning augmentation program met Requirement 8. Over the past two semesters, the program was developed exclusively using MATLAB software. Since this was the case, the algorithm was fully functional on MATLAB software. As was discussed in the Critical Design Review, the group determined that the focus of the project would be to develop the algorithm itself. After the development of the program, software engineers and programmers would be responsible for compiling the code into a format for use on smartphones. As a result, the code was fully functional and compatible with the MATLAB programming language.

5.9 Requirement 9

To the team members' best knowledge, Requirement 9 was satisfied by the relative positioning augmentation program. The team members made all corrections to the satellite parameters based on the GPS and Galileo ICDs. As a result, it appeared that the corrections were made accurately for each separate GNSS constellation. Furthermore, the mean positional accuracies in the easting and northing were at the metre level. If there was a significant correction error applied, the accuracies likely would have been poor to reflect such an inaccuracy. Therefore, it appeared that the solution was in compliance with Requirement 9.

5.10 Requirement 10

When examining the developed GNSS positioning algorithm, Requirement 10 was satisfied by the program. The GNSS signals used were on the L1 and L5 frequencies and were based on civilian code. No precise P code measurements were used in the positioning algorithm. Likewise, no velocity measurements were used in the positioning algorithm. Through using both GNSS and IMU technologies, missile navigation systems can be developed. As a result, the group decided to only process GNSS measurements. Doing so ensured that all navigation systems developed using this algorithm would not be applicable to weapons development. As a result, the group was in compliance with Requirement 10.

5.11 Requirement 11

The algorithm appropriately met Requirement 11. While planning the design for the project, the team consulted Canada's Export Control List released in 2016. After reviewing the 2016 version of Canada's Export Control List, concerns for legal violation exist when using precise GNSS signal code such as P code and incorporating IMU measurements. A blend of GNSS and IMU positioning technology using GNSS signals and velocities can be used to develop missile guiding systems. Since the team did not use velocity measurements, no acceleration calculations could be used in conjunction with GNSS measurements. Therefore, the group was in compliance with Requirement 11.

5.12 Requirement 12

While the relative augmentation positioning algorithm provided metre level positioning, it satisfied Requirement 12. Civilian GNSS codes were used in the computations of positions based on the L1 and L5 frequencies. No precise codes were readily available to the students in the RINEX measurement files nor would the students have used them. Therefore, Requirement 12 was complied with.

5.13 Requirement 13

Requirement 13 was met in the developed algorithm. The software was coded, tested, and edited on a laptop with an I5 processor. In the same way, the computer had a 64 GB memory system. Since the team determined that the program would be coded on a computer and compiled in the future by a software engineer, the group did not concern themselves with compiling the code for a smartphone. Consequently, the relative positioning augmentation algorithm met Requirement 13.

6 Work Breakdown Structure

In order to complete the project effectively and in a timely manner, the team broke down the tasks that needed to be completed. The group started by completing a work breakdown structure. The final work breakdown structure was very similar to the one presented in the Critical Design Review. After consideration, the only change that was made was that the double differencing work package was removed. This was done since the team decided to focus more on outlier residual removal and there was not enough time to complete the double differencing effectively. Furthemore, another edit was made to one of the work packages that changed from “Debug Program” to “Comment And Complete Program”. This was completed to ensure that two work packages to debug the program were not side by side (Figure 1). In the same way, the package was made to set aside time to comment the code and handle any last edits. By doing so, the team would be able to submit a better quality product and provide adequate documentation of the steps taken to complete the project.

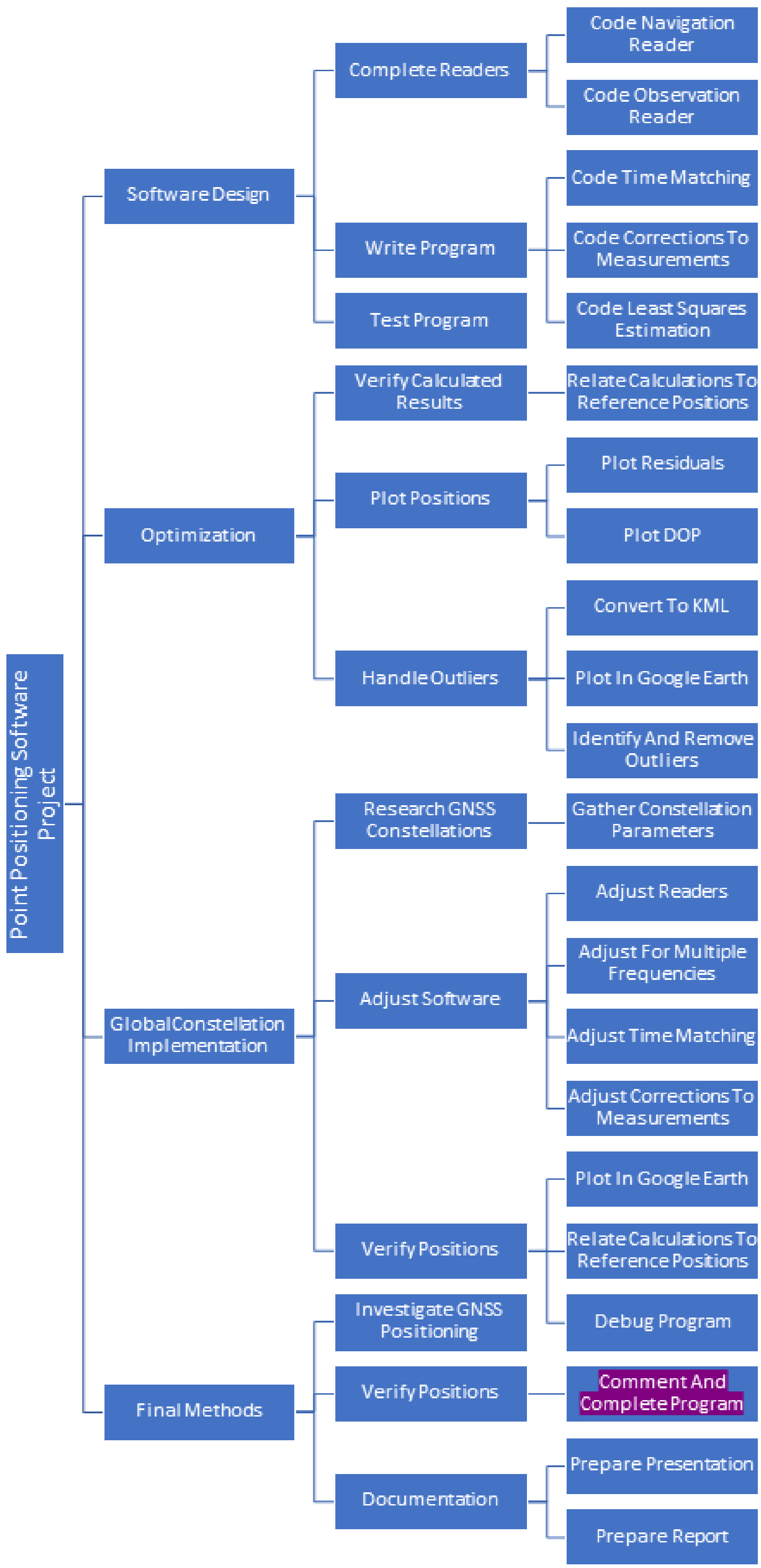
Point Positioning Software Project
Software Design
Complete Readers
Code Navigation Reader
Code Observation Reader
Write Program
Code Time Matching
Code Corrections To Measurements
Code Least Squares Estimation
Test Program
Optimization
Verify Calculated Results
Relate Calculations To Reference Positions
Plot Positions
Plot Residuals
Plot DOP
Handle Outliers
Convert To KML
Plot In Google Earth
Identify And Remove Outliers
Global Constellation Implementation
Research GNSS Constellations
Gather Constellation Parameters
Adjust Software
Adjust Readers
Adjust For Multiple Frequencies
Adjust Time Matching
Adjust Corrections To Measurements
Verify Positions
Plot In Google Earth
Relate Calculations To Reference Positions
Debug Program
Final Methods
Investigate GNSS Positioning
Verify Positions
Comment And Complete Program
Documentation
Prepare Presentation
Prepare Report

Figure 1: Work Breakdown Structure

7 Work Packages

Based on the work breakdown structure, the group generated work packages for each step within the project. In a similar manner to the work breakdown structure, the team only removed the work package regarding double differencing and edited another to discuss commenting and completing the program. Since a work package was deleted, the time frames for the surrounding work packages were edited. Debugging the program was extended to be 18 days while commenting and completing the program was also moved to be 10 days long. The team decided to adjust these work packages as the group thought that additional time would be needed to finish the last edits and verification of the program. Likewise, the team adjusted some of the wording for work packages that involved GLONASS and BeiDou. Such an edit was made as the team was able to end up completing GPS and Galileo GNSS processing. All the work packages that were used for the project were included in the Appendix of the report.

7.1 Gantt Chart

Once the work packages were finalised, a Gantt chart was created to visually show the schedule for the different work packages needed to complete the project (Figure 2). As was the case with the work breakdown structures and the work packages, the only edits to the Gantt chart were made to debugging the program and completing and commenting on the algorithm. Both of the time frames for the work packages were extended.

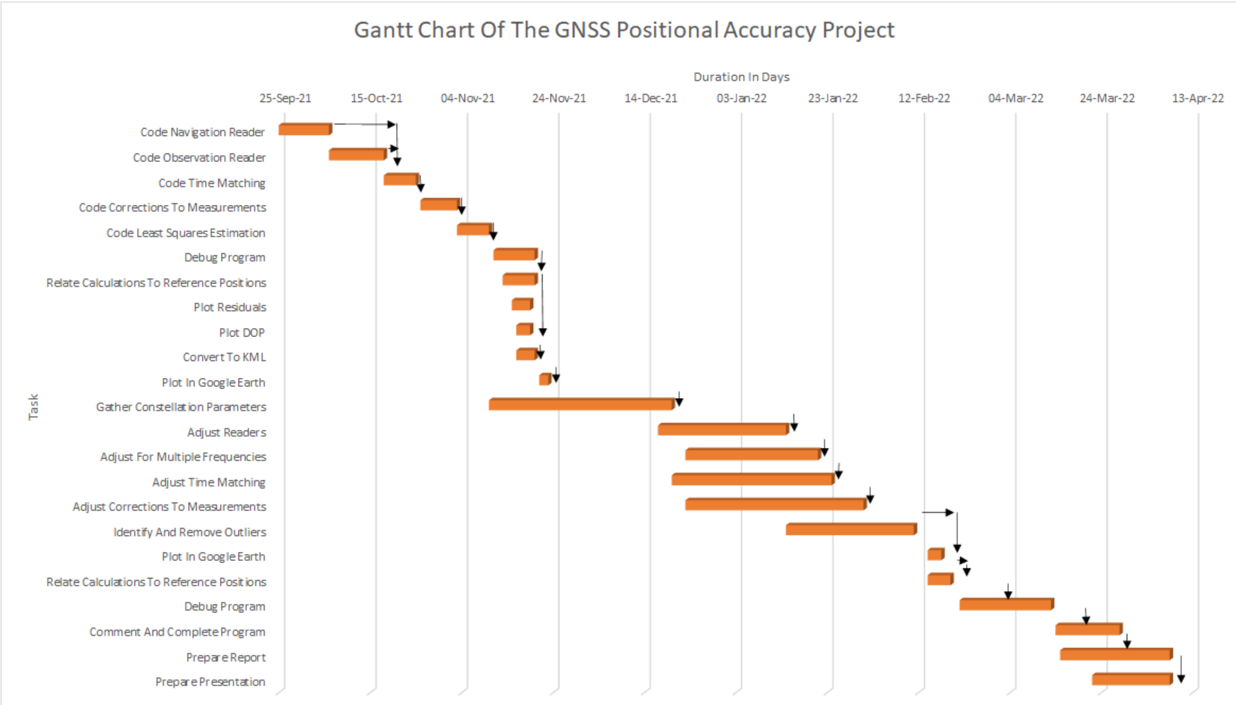

Figure 2: Gantt Chart

8 Materials

The list of materials did not change from the Critical Design Review. Since the project was primarily a software design, the main materials required were the reference files to be read into the program, the programming language software, Google Earth, and a KML online converter. The observation and navigation readers from Kaggle provided the measurements and corrections while the reference position files offered a means to verify how accurate the estimated positions were. MATLAB software was used to code the algorithm and Python was considered for use to organize the code at the end. GitHub software was considered for use by the team in order to track member edits and progress on the code. The team also used pseudocode from the ESSE3670 they had developed as a reference of the coding component of the project with full disclosure and approval from Dr. Bisnath. GNSS correction parameters were also needed from interface control documents (ICDs) to correct the GNSS measurements according to the current GNSS global constellation being considered. Google Earth, Excel, and the MyGeodata convertor were also required to visualise the results and verify their positions.

The only material that was added to the list was Excel. During the testing stage of the project, the estimated coordinates were exported to Excel so that they could be saved in a .csv format. Doing so allowed the file to be uploaded to an online KML convertor that would transform .csv files into .kml files. As a result, the team was able to visually evaluate their work on Google Earth and compare the files to the original reference points. All of the required materials were listed in the following table.

Table 2: Required Materials

Equipment	Type	Source
Satellite Navigation GNSS Parameters	Development	York University GNSS Laboratory
Observation GNSS Files	Input	Kaggle Competition Data
Known Positions For Comparison	Input	Kaggle Competition Data
MATLAB	Programming	MATLAB Services At York University
Python	Programming	Python Services Online
GitHub Software	Programming	GitHub Services
GNSS Correction Parameters	Development	Interface Control Documents For GNSS Constellations
Google Earth	Testing	Google Earth Services
KML Convertor	Testing	MyGeodata Convertor
Excel	Testing	Microsoft Excel

9 Budget

Based on the materials needed for the project, there did not appear to be any expenses related to the development and testing of the relative positioning augmentation algorithm. All of the required observation and navigation data was provided by the Kaggle competition on their website. In the same way, the Kaggle competition website also offered reference coordinates that could be referred to. While use of MATLAB requires a paid subscription, York University offered students the software free of charge. Consequently, no expenses were related to MATLAB. Similarly, Excel software was also freely available because York University also offered the software to students without charge. Google Earth was also free to download along with the GitHub software and the KML file convertor software. As a result, the proposed budget was $0.

In the Critical Design Review, the team budgeted $150 toward any unforeseen expenses that might have been incurred during the implementation and testing of the design. The team did not end up having any expenses in the project. Therefore, the final budget of $150 was met since the team spent $0. Consequently, the final expenses of the project had a final sum of $0.

Table 3: Project Expenses

Expense	Value
GNSS Observation Data	$0
MATLAB Software	$0
Python Software	$0
Google Earth Software	$0
KML Convertor Software	$0
GitHub Software	$0
Excel	$0

10 As-Built Design

Based on the results of the trade identification matrix, the team determined that the relative positioning augmentation solution would be the most appropriate proposed solution in terms of time, quality, and cost. In relative positioning, the vectors between a known and unknown point have to be determined. By doing so, the coordinates of the unknown point can be found (Figure 3). The change in coordinates between two positions, abstractly known as A and B, can be determined. Likewise, the measurements can be combined between the rover station and the reference station. Such a practice can result in errors in the system being reduced.

A: Reference (known) point

B: unknown point

$\vec{b}_{AB}$: baseline vector

$\vec{X}_A, \vec{X}_B$: position vectors

$$\vec{X}_B = \vec{X}_A + \vec{b}_{AB}$$

$$\vec{b}_{AB} = \begin{bmatrix} X_B - X_A \\ Y_B - Y_A \\ Z_B - Z_A \end{bmatrix} = \begin{bmatrix} \Delta X_{AB} \\ \Delta Y_{AB} \\ \Delta Z_{AB} \end{bmatrix}$$

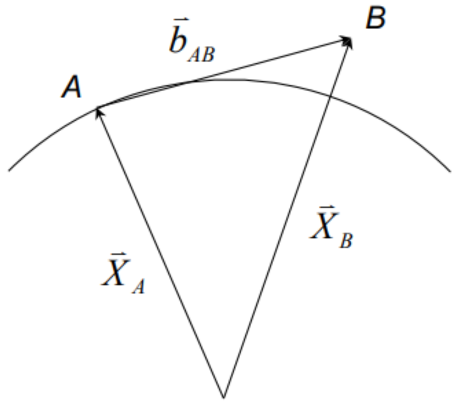

Figure 3: Relative Positioning (Bisnath, 2021)

In the relative positioning augmentation, there were three key methods the group used to reduce error in the measurements. The first practice was including as many measurements from different global GNSS systems as possible. By having a large number of measurements per epoch, potential outliers could be better identified in the least squares estimation (LSE) process and be isolated. Furthermore, the measurements would better fit the estimation model. The second method was to correct for ionospheric refraction. GNSS measurements are impacted by refraction due to the charged particles in the ionosphere above the earth. The best method was to linearly combine two different frequencies and pseudoranges. The resulting pseudorange would have the ionospheric error cancelled out. If there was only frequency, the group used the Broadcast (Klobuchar) model to model the ionospheric error for the epoch. Afterwards, the correction would be added to the original pseudorange measurement. The third method was to remove outliers in the system. Through doing so, the positional accuracy of the smartphones could be improved.

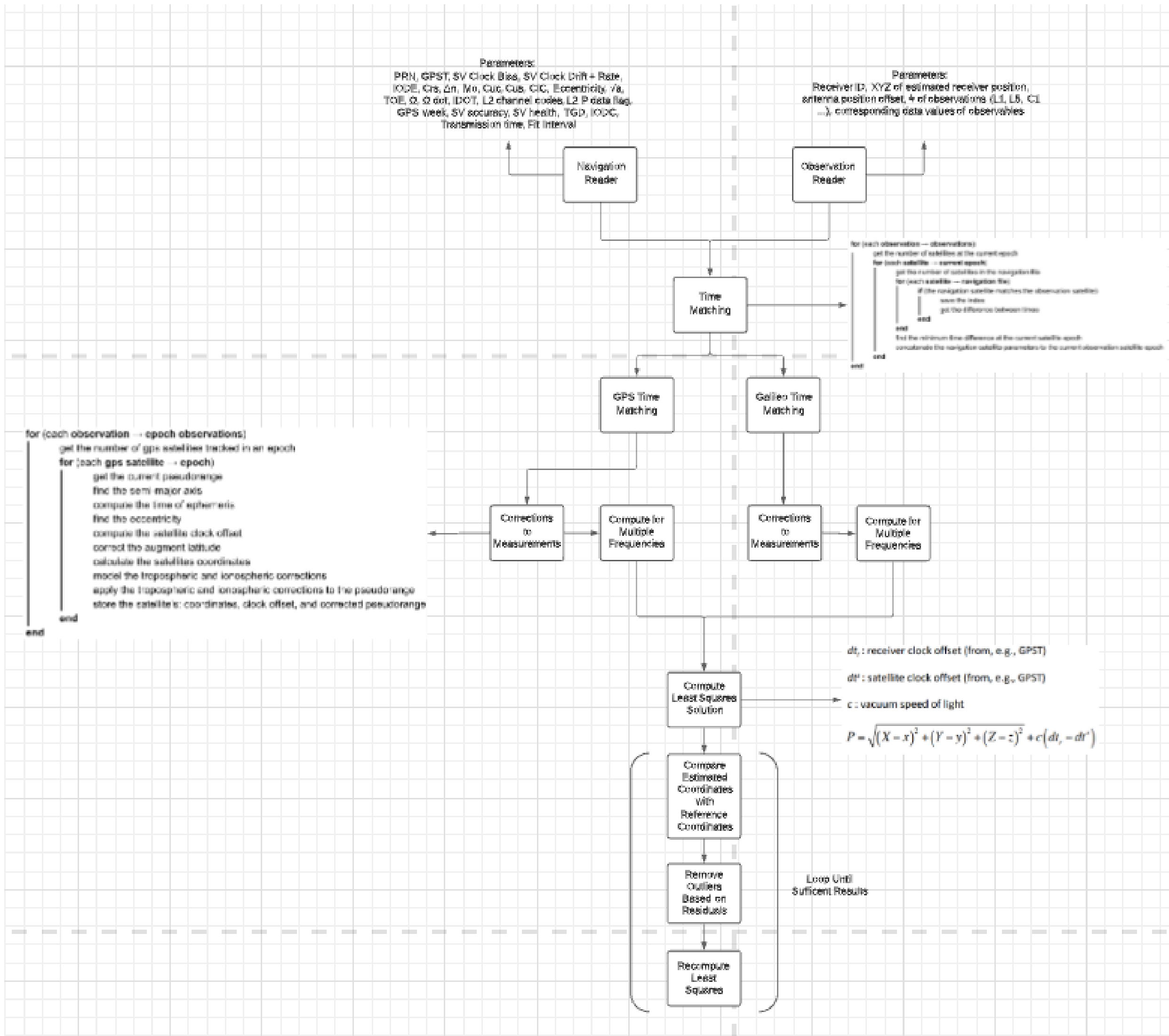

Figure 4: Design Workflow

The final design workflow remained basically the same as the one proposed in the critical design review. However, to add more clarity to the design structure the team added multiple new steps (Figure 4). The processes added were time matching for GPS and Galileo, comparing estimated coordinates with the reference coordinates, removing outliers based on residuals, and recomputing the least squares estimation. Each design component was described with further detail below.

10.1 Code Navigation Reader

The navigation reader was designed to iterate through and store the parameters associated with the current satellite and epoch in the RINEX 3.04 file. For each satellite, there were a number of parameters such as the clock bias, eccentricity, and the transmission time that had to be used to compute the satellite position and correct the pseudorange (Figure 5). Consequently, the reader was designed to convert the time epoch to GPS seconds and store the correction parameters in the navigation file to be used later.

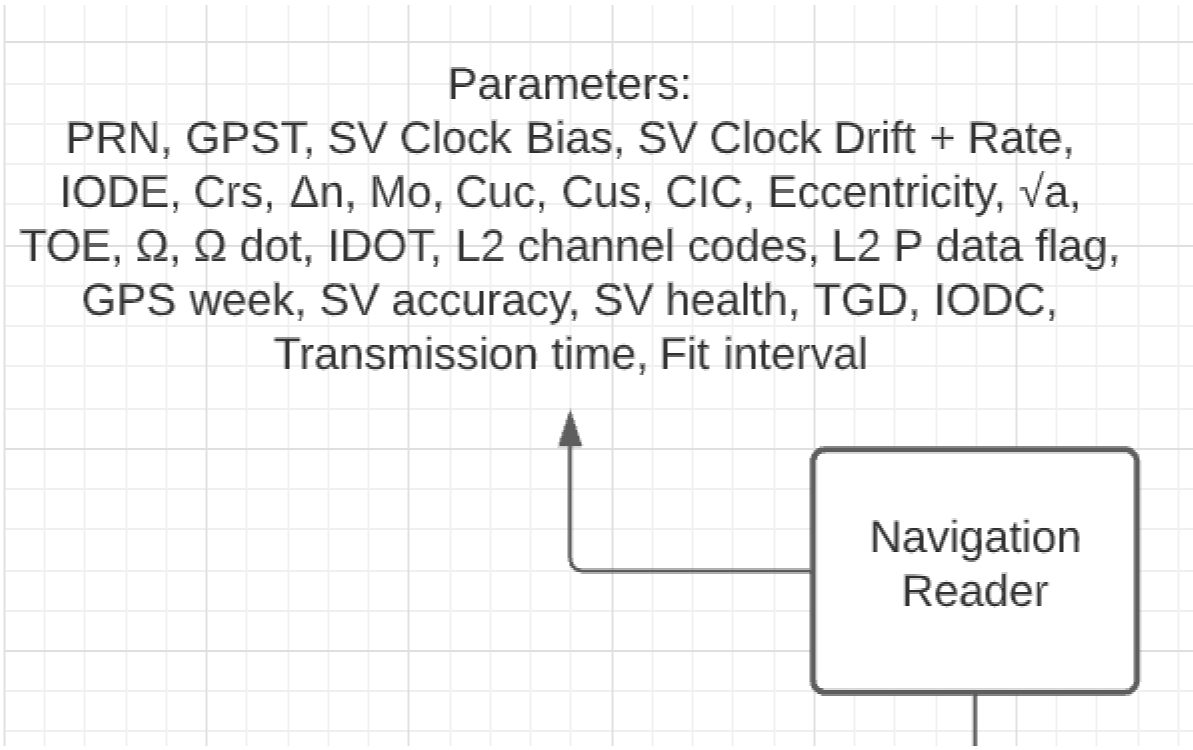

Figure 5: Navigation Reader Design

10.2 Code Observation Reader

In this section of the design, code was developed to read in the parameters from the observation file in the RINEX 3.4 format. The epoch, satellite constellation, satellite number, and pseudorange were key to extract, separate, and store for further processing (Figure 6). Furthermore, the time was computed to be in GPS seconds. The reader also was able to store the pseudorange and relevant data for both the L1 and L5 frequencies if observations for both were on the same satellite.

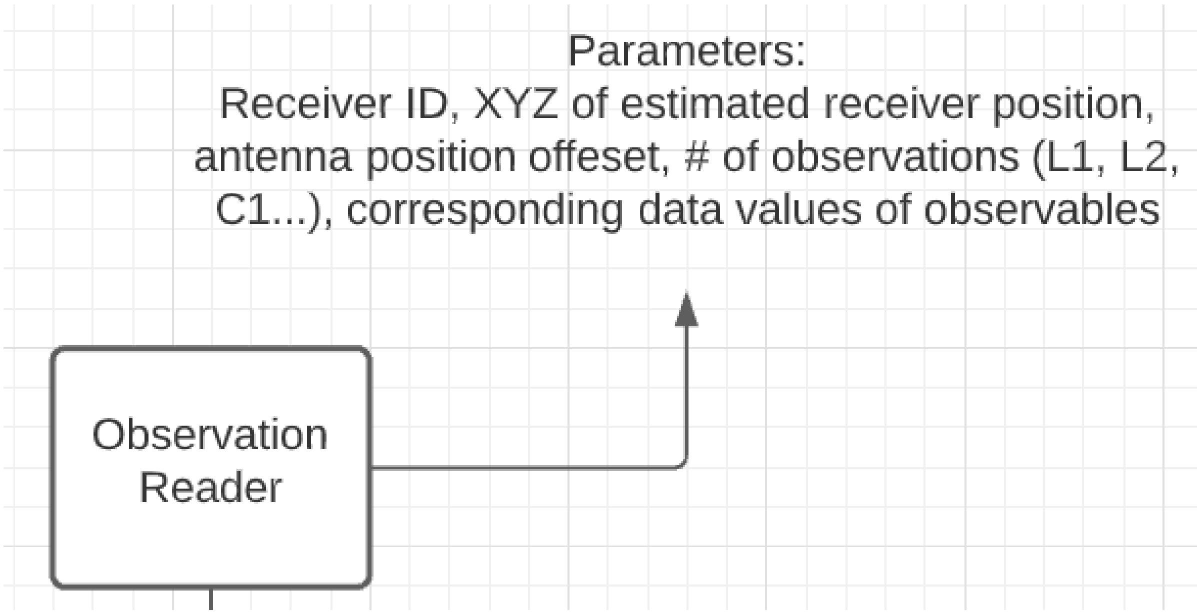

Figure 6: Observation Reader Design

10.3 Code Time Matching

A partial algorithm was designed to loop through the navigation data and compare the navigation time to the current observation data time being analysed. In the code, the smallest difference between the navigation epoch and observation epoch would be found (Figure 7). Afterwards, the index of the navigation data associated with the closest navigation epoch would be saved. The navigation parameters at that epoch would be saved to the current observation epoch. As a result, the closest match between the navigation and observation times would be made so that the correct satellite parameters would be saved to the correct observation epoch. Consequently, the proper corrections could be applied to the observation data.

An additional step was added when processing both GPS and Galileo measurements. In such a situation, an if-statement was included. If the current satellite was part of the GPS constellation, the time matching would occur within the GPS navigation data. If the current satellite was a member of the Galileo constellation, the time matching would be conducted in the Galileo navigation data. The index for the GPS navigation data would be stored for GPS satellites while the index for the Galileo navigation parameters would be kept for Galileo satellites.

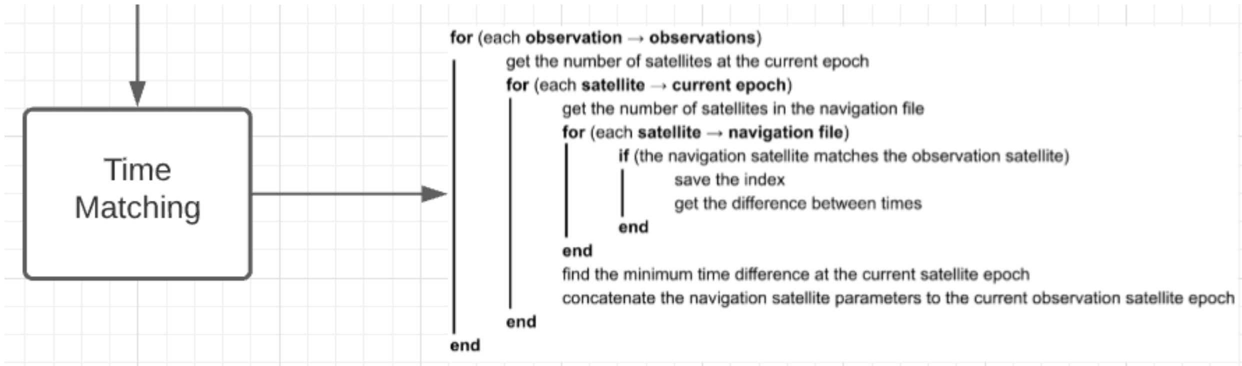

Figure 7: Time Matching Design

10.4 Code Correction to Measurements

In this section of the design, the pseudorange and observations would be corrected for errors. Likewise, the coordinates of the satellite position would be computed from the corrected satellite parameters (Figure 8). Each satellite had its own correction values and parameters provided by the closest navigation epoch. Through implementing the corrections to the clock offsets and pseudorange, the systematic and natural errors stemming from Earth's rotation and processes could be mitigated. Furthermore, the satellite coordinates and time estimates could be determined to feed into the least squares process.

An if-statement was used to differentiate between the GPS and Galileo measurements. If the current satellite being processed was from the GPS system and had a GPS index reference, the correction would be applied from the GPS navigation data. Meanwhile, a current satellite from the Galileo system with a Galileo index reference would have corrections extracted from the Galileo navigation data.

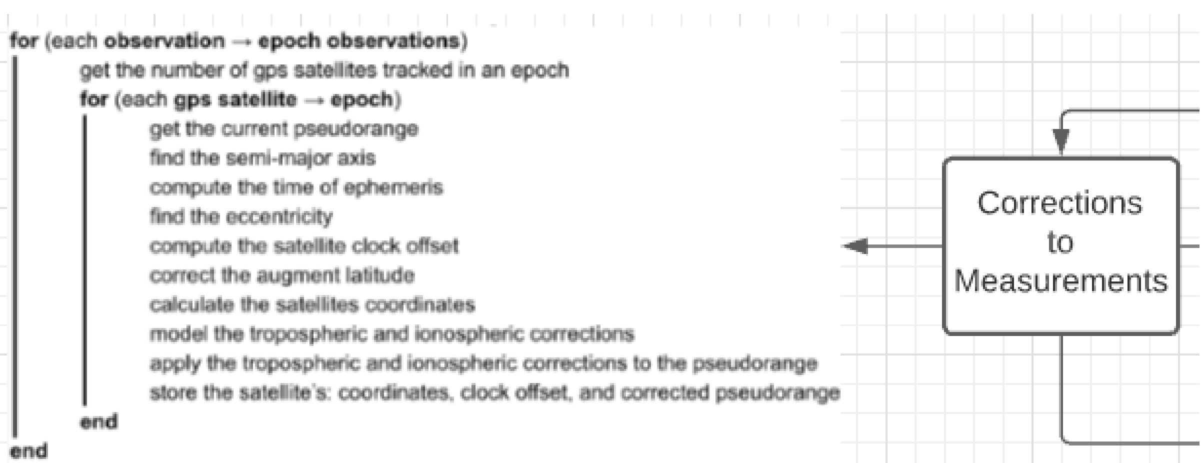

Figure 8: Corrections to Measurements Design and Code

10.5 Compute For Multiple Frequencies

Further consideration had to be given to correcting the ionospheric refraction. If there were both L1 and L5 pseudoranges for the current satellite, the two pseudoranges were linearly combined (Figure 9). By having two pseudoranges, the ionospheric error could be completely eliminated.

$$\rho^* = \frac{f_1^2}{f_1^2 - f_2^2}\rho_1 - \frac{f_2^2}{f_1^2 - f_2^2}\rho_2$$

Figure 9: Linear Combination of Pseudoranges (Bisnath, 2021)

However, if there was only one frequency, the process was more complex. The Broadcast (Klobuchar) Model could be used to estimate the ionospheric refraction based on the satellite latitude and longitude, the elevation angle, and the azimuth (Figure 10). Since the Broadcast (Klobuchar) Model was only an approximation, it was not as effective as the linear combination method. Nevertheless, the model corrected for the majority of the ionospheric error.

$$T_{iono} = \begin{cases} F * \left[5.0 * 10^{-9} + (AMP)\left(1 - \frac{x^2}{2} + \frac{x^4}{24}\right)\right], & |x| < 1.57 \\ F * \left(5.0 * 10^{-9}\right), & |x| \geq 1.57 \end{cases} \quad \text{(sec)}$$

Figure 10: Broadcast Model for eliminating Ionospheric Error (Bisnath, 2021)

10.6 Compute Least Squares Solution

Once the proper corrections were made and the satellite positions were calculated, least squares estimation was applied to estimate the locations of the smartphone positions. The estimation process was based on the pseudorange measurement equation (Figure 11). In the equation, the satellite X, Y, and Z positions, speed of light in a vacuum, and the satellite clock offset were known quantities. Meanwhile, the receiver x, y, and z positions and the receiver clock offset were not known. Each epoch had to have pseudorange measurements from at least five satellites to ensure that the system had a level of redundancy.

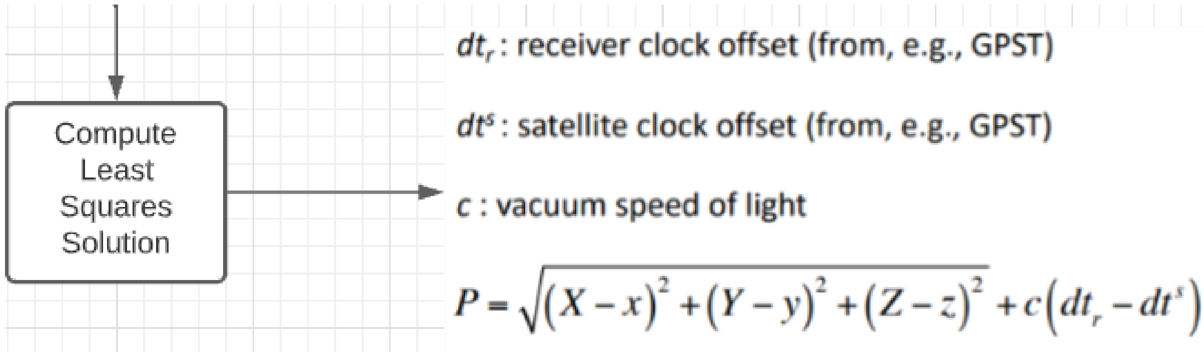

Figure 11: Least Squares Design

In the least square estimation, the partial derivatives were taken with respect to each of the four unknown parameters based on the observation equation. The resulting expressions were used to populate the first design matrix (A). Misclosures were also calculated based on subtracting the estimated pseudoranges from the actual pseudoranges. The least squares computation results in the computation of an incremental value (delta) that could be added to the current position estimates of the receiver and receiver clock offset. The least squares process would iteratively continue the computation process unless the updates to the coordinates were very small or less than 0.01. At this point, the best estimation possible under the circumstances was conducted and the resulting receiver positions would be computed. The final incremental value was added to the estimates of the receiver position to determine the final estimate position.

Observation equation: $P = \sqrt{(X-x)^2 + (Y-y)^2 + (Z-z)^2} + c(dt_r - dt^s)$

$$\frac{\partial P}{\partial x} = \frac{(X - x_o)}{\sqrt{(X-x_o)^2 + (Y-y_o)^2 + (Z-z_o)^2}} = \frac{(X - x_o)}{\rho_o}$$

$$\frac{\partial P}{\partial y} = \frac{(Y - y_o)}{\sqrt{(X-x_o)^2 + (Y-y_o)^2 + (Z-z_o)^2}} = \frac{(Y - y_o)}{\rho_o}$$

$$\frac{\partial P}{\partial z} = \frac{(Z - z_o)}{\sqrt{(X-x_o)^2 + (Y-y_o)^2 + (Z-z_o)^2}} = \frac{(Z - z_o)}{\rho_o}$$

Figure 12: Partial Derivatives with respect to x, y, and z (Bisnath, 2021)

$$A = \begin{bmatrix} \frac{\partial P_1}{\partial x} & \frac{\partial P_1}{\partial y} & \frac{\partial P_1}{\partial z} & c \\ \frac{\partial P_2}{\partial x} & \frac{\partial P_2}{\partial y} & \frac{\partial P_2}{\partial z} & c \\ & \cdots & & \\ \frac{\partial P_n}{\partial x} & \frac{\partial P_n}{\partial y} & \frac{\partial P_n}{\partial z} & c \end{bmatrix} \qquad w = \begin{bmatrix} P_1 - \rho_{o_1} \\ P_2 - \rho_{o_2} \\ \cdots \\ P_n - \rho_{o_n} \end{bmatrix}$$

$$\delta = \left(A^T A\right)^{-1} A^T w$$

Figure 13: Least Squares Estimate (Bisnath, 2021)

$$x = x_0 + \delta$$
$$y = y_0 + \delta$$
$$z = z_0 + \delta$$

Figure 14: Final Estimated Position (Bisnath, 2021)

10.7 Comparison of Estimated Coordinates

After the least square estimation, the residuals of the estimated coordinates were examined. Measurements with a residual value greater than 20 m were assigned an index. Likewise, epochs that had residual values that were greater than 45 m were also identified and were assigned an index.

10.8 Remove Outliers Based on Residuals

The measurements were looped through again to remove the measurements with high residuals. Measurements that were flagged as having residuals over 20 m were removed from the estimation process. In the same way, epochs that were flagged as having residuals over 45 m were taken out as well. By doing so, the outliers were removed from the system.

Table 4: Flagged Epoch Residual Data

Obs_data.data

	1	2
1	1.3037e+09	12x10 double
2	1.3037e+09	13x10 double
3	1.3037e+09	13x10 double
4	1.3037e+09	13x10 double
5	1.3037e+09	13x10 double
6	1.3037e+09	13x10 double
7	1.3037e+09	13x10 double
8	1.3037e+09	13x10 double
9	1.3037e+09	12x10 double
10	1.3037e+09	13x10 double
11	1.3037e+09	13x10 double
12	1.3037e+09	13x10 double
13	1.3037e+09	13x10 double
14	1.3037e+09	12x10 double
15	1.3037e+09	14x10 double

In the figure of the table, the first column represents the epoch in the GPS seconds (Table 4). Meanwhile, the second column represented the measurements of the available satellites at that epoch. Each satellite that was available provided information such as the pseudorange, clock error, the constellation the satellite belonged to, and the satellite number. For epochs with residuals that were greater than 45 m, the entire row was removed. In that way, the quality of the other measurements could be enhanced.

Table 5: Flagged Observations with Satellite Association

	1	2	3	4	5	6	7	8	9
1	1	5	2.3736e+07	-2.9393e+05	2.7760e+03	38.2661	0	0	0
2	1	6	2.2545e+07	3.7812e+05	-3.5834e+03	42.3591	2.2545e+07	2.8464e+05	-2.6766e+03
3	1	12	2.0114e+07	3.4739e+04	-350.3071	37.9301	0	0	0
4	1	19	2.4332e+07	3.7051e+05	-3.5006e+03	36.2171	0	0	0
5	1	24	2.3500e+07	3.8285e+05	-3.8183e+03	26.0801	0	2.8589e+05	-2.8468e+03
6	1	29	2.3834e+07	-1.7907e+05	1.8170e+03	26.3611	0	0	0
7	3	1	2.5782e+07	-1.5531e+05	1.6070e+03	30.0661	0	1.1601e+05	1.2007e+03
8	3	13	2.3331e+07	1.5494e+05	-1.5277e+03	35.3211	2.3331e+07	1.2079e+05	-1.1405e+03
9	3	15	2.6554e+07	3.0726e+05	-3.2269e+03	41.2751	2.6554e+07	2.2934e+05	-2.4080e+03
10	3	21	2.3452e+07	7.3223e+04	-719.9221	33.3561	2.3452e+07	5.6069e+04	-538.1422
11	3	26	2.4021e+07	-1.0667e+05	1.0752e+03	30.5191	0	-7.9688e+04	802.5322
12	3	27	0	0	0	0	2.5456e+07	0	0

Going into one of the indexes in the second column, the parameters in the observation file can be seen. The first column represented the constellation number with GPS being 1 and Galileo being 3 (Table 5). Meanwhile, the second column was the satellite number and the third column was the GPS time epoch in seconds. Whenever a row had a residual value that was greater than an absolute value of 20 m, the observation in the row was removed. By doing so, the quality of the estimation process could be maintained.

10.9 Recompute Least Squares

Once the outliers were removed from the observations, the least squares estimation was recomputed (Figure 15). The same process was applied as discussed in the "Compute Least Squares Solution" section of the report. Through doing so, the least squares estimation process was performed without the outliers.

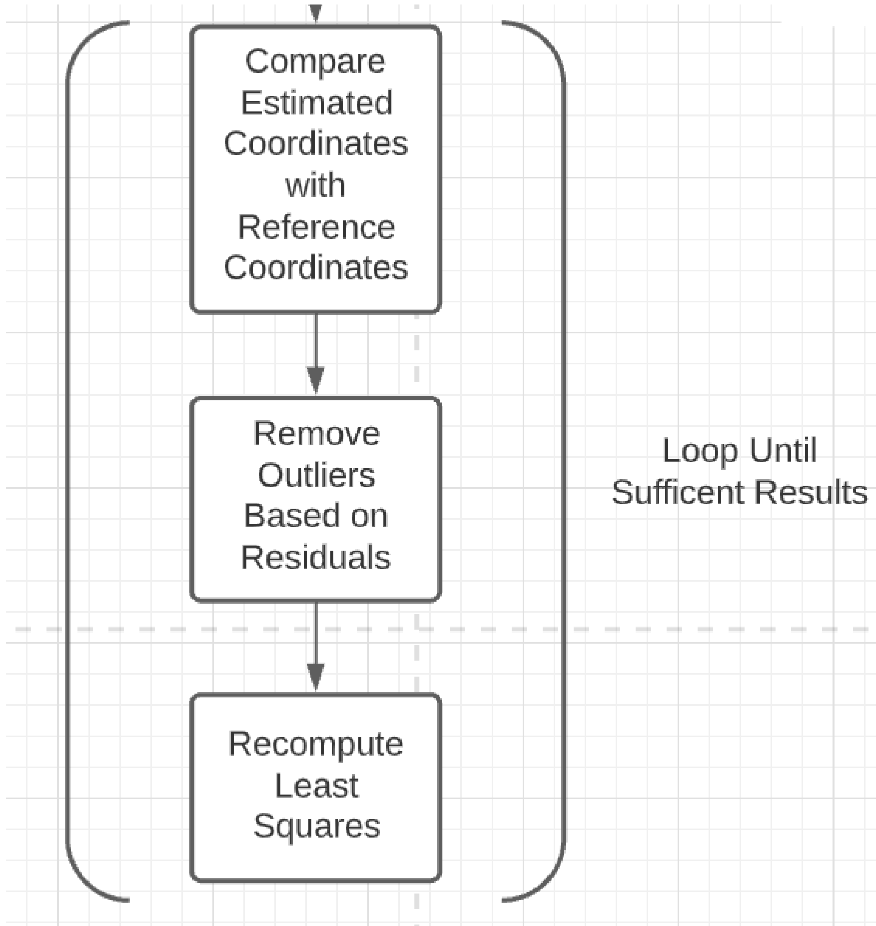

Figure 15: Least Squares Recomputation Design

Once the final least squares estimation process was completed, the coordinates were transformed from being in the Earth Centred Earth Fixed (ECEF) XYZ coordinate system to the geodetic ECEF system with latitude, longitude, and height. By doing so, the computed latitude, longitude, and height could be compared to the reference latitude and longitude values for the coordinates provided by the Kaggle competition organizer. Such a process provided a means by which the quality of the estimated positions could be verified.

10.10 Residual Outlier Removal Discussion

When considering the removal of poor measurements, the team had to be careful not to remove too many observations. Not all measurements with high residuals were outliers. Based on the least squares estimation process, the residuals illustrate how far the observations are from the estimated positions. In order to illustrate this concept, a group of measurements can be considered. A figure was drawn showing measurements by blue points and a best fit line approximating the points (Figure 16). Two noticeable outliers far away from the approximating line were circled in red. Meanwhile, two potential outliers at the far ends of the approximating line were circled in blue.

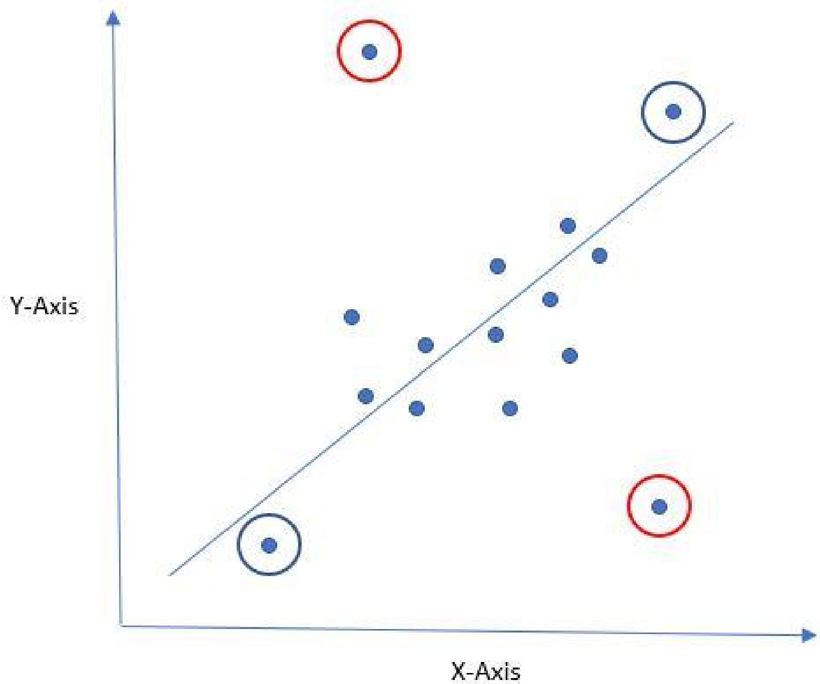

Figure 16: Best fit Model with Potential Outliers

Noticeable outliers can increase the residuals values and potential error in the system. By estimating unknown positions by measurements that are outside the normal ranges, the estimated values can be skewed incorrectly to the biassed measurements. As a result, the final estimated positions are not as accurate as they should be. In the case of the example figure, the observations that were circled in red were noticeable outliers (Figure 17). They were far from the best fit line and from clusters of other measurements. Due to this reason, removing such measurements can greatly aid in improving the estimated positional accuracy.

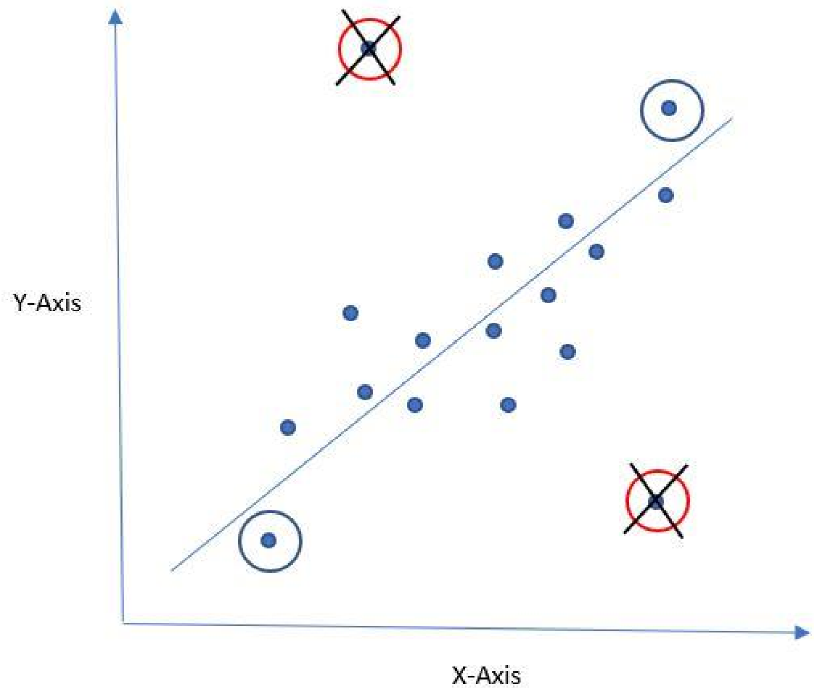

Figure 17: Best fit Model with Outliers Removed

Removing too many measurements by blindly looking at the residual values can end up having a negative impact on the quality of the measurement result. In the case of the example figure, let the two debatable circled outliers far from other measurements be removed and the position of the best fit line be estimated again (Figure 18). As a result, the estimation computation will focus more heavily on the measurements at the centre of the best fit line.

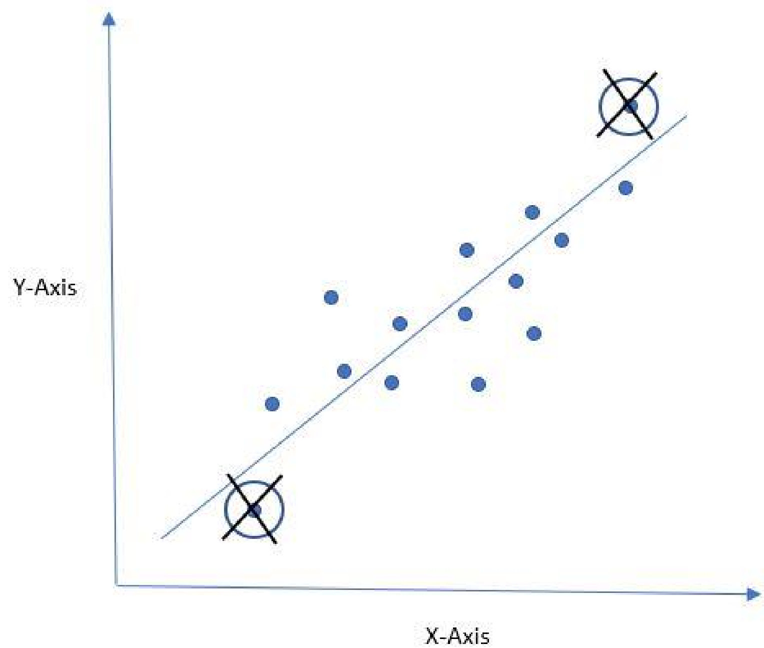

Figure 18: Best Fit Model with Potential Outliers Removed

In such a case with the two points at the ends of the line removed, the removal of the points can actually create a bias in the estimation system. The two points may have higher residuals but indicate the trend pattern of the estimation. By blindly removing such points, the team would end up having really small residuals for the points in the centre of the graph around the centre of the line. However, the tails of the line could shift one way or the other since there would be no reference points to use in the estimation process (Figures 19, 20). Therefore, the low residuals would reflect a false sense of accuracy. Even though the residuals fit really well for a certain estimated positions, other positions along the route could be estimated really badly.

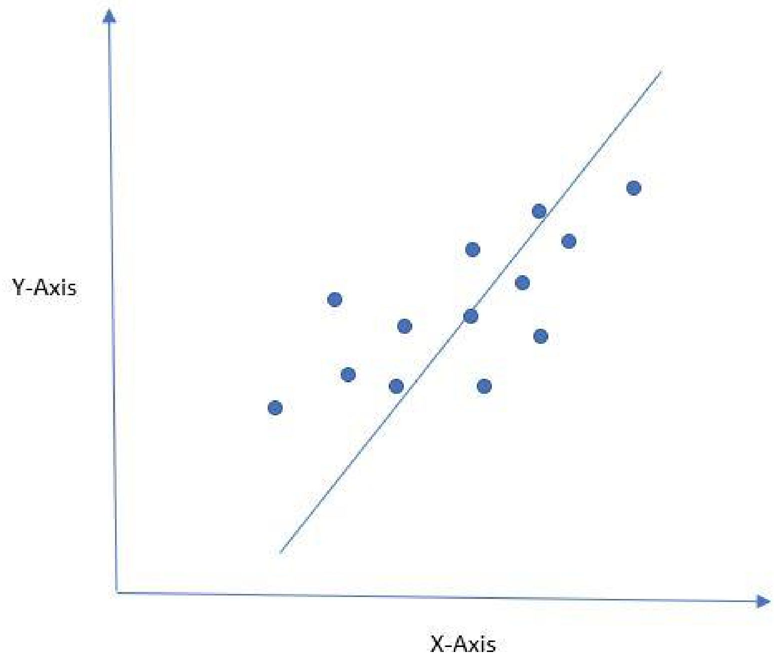

Figure 19: Impact of Removing Too Many Outliers 1

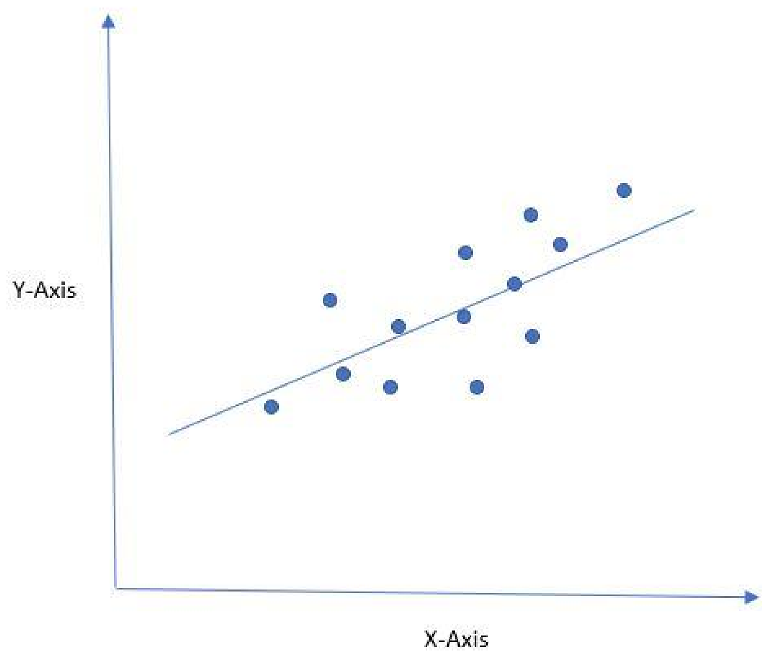

Figure 20: Impact of Removing too many Outliers 2

Due to the concepts illustrated in the previous figures, the team decided to compromise on including more measurements with higher residuals. Such a decision had the potential for increasing residual values and having measurements not fit the model equation quite as well.

Nevertheless, having more measurements ensured that important observations showing the path of travel were not removed and would end up likely improving the overall accuracy of the estimation of the position. After careful consideration, the team decided to remove all observations that had residuals value over an absolute value of 20 m. Generally, the majority of the residuals were below an absolute value of 20 m. Therefore, the removal of such measurements was not expected to impact key measurements or compromise the quality of the estimation.

The group established two different thresholds by which residuals would be removed. The first threshold was for observations with residuals that had an absolute value greater than 20 m. In this case, the observation itself within the time epoch would be removed. As a result, the epoch would still be kept in the estimation process with one less observation. On the other hand, a second threshold was set for residual values greater than 45 m. In such a case, the measurement was deemed to be such an obvious outlier that the entire epoch with its respective observations was removed. This was done so that the very poor measurement would not skew the result toward error.

Testing Process

In order to validate the measurement design, a series of tests were run on the data. Since the Pixel smartphone files did not work with GPS or Galileo with the L5 frequency, the Samsung smartphone files were used in the testing process. Three different days were tested (Table 7). The positions on April 28, 2021 were taken along the same route on a main street. Meanwhile, the positions on April 15, 2021 were taken along the highway for some of the route. Similarly, the positions on April 22, 2021 were observed in a very urban setting with skyscrapers and other tall buildings.

Table 7: Smartphone Data Usage

Phone	Date	Constellation	Navigation	Observation
Samsung	April 28, 2021	GPS	brdm1180.21p	SamsungS20Ultra_GnssLog.21o
Samsung	April 28, 2021	Galileo	brdm1180.21p	SamsungS20Ultra_GnssLog.21o
Samsung	April 28, 2021	Galileo And GPS	brdm1180.21p	SamsungS20Ultra_GnssLog.21o
Samsung	April 15, 2021	Galileo	brdm1050.21p	SamsungS20Ultr

				a_GnssLog.21o
Samsung	April 15, 2021	GPS	brdm1050.21p	SamsungS20Ultr a_GnssLog.21o
Samsung	April 15, 2021	Galileo and GPS	brdm1050.21p	SamsungS20Ultr a_GnssLog.21o
Samsung	April 22, 2021	GPS	brdm1120.21p	SamsungS20Ultr a_GnssLog.21o
Samsung	April 22, 2021	Galileo	brdm1120.21p	SamsungS20Ultr a_GnssLog.21o

Before the main testing procedure, the positions were plotted on Google Earth to visually verify if they were along the roadways. In the same way, the positions were tested with and without ionospheric corrections for the single and multiple frequencies. Next, the program was verified with the removal of large residuals (Figure 21). The process was outlined in the figure below:

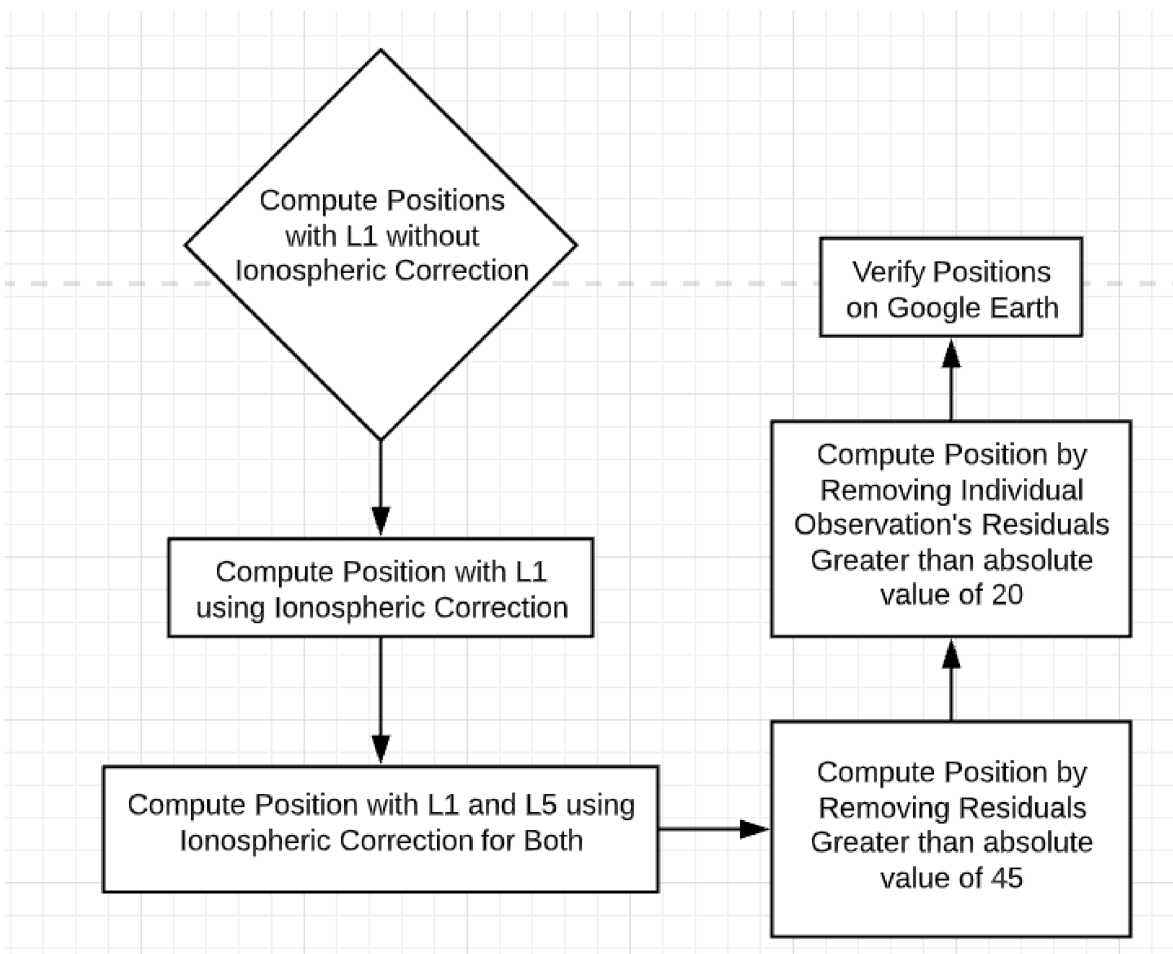

Figure 21: Computation and Verification Process of Positions

The testing procedure also tested different position estimation steps in the algorithm. One of the first tests was using different observation files with the same navigation file on the same day (Figure 22). Afterwards, multiple days were tested with different observation files on different days with the appropriate navigation files. The different observation and navigation files were tested with just one global constellation and then with the combined constellation program with both GPS and Galileo measurements. By doing so, the quality of the program was able to be validated.

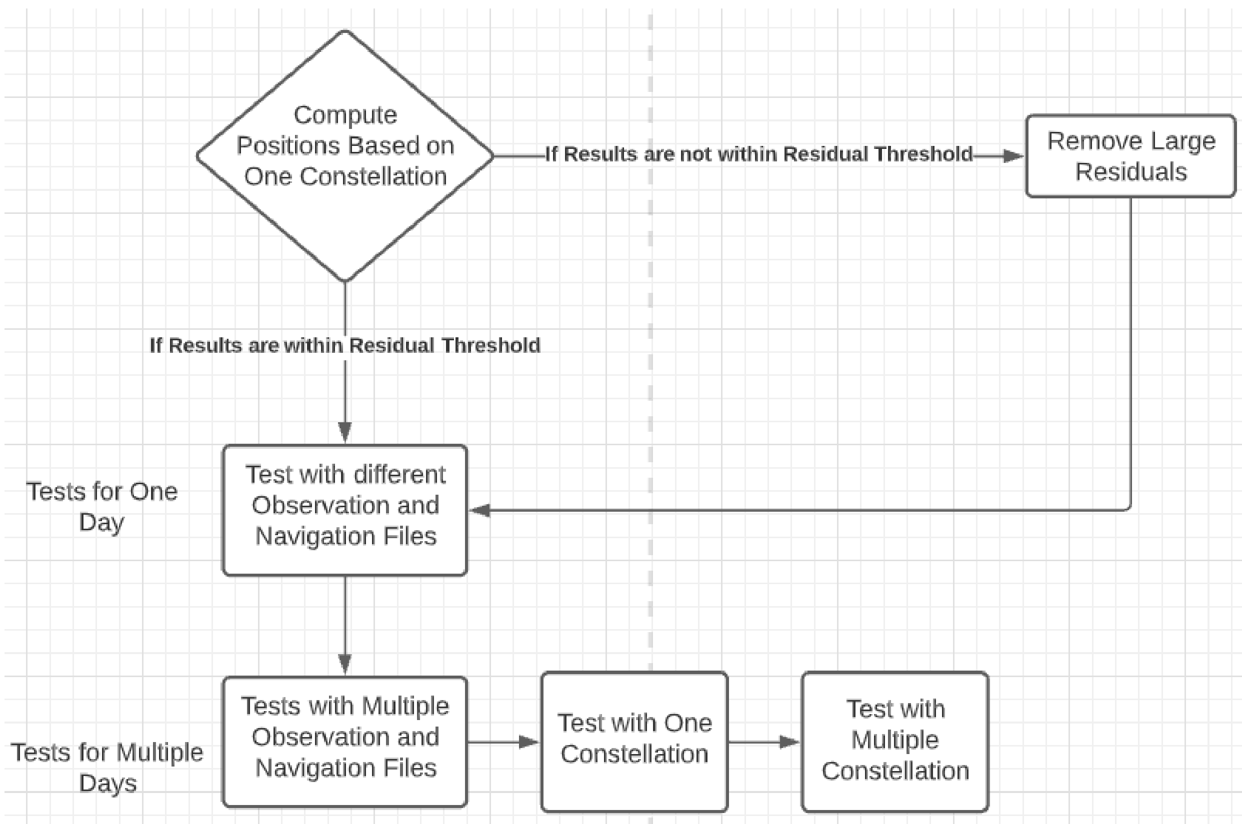

Figure 22: Reapplying Residual Thresholds over Days and Constellations

11 Project Results

The final estimated coordinates were computed based on the WGS84 reference ellipsoid in the geodetic coordinate system. All estimated positions were for each epoch and were shown in terms of latitude, longitude, and height. Likewise, the epoch was also stored along with the other values. The following figure showed the structure of the final results (Table 8). Each individual row represented an estimated smartphone position at each epoch.

Table 8: Smartphone Positions at each Epoch

	1 Lat	2 Long	3 Height	4 Epoch
1	37.3959	-122.1027	33.1049	1.3037e+09
2	37.3958	-122.1027	31.5307	1.3037e+09
3	37.3958	-122.1028	21.0560	1.3037e+09
4	37.3958	-122.1028	18.4280	1.3037e+09
5	37.3959	-122.1029	13.7583	1.3037e+09
6	37.3958	-122.1028	23.1497	1.3037e+09
7	37.3958	-122.1028	24.6560	1.3037e+09
8	37.3958	-122.1027	29.8769	1.3037e+09
9	37.3959	-122.1027	29.6597	1.3037e+09
10	37.3958	-122.1027	35.4624	1.3037e+09
11	37.3958	-122.1027	38.7064	1.3037e+09
12	37.3958	-122.1027	40.8936	1.3037e+09
13	37.3958	-122.1027	32.7378	1.3037e+09
14	37.3958	-122.1027	33.2469	1.3037e+09
15	37.3958	-122.1028	31.5156	1.3037e+09

To get a better understanding of the quality of the data, the positions were computed from the ECEF geodetic system to easting and northing in the UTM coordinate system. Similarly, the reference coordinates were converted into the UTM coordinate system as well. This was done so that the differences between the estimated and reference coordinates could be recorded in metres. Consequently, the team could understand how the error levels were distributed across all epochs.

In order to show the improvement of the results with the introduction of multiple constellations and residual removal, the measurements results were shown incrementally. The distribution of the positional errors in the Easting and Northing, the Dilution of Precision (DOP), the residual plots, and the number of satellites were displayed for each incremental measurement shown. Similarly, images from Google Earth of the estimated positions were also shown and evaluated.

12 Google Earth Images And Discussion

The resulting positions were plotted in Google Earth for three different dates. The first date was April 28, 2021 along a residential roadway (Figure 23a). While there were buildings on either side of the road, most of the buildings were not too tall. The observation data on April 28, 2021 was the main file that was used for testing as the results were fairly stable.

Figure 23a: Mountain View, California (April 28, 2021)

When the estimated smartphone positions on the road were examined, they were fairly reasonable (Figure 23b). Most of the points were along the roadway and within the lanes. When zooming further into Google Earth, the estimated positions continued to be located within the lanes. Overall, the estimated positions of the smartphones appeared to be realistic and appropriate.

Figure 23b: Mountain View, California (April 28, 2021)

While positions observed on April 28, 2021 were fairly good, there were some issues with some plotted positions. At times, there were issues with some estimated positions being clustered near taller buildings (Figure 23c). Such a result was likely due to the impact of multipath. Multipath occurs when the GNSS signal does not directly reach the GNSS receiver on the smartphone but bounces and reflects off of other obstructions before reaching the smartphone. Therefore, such measurements were of a poor quality. There was not much the group could do to improve the poor measurements. The group could only work with the measurements and attempt to eliminate outliers. As a result, the clustering of estimated positions around tall buildings resulted in a compromise in the positional accuracy of the estimated smartphone positions.

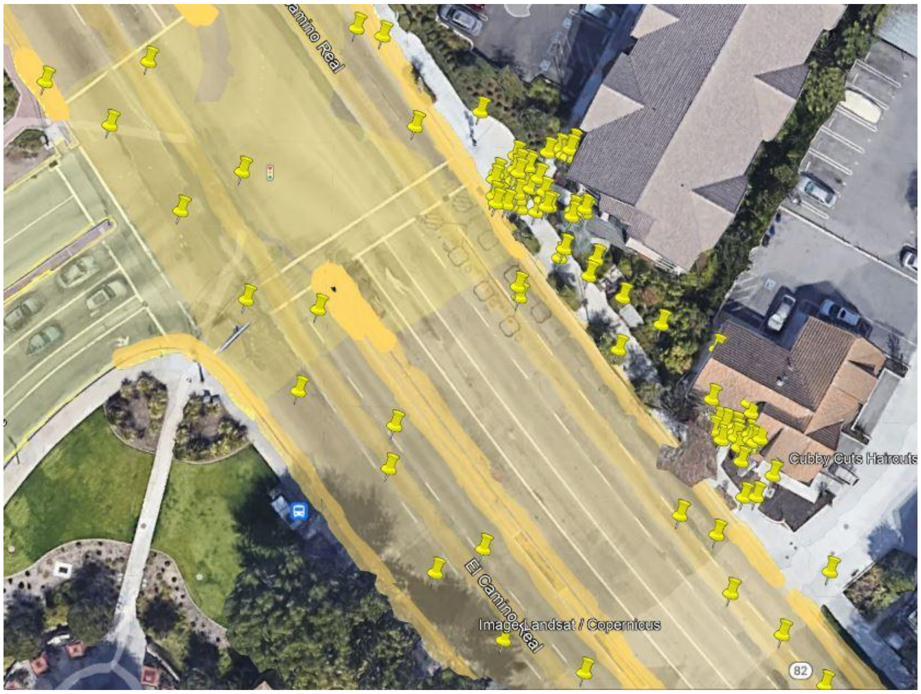

Figure 23c: Mountain View, California (April 28, 2021)

While the estimated positions along the roadways were fairly good, there appeared to be a slight locational bias toward the sidewalk at times. In the figure below, some of the points were drifting toward the sidewalk (Figure 23d). The building right along the sidewalk would likely have influenced this due to the impact multipath has as discussed in the previous paragraph. Nevertheless, most of the points were located along the roadway. Even at the current resolution of the Google Earth software, the points were visibly located within the lanes. Despite the issue with the estimated positions being biassed toward the sidewalk at times, the estimated points along the roadway were reasonable.

Figure 23d: Mountain View, California (April 28, 2021)

While there were sections of the street that were fairly open, some areas were partially covered by trees. Trees can present similar issues related to multipath as tall buildings. Nevertheless, the estimated positions along some portions of the road in the midst of trees were fairly reasonable (Figure 23e). As displayed in the figure below, the estimated positions were mostly along the roadway. The estimated positions were in a fairly straight lane and did not waver too much in between the tree pattern. The quality of the estimated positions in this section of the road was likely improved by there not being too many tall buildings around. While trees decrease the quality of GNSS receiver measurements, there is a greater potential for a GNSS signal to reach the receiver through a porous membrane of leaves on trees rather than through a solid building. Consequently, the accuracy of some estimated positions was maintained even in the presence of increased tree coverage.

Figure 23e: Mountain View, California (April 28, 2021)

While some sections of the roadway covered with trees did not negatively affect the estimated smartphone positions along the roadways too much, other areas had a worse impact. In a residential section with only two lanes for opposite travel, the estimation accuracy grew worse (Figure 23f). The estimated positions bounced around on the sidewalk or clustered up in certain areas. This was likely due to multipath from the tree foliage and the many buildings present along the roadway. The fact that the number of lanes decreased likely had an impact on the poorer quality of some measurements. As the lanes decreased, the open sky space above the street also decreased. This meant that the buildings and trees on the sides of the road had a greater likelihood of blocking or interfering with GNSS signals reaching the receiver from satellites. Therefore, the number of lanes in the roadway seemed to have an impact on the quality of the estimated smartphone positions along the roadway.

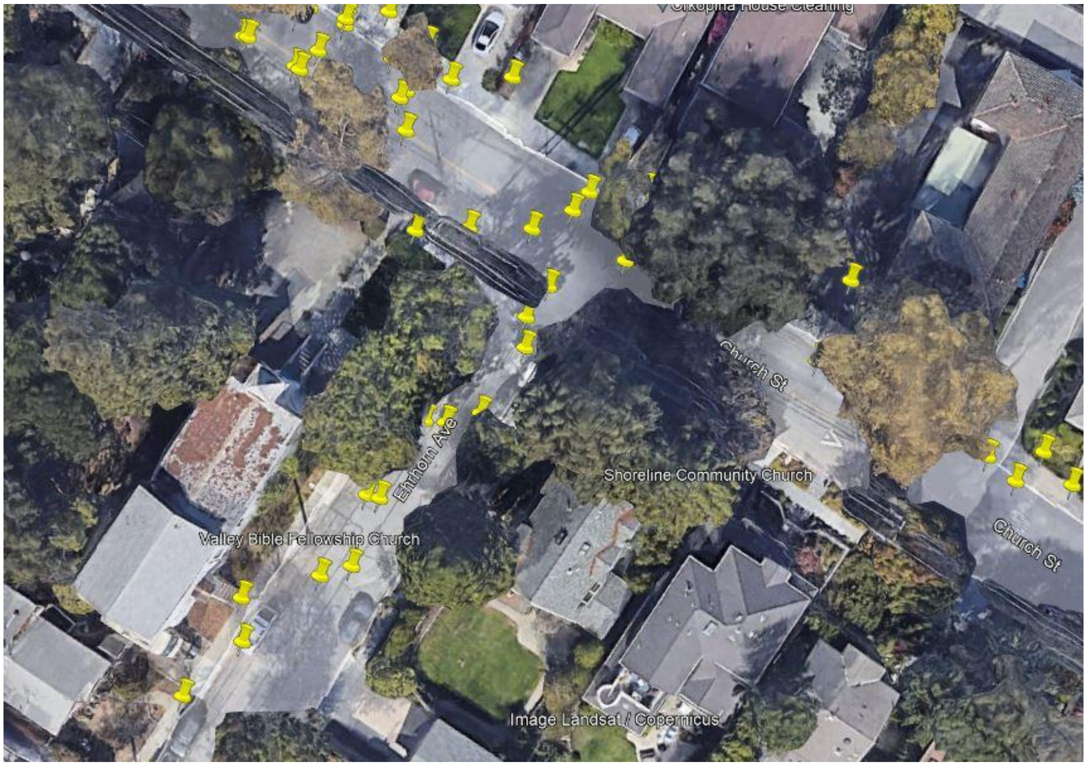

Figure 23f: Mountain View, California (April 28, 2021)

In order to better understand the quality of the relative position augmentation algorithm, observations taken on April 15, 2021 were processed. The data was taken along a longer stretch of highway (Figure 24a). Roadways alongside the highway were also travelled along within some on and off ramps to the freeway. A small amount of travel was done in urban areas with buildings around. However, the main focus of using the observation file taken on April 15, 2021 was to understand the quality of the estimated positions along an open highway.

Figure 24a: Mountain View, California (April 15, 2021)

When examining the estimated positions along the highway, the quality of the smartphone positions appeared to be fairly good. The estimated positions were along the highway and reasonably within lanes along the highway (Figure 24b). Furthermore, the estimated positions along East Bayshore Road looked quite good. The plotted positions were within the roadway. One area of interest was the locations of the estimated positions within the lanes. There appeared to be some differences in the lane location between some estimated positions located one after the other. While this may have been due to the lane changes, the variation could have been from the actual estimated positions changing by a metre. As a result, it appeared that the vehicle carrying the smartphone changed lanes when it really did not. The quality of the estimated positions on West Bayshore Road appeared to be lower. While some positions were along the roadway, others were more biassed toward the sidewalk. Such a result was likely due to the taller buildings located nearby.

Figure 24b: Mountain View, California (April 15, 2021)

When examining a location at the junction of a street and an on-ramp to the freeway, the quality of the estimated smartphone positions were varied. There appeared to be some unusual clustering of estimated positions around the curve toward the on-ramp that likely was not natural (Figure 24c). While the car could have been idling due to a red light, there was probably some error from poor quality measurements in this location. Furthermore, some of the estimated positions along the curve in the on-ramp to the freeway appeared skewed toward the side. Such a result could have been from the speed at which the car was travelling along a curve. As a result, the satellite timing and positioning were slightly thrown off. Nevertheless, the estimated positions of the smartphone within the vehicle on the street appeared to be fairly reasonable.

Figure 24c: Mountain View, California (April 15, 2021)

Along the route taken on April 15, 2021 on the roadway in the urban setting, the estimated positions were not too poor. With all of the buildings and vertical obstructions, the group expected the results to be quite poor. While the quality of the measurements was not the best, the estimated positions were mostly within the roadway (Figure 24d). The positions tended to float across the roadway as can be seen in the image below. Nevertheless, the estimated positions were in the roadway and were accurate to the metre level.

Figure 24d: Mountain View, California (April 15, 2021)

Another dataset provided by Kaggle was examined by the group. The data was taken on April 22, 2021 in downtown San Jose (Figure 25a). As a result, the roadways were mostly surrounded by tall structures. The group included these observations in the testing process in order to determine how well the accuracy of the estimated positions were maintained when the measurements suffered from high levels of multipath.

Figure 25a: Mountain View, California (April 22, 2021)

When Google Earth was zoomed in within the downtown area, the estimated positions were quite poor. The positions were clustered around some tall buildings and fluctuated in position across the street (Figure 25b). All estimated positions were not in a straight line along the roadway and tended to trend toward the sidewalks. Some were even estimated off of the roadways in buildings. Such poor measurements resulted from multipath.

Figure 25b: Mountain View, California (April 22, 2021)

The estimated positions of the smartphones were not even on the roadway at times. Estimated positions were on sidewalks, parking lots, and even within surrounding buildings (Figure 25c). Consequently, the algorithm was rather limited in highly urbanised areas with many tall structures.

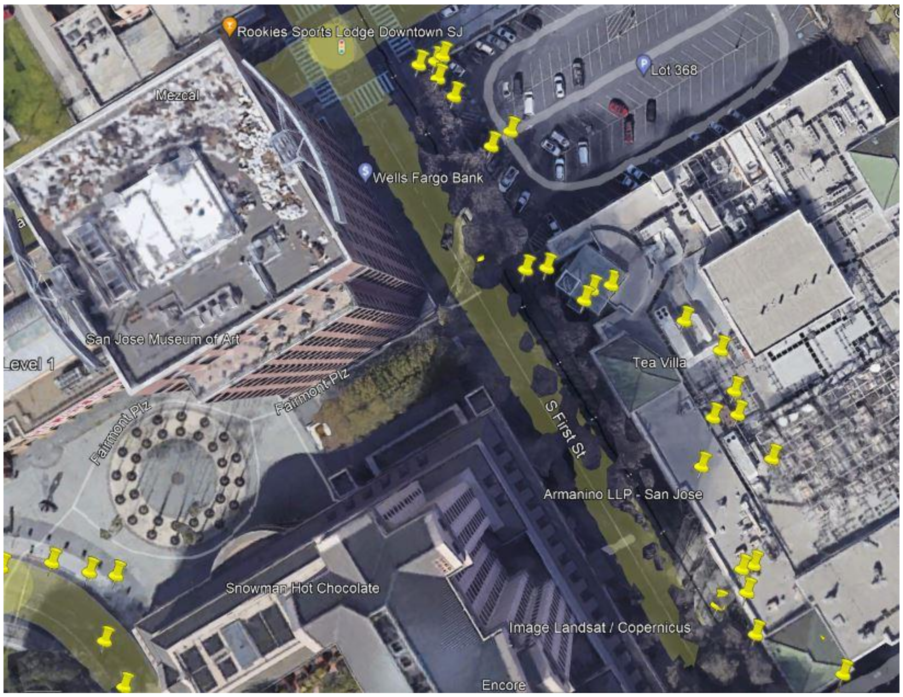

Figure 25c: Mountain View, California (April 22, 2021)

The error due to multipath was further compounded by there being trees along some portions of the roadway (Figure 25d). As a result, GNSS signals that made it through the buildings without high levels of multipath were further impacted by the trees along the roadways. An important aspect of the algorithm to comprehend in light of multipath was that the algorithm was only able to determine positions based on the measurements available. If the measurements were poor, the algorithm would generate a lower quality solution. In real time navigation, it is very challenging to verify if the estimated position is correct using sources outside of outliers within the measurements. If all of the measurements were technically outliers, the algorithm was not able to detect this as it compared the measurements within themselves to each other.

Figure 25d: Mountain View, California (April 22, 2021)

The negative impact multipath had on the high accuracy positioning of the smartphone was further evidenced by testing an area with fewer tall buildings. Along this section of the roadway, many more estimated positions were along the roadway and within the lanes (Figure 25e). This was likely due to there being more higher quality measurements from the satellites along the road with more open skies. Therefore, multipath had a significant negative impact on the estimated positions taken on April 22, 2021.

Figure 25e: Mountain View, California (April 22, 2021)

13 GNSS Result And Analysis

For the solution just using GPS measurements, there were a fairly reasonable number of satellites present in the system at each epoch. In order for the least square estimation process to work, there had to be at least five satellite measurements at each epoch. This was the case since there were four unknown parameters that had to be estimated and one more to introduce redundancy into the system. As there were at least five satellites at each epoch, the estimation system was overdetermined (Figure 26).

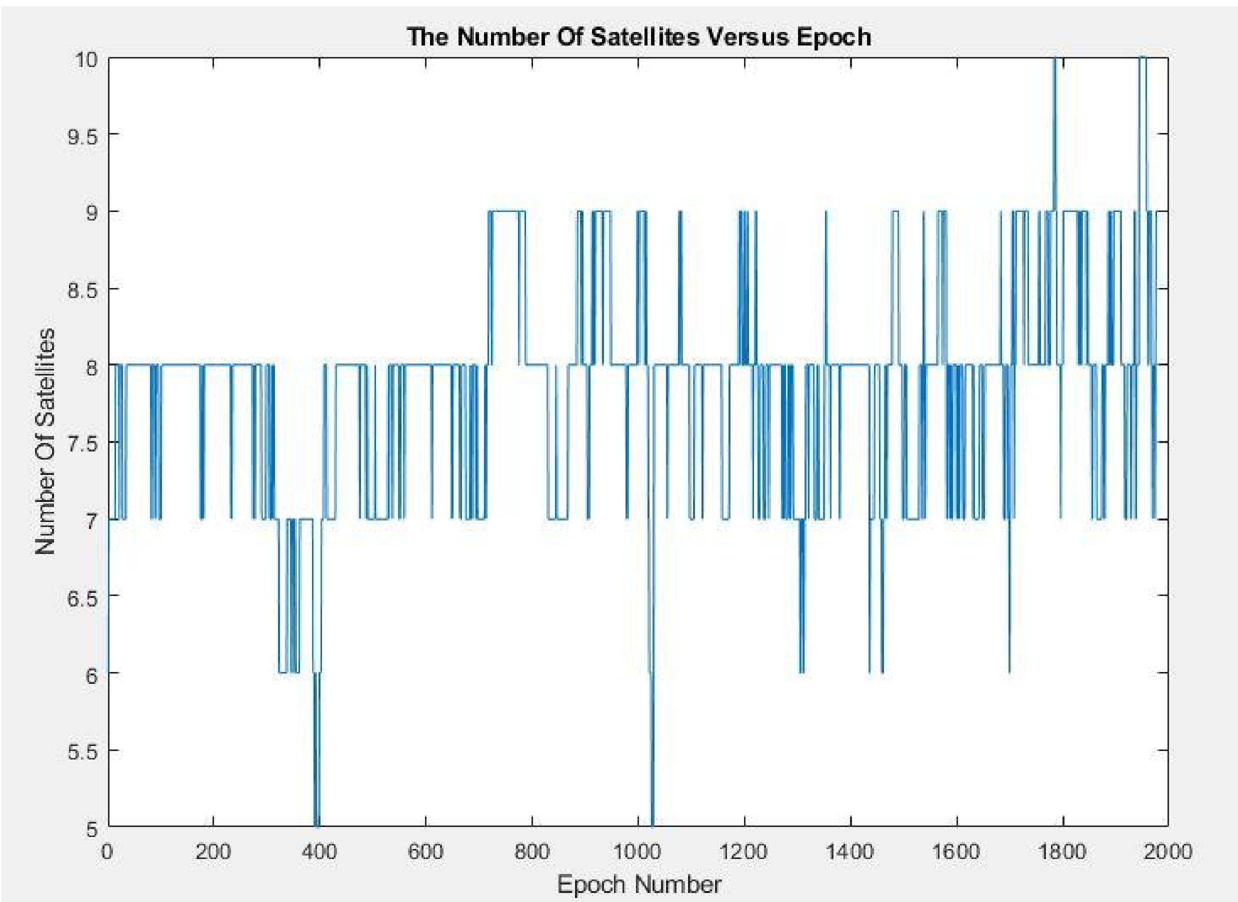

Figure 26: Dilution of Precision Plot for Satellite Coverage

The DOP plot of the GPS measurements was fairly appropriate. In order to have a reasonable level of precision in the system, the DOP has to be below 7. Looking at the following plot, the DOP values were all below 7 (Figure 27). There were particular spikes at some epochs, notably around epochs 1020 and 1710. This could indicate that the measurements made at these times were of a poorer quality. Nevertheless, the DOP plot appeared to be reasonable.

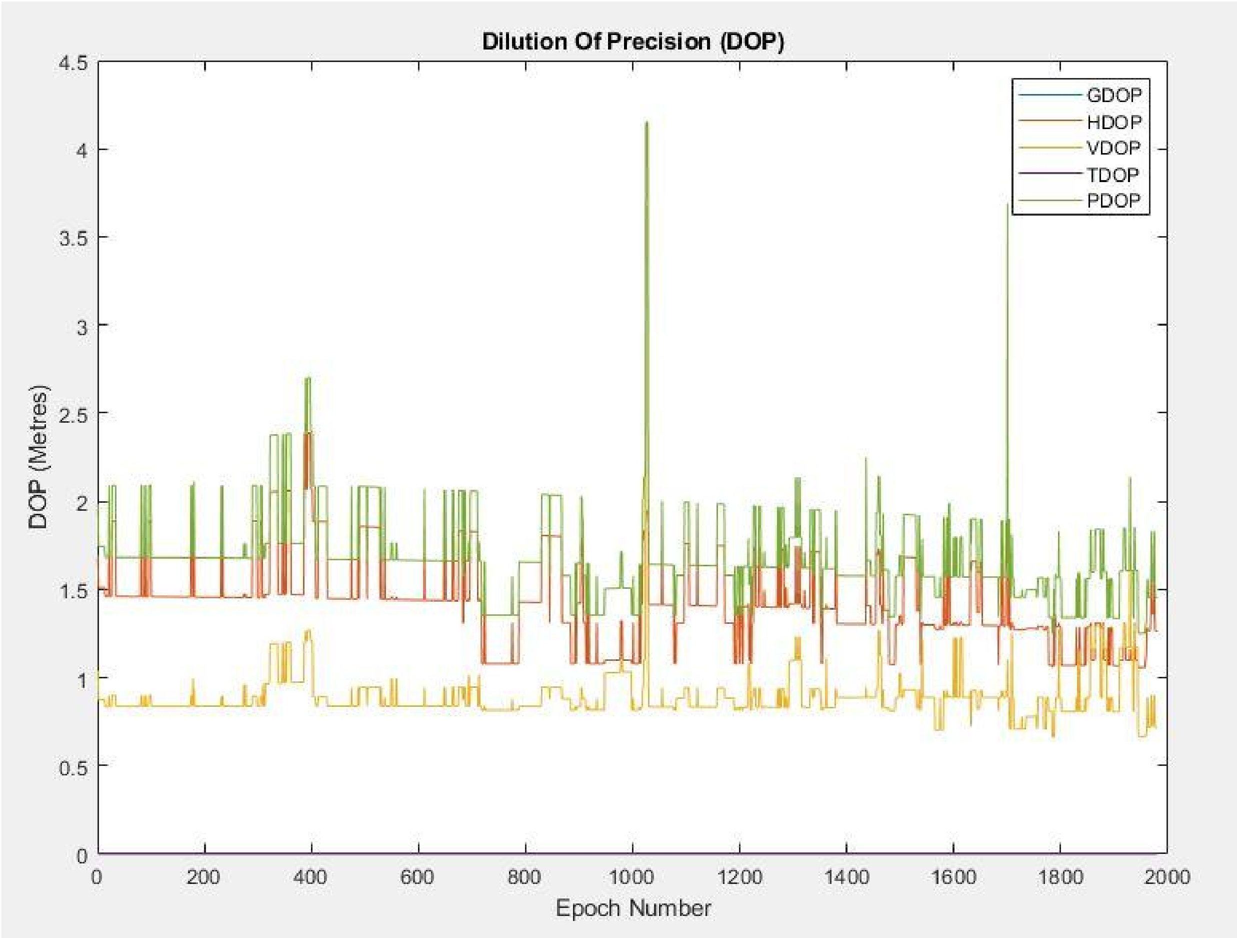

Figure 27: Dilution of Precision Plot across Geometric, Horizontal, Vertical, Time, and 3D Precision

While the DOP plot was fairly reasonable, the GPS positional errors were larger than expected. The average positional error in the easting seemed to be around 10 m (Figure 28). There were noticeable spikes around epochs 350, 530, 1160, 1440, 1650, and 1970. Overall, the error levels in the easting were larger as just GPS measurements were involved in the estimation process.

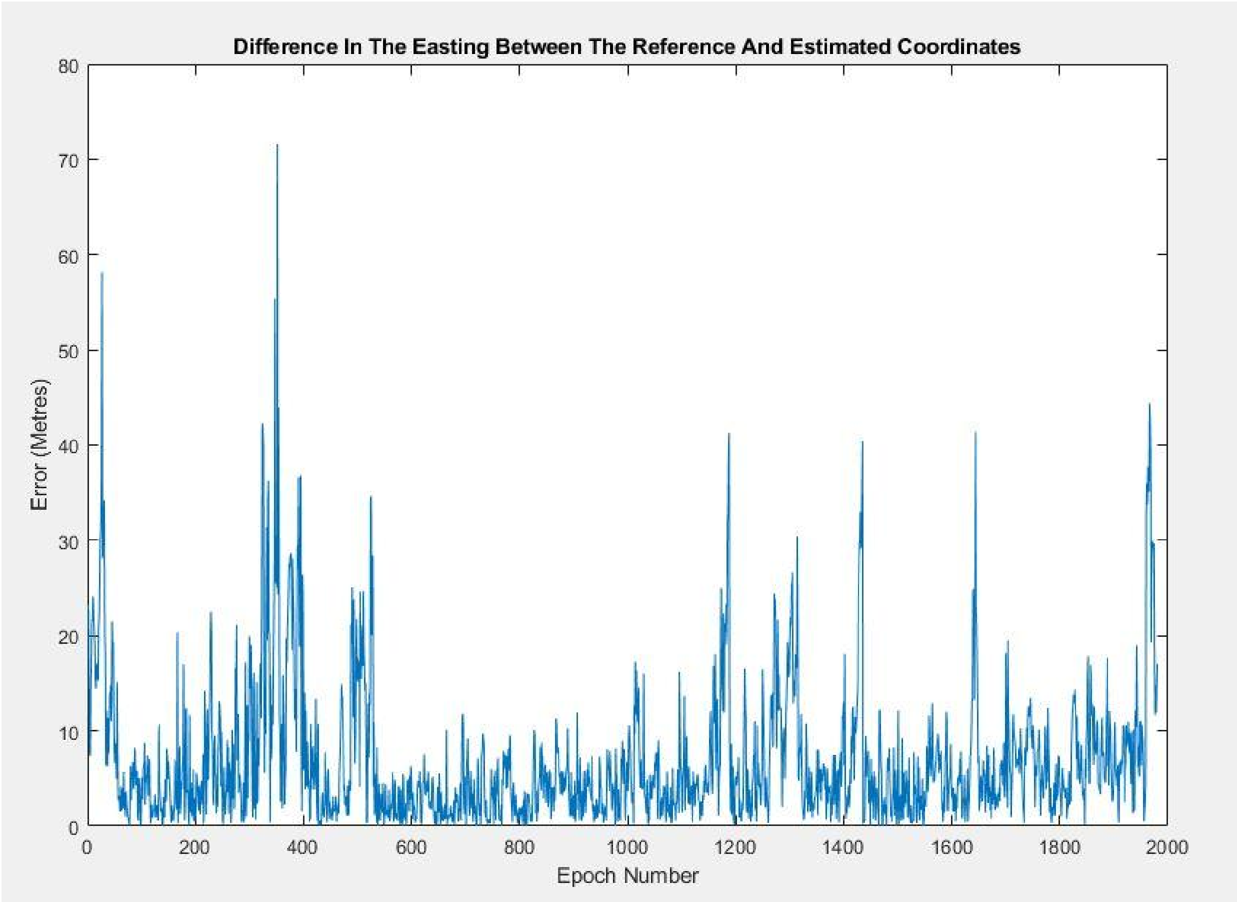

Figure 28: Dilution of Precision Plot over Eastings

The GPS error in the northing between the reference and estimated coordinates was similarly larger than expected. While the error in the easting had larger peaks of error, the overall northing errors appeared to be slightly larger than those in the easting (Figure 29). Based on the plot of the errors, there appeared to be a mean error around or just above 10 m. There were noticeable spikes around epochs 350, 520, 1130, 1310, 1640, 1810, and 1910. Therefore, more improvements had to be made to ensure that the error levels were at the metre level.

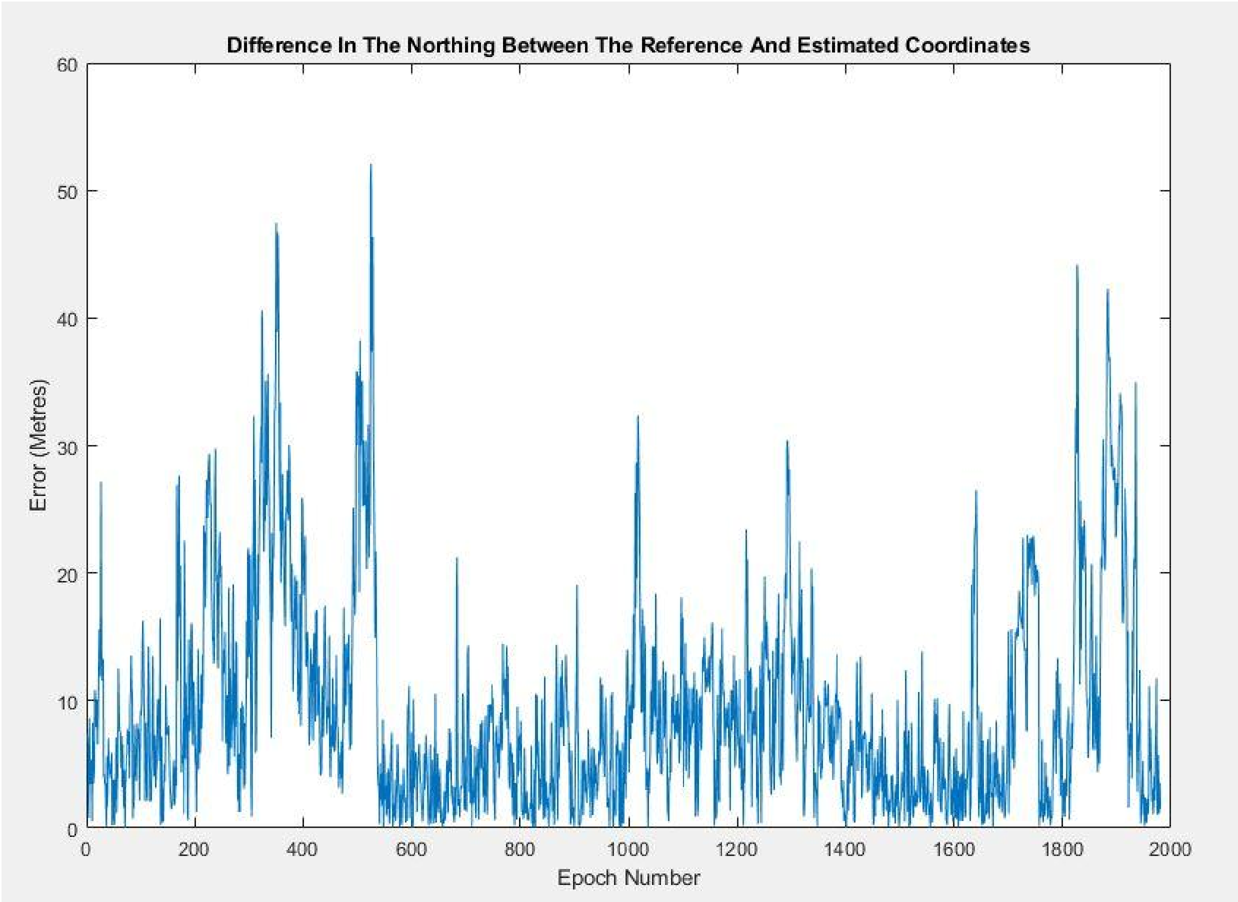

Figure 29: Dilution of Precision Plot over Northings

The GPS residuals were also plotted so that the team could have a better understanding how the estimated values fit the measurement model. Residuals provide a metric of how close the least squares estimated values are to the measurement equation being used as a reference. After examining the residual plot, the majority of the residual values appeared to have an absolute value of 15 m (Figure 30). In the same way, there were about 30 to 40 observations that had residuals around or above 45 m. Therefore, the team decided that removing outliers could help the other measurements be more closely estimated to the measurement equation.

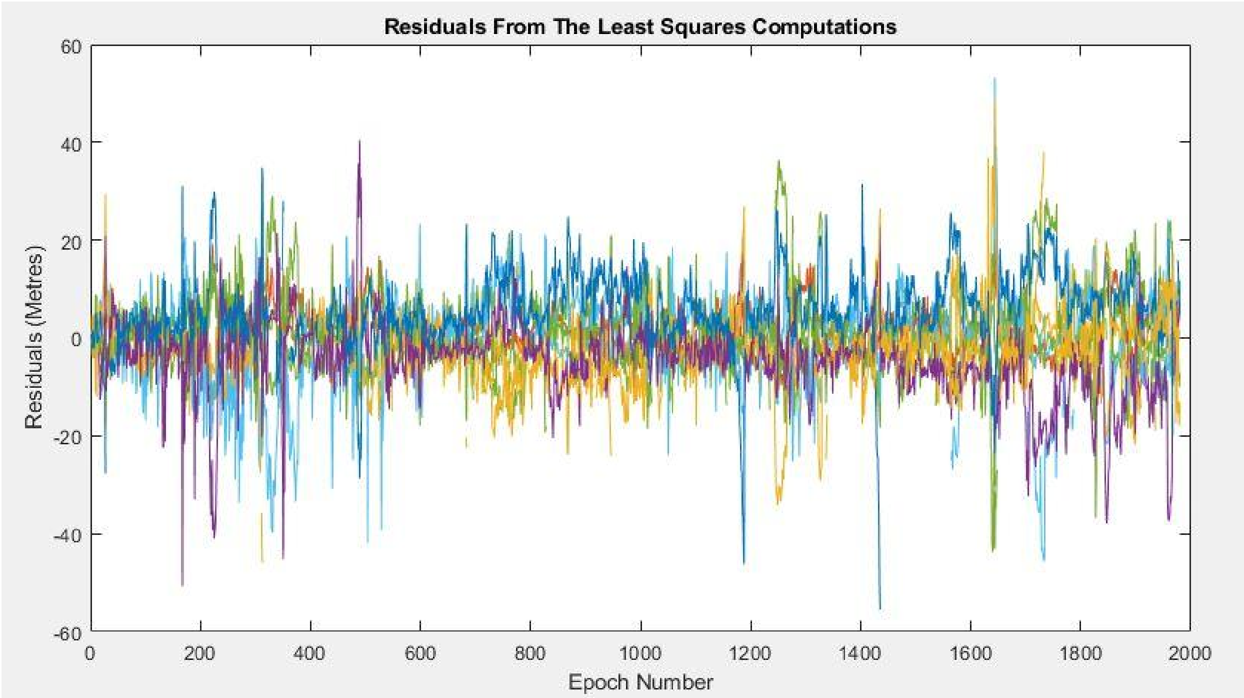

Figure 30: Metre Residuals across Epochs

GPS With Residual Outliers Removed Result

In the next iteration, the GPS measurement processing was completed again but observations with residuals with an absolute value of 20 m were removed. There were the exact same number of satellites as in the previous iteration as no satellites were removed (Figure 31). Consequently, there were an appropriate number of satellites at each epoch to have an overdetermined system.

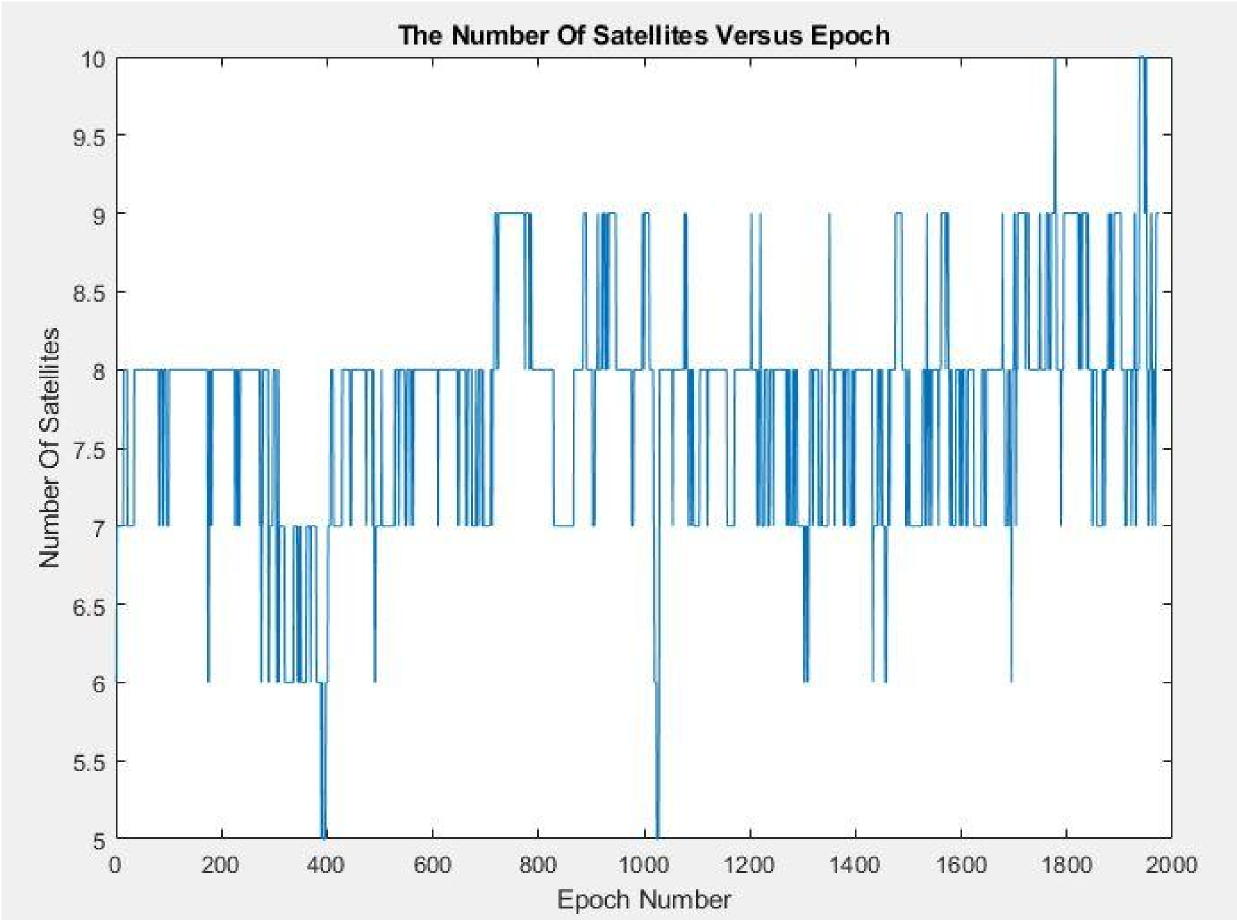

Figure 31: Number of Satellites at each Epoch

The DOP plot resulting from the precision levels in the system was fairly good. All of the DOP values at each of the epochs were below 7 (Figure 32). When comparing the DOP to the results from just processing GPS measurements, the DOP values were basically the same. As a result, removing outlier measurements from just GPS measurements did not appear to significantly improve the DOP.

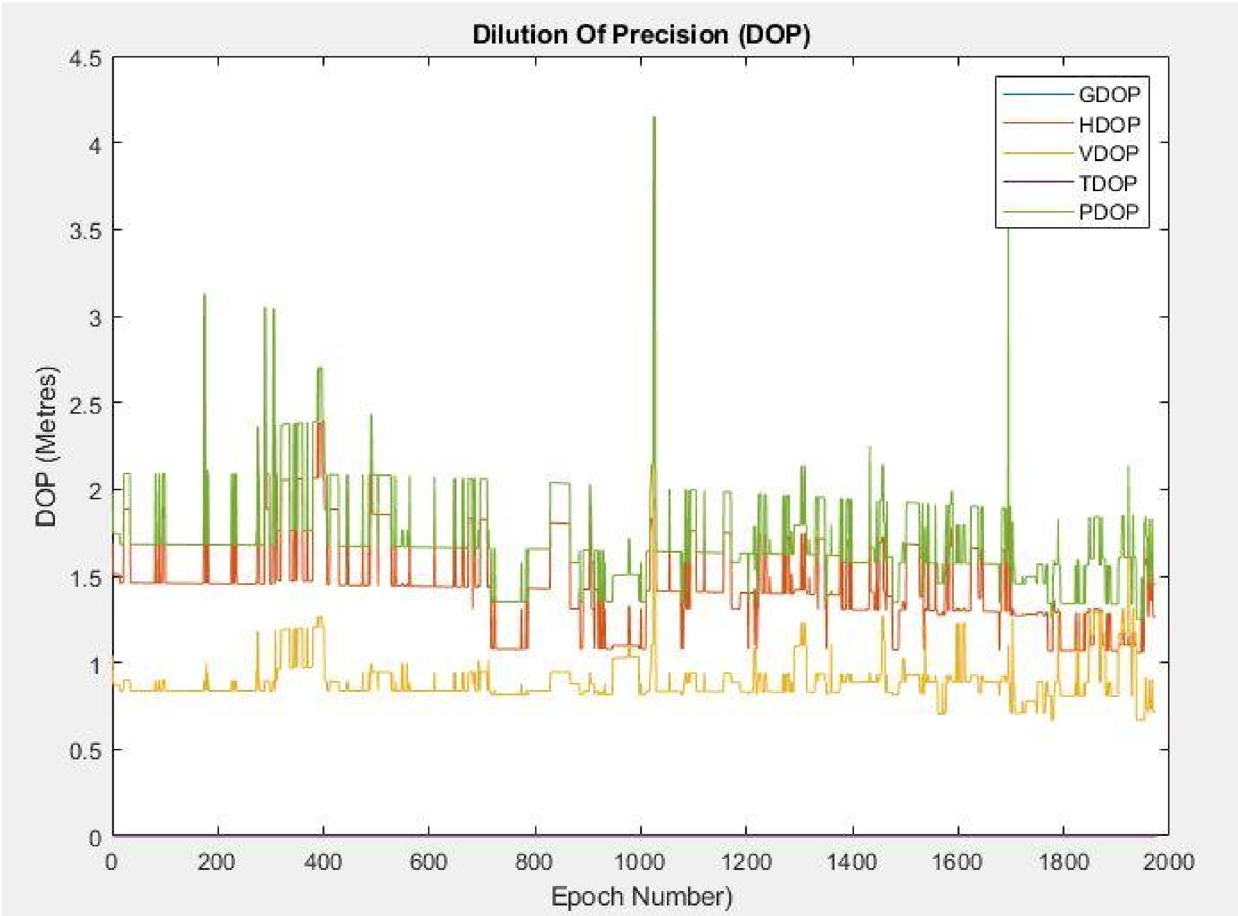

Figure 32: Dilution of Precision Plot across Epochs

The easting errors between the reference and estimated coordinates were still large. Some of the errors were actually greater than those just for GPS results (Figure 33). The peak of the errors seemed to be around the 350 epoch. On the other hand, the other errors appeared to be a bit more manageable. Much in the same way as with the GPS results, the average error result seemed to be between 10 m and 15 m. Nevertheless, the mean values were swayed by the large outliers.

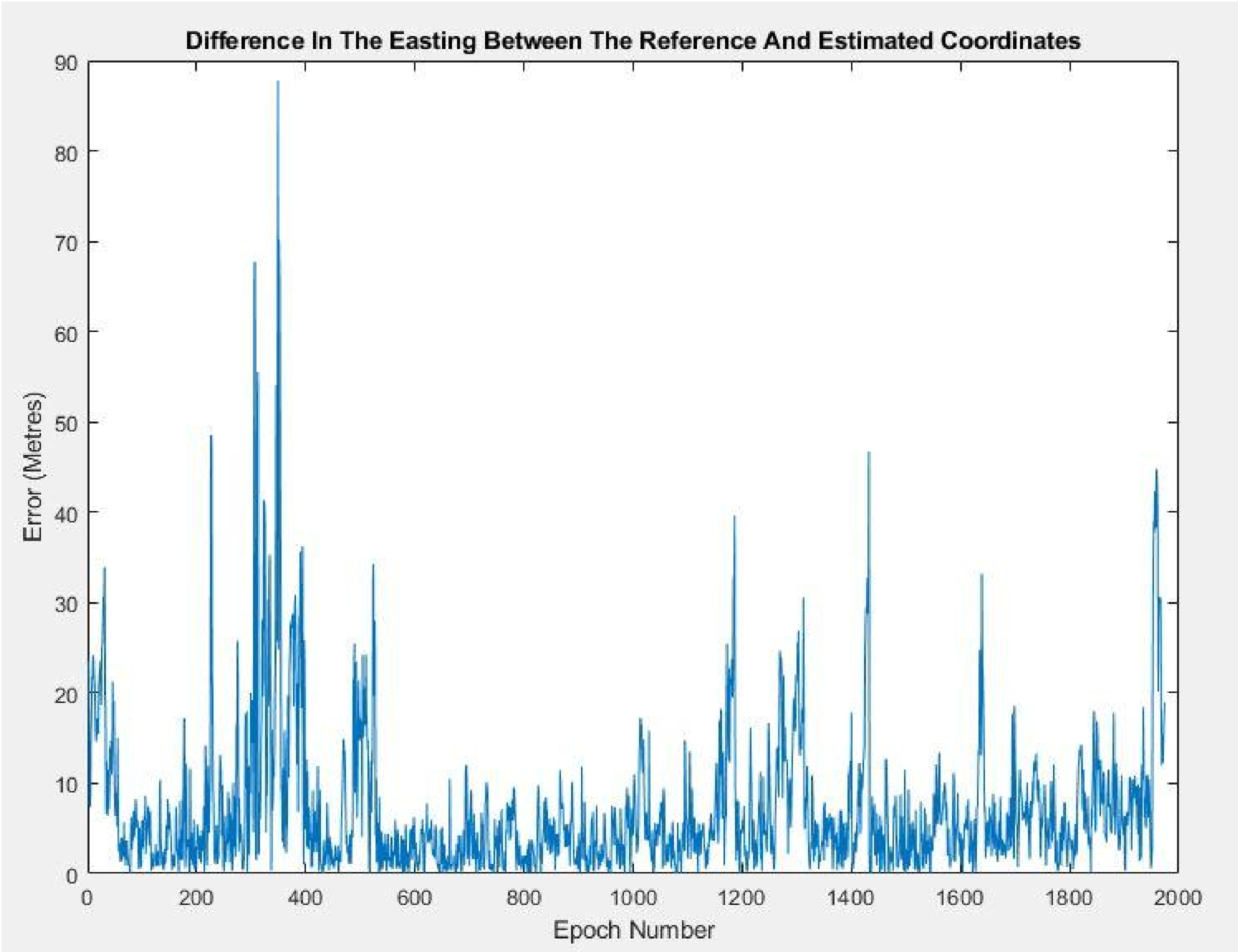

Figure 33: Easting Error

The differences in the northing between the reference and estimated coordinates had fewer significant error spikes than those in the easting. As a result, the removal of outliers seemed to have a poorer impact on the easting results than the northing results (Figure 34). Based on the estimated position errors, the group concluded that the estimation process with residual outlier removal did not have a significant impact on improving positional accuracy.

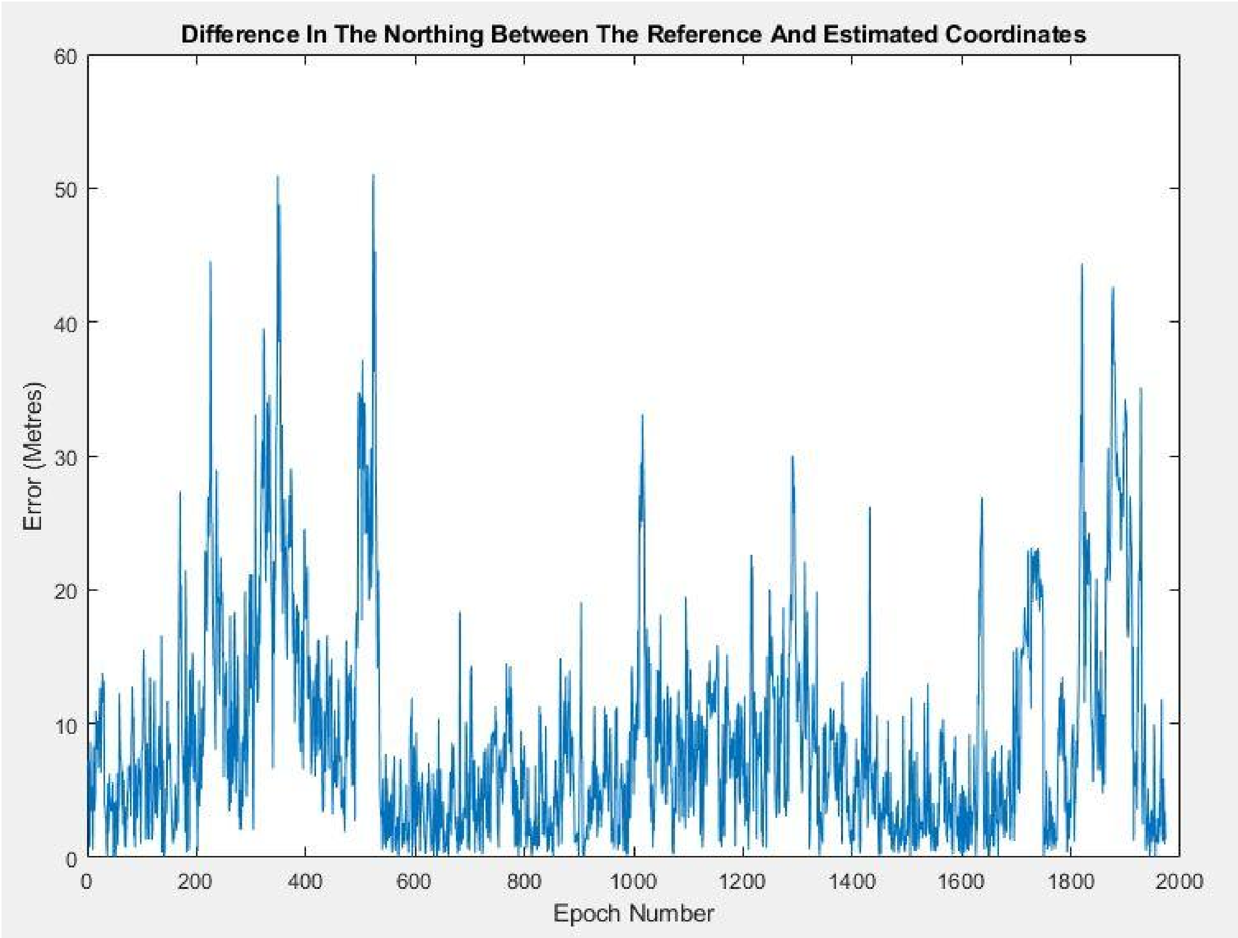

Figure 34: Northing Error

As was expected, the residual values somewhat improved. Major outlier residuals were noticeably removed when compared to the original GPS residuals (Figure 35). However, there were still outliers in the system. This was the case because of better approximation of the estimates to the model equation. As was explained earlier in the report, residual values will change each time values are removed. The residual values did decrease somewhat but not enough to have a significant positive impact on the system.

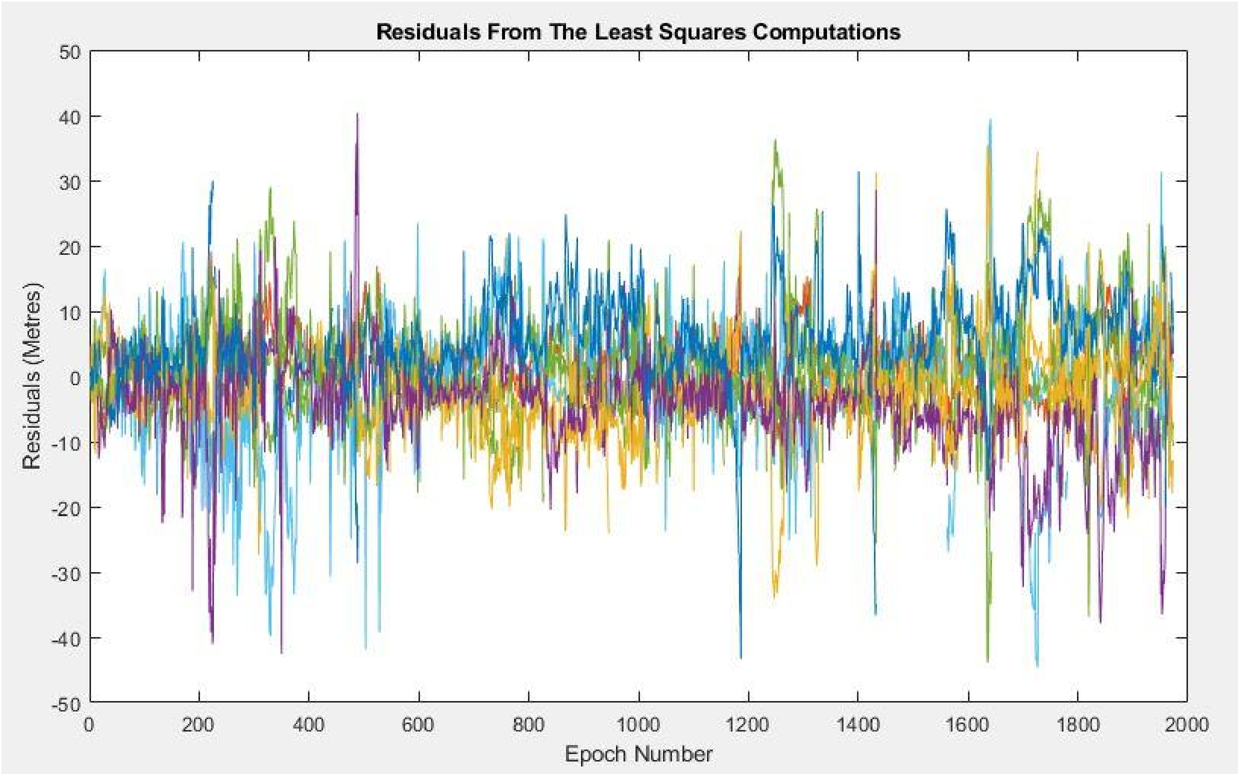

Figure 35: Residuals across Epochs after Least Squares Filtering

Galileo Result

When just the Galileo measurements were processed, the results were similar to those from the GPS processing. There were fewer satellites per epoch for Galileo which reduced the level of redundancy in the system (Figure 36). As a result, the overall estimation quality was lower. In the same way, Galileo is a newer global GNSS system. Consequently, it would be expected that there would be fewer satellites across the globe than for the older GPS system.

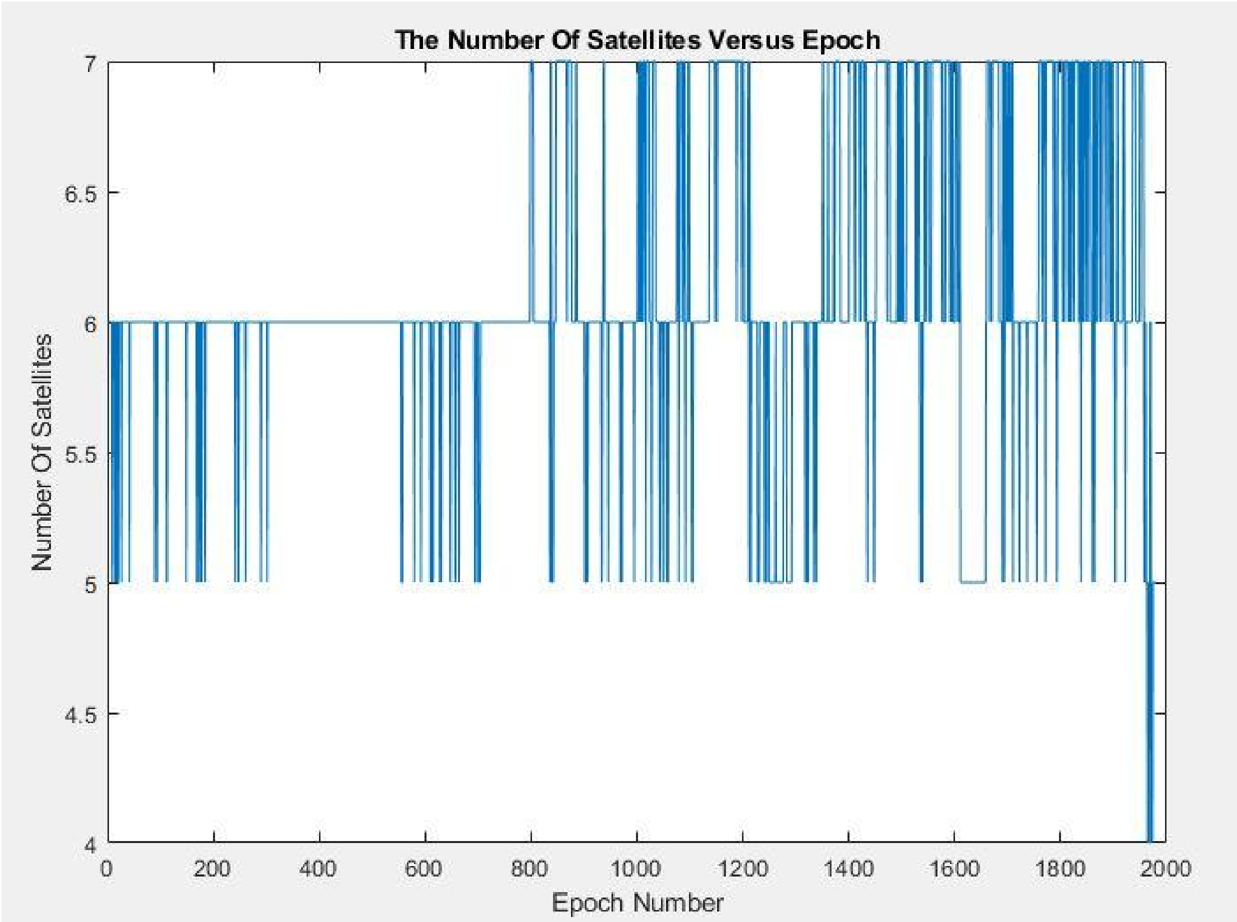

Figure 36: Available Galileo Satellites across Epochs

The DOP values for the positions estimated using only Galileo measurements were poorer than those with GPS. For most of the epochs, the DOP values were below 4 (Figure 37). In the last few epochs, the DOP values significantly spiked. This was likely due to there being only four satellites for a time epoch. In such a case, there was no redundancy in the least squares estimation measurement equation which increased the error in the system.

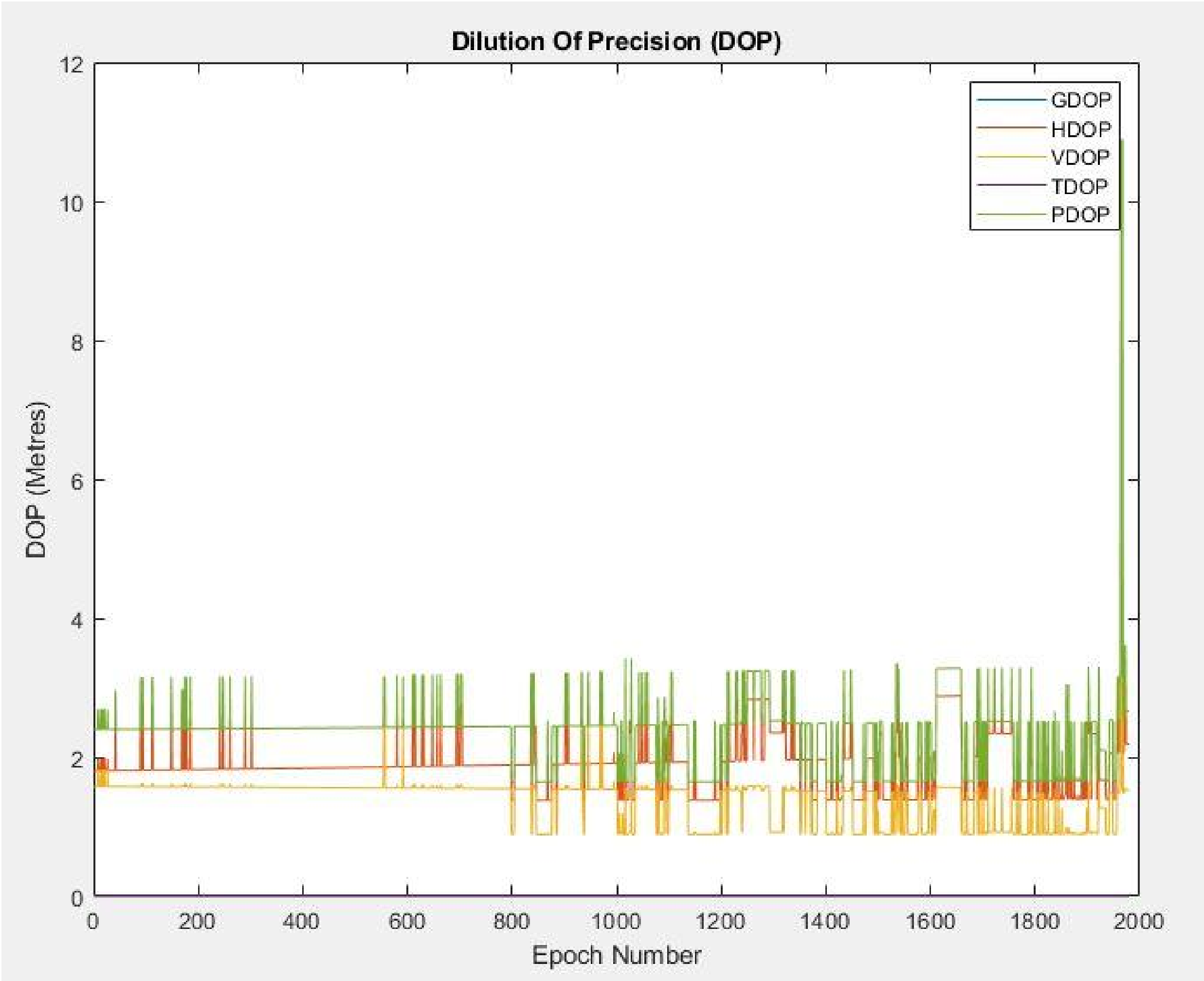

Figure 37: Dilution of Precision with Galileo

The error in the easting for Galileo measurements was similar to that for GPS. Overall, the only significant difference was a high level of error for the last few epochs (Figure 38). Such a result was understandable due to the lack of satellites at that time. Consequently, the estimation system was compromised which generated poorer quality estimated positions.

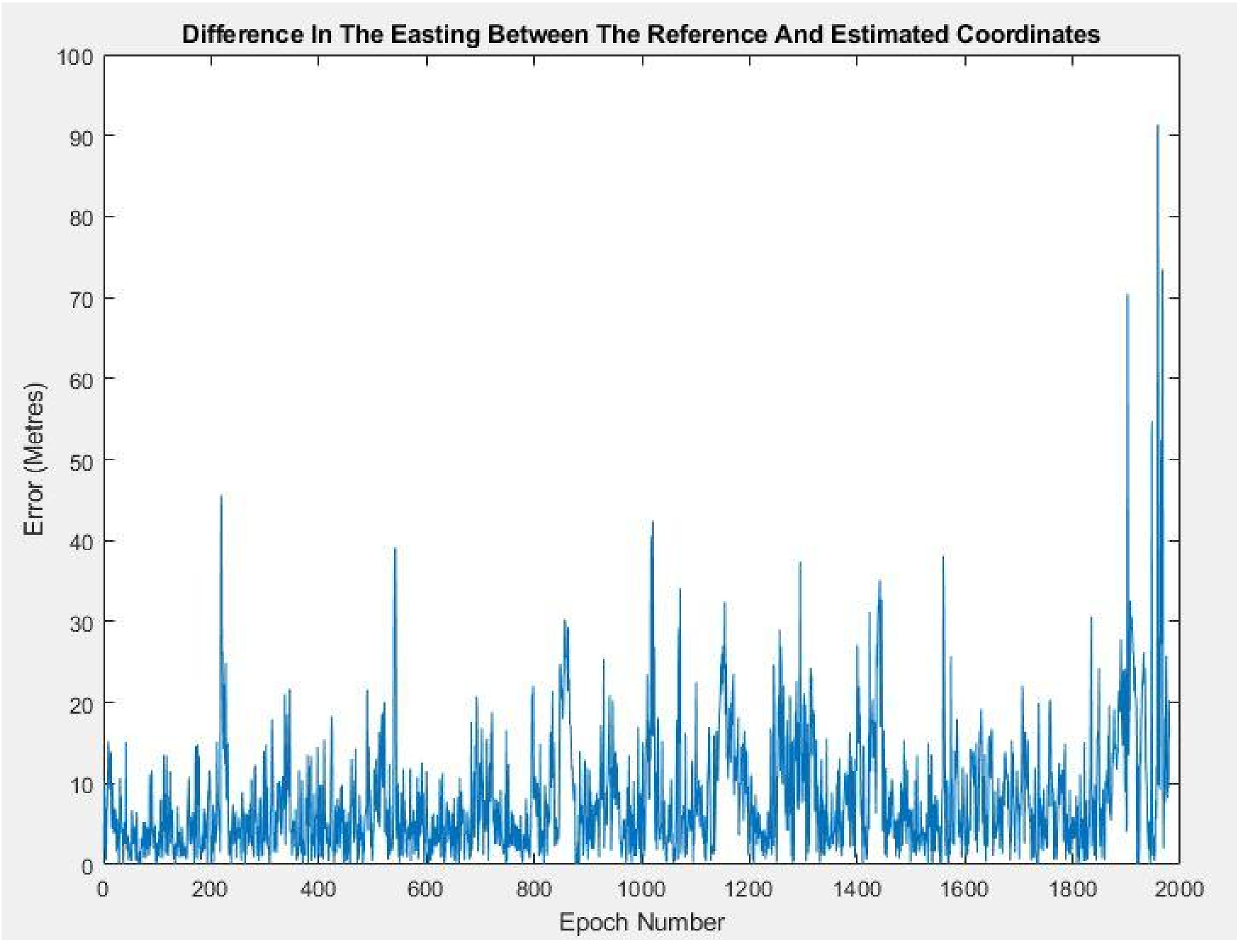

Figure 38: Easting Error with Galileo

The error in the northing between the reference and estimated coordinates from Galileo measurements was a bit better than that in the easting. However, the errors in positions were still rather noticeable (Figure 39). The mean error appeared to be between 10 m and 15 m. There was a spike in error toward the last few epochs as a result of there being fewer satellite measurements.

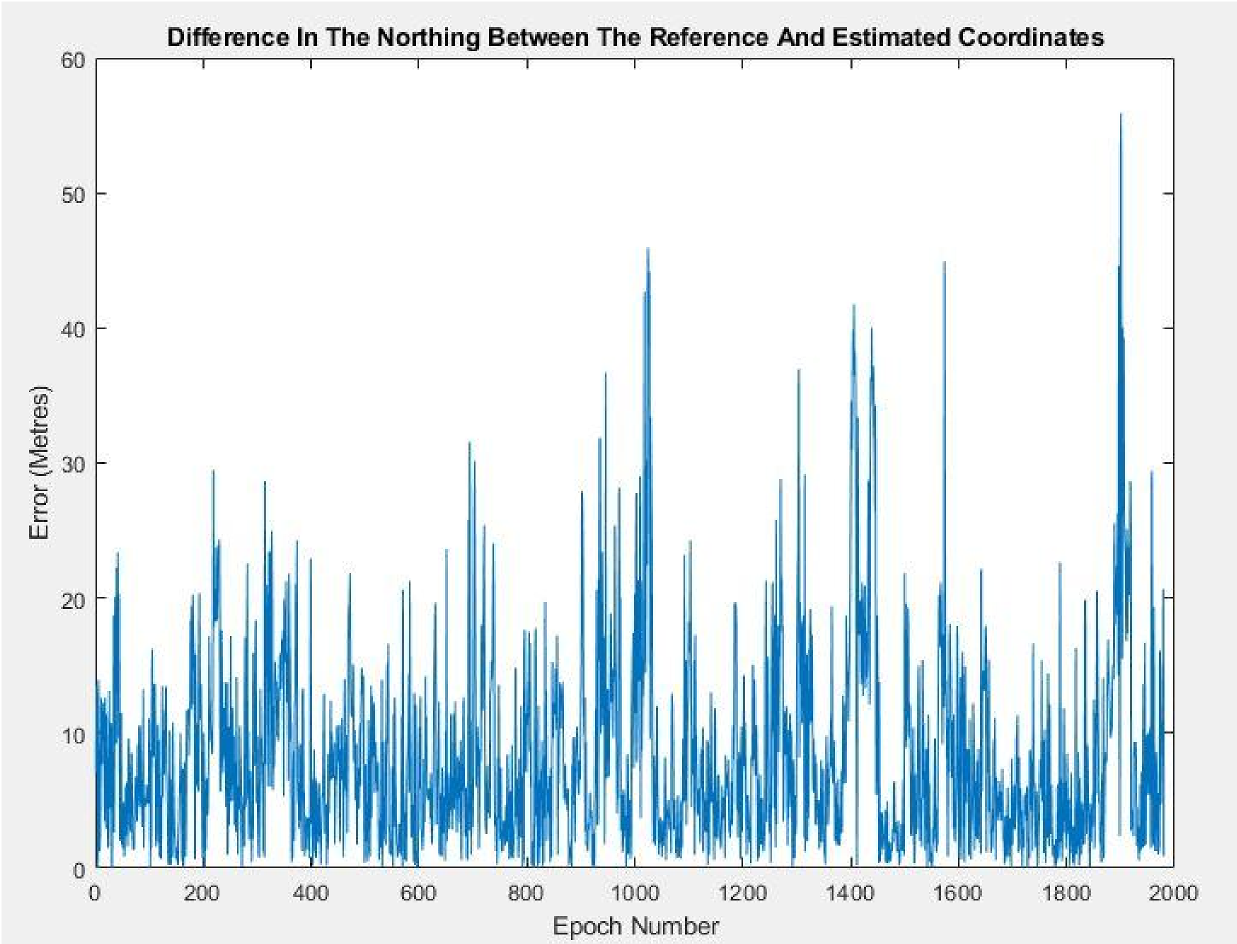

Figure 39: Northing Error with Galileo

The residual values for Galileo were fairly reasonable when compared with those from GPS. While there were some obvious outliers, the majority of the residual values were between 0 and an absolute value of 10 m (Figure 40). There were few very large outliers. The only major outliers appeared to be just after epoch 1000 and around epoch 1920. The large residuals for the last few epochs made sense due to the fewer number of satellites and poor DOP values. Nevertheless, the magnitude and distribution of the residuals appeared to be fairly understandable and expected.

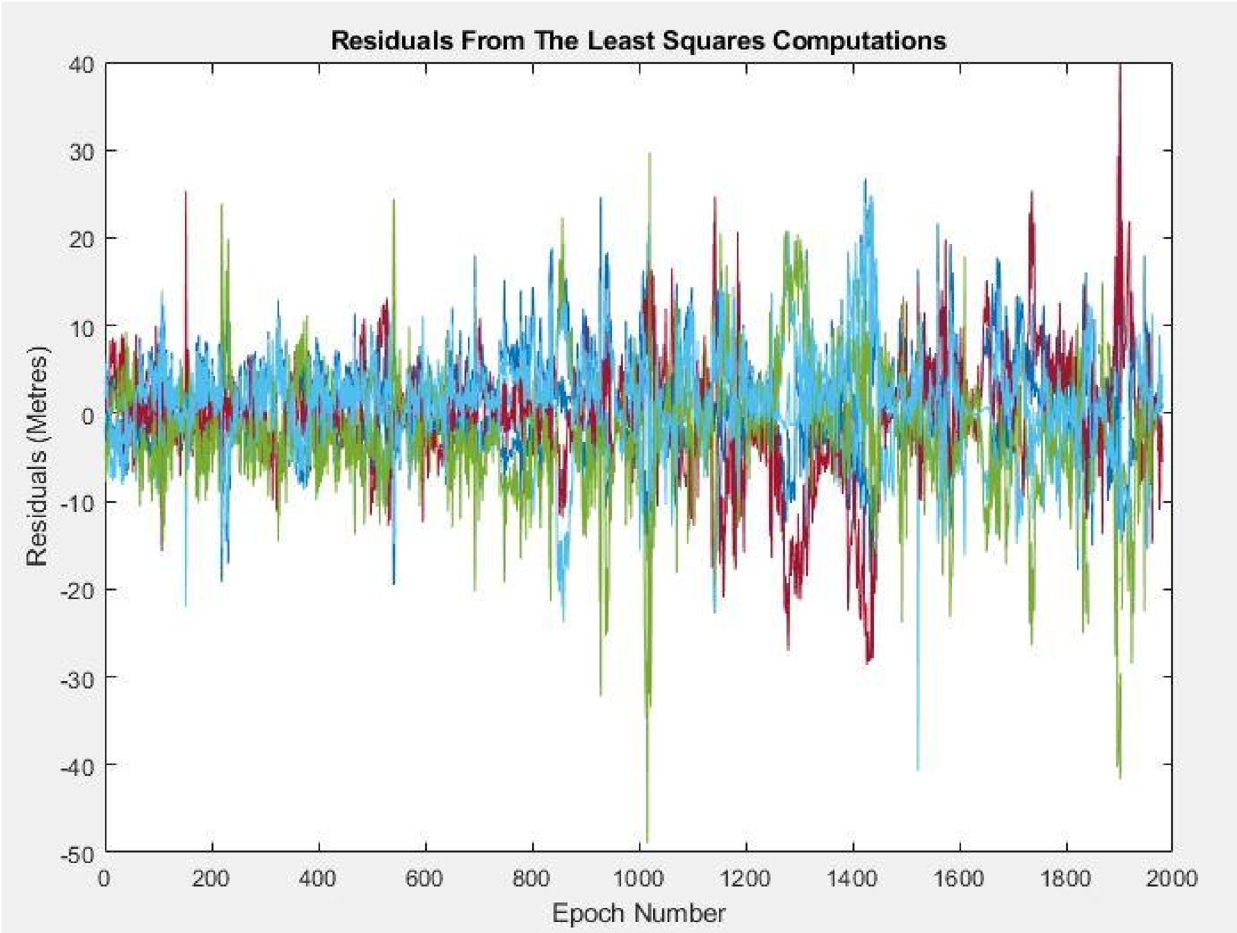

Figure 40: Residuals after Least Squares Filtering with Galileo

GPS And Galileo

The combination of both the GPS and Galileo measurements had a very positive impact on the number of satellites. Since both GPS and Galileo satellites were used in the estimation process, the number of satellites per epoch increased significantly (Figure 41). The fewest number of satellites per epoch was 10 while the most satellites per epoch was 17. As a result, there was a high level of redundancy for the estimation process.

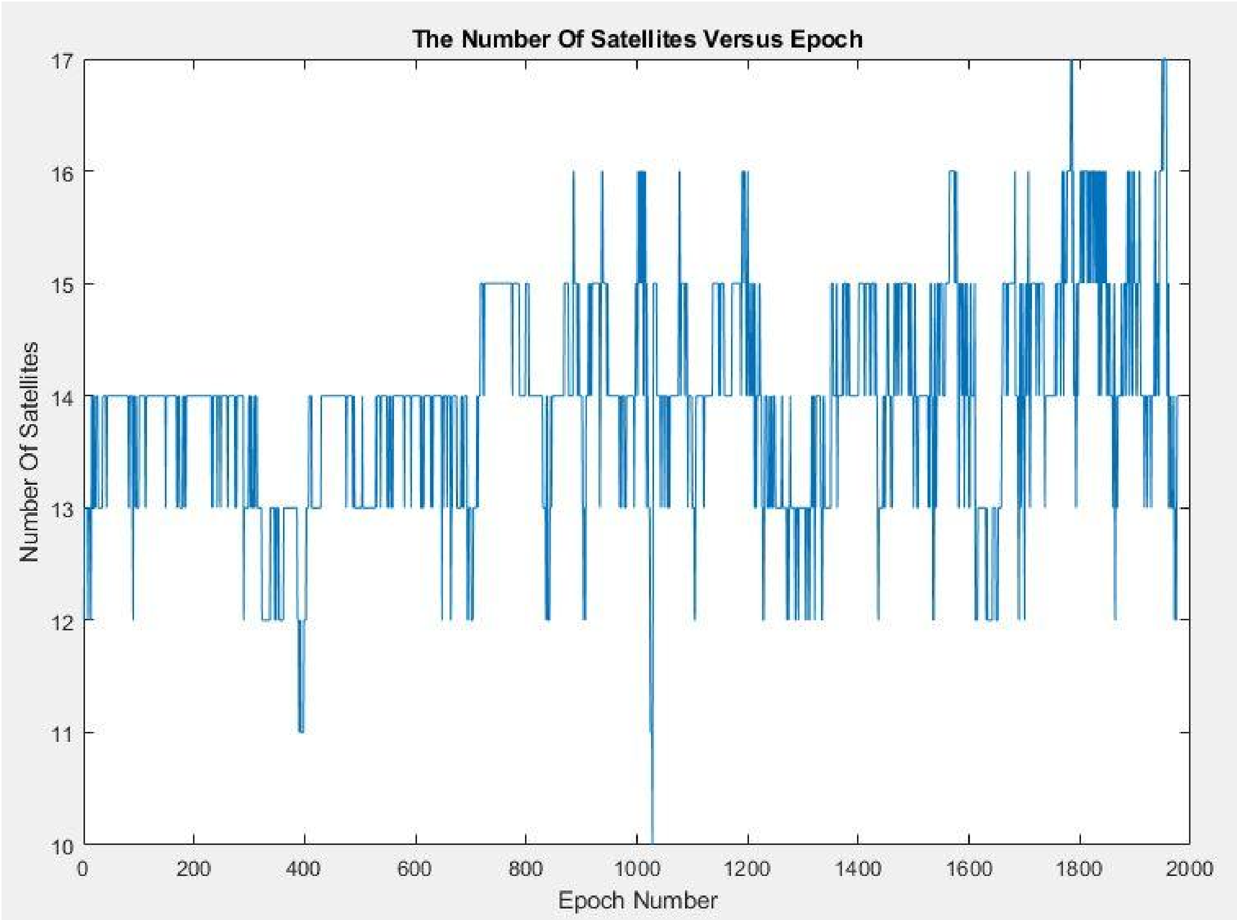

Figure 41: Available Satellites with GPS and Galileo

The DOP values significantly improved when compared to those for just the GPS results and the Galileo results. Across all epochs, the DOP values dropped. The greatest DOP value was just over 2 (Figure 42). Such a result was a noteworthy improvement from the single constellation processing for GPS where the maximum DOP value was just over 4. Furthermore, the DOP plot contained fewer spikes in DOP values. The consistent pattern appeared to represent precision stability across all epochs. Therefore, adding more measurements from different global GNSS constellations to the GPS solution greatly improved the DOP values.

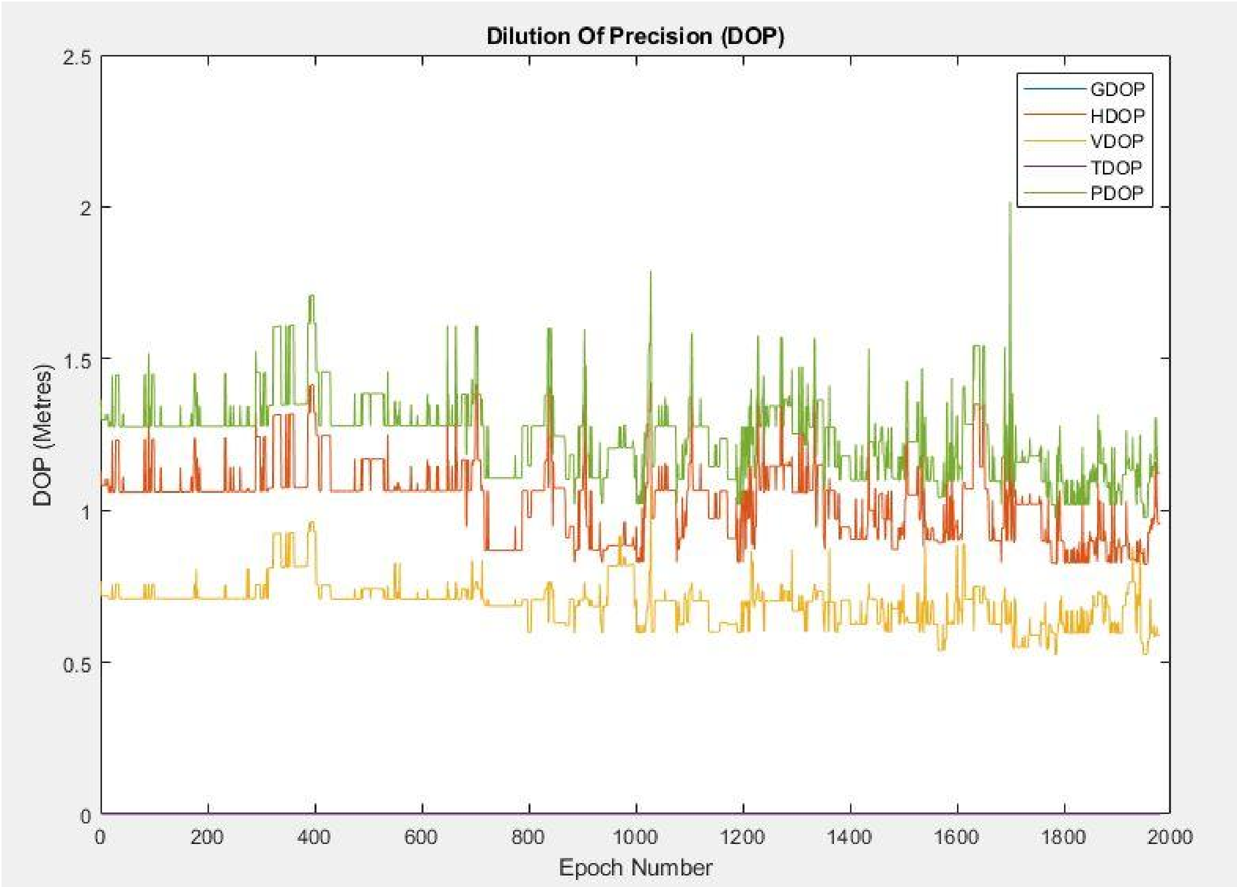

Figure 42: Dilution of Precision with GPS and Galileo

When comparing the error in the easting between the combined constellation result and the GPS result, the combined error was noticeably lower. The maximum error in the easting position for the GPS and Galileo processing was just over 45 m (Figure 43). Meanwhile, the maximum error was over 75 m for the GPS processing. Furthermore, the spikes in easting error for the combined result were significantly reduced. Based on the positional error calculations, adding more constellation measurements had a very good impact on improving the positional accuracy of estimated positions. Introducing more measurements increased the level of redundancy in the least squares estimation process. Consequently, outliers were able to be better isolated from the system and the resulting estimated positions were better able to fit the measurement equation model.

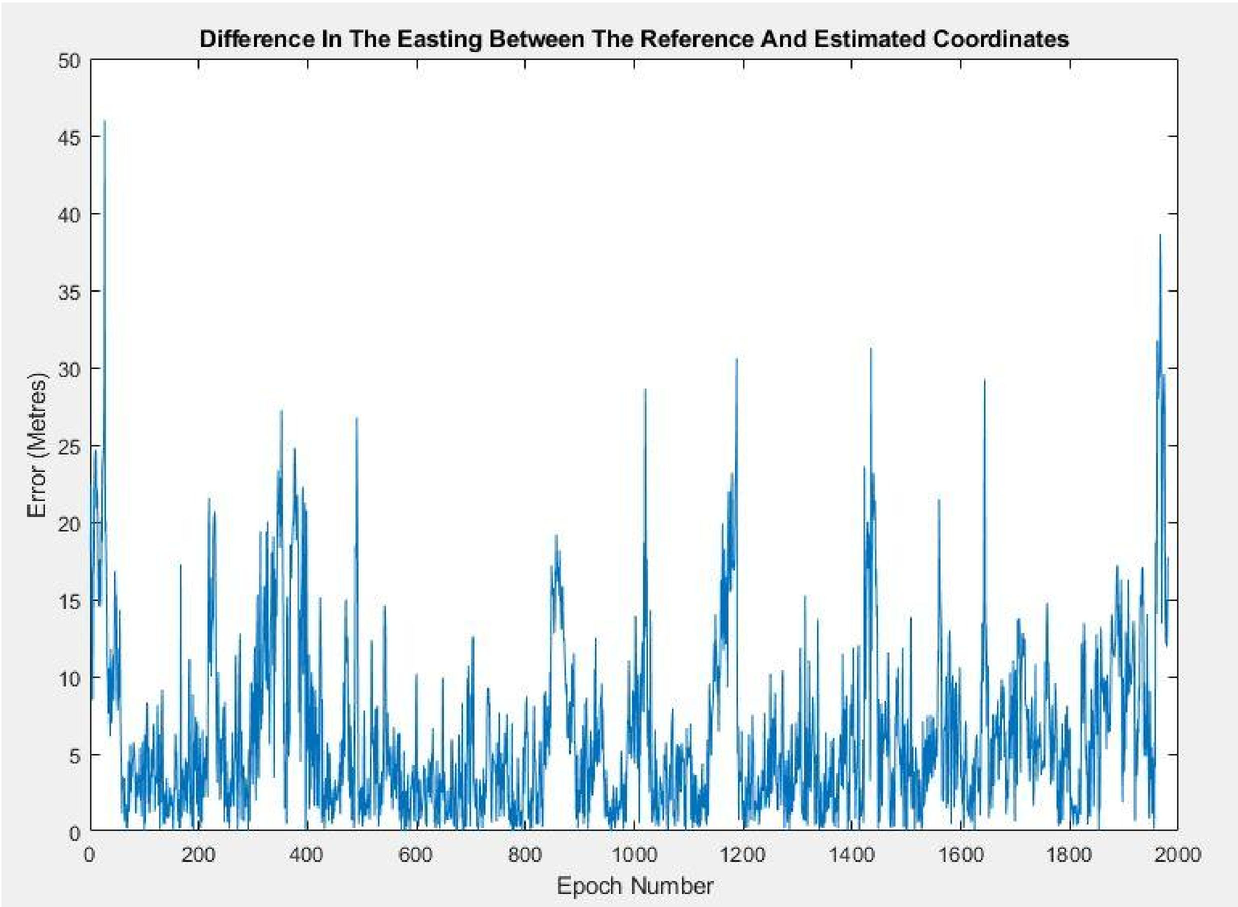

Figure 43: Easting Error over Epochs with GPS and Galileo

The positional error in the northing between the reference and estimated coordinates for the combined constellation processing significantly improved. Error spikes between epochs 300 and 500 noticeably decreased when compared to the original error levels in the exclusive GPS program (Figure 44). Furthermore, the maximum error levels were significantly different between the combined and original GPS results. In the GPS estimation, the maximum error was just over 50 m. Meanwhile, the maximum error in northing for the combined GPS and Galileo results was just over 45 m. When analysing the errors in northing between the reference and estimated coordinates, including more measurements increased the positional accuracy of the results. Therefore, increasing the number of measurements had the greatest positive impact on improvising the estimates of the northing of the unknown positions.

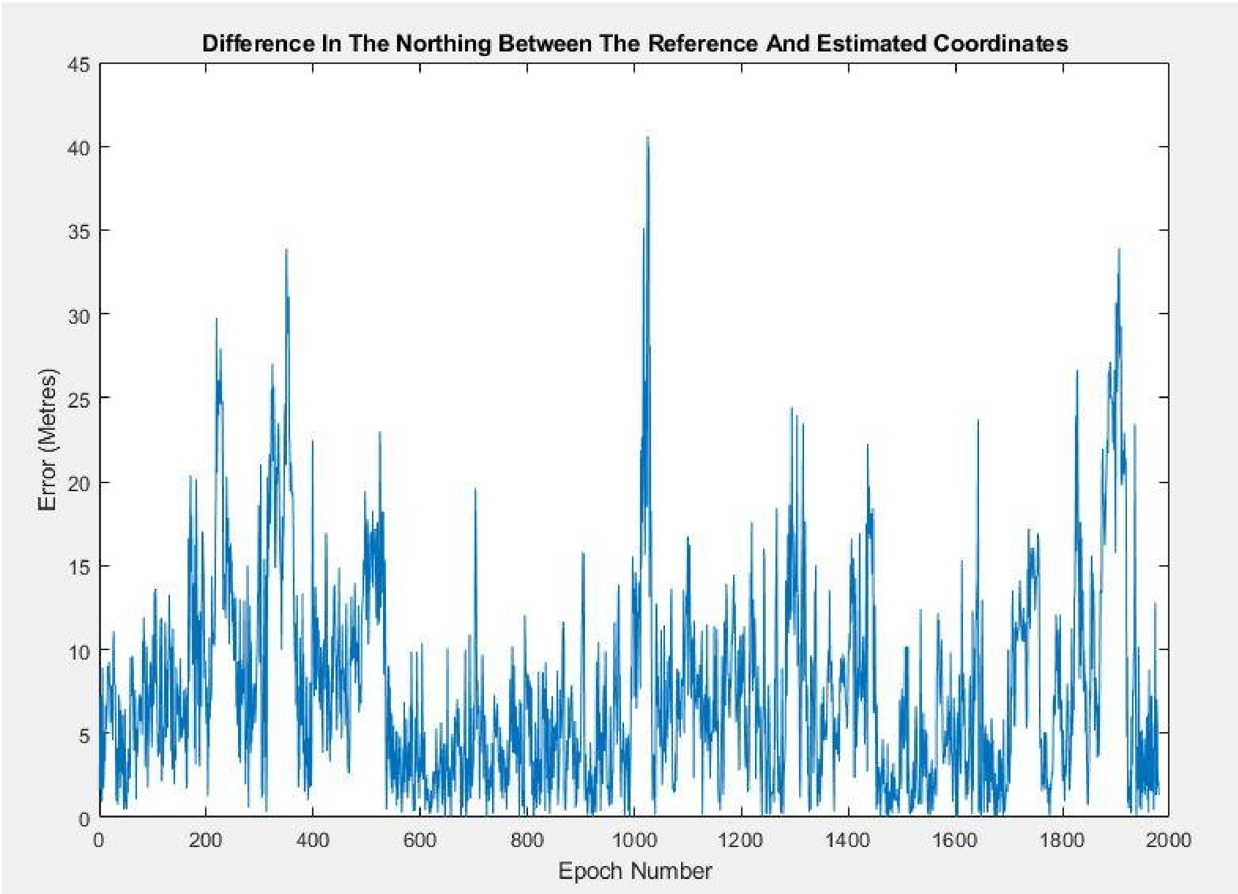

Figure 44: Northing Error over Epochs with GPS and Galileo

The residual distribution for the estimation process resulting from both the GPS and Galileo measurements was fairly reasonable. While the residual values were larger than GPS and Galileo individual results, such a situation was understandable (Figure 45). As there were more measurements, more outliers from the model measurement equation were detected. Due to this reason, the outliers could be better isolated from the estimation process and were more noticeable. However, they impacted the estimated positions less. Consequently, increasing the measurements per epoch in the system by including measurements from multiple GNSS constellations had a very positive impact on improving position estimation quality.

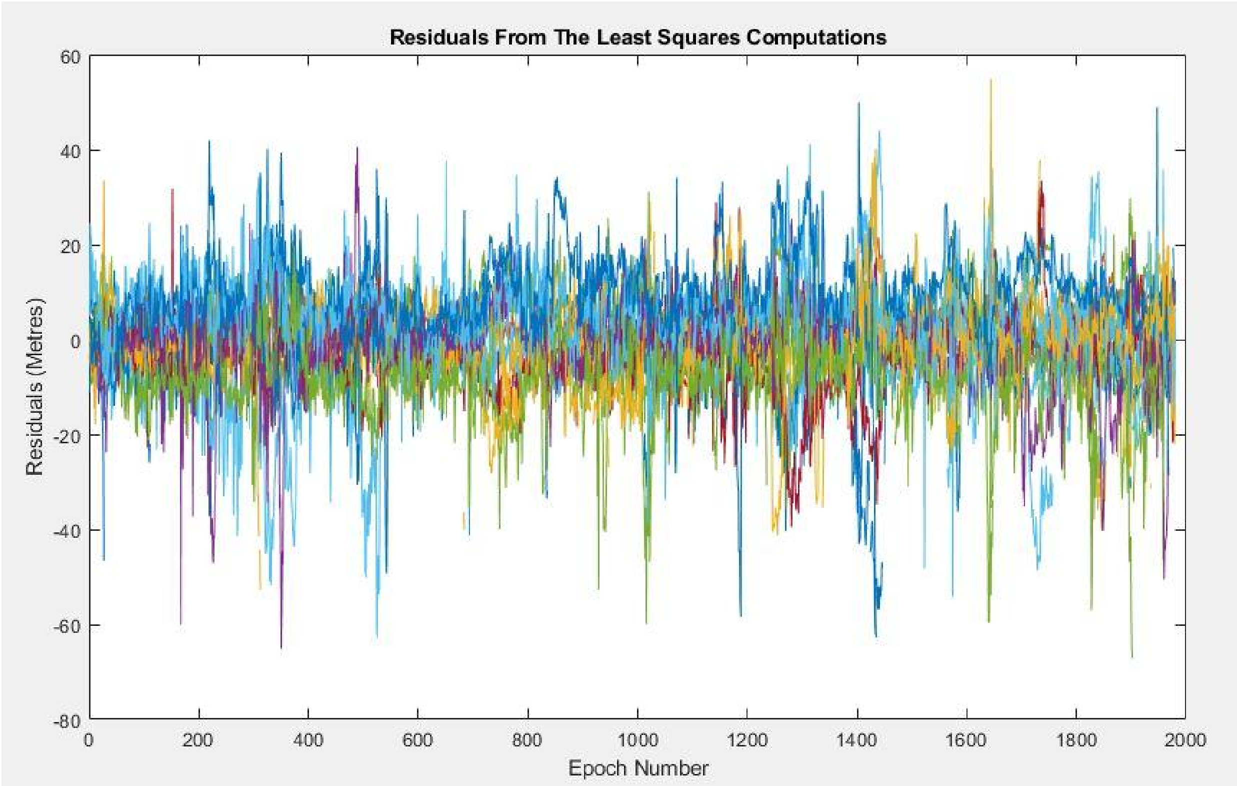

Figure 45: Residuals after Least Squares Filtering with GPS and Galileo

13.1 GPS And Galileo With Residual Outlier Removal

There were a high number of satellites per epoch in the system. By having measurements from two constellations in the least squares estimation process, the level of redundancy in the system increased dramatically. Since the lowest number of satellites at a current epoch was 10 satellites, there were many more measurements than unknowns in the system (Figure 46). This helped improve the quality of the estimation of the positions. The removal of residual outliers did not increase or decrease the number of satellites. Such a result was understandable since no additional measurements were added to the system. Therefore, including measurements from multiple constellations had a significant positive impact on the positional accuracy of the estimated locations.

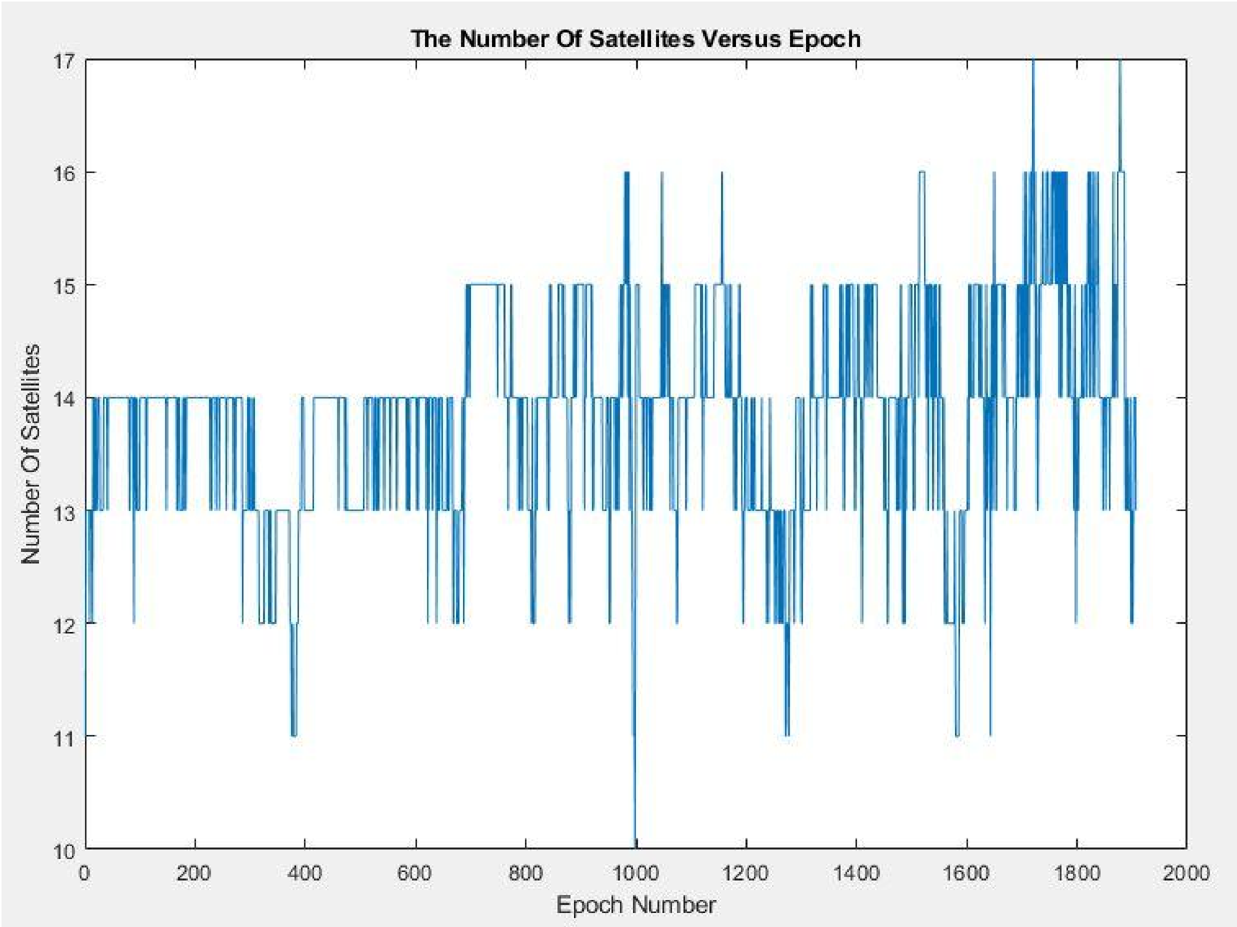

Figure 46: Available Satellites after Outlier Removal with GPS and Galileo

The DOP values were very good for the estimation process. All DOP values were well below the threshold of 7 (Figure 47). Such low DOP values meant that the precision of the system was quite good. The removal of some outliers did not have as large a positive impact on the DOP as the introduction of more constellation measurements did. Such a result was understandable as having more measurements improved the estimation process and introduced more redundancy. Meanwhile, residual outlier removal would have a better improvement on the residuals and error plots rather than the DOP.

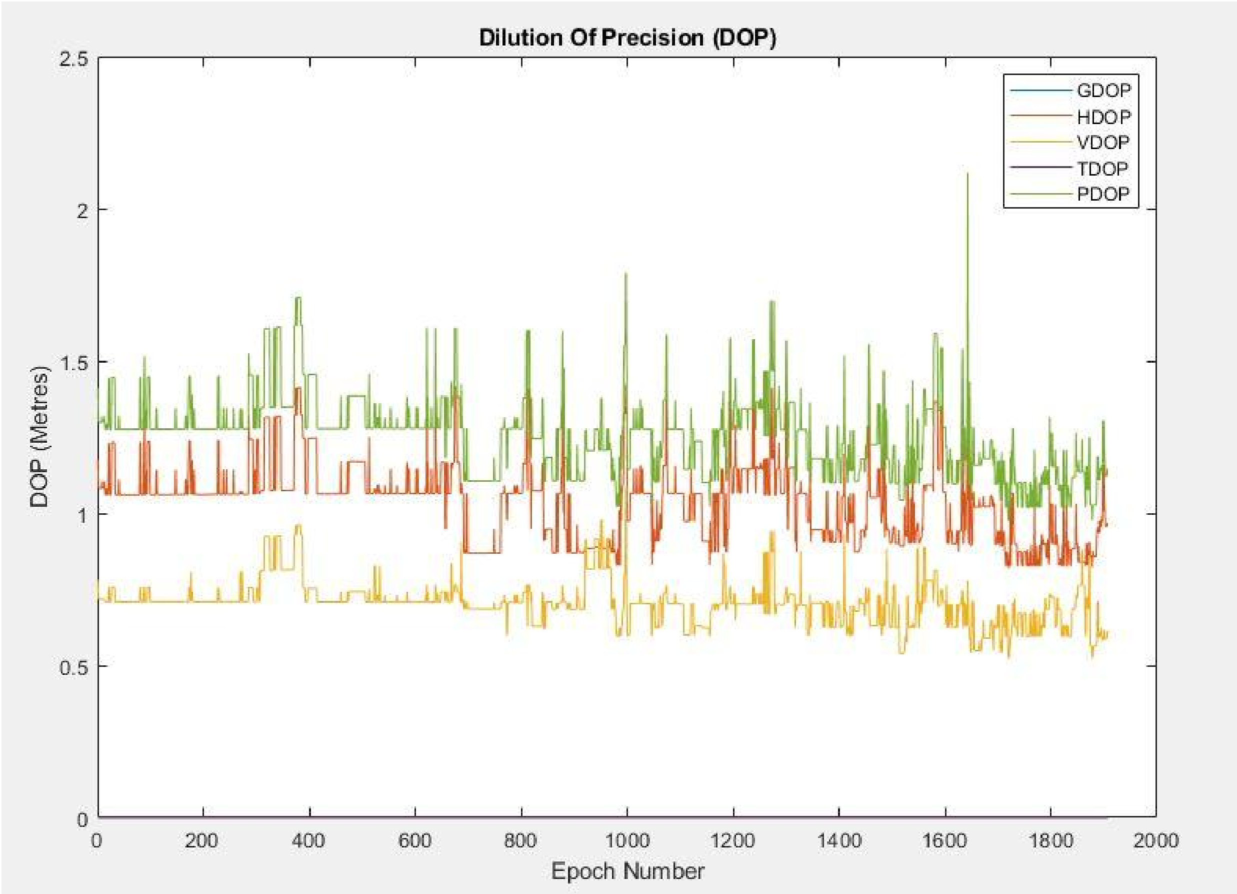

Figure 47: Dilution of Precision after Outlier Removal with GPS and Galileo

The easting positional accuracy for the unknown positions in the combined GNSS constellation program with residual removal improved by a small amount when compared to the easting errors in the combined constellation iteration. The most significant difference was the reduction of the significant 45 m error at about epoch 50 (Figure 48). Furthermore, some spikes in positional error of around 30 m around epochs 1200, 1450, and 1680 were reduced to having errors below 25 m. Overall, the removal of outlier residuals seemed to have a greater positive impact on the easting errors than the northing errors. The removal of outlier residuals did not have as great an influence on the easting errors as introducing Galileo measurements did. Nevertheless, the addition of outlier residuals still positively impacted the estimated positions.

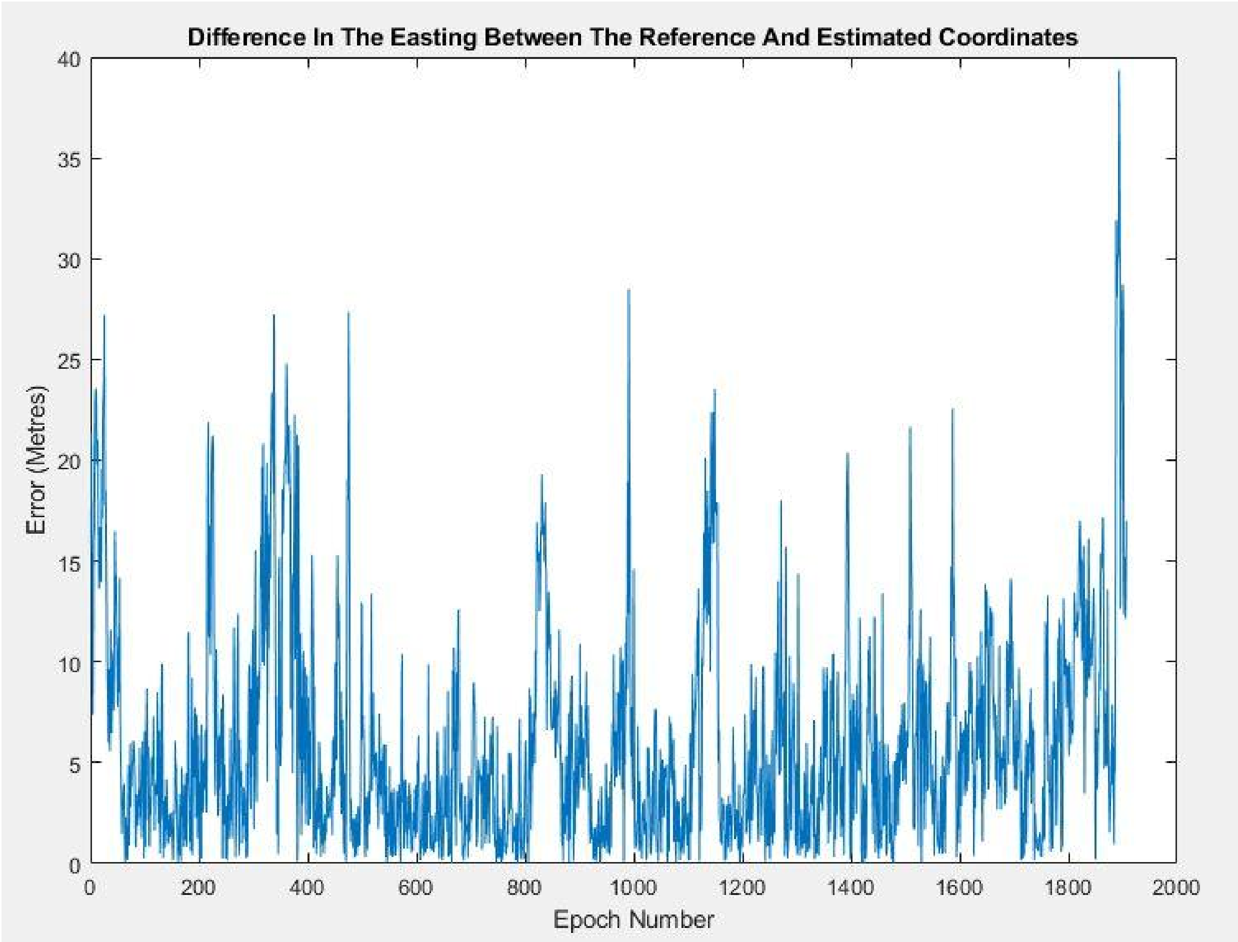

Figure 48: Easting Error after Outlier Removal with GPS and Galileo

The error in the northing between the reference and estimated coordinates for the combined constellation program with residual removal improved slightly over the iteration with just the combined constellations. When comparing the northing error results, the maximum error in the northing decreased slightly (Figure 49). The overall error distribution remained relatively the same. Nevertheless, the error values for the epochs with errors around 10 m or lower may have slightly improved. Therefore, removing residual outliers from the program with multiple GNSS constellation measurements appeared to slightly improve the positional accuracy in the northing. However, it did not improve the accuracy as decisively as when the GPS and Galileo measurements were combined. Such a situation was understandable as the residual outlier algorithm was designed to only remove significant outliers. Since there were not too many major outliers, the impact on the program appeared to be minimal but was positive.

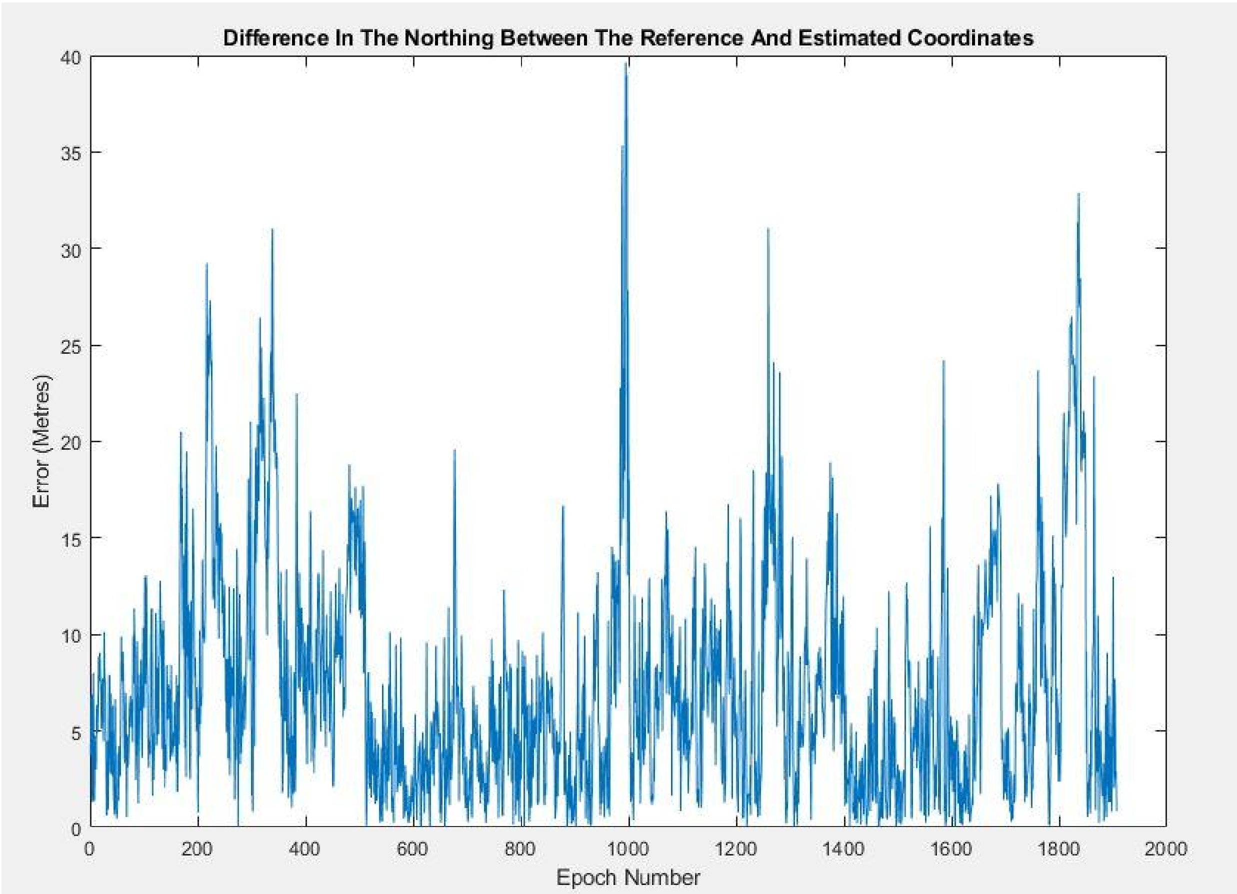

Figure 49: Northing Error after Outlier Removal with GPS and Galileo

The residual distribution from the estimation process involving GPS observations, Galileo measurements, and residual outlier removal improved over the previous iteration. Such an improvement was understandable as epochs at which the residual values were greater than an absolute value of 45 were removed. Overall, the residual distribution seemed to be centred between 0 and an absolute value of 20 (Figure 50). While there were still larger peaks of residuals, such a result was anticipated. As was discussed earlier in the report, removing too many measurements with poor residuals actually can reduce the quality of the estimated positions. Therefore, the residual plot from the combined constellation and outlier edited program was appropriate.

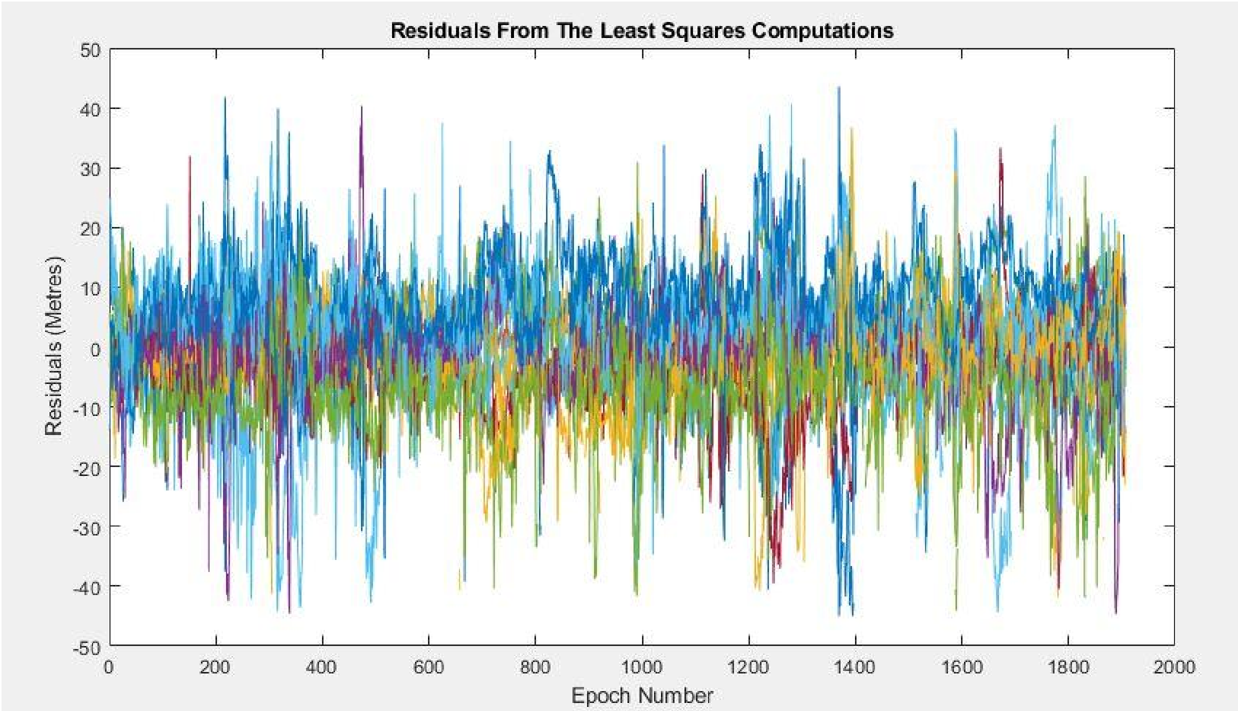

Figure 50: Residuals after Outlier Removal with GPS and Galileo

The means of the positional errors for each of the previously discussed measurements were taken for both the eastings and northings of the positions. Likewise, the mean horizontal error was determined as the square root of the sum of the squared mean error in easting and the squared mean error in northing. Furthermore, the root mean square error (RMSE) of the easting and northing positional errors from the combined constellation program with outlier residual removal was taken. The mean errors in the northing and easting were considerably smaller than the mean horizontal errors and the RMSE. Such a discrepancy was understandable as the horizontal error was a combination of the easting and northing errors. As a result, the horizontal error would always be larger. Furthermore, the RMSE values also were reasonable. Since the RMSE accounts for the residuals, using the RMSE showed that the estimated positions were more skewed to be closer to 9.0 m than 5.0 m. Consequently, the RMSE and mean error values showed different aspects of the average quality of the estimated positions.

Based on the final program combining both GPS and Galileo observations with residual outlier removal, the system reached metre level positioning. This requirement was based on the concept of the metre level being greater than 1 m and less than 10 m. The individual easting and northing errors of 6.0 m and 7.5 m respectively were well below the maximum 9.999 m threshold (Table 9). However, the combined positional error was closer to 9.999 m at 9.5 m. Such an increase in error was expected as the horizontal positional error was a combination of both the error in northing and easting. Nevertheless, the required positional accuracy at the metre level was achieved.

Table 9: Smartphone Positional Mean Error

Constellations and Modifiers	Mean Error In Easting (Metres)	Mean Error In Northing (Metres)	Mean Horizontal Error (Metres)
GPS	6.9	9.1	11.4
GPS With Residual Outliers Removed	7.1	9.0	11.5
Galileo	7.9	8.4	11.5
Galileo With Residual Outliers Removed	8.0	8.8	11.9
GPS And Galileo	6.3	7.6	9.9
GPS And Galileo With Residual Outliers Removed	6.0	7.4	9.5

Table 10: Smartphone Positional Root Mean Square Error

Constellations and Modifiers	Root Mean Square Error In Easting (Metres)	Root Mean Square Error In Northing (Metres)
GPS And Galileo With Residual Outliers Removed	8.1	9.3

As the group anticipated, the greatest improvement to the positional accuracy of the system was the introduction of more measurements from different GNSS constellations into the processing procedure. The positional error in both the northing and easting dropped by about 0.7 m in the easting and by about 1.0 m in the northing. This improvement likely resulted from the introduction of more measurements. By having more measurements in the system, outliers could be better isolated from the system.

The removal of residual outliers seemed to have an interesting impact on the system. When outliers were removed from the pure GPS processing systems, the mean positional error in easting and northing sometimes became worse. As a result, the mean horizontal positional error also became worse. However, the removal of residual outliers from the GPS and Galileo combined processing system actually had a very positive impact on reducing the error in the system. In particular, the positional error in the easting improved by 0.3 m. Similarly, the positional error in the northing improved by 0.2 m. Due to this reason, the overall positional error in the horizontal improved. After analysis, it seemed like the residual outlier removal was better suited for removing lower quality measurements when there were more observations in the system. The processing of both GPS and Galileo observations resulted in there being significantly more measurements in the least squares estimation measurement system than were necessary. As a result, the outliers in the system could be better identified and isolated. Therefore, the added code to remove the measurements with significant error was able to work more effectively in the combined system. Furthermore, such outlier measurements would have noticeably had a greater negative positional accuracy since they did not fit the model even more so with more observations. This likely meant that the residual outlier removal in the separate GPS and separate Galileo systems did not have as great an impact as there were less measurements in the system. Therefore, the outliers did not appear to be as isolated from the model as being poor measurements. This meant that when they were removed, those measurements had a greater impact on the measurement system than initially anticipated. Even though the removal of residual outliers did not seem to greatly improve positional accuracy estimation for the separate constellation processing, it proved to be of great worth for improving positional accuracy for the combined constellation processing.

In summary, implementing ionospheric and tropospheric corrections aided in reducing environmental errors impacting the GNSS measurements. Likewise, the introduction of measurements from both the GPS and Galileo constellations significantly improved the accuracy of the positioning solution. By increasing the redundancy in the system, the positions were better estimated in an overdetermined system with a high number of measurements. Furthermore, the removal of measurements with residual outliers ended up slightly improving the estimation of the unknown positions and thereby the accuracy of the positions. Therefore, the steps taken to reduce the error in the positional estimation process were successful.

14 Compliance with Professional Engineers Act

It is important that this project complies with the Professional Engineers Act because it establishes a standard of care whether it be to the public, client, respecting intellectual property, or laws and regulations. The Professional Engineers Act defines the ethics or code that an engineer must follow when designing a project which proves an engineer’s competency. The main goal is to ensure that an engineer performs adequately and his/her project deliverables meet defined requirements and specifications. These requirements and specifications should be defined by respecting laws and regulations so that the project does not negatively impact society. It is a duty for an engineer to work towards the good of the public as evidenced in following patterns such as the 17 Sustainable Development Goals. This ensures a sense of trust from all licensed engineers that they are to deliver expectations effectively. The Professional Engineers Act also proves an engineer's integrity by abiding to intellectual property laws. Such integrity ensures that an engineer's work is unique and original and does not replicate the methods and procedures of patents. This legislation prevents carelessness in an engineer's competency which is complementary to the trust between clients and the public.

Strategies for Team Performance

In the team's first gate review discussing problem definition and requirements review, the group stated each team member's strengths and weaknesses. Based on individual qualities, the team assigned each other roles for the project that would be comfortable and effective. Nathan and Jared were to develop the different positioning systems using different constellations, Melvin was to convert the MATLAB code to KML to represent the results, and Ryan created plans and kept track of what needed to be submitted. Ryan also focused on writing reports and developing video presentations of the results of the project. The team followed their role where everyone was to update their results on the report according to the gate deliverables. Nathan and Ryan met weekly to talk about the gate deliverables, the Critical Design Review, and final report to discuss its contents. Meanwhile, Jared and Melvin continued working on the project. This was done in order to visualise how the deliverables would look and form an idea of where the content would be placed to flow well. Nathan and Ryan would assign sections of the report as team members had no preference and preferred to be straightforward with course requirements. Updating group progress through draft reports to Professor Bisnath provided good feedback on the team's performance. Professor Bisnath had extensive knowledge in GNSS and provided the group with guidance, motivation, and things to avoid. The team felt good about the performance of the developed algorithm. Furthermore, the team appreciated the feedback they received from Professor Bisnath regarding his request to use the group's trailer to explain GNSS concepts for their lab's website.

15 Lessons Learned And Problems Resolved

During the testing stage of the project, a number of problems were encountered. One of the issues was the fact that some of the observation files had considerably different pseudorange values for the L1 and L5 frequencies. While it was expected that the pseudoranges between L1 and L5 would not be the same, this difference should have been a maximum of a factor of 5. The Pixel phone observation files presented this issue. Pseudoranges on the L1 and L5 frequencies were different by a factor of over 1,000. As a result, this presented a significant problem when attempting to linearly combine the pseudoranges and frequencies of L1 and L5 to cancel the ionospheric refraction.

There was a specific reason why only Pixel observation files had significantly different pseudoranges between L1 and L5. After discussing the problem with Dr. Bisnath, the group learned that the GNSS chips on Pixel smartphones had different oscillation rates between L1 and L5. Even though both L1 and L5 frequencies could be read, they could not be properly processed because the measurements on the two frequencies were on independent oscillators. After reflection with Dr. Bisnath and other graduate students from the GNSS laboratory at York University, the group determined that the correction was too time consuming and complex to implement for the project. Therefore, Pixel files that only contained one frequency were used with a single frequency ionospheric correction based on the Broadcast (Klobuchar) Model. Multiple frequency corrections were skipped due to the incompatibility of the pseudoranges in the different frequencies. Due to this reason, Samsung observation files were primarily used in the development and testing phases of the project since all GPS measurements had compatible L1 and L5 pseudoranges.

In February, the team had issues with positional error compared to the reference smartphone positions. At that time, the mean error seemed to be centred between an absolute value of 10 m and 20 m. This was unexpected and indicated that there was likely a programmatic error. After considerable testing, the team discovered that the error was with the conversion between the Earth Centred Earth Fixed (ECEF) WGS84 ellipsoid coordinate system and the Easting, Northing, and Up (ENU) WGS84 ellipsoid coordinate system. In the current function, the reference origin being used was “0,0,0”. This seemed to throw off the conversion process and caused the shift in the mean error. In order to solve this issue, the team found a MATLAB function “deg2utm” which converted the latitude and longitude of the current position into easting and northing in the WGS84 Universal Transverse Mercator (UTM) coordinate system. When this was completed, the errors between the estimated and reference positions were centred around 0. Through this experience, the team learned about the importance of the origins of coordinate systems. By having an origin that might be slightly incorrect, the transformation of coordinates from one system to another can be significantly changed.

The team also encountered an issue with comparing the estimated and reference positions when there were residuals removed. In order to improve the accuracy of the estimated positions, some measurements and epochs were removed from the system when they significantly had a negative impact on the estimation. However, such measurements were not removed from the reference positions. Therefore, the team originally had larger errors being calculated between the estimated and reference positions. The program was blindly computing the difference between the two values based on the index in the table of measurements, not the value of the epoch. To solve this problem, the team programmed a loop to take the difference between the current epoch and all the reference epochs. Afterwards, the minimum time difference was determined and the index of that epoch within the reference table was saved. When the differences between the estimated and reference coordinates were taken, the saved index of the minimum index was used to take the difference between the estimated position and the reference position with the closest time epoch. This helped to avoid larger errors than there really were from being falsely calculated and displayed.

15.1 Nathan

In the project, I learned a considerable amount of GNSS positioning. The importance of having multiple constellations really stood out to me. Adding multiple measurements to each epoch from different GNSS constellations significantly improved the position estimation. Even though nothing was done to improve the additional GNSS measurements, just the fact that there were more observations made the final accuracies better. Furthermore, I also learned about how complicated the smartphone positioning problem actually is. Due to the poor quality of GNSS receivers on current smartphones, there is only so much that can be done to improve position estimation accuracy without changing the actual hardware and expense of the smartphone. Having a clear grasp of this concept better helped me appreciate what our group had accomplished and increased my satisfaction with the final solution developed in the project. Over the course of the semester, I also appreciated gaining and practicing project management skills. It was a good experience having such a large project over a long space of time. This helped develop perseverance and practice how to break up a task into smaller projects. Since this was the case, I appreciated having the capstone project experience and spending more time learning about GNSS positioning in relation to smartphones.

15.2 Ryan

I learned how much more complex this project is and the importance of the quality of GNSS receivers needed for a positioning solution. In previous courses, I developed a GPS positioning solution. For this project, I was to develop a GPS positioning solution, GPS positioning solution with refined measurements (without residuals), a Galileo positioning solution, Galileo without residuals, multi-constellation positioning solution and multi-constellation without residuals. Unfortunately, we did not get to BeiDou or GLONASS but they would have been more complicated as GLONASS had its own position solution formulas. The project also felt a bit futile because none of the measurements of the smartphones were of decimeter standard. Which is why we opted for submetre positioning solution because our solution was to have more weighting/meaning towards the best performing measurements (where the best were not good enough).

15.3 Jared

As the one responsible for processing GLONASS observations, I learned that in the future I should be more systematic in coding problems and this advice could be extended to problems in general. When coding GLONASS, I did not plan out the code enough and tried to figure out details as I went along leading to many unnecessary revisions to code which could have been avoided. Mainly writing the pseudocode solved this problem since many problems have to be sorted out at this stage such as the data type, data size and the logic of the algorithm. Consistent testing as one completes parts of a project is also an important lesson since failing to do so regularly can lead to confusion when parts go wrong. I had done testing irregularly, testing some parts thoroughly after they were completed and not testing some upon completion assuming they worked as expected. When testing the overall system performance, a blunder in one of the untested parts was apparent and I was confused as to which part had failed and was not found in time.

15.4 Melvin

This project allowed me to experience the complexities of documenting and gauging software requirements. My role in the project was to help produce and improve the software solution. Specifically, I was tasked with improving data processing algorithms and visualizing the results in a number of user-friendly ways. A lesson I learned in terms of planning was to make detailed documentation of software ahead of time. I tried to improve the Least Squares regression model with a more flexible machine learning model, however, I found that my results were incompatible with the current project. I wasn't able to leverage the work already produced as I was unaware of specific functionality at the time. Another lesson I learned was the importance of developing short term tasks and setting clear milestones. As I was attempting to make the project a portable executable on command terminals, I found myself getting overscheduled developing unnecessary features like: automating scripts and explanatory markdown files. If I could redo the work, I would divide a week's worth of work into at least 10 tasks to prevent sidetracking.

16 Deviations From The Original Design

Due to time constraints and skill levels of the students, some deviations were made to the original project design. One of the major modifications made was not incorporating double differencing. During the initial project planning, the team thought that the four global constellations would be coded and tested faster than they actually were. The least squares process tends to produce a better estimation of the true positions when there are more measurements in the system. Due to this reason, incorporating measurements from more global constellations can improve the least squares estimation. Furthermore, double differencing was a new concept for the team. While some team members were geomatics students, they had only heard about double differencing in their ESSE3670 course and did not have any practical experience implementing it. Such a situation could result in an error-filled double differencing program being developed that would extract more time and effort than it would contribute to a good solution. After careful consideration, the team determined that it would be more efficient for the team to focus on developing and perfecting processing algorithms for the global constellations than double differencing.

Another deviation from the original design was in regards to the correction for ionospheric refraction. In the original design consideration, it was assumed that all observations would be on both the L1 and L5 frequencies. With this assumption, the team thought that they could get rid of the ionospheric refraction through a standard linear combination. In reality, there was great variation in the phone data. For Pixel files, some only had measurements in the L1 frequency. Even with fairly comprehensive phones like Samsung, sometimes the observations were only with L1, only L5, or both L1 and L5. As a result, the team had to adjust their design to incorporate the Broadcast (Klobuchar) model. This model is not as effective for correcting for ionospheric refraction as the linear combination with two frequencies is. Nevertheless, the Broadcast (Klobuchar) model offers a better correction than just ignoring the ionospheric effect. As a result, the team adjusted the ionospheric correction design to account for the presence of one or more frequencies in the system. If there were measurements from only L1 or only L5, the Broadcast (Klobuchar) model was used to approximate the ionospheric error and remove it from the system. Otherwise, a linear combination was conducted with the pseudoranges and frequencies of both L1 and L5 to compute a new pseudorange for processing.

17 Project Failures

Due to the time available to complete the project and challenges encountered, BeiDou measurements did not end up being included in the final solution. The team made a strong effort to include BeiDou observations. The interface control document (ICD) for BeiDou was reviewed and the changes in satellite parameters were performed compared to the reference parameters for the GPS system. Since the BeiDou system is based on GPS, not too many edits were made and the team thought that the BeiDou system would be relatively straightforward to implement. Nevertheless, the team had considerable issues with adjusting the time in the system. BeiDou timing is off from the UTC system by 4 leap seconds while GPS time has 18 leap seconds. Therefore, adjustments were made to the timing by adding 14 leap seconds to the BeiDou time to put it into GPS time. Furthermore, the time per week was changed from starting on January 6, 1980 for GPS to January 1, 2006 for BeiDou in the Julian date format. However, the estimated coordinates were not in the right area and the residuals were very large. Furthermore, the residuals followed an unusual pattern in that they seemed to significantly increase in both the positive and negative as the epoch number increased. The team reached out to the York University GNSS laboratory and tried further adjustment methods to the BeiDou time to correct the error. However, the same strange residual pattern continued regardless of the adjustment of the GPST time. After working on the problem for over two weeks with no progress, the team decided to stop work on BeiDou. The deadline for the project was drawing near and the team was concerned about resolving bugs in their current software. After analysis, the team decided that the results would be better improved by focusing efforts on resolving issues with the current GPS and Galileo combination rather than continuing futile efforts on BeiDou. Furthermore, BeiDou is a relatively new constellation. As a result, there are generally not as many measurements available as a more established constellation like GPS. Therefore, excluding BeiDou software would likely prove to be less destructive than including BeiDou estimated positions with significant errors that would throw off the results.

GLONASS, the Russian GNSS constellation, was attempted to be integrated which tends to offer a greater satellite availability with its added 24 satellites in orbit. Integrating another constellation would have added more redundancy to the positioning solution and increased its performance. Although the majority of the algorithm for processing GLONASS code pseudorange observations was completed, a blunder in the code was apparent once testing of the algorithm happened and was not found in time for the final deliverables. There were some indications the algorithm had done most of the work save for the blunder. Testing the GLONASS algorithm on a high-accuracy, geodetic GNSS receiver at the Algonquin Radio Observatory where the position was well-known and considered as truth, the performance of the GLONASS algorithm was tested by using the publicly available observation files from the geodetic receiver. The residuals between the true position and the position from the GLONASS algorithm was on the order of 100km but were centred about 0. Such a result suggested a computation bias on that order was affecting the final solution but that the algorithm had still done something right since the positions were centred around the true position.

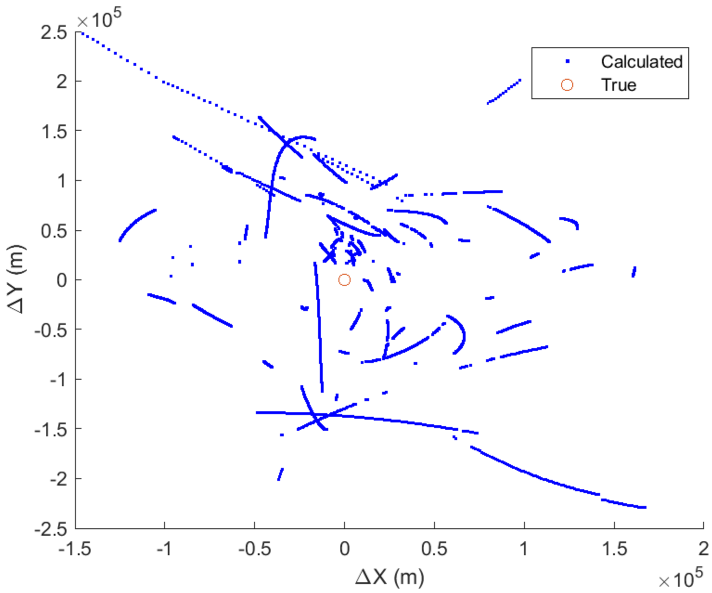

Figure 51: Calculated position of reference station vs. True position

18 Future Steps

As was stated in the Critical Design Review when the project was created, the team was focused on developing the relative positioning augmentation algorithm to estimate position. The final algorithm had to be compiled by a software engineer from MATLAB to a format that was directly transferable to smartphones. Furthermore, the current format of the algorithm was on a laptop computer with an I5 processing system with a 64 GB memory. The software developer would have to format the code in such a way that the algorithm would not overwhelm the memory on a smartphone. Consequently, a software developer would have to spend time working with the format of the algorithm before it would be ready for use on smartphones.

Unlike how the Kaggle competition organizers described the challenge, decimetre level positional accuracy cannot be achieved on smartphones at the current time. The navigation GNSS receiver chips and other hardware on smartphones are not of the proper quality to take such accuracies. As a result, the algorithm should not be blindly trusted by the user. In open areas such as highways, the tracking accuracy would be expected to be between 2 m and 5 m. It can be used as a tool to improve the user's knowledge about their positions but not as the ultimate navigator. Meanwhile, the positional accuracy in areas with tall obstructions such as the downtown area or underneath a large number of trees would be significantly worse. In such areas, the user would have to take more ownership for navigating themselves and put less weight on their device.

Further developments could be made to the project to improve the positioning accuracy. If the students had more time and expertise, the errors with GLONASS and BeiDou measurements could be fixed so that the observations could be included in the estimation process. Likewise, the team could investigate estimating carrier phase ambiguities. By modelling such ambiguities in the carrier phases, the error in the system could potentially drop. In a similar way, map matching could also be investigated and incorporated into the solution. Map matching would involve matching measured positions to a known map. Whenever a position is off of the road, the map matching software would identify that and move the estimated position to be on the road. Therefore, further efforts could be made to attempt to improve the accuracy of the positioning estimates in the system.

While the team did their best to improve the positioning accuracy, there were steps that could have been taken earlier on in the project. The team was quite successful in developing the first prototype with GPS processing. However, the group could have started earlier on coding the GLONASS processing component of the algorithm. While the team began in December, there were many unknowns with GLONASS processing. As a result, additional time would have been helpful in order to ask for further assistance from the York GNSS laboratory. Furthermore, the team could have incorporated the error evaluation between the reference and estimated coordinates earlier on in the project. At the beginning, the team primarily used Google Earth to visually verify the positions of the estimated coordinates. While the reference points were successfully read in and evaluated within ample time, reading them into the program earlier would have helped the group catch some errors in a more time efficient manner. Despite these changes that could be made in the future, the project was still successfully completed. Consequently, the team could have taken further measures to improve the quality and schedule of the project if the work was completed again.

19 Conclusion

High accuracy positioning based on GNSS measurements from smartphones has great potential for good in society. From autonomous vehicle positioning to augmented reality, there are many practices and tasks that can benefit from localized high accuracy positioning. Nevertheless, the current GNSS measurements from smartphones are quite noisy due to the environment in which they were observed, poor hardware, multipath, and movement during the recording of the observation. Despite such challenges, the poor data can be processed using relative positioning augmentation to estimate positions at the metre level of accuracy. By increasing the redundancy of the measurement system with observations from different GNSS constellations, correcting for ionospheric refraction, and removing noteable residuals, the estimation of the position can be improved. While there is still progress to be made in improving smartphone GNSS measurements, the methods developed in the project offer a means to achieve metre level positioning working with the current system available.

20 Appendix

20.1 References

Bisnath, S., et. al. (2021). “Changing Lanes With Smartphones Technology.” Toronto, Ontario, Canada: York University

Bisnath, S., et. al. (2021). “Resilient Smartphone Positioning Using Native Sensors And PPP Augmentation.” Toronto, Ontario, Canada: York University

Bisnath, S. (2021). “Topic 6 - Introduction to GNSS hardware. Global Navigation Satellite Systems.” Toronto, Ontario, Canada: York University

Bisnath, S. (2021). “Topic 7 - Introduction to GNSS observables. Global Navigation Satellite Systems.” Toronto, Ontario, Canada: York University

Bisnath, S. (2021). “Topic 8 - GNSS point positioning. Global Navigation Satellite Systems.” Toronto, Ontario, Canada: York University

Bisnath, S. (2021). “Topic 9 - GNSS error sources. Global Navigation Satellite Systems.” Toronto, Ontario, Canada: York University

Bisnath, S. (2021). “Topic 10 - Relative positioning. Global Navigation Satellite Systems.” Toronto, Ontario, Canada: York University

China Satellite Navigation Office. (2020). “BeiDou Navigation Satellite System Signal In Space Interface Control Document.” China: China Satellite Navigation Office: http://en.beidou.gov.cn/SYSTEMS/ICD/202008/P020200803539206360377.pdf

Coordination Scientific Information Center (1998). “GLONASS Interface Control Document.” Moscow, Russia: Coordination Scientific Information Center: https://www.unavco.org/help/glossary/docs/ICD_GLONASS_4.0_(1998)_en.pdf

Data Design Group, Inc. (2022). “Convert CSV to KML.” : Data Design Group, Inc: https://www.convertcsv.com/csv-to-kml.htm

European GNSS (Galileo) Open Service. (2021). “Signal-In-Space” Interface Control Document.” European Union, Europe: European Union: https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf

Flores, A. (2021). “NAVSTAR GPS Space Segment/Navigation User Interfaces.” El Segundo, California, United States of America: Interface Control Contractor: https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf

Government of Canada (2016) “A Guide To Canada’s Export Control List.” Ottawa, Ontario, Canada: Government of Canada: https://www.international.gc.ca/controls-controles/assets/pdfs/documents/guide-2016-eng.pdf

Industrial Pixel VFX Staff (2022). “Industrial Pixel.” Vancouver, British Columbia, Canada: Industrial Pixel: https://industrialpixelvfx.com/portfolio/

International GNSS Service. (2018). “RINEX: The Receiver Independent Exchange Format: Version 3.04.” Boulder, Colorado, United States of America: International GNSS Service (IGS): http://acc.igs.org/misc/rinex304.pdf

Palacios, R. (2006). “Deg2utm.” MATLAB, Mathworks: The Mathworks, Inc: https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/44242/versions/1/previews/deg2utm.m/index.html

TechTarget Custom Media (2012). “Reveal More Value In Your Data With Location Analytics.” Redlands, California: ESRI: https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/reveal-more-value.pdf

20.2 Code Download

The algorithm and the associated functions needed to run the program can be accessed by the link shared below. The main program is named “relativepositioningaugmentationprogram.m”. https://drive.google.com/drive/folders/1slyghY42k30A-5Wsbz223DjX_F49npff?usp=sharing

The observation file, navigation file, and the reference coordinate file can be downloaded from the Kaggle Competition website provided by the link below. The observation file has a file format ending with “.o”, the navigation’s file format ends with “.p”, and the reference coordinate file is in the “.csv” format. Sample observation, navigation, and reference coordinate files were provided in the Google link drive above for April 28, 2021 as a reference.
https://www.kaggle.com/c/google-smartphone-decimeter-challenge

The “.o” observation file has to be included in the “gpsgalobsreader.m” function while the “.p” navigation file has to be present in the “gpsgalnavreader.m” function. Meanwhile, the “.o” observation file, “.p” navigation file, and the “.csv” reference coordinate file have to be included in the “relativepositioningaugmentationprogram.m” program.

20.3 Work Packages

WP ID: 1	WP Name: Code Navigation Reader
Expected effort in hours: 55 hours Expected effort in days: 11 days	WP manager: Nathan Stachow, 45 hours Support: Melvin Gagarao, 3 hours; Ryan Nguyen, 1 hour; Jared Yen, 6 hours
Expected start date: September 25, 2021	Expected end date: October 6, 2021
Inputs needed: Navigation file from the CDDIS website, MATLAB software, ESSE3670 GNSS course slides, ESSE3670 GNSS pseudocode.	
Tasks to be performed: Read in the navigation file, convert time to GPS time, sort the parameters into columns corresponding to a specific satellite number and epoch.	
Output generated: Struct containing the navigation satellite parameters in the GPS GNSS system.	

WP ID: 2	WP Name: Code Observation Reader
Expected effort in hours: 63 hours Expected effort in days: 12 days	WP manager: Nathan Stachow, 49 hours Support: Melvin Gagarao, 3 hours; Ryan Nguyen, 1 hour; Jared Yen, 10 hours
Expected start date: October 6, 2021	Expected end date: October 18, 2021
Inputs needed: Observation GPS file from Kaggle, MATLAB software, ESSE3670 GNSS course slides, ESSE3670 GNSS pseudocode.	
Tasks to be performed: Read in the observation file, convert time to GPS time, sort the satellite observations to correspond to each epoch, store observations in columns relating to the current satellite at the current epoch.	

Output generated: Structure containing the observation satellite parameters in the GPS GNSS system.

WP ID: 3	WP Name: Code Time Matching
Expected effort in hours: 40 hours Expected effort in days: 7 days	WP manager: Nathan Stachow, 38 hours Support: Melvin Gagarao, 1 hour; Jared Yen, 1 hour
Expected start date: October 18, 2021	Expected end date: October 25, 2021
Inputs needed: Observation reader, navigation reader, observation GPS file from Kaggle, navigation file from the CDDIS website, MATLAB software, ESSE3670 GNSS course slides, ESSE3670 GNSS pseudocode.	
Tasks to be performed: Loop through the observation files and navigation files. Take the differences between the current observation time and the navigation time and find the minimum time difference. Associate the navigation parameters with the navigation time with the minimum difference to the observation time and measurements.	
Output generated: Structure with navigation parameters attached to each observation epoch.	

WP ID: 4	WP Name: Code Corrections To Measurements
Expected effort in hours: 40 hours Expected effort in days: 7 days	WP manager: Nathan Stachow, 38 hours Support: Melvin Gagarao, 1 hour; Jared Yen, 1 hour
Expected start date: October 26, 2021	Expected end date: November 3, 2021
Inputs needed: Time matching code, observation reader, navigation reader, observation GPS file from Kaggle, navigation file from the CDDIS website, MATLAB software, ESSE3670 GNSS course slides, ESSE3670 GNSS pseudocode.	

Tasks to be performed: Apply corrections to observation parameters, apply tropospheric and ionospheric corrections to pseudorange, correct for the Earth’s rotation, calculate the northing, easting, and up position, calculate the geographic position, determine the zenith and azimuth angles, estimate satellite clock offset.
Output generated: Corrected pseudorange, geographic coordinates of satellite, northing, easting, and up of the satellite, corrected satellite clock offset.

WP ID: 5	WP Name: Code Least Squares Estimation
Expected effort in hours: 40 hours Expected effort in days: 7 days	WP manager: Nathan Stachow, 38 hours Support: Melvin Gagarao, 1 hour; Jared Yen, 1 hour
Expected start date: November 3, 2021	Expected end date: November 10, 2021
Inputs needed: Time matching code, corrections to measurements program, observation reader, navigation reader, observation GPS file from Kaggle, navigation file from the CDDIS website, MATLAB software, ESSE3670 GNSS course slides, ESSE3670 GNSS pseudocode.	
Tasks to be performed: Estimate the positions of the receivers and receiver clock offset based on least squares principles, continue with the least squares estimates until an appropriate threshold is met, determine the residuals and a-posteriori variance factor.	
Output generated: Estimates of the receiver X, Y, and Z position components, the receiver clock offset, and the residuals produced from the adjustment.	

WP ID: 6	WP Name: Debug Program
Expected effort in hours: 55 hours Expected effort in days: 9 days	WP manager: Nathan Stachow, 45 hours Support: Melvin Gagarao, 5 hours; Jared Yen, 5 hours
Expected start date: November 11, 2021	Expected end date: November 20, 2021

Inputs needed: Time matching code, corrections to measurements program, least squares estimation code, observation reader, navigation reader, observation GPS file from Kaggle, navigation file from the CDDIS website, MATLAB software.
Tasks to be performed: Debug program for any issues, analyse residuals.
Output generated: GNSS positioning program without as many programmatic issues.

WP ID: 7	WP Name: Relate Calculations To Reference Positions
Expected effort in hours: 40 hours Expected effort in days: 7 days	WP manager: Jared Yen, 38 hours Support: Nathan Stachow, 1 hour; Melvin Gagarao, 1 hour
Expected start date: November 13, 2021	Expected end date: November 20, 2021
Inputs needed: Estimated receiver positions in the geographic coordinate system, GPS positioning program, MATLAB software, reference receiver positions from Kaggle, Excel software.	
Tasks to be performed: Export receiver positions to Excel, read in reference receiver positions, calculate the differences between the positions, find the root mean square error (RMSE)	
Output generated: Calculated differences between the calculated and reference receiver smartphone positions.	

WP ID: 8	WP Name: Plot Residuals
Expected effort in hours: 30 hours Expected effort in days: 4 days	WP manager: Nathan Stachow, 29 hours Support: Jared Yen, 1 hour
Expected start date: November 15, 2021	Expected end date: November 19, 2021

Inputs needed: Residuals from the least squares adjustment, GPS positioning program, MATLAB software, ESSE3670 GNSS course slides.
Tasks to be performed: Plot the residuals in MATLAB software.
Output generated: Plot of the residuals from the least squares adjustment.

WP ID: 9	WP Name: Plot DOP
Expected effort in hours: 15 hours Expected effort in days: 3 days	WP manager: Nathan Stachow, 14 hours Support: Melvin Gagarao, 1 hour
Expected start date: November 16, 2021	Expected end date: November 19, 2021
Inputs needed: GPS positioning program, first design matrix, MATLAB software, ESSE3670 GNSS course slides.	
Tasks to be performed: Calculate the dilution of precision (DOP) values, calculate the horizontal DOP (HDOP), compute the vertical DOP (VDOP), time dilution of precision (TDOP), calculate the position dilution of precision (PDOP), plot all DOPs in MATLAB.	
Output generated: Plot of all DOP values from the positioning software.	

WP ID: 10	WP Name: Convert To KML
Expected effort in hours: 10 hours Expected effort in days: 4 days	WP manager: Jared Yen, 9 hours Support: Melvin Gagarao, 1 hour
Expected start date: November 16, 2021	Expected end date: November 20, 2021
Inputs needed: GPS positioning program, first design matrix, MATLAB software, online geographic coordinate to KML convertor.	

Tasks to be performed: Export the estimated receiver positions in the geographic coordinate system, upload the exported coordinates to a KML online convertor.
Output generated: KML file of the estimated receiver positions.

WP ID: 11 WP Name: Plot In Google Earth
Expected effort in hours: 8 hours WP manager: Melvin Gagarao, 7 hours Support: Ryan Nguyen, 1 hour Expected effort in days: 2 days
Expected start date: November 21, 2021 Expected end date: November 23, 2021
Inputs needed: KML file of the estimated receiver positions, Google Earth software.
Tasks to be performed: Import the estimated receiver positions into Google Earth.
Output generated: Plot of the estimated receiver positions on Google Earth.

WP ID: 12 WP Name: Gather Constellation Parameters
Expected effort in hours: 100 hours WP manager: Ryan Nguyen, 50 hours Support: Jared Yen, 25 hours; Expected effort in days: 61 days Melvin Gagarao, 15 hours Nathan Stachow, 10 hours
Expected start date: November 10, 2021 Expected end date: December 20, 2021
Inputs needed: ESSE3670 GNSS course slides, GLONASS website, Galileo website, BeiDou website.
Tasks to be performed: Locate interface control documents (ICDs) for global GNSS systems, find the parameters for the GLONASS, Galileo, and BeiDou satellite systems to correct the observation measurements.

Output generated: Parameters from global GNSS satellite systems to correct observations measurements.

WP ID: 13	WP Name: Adjust Readers
Expected effort in hours: 60 hours Expected effort in days: 28 days	WP manager: Nathan Stachow, 40 hours Support: Jared Yen, 10 hours; Melvin Gagarao, 10 hours
Expected start date: December 17, 2021	Expected end date: January 14, 2022
Inputs needed: Observation reader, navigation reader, observation file from Kaggle, navigation file from the CDDIS website, MATLAB software.	
Tasks to be performed: Adjust observation code to read in other GNSS observations besides the GPS results, adjust navigation code to read in other GNSS parameters beside the GPS results.	
Output generated: Navigation and observation readers that bring in observation measurements and navigation parameters for GPS, GLONASS, Galileo, and BeiDou systems.	

WP ID: 14	WP Name: Adjust For Multiple Frequencies
Expected effort in hours: 65 hours Expected effort in days: 29 days	WP manager: Nathan Stachow, 45 hours Support: Jared Yen, 10 hours; Melvin Gagarao, 10 hours
Expected start date: December 23, 2021	Expected end date: January 21, 2022
Inputs needed: Observation reader, navigation reader, observation file from Kaggle, navigation GPS file from the York University GNSS laboratory, GPS positioning program, MATLAB software.	

Tasks to be performed: Adjust observation reader code to read in L5 and L1 frequencies, adjust time matching code to use the Broadcast (Klobuchar) model to approximate ionospheric correction when there is only one frequency, adjust corrections to satellite parameters code to have a combination of L5 and L1 frequencies.
Output generated: Corrected pseudorange measurements that result from a linear combination of L1 and L5 frequencies when possible. In cases where there is only one frequency, a correction for ionospheric delay will be generated based on the Broadcast (Klobuchar) model.

WP ID: 15	WP Name: Adjust Time Matching
Expected effort in hours: 110 hours Expected effort in days: 35 days	WP manager: Nathan Stachow, 65 hours Support: Jared Yen, 40 hours; Ryan Nguyen, 5 hours
Expected start date: December 20, 2021	Expected end date: January 24, 2022
Inputs needed: Adjusted observation reader for global GNSS, adjusted navigation reader, observation file from Kaggle, navigation GPS file from the York University GNSS laboratory, GPS positioning program, MATLAB software.	
Tasks to be performed: Adjust program to be able to match the satellite parameters from the Galileo satellite with the smallest time difference to the current observation epoch, adjust program to be able to match the satellite parameters from the Galileo satellite with the smallest time difference to the current observation epoch and the same process for GPS matching at the same time.	
Output generated: Every satellite in the Galileo GNSS system at each observation epoch has navigation satellite correction parameters associated with it. Likewise, another script will be generated where each epoch has both GPS and Galileo satellites and navigation satellite correction parameters associated with it.	

WP ID: 16 WP Name: Adjust Corrections To Measurements
Expected effort in hours: 120 hours WP manager: Jared Yen, 70 hours Support: Melvin Gagarao, 40 hours; Expected effort in days: 39 days Nathan Stachow, 10 hours
Expected start date: December 23, 2021 Expected end date: January 31, 2022
Inputs needed: Adjusted observation reader, adjusted navigation reader, observation file from Kaggle, navigation GPS file from the York University GNSS laboratory, adjusted time matching code, GPS positioning program, MATLAB software.
Tasks to be performed: Adjust corrections to measurements from Galileo satellites based on the Galileo ICD, adjust corrections to measurements based on the Galileo and GPS satellites using the parameters provided in both the Galileo and GPS Interface Control Documents (ICDs).
Output generated: Corrected pseudorange, geographic coordinates of satellite, northing, easting, and up of the satellite, corrected satellite clock offset for Galileo GNSS measurements and similarly for both Galileo and GPS measurements simultaneously.

WP ID: 17 WP Name: Identify And Remove Outliers
Expected effort in hours: 60 hours WP manager: Melvin Gagarao, 30 hours Support: Jared Yen, 10 hours; Expected effort in days: 28 days Nathan Stachow, 10 hours; Melvin Gagarao, 10 hours;
Expected start date: January 14, 2022 Expected end date: February 11, 2022
Inputs needed: Adjusted observation reader, adjusted navigation reader, observation file from Kaggle, navigation GPS file from the York University GNSS laboratory, adjusted time matching code, GNSS positioning program, MATLAB software, Google Earth, residual plot, DOP plots.

Tasks to be performed: Identify outliers, separate outliers from other calculated positions, remove outliers, recompute estimates of receiver positions.
Output generated: Estimated positions of smartphone receivers with outliers removed with a higher positional accuracy.

WP ID: 18	WP Name: Plot In Google Earth
Expected effort in hours: 7 hours Expected effort in days: 3 days	WP manager: Melvin Gagarao, 6.5 hours Support: Ryan Nguyen, 0.5 hours;
Expected start date: February 14, 2022	Expected end date: February 17, 2022
Inputs needed: KML file of the estimated receiver positions, Google Earth software.	
Tasks to be performed: Import the estimated receiver positions into Google Earth.	
Output generated: Plot of the estimated receiver positions on Google Earth.	

WP ID: 19	WP Name: Relate Calculations To Reference Positions
Expected effort in hours: 9 hours Expected effort in days: 5 days	WP manager: Jared Yen, 8.5 hours Support: Nathan Stachow, 0.5 hours;
Expected start date: February 14, 2022	Expected end date: February 19, 2022
Inputs needed: Estimated receiver positions in the geographic coordinate system, GNSS positioning program, MATLAB software, reference receiver positions from Kaggle, Excel software.	

Tasks to be performed: Export Excel reference positions to MATLAB, read in reference receiver positions, calculate the differences between the positions, find the mean error of the differences.
Output generated: Calculated differences between the calculated and reference receiver smartphone positions.

WP ID: 20	WP Name: Debug Program
Expected effort in hours: 55 hours Expected effort in days: 18 days	WP manager: Nathan Stachow, 45 hours Support: Jared Yen, 5 hours; Nathan Stachow, 5 hours;
Expected start date: February 21, 2022	Expected end date: March 13, 2022
Inputs needed: Time matching code, corrections to measurements program, least squares estimation code, observation reader, navigation reader, observation GPS file from Kaggle, navigation file from the CDDIS website, MATLAB software.	
Tasks to be performed: Debug program for any issues, analyse residuals.	
Output generated: GNSS positioning program without as many programmatic issues.	

WP ID: 21	WP Name: Comment And Complete Program
Expected effort in hours: 55 hours Expected effort in days: 10 days	WP manager: Nathan Stachow, 45 hours Support: Jared Yen, 5 hours; Nathan Stachow, 5 hours;
Expected start date: March 14, 2022	Expected end date: March 28, 2022

Inputs needed: Time matching code, corrections to measurements program, least squares estimation code, observation reader, navigation reader, observation GPS file from Kaggle, navigation file from the CDDIS website, MATLAB software.
Tasks to be performed: Debug program for any issues, analyse residuals.
Output generated: GNSS positioning program without as many programmatic issues.

WP ID: 22	WP Name: Prepare Report
Expected effort in hours: 130 hours Expected effort in days: 22 days	WP manager: Ryan Nguyen, 50 hours Support: Nathan Stachow, 40 hours; Melvin Gagarao, 20 hours; Jared Yen, 20 hours
Expected start date: March 15, 2021	Expected end date: April 8, 2021
Inputs needed: Positioning results from the GNSS positioning program, DOP plots, residual plots, Google Earth plots, MATLAB software, Word processing software.	
Tasks to be performed: Report on the process of making the positioning software, display and explain positioning accuracy results.	
Output generated: Professional report documenting the results of the accuracy positioning program.	

WP ID: 23	WP Name: Prepare Presentation
Expected effort in hours: 110 hours Expected effort in days: 20 days	WP manager: Ryan Nguyen, 50 hours Support: Nathan Stachow, 40 hours; Melvin Gagarao, 10 hours; Jared Yen, 10 hours

Expected start date: March 17, 2021	Expected end date: April 8, 2021
Inputs needed: Positioning results from the GNSS positioning program, DOP plots, residual plots, Google Earth plots, MATLAB software, PowerPoint processing software.	
Tasks to be performed: Create PowerPoint outlining the project process and displaying the results of the accuracy point positioning GNSS program.	
Output generated: Professional presenting the results of the accuracy positioning GNSS program.	