Thesis

Lydia Drabsch

28th October 2017

Instantaneous Relative Positioning of Multiple GNSS Receivers

Stefan Williams
University of Sydney

Contents

List of Figures

List of Tables

1. Abstract

2. Introduction

epoch synchrosation followed by two fold optimisation process
need instantanous relative positioning with minimal calibration or setup of external hardware

Instantaneous Relative displacement/position between GNSS receivers.
Simulation case studies are presented to validate the mathematical models.

The algorithm presented is designed to replace current methods.

With a local reference point with a well known global location, the relative position of the other receivers

3. Literature Review

3.1 Position Requirements

- how/why to use position for applications
- absolute vs relative
- why use relative positioning
- knowing where you are positioned is important for data gathering, motion detecting and tracking, path planning
- formation flying, drones

3.2 Relative Positioning Technology

- line of sight methods
- pre-setup requirements
- long calibration setup?

- last one being GNSS

3.3 GNSS Operational Components

3.3.1 Space Segment

- current GNSS: explain GPS, GLONASS, galelao, chinese one constellations and how it works - what orbits are they in and why?: altitude
inbetween the two radiation belts?

3.3.2 User Segment

- typical accuracy for civilian accessable gps
- military has more precise stuff
- lower cost receivers have only one frequency band, error in timing

Unfortunately, low cost GNSS receivers rarely provide official access to the GNSS raw data. Previous studies have used customised bluetooth headsets or customised android platform mobile phones to investigate algorithms on low-cost GNSS receivers. More expensive receivers do allow raw data to be utilised, however they also provide other mechanisms such as duel frequencies and more accurate clocks, rendering the new algorithm *obtuse*. The mindset of *crowd-sourcing*/customising/flexible technology is changing the way manufactures build GNSS receivers. The new Android OS platform Nougat 7.0 provides the developer raw GNSS data at the software level.

3.3.3 Control Segment

3.4 GNSS Satellite Signals

- civilian GNSS using duel frequency, send CDMA, how decryption works
- what is psudorange?
- what is carrier phase
- clock bias

3.5 The GNSS concept

The base concept behind identifying the position of something using GNSS is remarkably simple. Satellite W sends out a radio signal at time X and it's position Y which the user received at time Z. The time difference is used to calculate the distance from the satellite's position. With this information from multiple satellites, the position of the user is triangulated.

- timing comparison between satellite and receiver to find psudeorange - ECEF frame of reference

3.5.1 2D case

3.5.2 3D case

- NLLS solve spheres - need 4 satellites minimum -

3.6 GNSS Error Sources

- how large

3.6.1 Clock Errors

- timing of received signal because of low cost clock on receiver.

3.6.2 Receiver Noise

- antenna phase

3.6.3 Ephemeris Errors

- satellite position is approximated
- due to gravity effects of other gravitational sources/ non-spherical earth/ what model is used?
- how long is the ephemeris data accurate for?
- actual location of satellite, where it thinks it is is based on a prediction model so its not 100% correct. what uncertainty in this location therefore vector is there?

3.6.4 Atmospheric Effects

- ionosphere and troposphere refraction - speed of propagation changes which alters the time of flight

`http://www.trimble.com/gps_tutorial/howgps-error.aspx`

The distance from a satellite to a user is calculated by the time difference when the radio signal was sent and when it was received. However, the speed of light is reduced when in the atmosphere compared to that in space.

The ionosphere is the upper layer of the atmosphere ranging from 50 to 500 km

3.6.5 Mutlipath Interference

3.6.6 Sagnac Effect

3.6.7 Electrical Interference

- space weather
- jamming?

3.6.8 GNSS Error Summary

3.7 Multiple Receivers

- problems arising with multiple receivers -

3.8 Current GNSS algorithms

- just reference implementation papers? - algorithms to make it more accurate
- use for motion tracking
- performance vs cost trade off
(http://ieeexplore.ieee.org.ezproxy1.library.usyd.edu.au/document/7530542/)

3.8.1 Standard Positioning Service

- single frequency and multi frequency to remove atmospheric affects

3.8.2 Differential GPS

- explain what it is
- what setup is required
- abs vs rel
- degree of accuracy

3.8.3 WAAS DGPS

3.8.4 SBAS ?

3.8.5 Real Time Kinematic

3.8.6 Post Processing Algorithm

3.8.7 Single Frequency Precise Point Positioning (SF-PPP)

Rademakers how to say reference? at University of Delft in the Netherlands developed a solution for finding the absolute position in open areas to a horizontal accuracy of 0.5 m. It uses a single frequency, single antenna low cost GPS receiver by connecting to the internet and using real time information to model all errors. The errors they corrected with the potential improvements are outlined in Table 3.1.

3.8.8 Duel-Epoch, Double-Differencing Model

In the paper by the Institute of Software Integrated Systems, Vanderbilt University called *High-Accuracy Differential Tracking of Low-Cost GPS Receivers*, Hedgecock and party developed a new algorithm for relative motion tracking for multiple receivers. They used low cost GPS receivers with access to raw measurement data to produce centimeter-scale tracking accuracy. Each receiver was shared the whole networks data and ran the localisation algorithm independently to avoid having a single point of failure.

Table 3.1: Error Components and Potential Improvements for SF-PPP

Error component	**Potential Improvement**
Ionosphere: Klobuchar model	7 m
Troposphere: Saastamoinen model	2.5 m
Ephemeris data	1 m
Satellite clock drift	1.5 m
Differential code bias	50 cm
Phase windup: rotation of the antenna	dm
Sagnac effect	30 m
ROA: satellite orbit correction	up to 10 cm
Relativistic clock correction	up to 21 m
Moon-Earth interaction	5cm (Hor) and 30 cm (Ver)

The algorithm uses the change in carrier phase through time of each receiver to estimate the change in relative ranges between a satellite and two receivers. It does not require a reference satellite, a reference node or an integer ambiguity solution. It does require the clock bias for each receiver at each point in time as solved for by non-linear least squares for the absolute position before running the algorithm itself. To reiterate, it does not directly solve for the relative position but the relative motion. However, neither of the initial positions of the receivers need to be precisely known in order for he relative motion to be accurate. Due to the time dependency, consistent satellite locks of at least four satellites are required, otherwise reinitialisation must occur. The calculated change was projected onto the unit direction vector from receiver to satellite. The system of these tracking equations was solved via least squares optimisation.

It uses the assumption that all satellites in the constellation are such a great distance from the surface of the Earth that the unit vector from both receivers are parallel to each individual satellite, as long as it is in the same geographical region. How far apart the receivers can be for this assumption to hold was not stated.

- how many receivers?
- why and how it aligns epoch
- uses difference in time for a single receiver to find change in motion.
- have this one last as it is the most similar
- needs instantaneous relative distance for first point, to speed up processing and make the first few time steps more accurate, also when locking onto new satellites

3.8.9 Summary of Algorithms ?

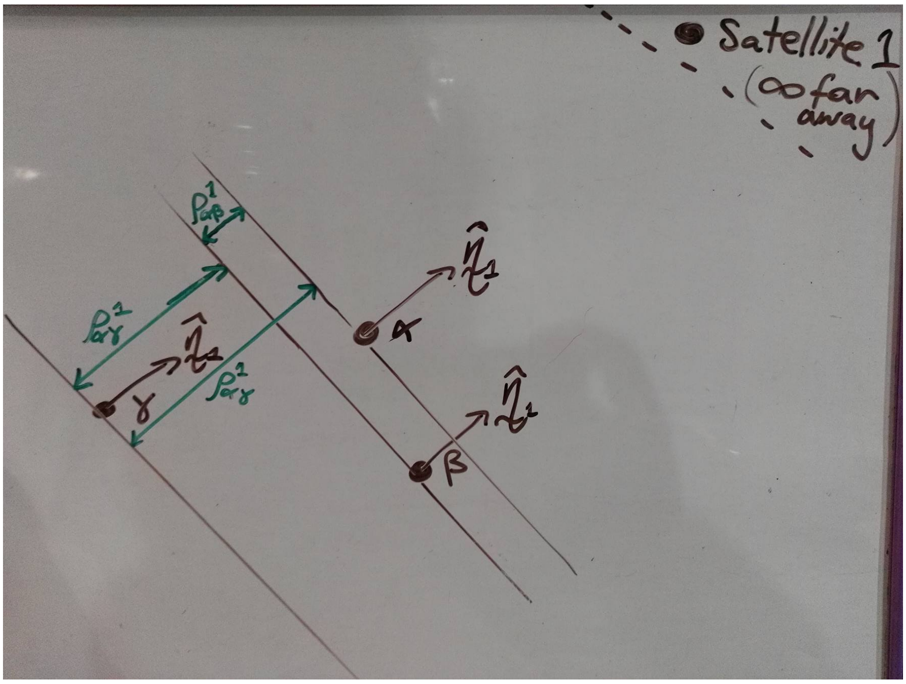
- dynamic tracking (need temporal measurements) vs static measurement - no temporal
- post processing vs pre-processing vs realtime
- ground structure vs free standing
- absolute vs relative
- accuracy (how much)
- computation time/space required
- what error is each method removing
- what piece of data it needs (if raw)
- calibration required
- robustness -¿ if a satellite goes out of view does it need to re-calibrate? passing information between receivers-¿ is one a reference? single point of failure

3.9 Proposed Planar Intersection Algorithm

Following the literature, the two main options for increasing the accuracy is extensive modelling of the errors, or some type of differencing algorithm. Error modelling requires external hardware, internet connection and extra computation time which increases the budget requirements. Whereas

This new algorithm is derived from taking the difference in pseudorange between multiple receivers from one satellite and expressing the distances as planes.

Figure 3.1: 2D representation



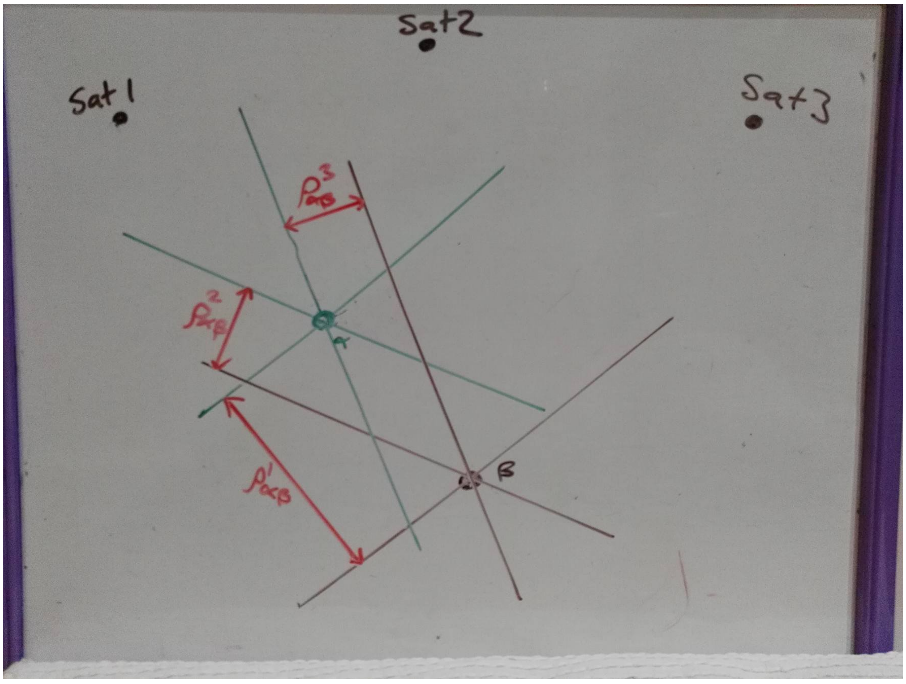
With multiple satellites in view, the intersection of planes for a particular receiver is the position of the receiver.

With this strategy, some of the errors that plague the absolute position are negated for the relative position.

$$\rho_i = \rho_n - cb_\omega + c(T_s + I_s + \nu_s + b_s) \tag{1}$$

where ρ_n is the real range with the following sources of error; b_ω is the receiver clock bias, T_s is the tropospheric error, I_s is the ionospheric error, ν_s is the relativistic error and b_s is the satellite clock bias.

Figure 3.2



4. Planar Intersection Algorithm

- assume satellite is at infinity for comparing difference in pseudorange for a particular reference satellite.
- use all satellites as reference satellite - no single point of failure, also not all satellites might be in view for all receivers
- get the normal vector between all receivers and each sat.
- Calculate the average normal vector.
- get the difference in pseudorange between all receivers along each normal vector
- create a plane with the normal vector with that distance
- solve via optimization (least squares)
- use clock adjustment from abs gps? or have as another optimisation variable
- antenna problems? misalignment?
- share clock bias's between solving for different reference sets? - do it one by one or all together?
- need to align the time of signal sent to the receivers before calculating average normal vector

- have weighted planes based on ? have weighted area on the planes?
- if one plane intercepts far away from the others then ignore it (multipath). hyperdimensional surface to minimise

how to send data between receivers? do it offline on a different platform?

https://www.e-education.psu.edu/geog862/node/1759 - errors in psedeuorange

http://www.insidegnss.com/node/2898 - how to get pseudorange from raw data

4.1 Assumptions

4.1.1 Static Receivers

All receivers are static for the time in between all receivers get a GPS lock. This makes for an easier transform to align the satellite positions to a common time. It also ignores the problem of how the pseudorange from each receiver would be sent to either all of the receivers or to a central device for computation and the time delay associated with that. The incorporation of moving receivers is an area to explore for future work on the algorithm.

4.1.2 Transform asynchronous time

any two receivers will not be synchronized. The earliest time between all the receivers will be used as the time reference point. The satellite position in the future time steps were backcalculated to find the difference in the pseudorange. As the time between receivers will be ≈ 1 second, this extra distance is only in the vacuum of space and is not affected by potential nonlinear affects such as ionosphere and troposphere errors that affect the speed of light.

The error in the normal vector due to the time difference is

4.1.3 Parallel plane assumption

It as assumed for the plane equations that all receivers point to a satellite along the same vector. This is valid for a dispersion of receivers for 10km for an error of XX. This is synonymous to if the satellites were at infinity and all the receiver vectors are parallel to a satellite

$$\delta = \tan^{-1}\left(\frac{d}{a}\right) \tag{2}$$

$$e = 2d\tan\delta \tag{3}$$

$$(2)\&(3) \Rightarrow e = \frac{2d^2}{a} \tag{4}$$

Where a is the altitude, d is the distance between two receivers and e is the error in the plane created. The worst configuration for error in the vector normal to the plane is if the satellite is directly above the receivers at the smallest distance from the Earth in orbit, $a > 20000\ km$. For d=5 km the perpendicular error is 2.5 m

4.2 Algorithm

4.2.1 Pre-Processing

4.2.1.1 Select reference receiver α

The receiver α is used as the reference location and common time in the NED frame.

4.2.1.2 Collect data of one timestep from all receivers

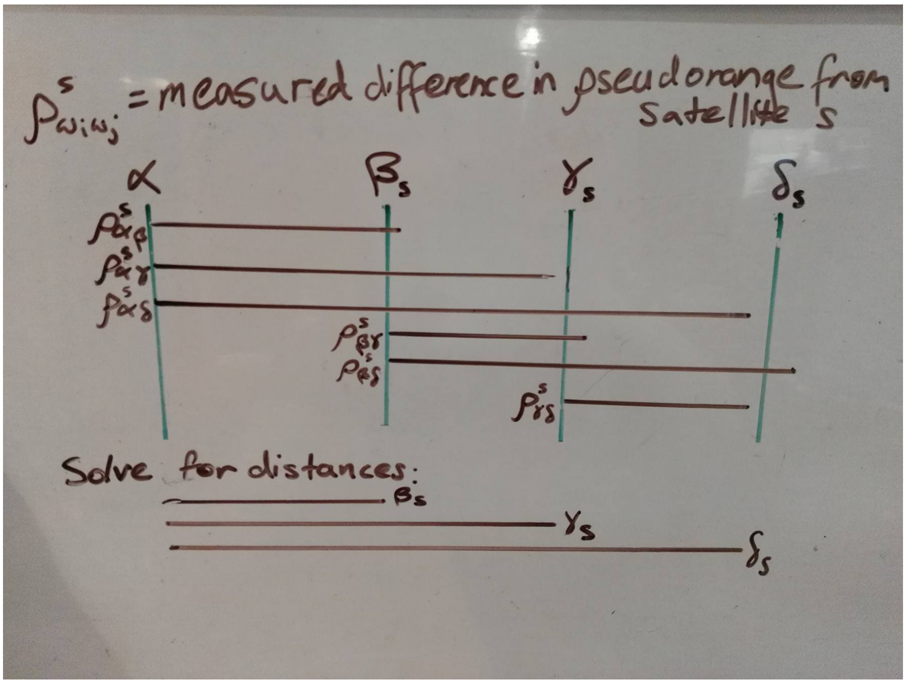
The raw data as well as the estimated absolute location and clock bias (what frame of reference is this?) from non-linear least squares optimisation is collected from all GNSS receivers.

4.2.1.3 Align to reference Epoch time

4.2.2 Distance Optimisation

By optimising the distance between each pair of receivers, the error in the whole system is minimised. This means that the position receivers are not only relative to the reference receiver α but between all receivers just with the reference frame origin located at α. It is because of this step a receiver does not need to have all the same satellites in view as all other receivers, including the designated α.

Figure 4.1: text



4.2.2.1 Average normal Vector

Find the average normal vector pointing to each satellite $\hat{\eta}_s$ from the receivers. The normal vector is calculated by using the position all of the satellites in view at the common time t_α as previously transformed in 4.2.1.3 and the estimated absolute position of all receivers. The average for each satellite is calculated by taking the mean across all receivers.

4.2.2.2 Difference in Pseudorange

The differences in pseudorange are calculated $\Delta\rho^s_{\omega_i\omega_j}$ where s is the satellite, ω_i and ω_j are receivers $(for i < j, i \neq j)$.

4.2.2.3 Optimise Pseudorange

The pseudorange between each pair of receivers along each normal vector $\hat{\eta}_s$ creates an overdetermined linear system that is solved via least squares.

$$\Phi = \begin{bmatrix} 0 & -1 & 0 & ... \\ 0 & 0 & -1 & ... \\ ... & ... & ... & ... \\ 1 & -1 & 0 & ... \\ 1 & 0 & -1 & ... \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & -1 & ... \end{bmatrix} \tag{5}$$

$$\Omega_s = \begin{bmatrix} \beta_s \\ \gamma_s \\ \delta_s \\ \vdots \end{bmatrix} \tag{6}$$

$$\rho_s = \begin{bmatrix} \rho_{\alpha\omega_1} \\ \rho_{\alpha\omega_2} \\ \cdots \\ \rho_{\omega_1\omega_2} \\ \rho_{\omega_1\omega_3} \\ \vdots \end{bmatrix} \tag{7}$$

$$\Phi \times \Omega_s = \rho_s \tag{8}$$

Solve by linear least squares for an overdetermined system by the pseudo inverse matrix

$$\Omega_s = (\Phi^T\Phi)^{-1}\Phi^T\rho_s \tag{9}$$

4.2.3 Point Optimisation

4.2.3.1 Create Planes

Create sets of planes for each receiver ω from the normal vectors $\hat{\eta_s}$ and the set of distances from the reference point α to receiver ω along each of the normal vectors denoted Ω_ω.

The equation of a plane is $Ax + By + Cz + D = 0$ where the coefficients [A,B,C] describe the normal vector of the plane and the coefficient D sets the plane in 3D space along the vector. As the normal vector is already calculated for each satellite, only the D coefficient must be solved for each receiver and satellite pair.

$$P^s_\omega = (i \cdot \hat{\eta_s})x + (j \cdot \hat{\eta_s})y + (k \cdot \hat{\eta_s})z + D^s_\omega \tag{10}$$
$$P^s_\omega = I \cdot H + D_\omega \tag{11}$$
$$\tag{12}$$

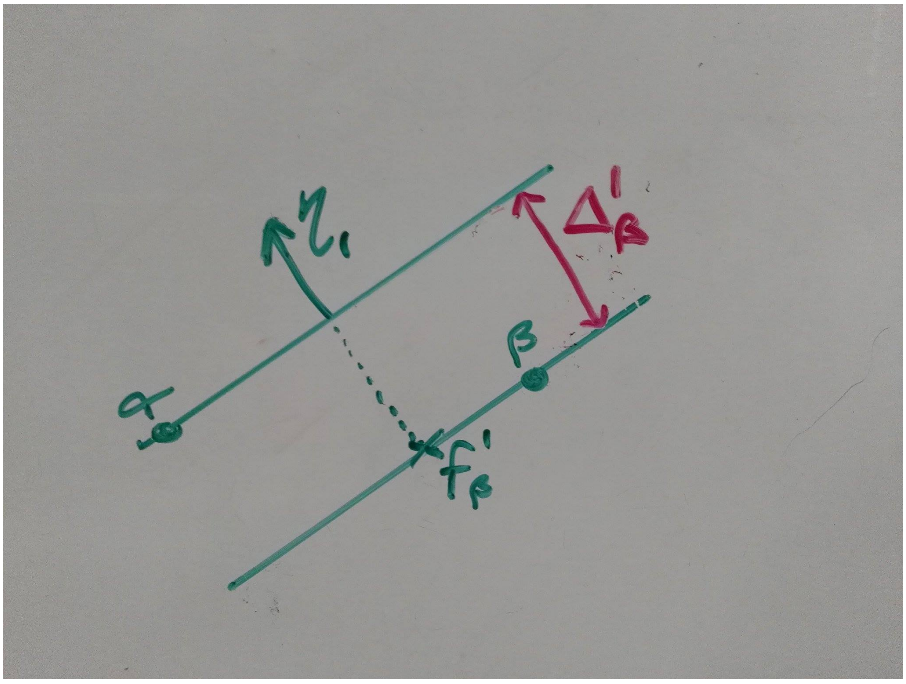
Where $I = x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}}$ is the *identity* vector and H is a matrix of normal vectors to each satellite:

$$H = \begin{bmatrix} \hat{\eta_1} \\ \hat{\eta_2} \\ \vdots \\ \hat{\eta_n} \end{bmatrix} \tag{13}$$

The coefficient D can be calculated by finding a point on the plane f^s_ω, then substituting it into (10) for x,y,z. The point of the plane is calculated by moving along the normal vector by the optimised pseudo distance from the reference point (14).

$$
\begin{aligned}
f^s_\omega &= \Delta^s_\omega \hat{\eta_s} &(14)\\
P^s_\omega &= \hat{\eta_s} \cdot f^s_\omega + D^s_\omega = 0 &(15)\\
D^s_\omega &= -\hat{\eta_s} \cdot f^s_\omega &(16)\\
D^s_\omega &= -\Delta^s_\omega ||\hat{\eta_s}|| &(17)\\
||\hat{\eta_s}|| &= 1 &(18)\\
D^s_\omega &= -\Delta^s_\omega &(19)\\
\Rightarrow P^s_\omega &= I \cdot H - \Omega_\omega &(20)
\end{aligned}
$$

Figure 4.2: Find position f^s_ω on the plane



Ω_ω is a vector of optimised pseudo-distances from reference α to receiver ω for all satellites $s \in 1, 2...n$

$$\Omega_\omega = \begin{bmatrix} \Delta^1_\omega \\ \Delta^2_\omega \\ \vdots \\ \Delta^n_\omega \end{bmatrix} \tag{21}$$

Where Ω_s is the vector of optimised pseudo-distances from α to each receiver $\omega \in 1, 2...m$ for a single satellite s:

$$\Omega_s = \begin{bmatrix} \Delta^s_{\omega_1} \\ \Delta^s_{\omega_2} \\ \vdots \\ \Delta^s_{\omega_m} \end{bmatrix} \tag{22}$$

4.2.3.2 Solve for Intersection

As the system of homogeneous linear equations is overdetermined, it can be solved using singular value decomposition to find a point that has the minimum residuals from all of the planes in its set

P_ω. Each set of planes for a particular receiver is independent to all other receivers. The vector X_ω describes the position of receiver ω in NED coordinates and τ_ω describes a final receiver clock bias that alters the displacement of all the planes in the set P_ω by the same parameter.

$$X_\omega = \begin{bmatrix} x_\omega \\ y_\omega \\ z_\omega \\ \tau_\omega \end{bmatrix} \quad (23)$$

$$P_\omega X_\omega = D_\omega \quad (24)$$

$$(25)$$

In order to solve all of the receivers with the least amount of error in the whole system, all of the position vectors X_ω are solved at the same time. The reference planes of α must be included as a constraint on the system. All of the clock biases are also constrained with the clock bias from τ_α, see (26). The receiver clock bias only affects the equation of the planes by altering the constant as a change in the pseudorange has no affect over the angle of the plane. Each receiver clock bias alters all the planes associated with that receiver proportionally.

$$P^s_\omega = (i \cdot \hat{\eta_s})x + (j \cdot \hat{\eta_s})y + (k \cdot \hat{\eta_s})z + D^s_\omega + (\tau_\omega - \tau_\alpha) \quad (26)$$

$$(27)$$

5. Method

5.1 Creating a Simulation

Matlab was chosen as the platform to simulate and evaluate the program due to the ease of matrix manipulation and graphical interaction.

5.1.1 Simulate Satellite Locations

Using real ephemeris data for the GPS constellation, the location of all of the satellites were calculated, see Table 5.1. The approximate reference location of the network of receivers was inputed into the program as longitude, latitude, height geocentric coordinate system. The satellite positions were transformed to the local tangent plane of the reference location in polar coordinates. The satellites that had an elevation of above 12deg were selected as potential visible satellites. This lower elevation limit was selected to minimised multipath effects that would likely occur at ground level REF.

5.1.2 True Location of Receivers

The dispersion of receivers from α is a variable to the program. The actual displacement is calculated by multiplying the dispersion magnitude by a uniformly distributed random vector ranging from [0,1] in three dimensions for all remaining receivers in ECEF frame localised at α. The positions were then transformed to the global ECEF frame.

5.1.3 Calculation of Pseudorange

In ECEF frame, the instantaneous distance between each visible satellite and each receiver was calculated. The errors were simulated by adding random distance proportional to error models in the

Table 5.1: Ephemeris data for GPS constellation

Orbital parameters	Value

Table 5.2: Magnitude of simulated errors

Error Source	meters

literature to each individual satellite, see Table 5.2. The errors followed the structure in Eq In the literature, the errors were expressed in meters which were converted to seconds.

$$Error(seconds) = (Error) \tag{28}$$

5.1.4 Planar Algorithm

The normal vector was
- what data is it using from receiver? psudorange, time

- what errors to include and how to incorporate into the simulation.
- how to include the different(asynchronous) time received for all receivers-¿ is for the one receiver
- how extra receivers affects computational time/ accuracy
- how number of sats affect comp time/accuracy
- configuration of sats
- large multipath affects
- no receiver sees the same sat? - does it just output the relative difference between abs values? -¿ incorrect? just have it fail? not actually implementing, can control the environment
- distance of receivers apart
- configuration of receivers
- what data received and how to simulate misaligned timing between receivers
- what magnitude are the errors and how to simulate them
- simulate the errors individually (to see how each type affects the sim - convergence time and accuracy) and/or all errors at once

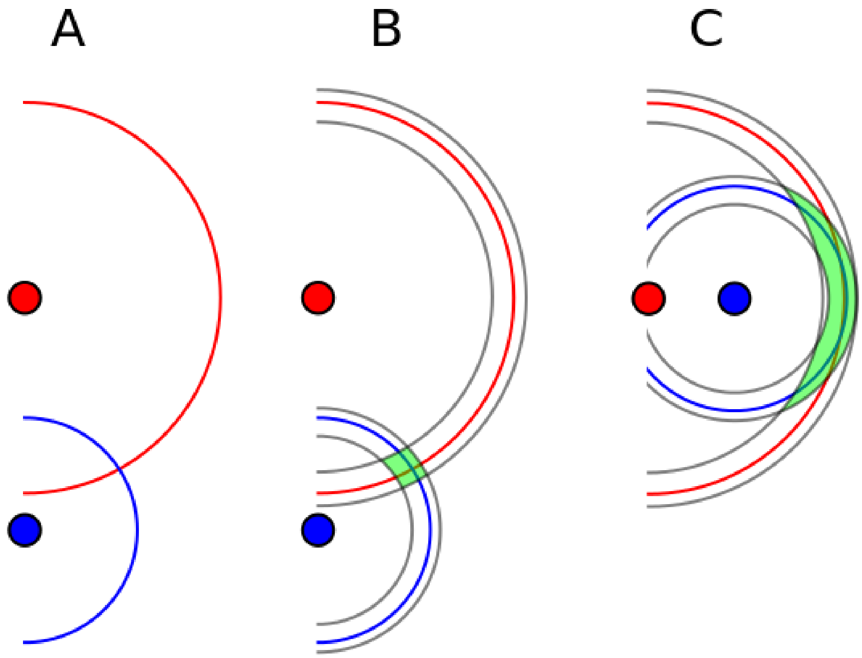
5.2 Evaluation

The expected error due to the assumptions were outlined in Section XX. To evaluate the parallel plane assumption, two receivers were a set distance apart.

5.2.0.1 Dilution of Precision

The dilution of precision (DOP) is a measure of how the geometry of the satellites affect the position measurement precision. It is used in GNSS to evaluate how much error there may be in a measurement. Figure 5.1 shows two different configurations of satellites in the 2D circular solution case. With some error bounds as shown in B and C, the solution can lie anywhere in the green area. Due to the geometry configuration of the satellites in C, the area is considerably larger even though the error bounds on the signals are the same.
- fake gps data
- how to simulate noise - what level SNR
- to calculate your own GPS location using the normal algorithm? - space 3
- use real GPS locations? (and through time) -space 3
- vary number of satellites in view
- vary GDOP (good GDOP and bad)
- when receivers don't see the exact same satellites
- vary number of receivers
- simulate a multipath error and how does it account for it or how much error does it introduce

Figure 5.1: Geometric Dilution of Precision



How to evaluate?: - accuracy in relative space
- compare to just taking differences in absolute position
- between individual receivers and the total error in the whole system
- markov? error analysis -¿ cannot do precision without statistical analysis but isolate errors in x,y,z. how much worse is z than horizontal?
- computational time-¿ how does more receivers/satellites affect the comp time -¿ what time and space complexity?

6. Result

References