**Small batteryless wireless
solar powered GPS
receivers for subcentimeter
land deformation
monitoring**

by

Jonathan Paul Simon Olds

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the requirements for the degree of
Master of engineering

Victoria University of Wellington
(2015)

Abstract

We design, implement, and validate a unique permanently deployed land deformation monitoring system using small (brick sized), cheap (approximately \$100 USD), batteryless, solar powered singleband GPS wireless sensor nodes. Both hardware and software were designed, implemented, and validated by us. Constraints by our hardware and application prompted us to design a unique distributed relative static positioning algorithm designed for intermittent poor quality phase observable measurements, for sites with high multipath and high node densities requiring good solution accuracies; the static solutions were calculated on a daily basis. Our algorithm used a quarter of the bandwidth that would typically be required for an RF link used for a comparable application. GPS on time was observed to vary greatly from as little as 0.5 hours a day in winter to over 8 hours a day and summer in one of our tests. Typical solution precision was 4 mm 2DRMS. Simulations predicted an undesirable slowly changing solution bias that would repeat every year.

Acknowledgments

I would like to thank the following...

The kind people at NCNU for their warm hospitality and assistance during my stay with them in Taiwan.

My mother for her support during this endeavor.

Cat and Tink for their company and inspiration.

Tim for his weekly chats and ordering components.

EQC for funding me.

My supervisor, without whom, this couldn't have happened.

Contents

1	Introduction	1
1.1	Related work	13
1.2	Contributions	17
1.3	Thesis layout	19
2	The Code Float Fix Sidereal algorithm	21
2.1	Hardware influence on the algorithm design	21
2.2	General idea of solving	23
2.3	Algorithm	24
2.3.1	Required data	25
2.3.2	Data flow	25
2.3.3	Node section	27
2.3.4	Main section	29
2.3.4.1	Wrapped double difference creation algorithm	30
2.3.4.2	Code	30
2.3.4.3	Float	30
2.3.4.4	Fix	34
2.3.4.5	Sidereal	37
2.4	Independent LS and reference satellite selection	40

CONTENTS

3	Multipath and sidereal filtering	43
3.1	Satellite and multipath periodicity	46
3.2	Remaining errors	48
3.2.1	Measurement noise	49
3.2.2	Multipath error	50
3.3	Interday sidereal filtering	52
3.3.1	Simulation validation	63
3.3.1.1	Sidereal response	64
3.3.1.2	Sidereal solution improvement	65
3.4	Intraday sidereal filtering	68
3.4.1	Intraday sidereal filtering improvement	71
3.4.2	Intraday sidereal filtering response	74
3.4.2.1	5 mm permanent power	74
3.4.2.2	2 cm solar power	77
4	Algorithm comparison	81
4.1	Reduction of network bandwidth comparison	81
4.1.1	Real life reduction of bandwidth comparison	83
4.2	Solution comparisons	85
4.2.1	Comparison methodology	88
4.2.2	Continuous high quality data	89
4.2.3	Simulated intermittent high quality data	92
4.2.4	Algorithm performance using real life intermittent poor quality data	94
4.2.5	Solution comparison summary	97

CONTENTS

5	Hardware	99
5.1	Node	100
5.1.1	Node hardware design requirements	107
5.1.2	PMB	108
5.1.3	Node quiescent power	109
5.1.4	Peripheral power use	110
5.1.5	Active node power use	111
5.1.6	Solar power supply	112
5.1.6.1	Dark current and blocking diode	118
5.1.6.2	Solar panel supply summary	119
5.1.7	Super capacitor sizing	119
5.1.8	Precise solutions at 20 Wm^{-2}	122
5.1.9	GPS antenna	123
5.1.10	Housing	123
5.2	Base station	124
5.3	Repeater	127
5.4	Remaining hardware	129
6	Testbed results and analysis	131
6.1	Paekakariki testbed	131
6.2	LuShan testbed	135
6.2.1	Movements relative to node 274	142
6.2.2	277 and 276 relative to node 1	144
6.2.3	Comparison with retro reflector observations	149

CONTENTS

7	Conclusions	155
7.1	Conclusion summary	161
	Nomenclature	163
	Appendices	167
A	GPS basics	169
A.1	GPS signal	169
A.2	What the GPS receiver does to the received $W_{RX}(t)$ signal . . .	172
A.2.0.1	Obtaining the baseband signal $R(t)$	172
A.2.0.2	Stopping rotation	173
A.2.0.3	C/A Code alignment	173
A.3	Acquisition, tracking and NAV data extraction	174
A.3.0.1	Extraction of NAV data using filtering	174
A.3.0.2	A metric for C/A code alignment and rotation .	175
A.3.0.3	First-order linear approximations of unknown functions τ and Φ	177
A.3.0.4	Further simplifications to the first-order linear approximations of unknown functions τ and Φ in regard to acquisition	178
A.3.0.5	Acquisition	179
A.3.0.6	Tracking	181
A.4	Observables	182
A.4.1	The code observable	183
A.4.1.1	Calculating code based solutions	184

CONTENTS

A.4.1.2	Final code observable model	187
A.4.2	The phase observable	188
A.4.2.1	Final phase observable model	190
A.5	Selected proofs	193
A.5.1	Received phase using flight time approximation.	193
A.5.2	τ First-order linear approximation	193
A.5.3	Radial velocity with constant radial velocity offset ϵ . . .	194
A.5.4	Maximum radial velocity and acceleration of the satellite with respect to the receiver	194
B	Differencing the phase observable	197
B.1	Single difference	197
B.1.1	Single difference interpretation	202
B.2	Double difference	204
B.2.1	Double difference interpretation	205
B.2.2	Double difference cycle slips	207
B.2.3	Double difference residual	207
B.3	Wrapped double difference	208
B.3.1	Wrapped double difference interpretation	208
B.3.2	Wrapped phase observable	211
B.3.3	Unwrapping the wrapped double difference	213
B.3.4	Wrapped double difference residual	215
B.4	Selected proofs	216
B.4.1	Range rate calculation	216

CONTENTS

C Node functionality at night	219
C.1 Assisted GPS	219
C.2 Circadian energy distribution algorithm	224
C.3 Circadian energy distribution algorithm and AGPS system im- plementation results	226
D Network	231
D.1 Internet section	231
D.2 The WSN section	233
D.2.1 Repeater	233
D.2.2 Device server	234
D.2.3 Phasing	235
D.2.3.1 Phasing algorithm	236
Bibliography	241

Chapter 1

Introduction

Purpose: To determine displacement.

There are many such ways to measure the movements of the Earth; the American *Global Positioning System* (GPS) is just one such way. GPS is one of the *Global Navigation Satellite System* (GNSS) and is currently the most well known and ubiquitous. Others such as the Russian version *GLObalnaya NAVigationnaya Sputnikovaya Sistema* (GLONASS), China's BeiDou, and the European Galileo also exist with other systems planned. While differences exist amongst the systems, the basic principle remains; sending signals from the satellites to determine the range to the satellite.

Being able to measure the movements of the Earth is an important role for land deformation monitoring and we use this as the primary application for our work.

Land deformation monitoring is useful for mitigation of property damage as well as mitigation of injury to people. It is also useful from a scientific standpoint in gaining a deeper understanding of the processes that shape the Earth. Some examples for sites that might benefit from land deformation monitoring are volcanoes, slopes prone to landslides, glacial flows, fault lines, subsidence, and large man-made structures such as dams and buildings to name a few.

The number of different techniques to monitor land deformation are many and varied. Methods include GPS, tilt meters, aerial photography, *Interferometric Synthetic Aperture Radar* (InSAR), lidar and total stations. Methods such as

CHAPTER 1. INTRODUCTION

space or aerial InSAR and lidar can cover areas of hundreds of kilometers and can obtain a spatial resolution far better than any practical GPS deployment one could imagine, while still producing accurate solutions comparable to GPS. However, solutions obtained from space or aerial InSAR and lidar produce solutions relatively infrequently and there may be many weeks between solutions [43]. Tilt meters can provide sub daily estimates but their sensors drift and require calibration [43]. GPS has the advantage of being able to obtain accurate position solutions but permanently deployed networks can be expensive to install and operate and only produce solutions for discrete points where the GPS receivers are installed [43].

GPS survey campaigns can produce a point for every place visited but are time consuming, produce solutions infrequently and intermittently, are labour intensive, require a skilled operator, and may not even be possible for dangerous sites; for these reasons it is customary to use a network of permanently deployed GPS receivers. While permanently deployed GPS receiver networks can be expensive to implement and maintain, they can produce consistent frequent solutions circumventing some of the problems associated with survey campaigns but only supply solutions for a fixed set of positions. In this thesis we focus on permanently deployed GPS networks.

Permanently deployed GPS system design considerations:

The following lists some of the considerations one needs to take into account when designing a permanently deployed GPS system.

- Network scalability
- Observation quality
- Signal tracking reliability
- Power
- Maintenance
- Price
- Node density

CHAPTER 1. INTRODUCTION

- Node size
- Baseline length
- Ease of deployment
- Precision and accuracy
- Environment quality
- Communication network reliability
- Power supply reliability
- Operator skill
- Time
- Site danger
- Number of solutions per day

GPS receivers commonly are classified into three different types, recreational grade, mapping grade and survey grade. Survey grade GPS receivers are the most accurate of all three types of receivers while recreational grade are the ones such as found in cell phones to which people are most accustomed to and have the poorest accuracy. Dual band survey grade GPS receivers when using phase based differential static survey methods can produce solutions with accuracies less than 1 cm [14, 26]. However, with a pricetag of more than \$8,000 US each [18, 61] this type of receiver comes at a cost. Phase based differential static survey methods can produce the greatest accuracy but also requires the greatest period of site occupation; typically at least 30 minutes but can also be many hours [14, 26]. To obtain an idea of accuracy and time required for this accuracy we consider the ProMark 700 from Spectra Precision Division as an example. This GPS receiver is a multiband survey grade receiver capable of receiving GPS and GLONASS costing \$8279.17 USD from Cansel as of writing (Feb/2015) [1]. The datasheet states that high precision static processing can obtain a 2DRMS accuracy value of 10 mm and may take up to 24 hours to obtain this accuracy [51]. Non-static methods such as *Real Time Kinematic* (RTK) may not require extended occupation times but produce poorer quality

CHAPTER 1. INTRODUCTION

accuracies as can be seen in the 20 mm *2 times Distance Root Mean Square* (2DRMS) specification of the ProMark 700 [51].

A static surveying method is a method whereby many observations are used to calculate one solution; while taking these observations the GPS receivers must remain static for a period of time called the site occupation time. Differential techniques come in various forms but using the phase observable of the GPS satellites produces higher accuracy results if the phase ambiguity can be correctly resolved. This is because the wavelength of the carrier wave is about 1500 times shorter than that of chip length of the code observable [21], thus allowing a much more accurate determination of range to the satellites all things being equal. This of course is only valid if the phase ambiguities can be correctly resolved which in itself has been the field of much research with 300 papers being published on the subject during the decade of the 90s alone [38]. Differencing the phase observable between two satellites and two receivers can be done to produce what is known as a double difference removing most systematic errors if given a short baseline between the two receivers of less than say 10 km. With distances less than 10 km ionosphere error tends to cancel out due to the high correlation between the ionospheric error and receiver separation. For longer baselines dual band receivers can estimate the ionospheric error and hence the distance between receivers can be extended. Using double differences is a common differential phase technique used for high accuracy work. Differential techniques always result in relative solutions. Of the remaining errors the phase ambiguity and the multipath are the main errors left.

Whilst accuracy is good, the shortcomings of survey grade GPS receivers for land deformation monitoring are that they are high power and expensive. Because of their high power demands and that generally there is no mains power supply at land deformation sites, items such as solar panels and batteries for powering such devices can be very large, increasing cost and size. Increased size presents deployment issues, becoming more time consuming, expensive and difficult for long term monitoring sites. Therefore, the density with which such receivers are typically deployed is sparse. The sparseness of such systems then require higher power communication radios placing added demand on the power supply. Figure 1.1 shows a typical survey grade reference station set up at Paekakariki (New Zealand); it is an excellent site with a good view of the sky; it is fenced off to prevent vandalism from grazing animals and requires

CHAPTER 1. INTRODUCTION

regular maintenance such as lawn mowing, battery replacement, etc.

Figure 1.1: Typical survey grade GPS receiver reference station at Paekakariki (New Zealand). Picture sourced from GeoNet [4]

Consumer grade GPS receivers are singleband with uncalibrated antennas and are usually only intended for autonomous course acquisition code solutions with an accuracy of a few meters at best. However, effects such as the atmosphere and structures, inaccuracies in the ephemerides, timing, etc introduce other errors degrading accuracy even more. This type of receiver is typical in cell phones today. Being autonomous solutions, the solutions are absolute therefore useful for telling you where you are in the world, but not particularly suited for land deformation monitoring where relative solutions which are more accurate are better suited. While the power of consumer grade GPS receivers is generally low, autonomous solutions cannot be seriously considered as a viable alternative to differential phase based methods where accuracy is demanded. However, just because consumer grade GPS receivers are not intended for precise work does not mean that they are unable to do so; some modules output pseudorange and phase data such as SirfIV, NVS NV08C-CSM, Skytraq S1315F-RAW, Ublox LEA-6T, Ublox NEO-6P and Ublox NEO-7P thus allowing differential phase based methods to be used.

RTK is a real time differential phase based method. It is very popular due to its ability to obtain a solution each epoch in real time. RTK is not as accurate as static solutions. As an example of RTK and what a consumer grade receiver costing \$50 USD with a low cost uncalibrated antenna can do under good conditions, an antenna was placed on a roof, after 10 minutes the

ambiguities were fixed and then the antenna was moved a few centimeters as can be seen in Figure 1.2; each point represents one second.

Figure 1.2: RTK example: 4km baseline, singleband, 10 minute *Ambiguity Resolution* (AR) time, good conditions

Besides accuracy, one shortcoming of RTK is that it is very susceptible to poor quality environments and loss of continuous tracking of the phase observable. From our experience ambiguity resolution can be very difficult to achieve with low quality antennas in poor quality multipath prone environments; without ambiguity resolution solutions are considerably more inaccurate.

Generally the cost of mitigating one problem is at the expense of others. For example, accuracy can be improved at the expense of the number of solutions per day, size, etc. Accuracy is one of the aspects of land deformation that is regarded highly; because of this we do not entertain solutions that would significantly degrade solution accuracy such as not using phase observations. Both temporal and spatial resolution are also important factors for land deformation monitoring. We would like to improve spatial resolution. Spatial resolution requires many GPS receiver nodes for permanently deployed networks. To achieve this we need to design, cheaper, smaller equipment, that need less power, less maintenance, less communication bandwidth, and are easier to deploy. Temporal resolution depends on the scenario but one solution per day is typical for long term monitoring scenarios where accuracy is paramount, it is this rate that we aim for.

Initial problems to solve:

CHAPTER 1. INTRODUCTION

- Reduce price
- Reduce size
- Reduce weight
- Reduce hardware requirements
- Reduce deployment requirements
- Reduce maintenance requirements
- Able to obtain accurate solutions in poor quality environments with high multipath

The initial problems we desire to solve are to reduce price, size, weight, hardware, deployment and maintenance requirements of a system while still producing accurate solutions in poor quality environments with high multipath. We propose to remove the battery from the permanently deployed receivers (nodes) and use a small solar panel and a super capacitor to allow duty cycling of the GPS receiver even in very low light levels. Removing the battery means there is no need for battery maintenance, eliminating this as a cost. It also eliminates human exposure to hazardous sites for battery maintenance and allows the possibility for smaller lighter nodes to be created. We propose that the size of a node including solar panel be no larger than about the size of a brick, weigh a few grams and be self contained. Being self contained and small allows a cheaper node to be created as the solar panel and enclosure are small, requiring only one item to be installed in the field. In addition, such a node is easier to deploy both due to its small size, weight and that it is entirely self contained. Due to the size, cost and power constraints the hardware of the nodes has to be chosen accordingly. We propose to use small, low cost, low power singleband consumer grade GPS receivers, small low quality cheap uncalibrated GPS antennas and small, low cost, low power, short range communication transceivers. We propose to use phase based double differences to obtain static solutions calculated on a daily basis; such relative phase based solutions produce good accuracy. We propose to implement sidereal filtering to mitigate multipath.

Induced problems:

CHAPTER 1. INTRODUCTION

In solving our initial problems we create a set of induced problems more difficult to solve; the following lists the main problems.

- Poor observation quality
- Unknown *Phase Center Offset* (PCO) and *Phase Center Variation* (PCV)
- Unreliable GPS phase accumulation
- Unreliable intermittent power supply
- High node density
- Short baseline length less than 10 km
- Variable precision
- Poor communication network reliability

These are induced problems due to the following. Using consumer grade GPS receivers with poor quality antennas results in poor observation quality [25]. Cheap uncalibrated antennas have an unknown PCO and PCV. Duty cycling of the GPS receiver itself means the GPS receiver cannot track the phase of the satellites reliably. The power supply is unreliable and intermittent during the day while there is no power at all at night; the available power depends on the time of year and weather conditions. A cheaper smaller node that is easier to deploy and maintain allows networks of greater density to be created, this in turn is a problem as it means added pressure on the communication link from the nodes to the base station. Using singleband receivers generally limits baseline length to less than 10 km. Due to the unreliable nature of the power supply the quantity and distribution of observations used for a static solution in the day causes solution precision to vary. Due to both limited power supply and higher density of nodes the communication network reliability is expected to be worse than would typically be expected.

Conceivably a sparse *Continuously Operating Reference Station* (CORS) network of dual band GPS receivers could be used to increase baseline length but we do not attempt to address this problem. As we are interested in measuring movements rather than physical distance between points we ignore the

CHAPTER 1. INTRODUCTION

unknown PCO. We propose to use sidereal filtering to minimize the error induced by the unknown PCV.

With such a small form factor removing the battery puts a great constraint as to how the hardware and software are designed that would otherwise not be considered. This constraint produces innovative solutions and is the driving force for inspiration. In particular not having a battery means careful consideration of energy consumption at all levels from the GPS receiver and antenna, communication protocol, solution algorithm, to radio links. A cheaper smaller node that is easier to deploy and maintain also allows networks of greater density to be created, this in turn means added pressure on the communication link from the nodes to the base station; because of this the communication protocol and solution algorithm have to be reconsidered to allow such an increase in node density. To mitigate the pressure on the communication link some form of distributed solution algorithm makes sense.

A standard axiom when using the phase observable is that the GPS receiver continuously tracks the phase of each satellite; this continuous tracking is called phase accumulation. Without a battery, for very low light levels, duty cycling of the GPS receiver is needed whereby the GPS receiver cannot track the phase of the satellite when the GPS receiver is in an off state. This in turn causes cycle slips that typical GNSS solution processing software would try to reinitialise the phase ambiguity for every duty cycle. It takes time to fix the phase ambiguities to integers, thus duty cycling can effectively mean a typical GNSS solution processing software would be unable to fix such ambiguities and therefore be unable to provide an accurate solution. In addition, communication protocols using simple low power communication radios can be lossy; from our results in Chapter 4 all GNSS solution processing software that we tried failed to produce accurate solutions when observations were sent over even mildly lossy radio links. This means an entirely unique solution algorithm must be designed. A solution algorithm that does not require the GPS receiver to accumulate phase effectively means the solution algorithm must only use the positive fractional component of the phase observable and have no need of the receiver's ability to track the phase observable.

Another significant issue when not using a battery is that effectively observations are distributed randomly only during daylight hours. This differs significantly from the general method of obtaining observations. Observations

CHAPTER 1. INTRODUCTION

are usually gathered in consecutive blocks where one day is a fairly typical block. During this period observations are gathered at a consistent rate. As seen in Chapter 4, due to the repeating nature of the satellite constellation, observations gathered this way can produce solutions that are more accurate and precise than randomly obtaining observations only during daylight hours. To the best of our knowledge no one has investigated the solutions obtained when randomly obtaining observations solely during daylight hours. This also required investigation and we find that sidereal filtering at least overcomes some of the shortcomings of the degradation that is introduced by randomly taking observations only during daylight hours.

Objectives

1. Design and implement an entire GPS land deformation monitoring system
2. Design unique GPS and repeater nodes
 - Small form factor (size of a brick)
 - Light weighing a few grams
 - Low cost (about \$100 USD)
 - Batteryless
 - Low power
 - Use singleband consumer grade GPS receivers
 - Use cheap uncalibrated antennas
 - Solar powered
 - Able to produce sufficient observations for solutions in very low light levels
3. Design a unique solution algorithm
 - Distributed
 - Accurate and precise phase based static solutions
 - Singleband, baseline less than 10 km

CHAPTER 1. INTRODUCTION

- Able to function with lossy communication links between the nodes and the base station
 - Low bandwidth communication between the nodes and the base station
 - Able to function with poor quality observations in high multipath environments
 - Able to function with unreliable and intermittent power supplies
 - Does not require the GPS receiver to accumulate phase (use only the positive fractional component of the phase observable)
 - Implement multipath mitigation
4. Design the base station
 - Permanently powered
 - 3G Internet connection
 5. Design WSN communication protocol for the solar powered network
 - Energy aware protocol
 6. Implement the design
 7. Investigate the form that the solutions take
 8. Compare the solution algorithm with other designs
 9. Set up testbeds and analyze the results

Summary

The aim of this thesis is to design and implement a permanently deployed land deformation monitoring system using GPS where the baseline is less than 10 km. The node itself should cost a fraction of the price compared to a node in a current system using survey grade GPS receivers. The cost of such a node is anticipated to be approximately \$100 USD and be no larger than the size of a brick weighing a few grams with no external connections necessary. The node should not have a battery and should be entirely powered by solar

CHAPTER 1. INTRODUCTION

panels which must be part of the node itself. The solutions produced should have an accuracy far greater than that of autonomous code based solutions and be of the type more typically found with survey grade receivers. The solution algorithm should be able to deal with poor quality observations that are intermittent and does not require the GPS receiver to accumulate phase as the power to the GPS receiver itself is intermittent. The solution algorithm should be a distributed one and use minimal bandwidth between the nodes and the base station so as to scale better for high density networks.

The aim is to design a system and show that it works meeting our objectives. The aim of this thesis is not to determine quantitatively the accuracy of such a system, nor to calculate the total cost of such a system including maintenance and so forth. The aim is to stop when the solutions are obtained, not to process the solutions further using statistical analysis or any other method. Many other things such as accuracy, usability, visualization, further solution processing, total cost, life expectancy of node and so forth are left for future work; we design, implement, test, and analyze.

1.1 Related work

[50] propose using singleband receivers for monitoring volcanoes where the baseline is less than 10 km. The GPS receiver node they estimate will cost \$1500 AUD and uses 0.7 W. They cite that high quality antennas are needed for good repeatability of solutions and an extra ground plane should be considered. It appears that they use a battery pack and solar panels for recharging their batteries to power their nodes. They also previously made a design that they implemented in Indonesia in 1999 that they considered a low-cost system costing less than US\$3000 per node including monument. They discuss the technical and logical challenges here [47] in their testbed. They used two solar panels capable of supplying 75 W each; a GPS antenna mounted 1.5 m off the ground so as not to affect baseline determination as suggested by [37]. The antennas they use had their phase variation offset measured as this was believed to be important for accuracy. The antennas themselves cost \$300 making them 10% cost of the entire node. They noted because the antennas were stationary that they could use sidereal filtering to improve solutions.

[23] designed and implemented a singleband wireless GPS receiver node (Ublox based) that used approximately 138 mW. They claim with a 14,000 mAh battery (presumably a 5 V battery but do not say) that this should last 62 days with the GPS receiver on for 6 hours a day; that's about 17 to 29 AA batteries. Their prototype appears to be entirely powered by battery and no energy harvesting techniques are used to recharge the batteries. They claim that the system they created can obtain sub-centimeter accuracy but do not give a maximum baseline for this, although presumably a maximum baseline for such an estimate would be less than 10 km. The system they designed was a multi-hop network.

[42] also investigate the use of singleband GPS receivers (Ublox) for land deformation monitoring at a rock glacier in Switzerland obtaining 4 months of data; 17 cm of movement was detected over 11 days. The cost per node was €1300. They used a sealed lead acid battery weighing 8.6 kg and a 24 W solar panel measuring 0.5 x 0.5 meters. They obtained both daily static solutions and kinematic ones and used the processing software Bernese to obtain solutions.

[25] mention there is high interests in many fields regarding the cost, size and power benefits of using singleband receivers. Primarily their work is concern-

CHAPTER 1. INTRODUCTION

ing with the solution differences obtained between a singleband GPS receiver (Ublox) and a survey grade receiver where two antennas are used, one being a high quality geodetic antenna and another one being an uncalibrated low cost antenna. They use virtual reference stations to mitigate the problem of limited baseline length caused by using singleband GPS receivers and show that with a high quality geodetic antenna 10 minutes is necessary to fix the phase ambiguities and have centimeter level accuracy at least 80% of the time where the virtual reference station baseline is less than 10 km. This compares to their survey grade receiver able to fix ambiguities within 5 minutes and obtain centimeter level accuracy at least 95% of the time with baselines up to 40 km. They show that a high quality geodetic antenna improves solution accuracy particularly in the up direction. They conclude a singleband GPS receiver accuracy is comparable to dual band survey grade receiver given at least 10 minutes of acquisition time, and a baseline less than 1 km with a comparable antenna. In addition, they conclude that by using virtual reference stations, the baseline length between the node and each of the real reference stations is “not much of a problem” if the virtual reference station can be created close enough to the node.

TrackTag [22] is a GPS tracking unit capable of operating for two years with 64,000 observations which uses a unique approach by only sampling and saving 30 ms of the baseband signal for later processing and weighs 30 g. This eliminates the need for the GPS receiver itself to acquire the satellites etc. thus requiring an exceedingly small amount of energy per solution. It is not designed for high precision work and runs off batteries.

Small GPS receiver devices that run on solar power exist. Generally these devices are to save the battery life of cell phone GPS receivers. To the best of our knowledge none of these use phase data or are used in precise work such as land deformation monitoring. Such devices use the solar panels to recharge a battery and the device itself, for example [3]. All of these such devices as far as we are aware of use nothing but autonomous code based solutions and are therefore not suitable for precision work. One different device in this category is a small solar powered tracking device by Retriever [10] but is only in the pre-order phase and once again only does autonomous code based solutions.

Multipath mitigation at both a hardware and software level have been addressed. Typically an antenna with a choke ring is used to mitigate multipath

CHAPTER 1. INTRODUCTION

effects [29] and is the frontline in multipath defence. [30] proposes using dual polarisation antennas as a possible way of mitigating multipath. At a solution processing level, [49] use wavelet analysis to mitigate multipath error. [62] decompose double differences into single differences, aligning each single differences on different days by using the exact repeat time of each satellite as determined by the broadcast navigation data, correct the single differences and then transform them back into double differences. This technique circumvents the problem that standard sidereal filtering has of assuming that the repeat periods of the satellites used for a double difference are the same.

Ambiguity resolution is fairly mature with hundreds of papers written on the subject. [54] analyses the gain one can expect of precision going from a float solution to a fixed solution and shows that longer observation time produces less benefit. Most papers appear to concern computational speed. Currently the *Least-squares AMBiguity Decorrelation Adjustment* (LAMBDA) method first introduced by [53] appears to be somewhat of a de facto standard and is very fast. When using multiple epochs for ambiguity resolution to the best of our knowledge only the *Ambiguity Function Method* (AFM) and conceivable methods similar to it are invariant to cycle slips. AFM even with modification is extremely computationally intensive. However, despite this shortcoming due to its simplicity and its invariance to cycle slips it is commonly found in literature even today in such papers as [36] and [32]. Of the other methods, cycle slips must be corrected or ambiguities reinitialised. If cycle slips are missed or improperly corrected, the solution can be considerably degraded. The first line of defence from cycle slips is the hardware itself, both the antenna and the receiver. A good antenna ensures a strong signal, while a receiver continuously tracking the phase, doing its best to avoid cycle slips, reports to the user in the event of a possible cycle slip; both of which mitigate the number of cycle slips that occur and the number of cycle slips missed. Other methods during solution processing are also implemented such as instantaneous dual band methods [39], using time differencing techniques, and *Inertial Measurement Unit* (IMU) devices are all used [33].

Unaddressed issues

To the best of our knowledge there is no literature or analysis concerning intermittent random observations and receivers that cannot be powered con-

CHAPTER 1. INTRODUCTION

tinuously; this area we believe needs addressing. The reason why there is no work done in this field is probably due that to the best of our knowledge no one has removed the battery from a GPS node. All literature at least implicitly implies that a continuous stream of data is needed. Many such things as frequency analysis of residuals become difficult or meaningless when observations are intermittent and random. Sidereal filtering using stacking methods become unclear as to how to implement. Such assumptions that the GPS receiver is continuously powered, meaning cycle slips are at least mitigated, are not valid any more. If GPS nodes are powered using energy harvesting techniques, no longer can the assumption be made that the user has the choice when an observation is to be made; this may have an effect on the form the solutions take and may allow new methods to be devised that mitigates the degradation due to the more relaxed power supply assumptions. If the requirement on hardware power is allowed to be relaxed then new analysis is needed.

1.2 Contributions

To the best of our knowledge no one has created a GPS land deformation monitoring system such as ours using randomly intermittent poor quality singleband observations for precise GPS land deformation monitoring where the receiver was only powered intermittently, or looked into the form these solutions take. Neither could we find anyone who had designed a high precision GPS land deformation monitoring system using small self contained batteryless *Wireless Sensor Network* (WSN) nodes. The novel approach of treating the algorithm as a distributed algorithm over the nodes and the main processing computer allowed us to reduce data channel capacity requirements between the nodes and the main processing computer. The cheap small light nature of the system and the reduced data channel capacity requirements allow higher densities of wireless sensor networks to be deployed than would be possible with current solutions. Also, with the solar energy harvesting and lack of batteries on the wireless sensor nodes means that there is no need for costly battery replacement in the field. A method for solving solutions based only on the positive fractional component of phase observations was created in an attempt to mitigate the problems of using intermittent observations while at the same time reducing data channel capacity requirements. With techniques such as sidereal filtering to mitigate multipath errors, a highly precise and accurate GPS land deformation monitoring system was designed. Finally, an implementation of the system was created and deployed and the results compared and analyzed to that of other systems. The contributions of this thesis are itemized as follows.

- The use of random intermittent poor quality singleband observations without the need for receiver phase accumulation for precise GPS land deformation monitoring
- An analysis of solutions based on intermittent observations obtained solely during the day
- A distributed algorithm with reduced data channel capacity requirements
- An increase in the density of wireless sensor nodes due to lower cost, size, and data channel capacity requirements
- Energy harvesting nodes without batteries

CHAPTER 1. INTRODUCTION

- Methods for the use of only positive fractional component of phase observations for accurate solutions
- An implementation of the unique system with results compared to that of other systems
- Design of a circadian based energy distribution algorithm to allow GPS receiver activity at night (in appendix C)
- Network phasing algorithm (in appendix D)

1.3 Thesis layout

The thesis is organized as follows:

Chapter 1 “Introduction”: Introduces the thesis, related work, contributions, and layout.

Chapter 2 “The Code Float Fix Sidereal algorithm”: Describes the distributed solution processing algorithm.

Chapter 3 “Multipath and sidereal filtering”: Examines the form that the solutions take both before and after sidereal filtering has been implemented.

Chapter 4 “Algorithm comparison”: Compares the solutions obtained using an implementation of our algorithm with solutions obtained from other solution processing software.

Chapter 5 “Hardware”: Details the hardware and the design of it.

Chapter 6 “Testbed results and analysis”: Contains the results and analysis of two significant testbeds of our work.

Nomenclature: Contains the nomenclature used in this thesis.

Appendix A “GPS basics”: Intended for background reading and for readers who desire a deeper understanding into some of the basics of GPS signals.

Appendix B “Differencing the phase observable”: Explains more specific aspects of solution processing as it pertains to this thesis as well as notation used in this thesis; it is recommended that this be the first thing that is read even if the reader just skims through the section. This appendix is essential for an understanding of this thesis.

Appendix C “Node functionality at night”: Details experiments used to obtain observations at night; it is included as results of this was used in the hardware design chapter.

Appendix D “Network”: Briefly describes the structure and the algorithms used to create the communication network.

CHAPTER 1. INTRODUCTION

Chapter 2

The Code Float Fix Sidereal algorithm

The *Code Float Fix Sidereal* (CFFS) algorithm is a static relative positioning solution algorithm specifically for our scenario. It is a distributed algorithm spread over multiple hardware devices and is comprised of many smaller algorithms itself. This chapter is divided into three parts. First the hardware's influence on the algorithm design, second the general idea of solving solutions, and finally the algorithm itself. For notation and background reading refer to Appendix B. Appendix B introduces the concepts of wrapped and unwrapped phase observables and double differences, as well as solution planes.

2.1 Hardware influence on the algorithm design

Here we mention the hardware constraints that have directly influenced the design of the algorithm. While the algorithm does not require such hardware the algorithm should work with such hardware.

The node hardware is envisaged as being solar powered, wireless, batteryless, small, low power, cheap, accurate, does not move fast, and does not require human intervention.

Accurate For the hardware to produce accurate position solutions, the phase observable is the obvious choice due to the carrier’s much shorter wavelength compared to the chip length of the code observable.

Slow node movement speed The node we assume does not move fast hence static solutions where observations over an extended period of time are used for just one position solution makes sense. Also, due to its slow motion it makes sense to implement some form of sidereal filtering.

Wireless Being wireless there has to be some method for the data to somehow get from the hardware to a remote user; this is generally done using a *Radio Frequency* (RF) link. The RF link has a certain bandwidth which is a limited resource and has to be shared by all nodes. In addition to being a limited resource if the required data can be reduced then this produces lower epoch data loss thus mitigating the effect of a lossy channel.

Therefore, the algorithm should attempt to minimize the RF link network bandwidth requirement of each node so as to increase the number of possible nodes in the network, to reduce epoch data loss, increase transmission range, and reduce power. Therefore the node should preprocess the raw observables to reduce the data size and the algorithm should use this data rather than the raw observables. As described in Appendix B, the wrapped phase observable is the positive fractional component of the phase observable and is encoded in less bits of information compared to the phase observable itself. Therefore, it makes sense for the node to use the wrapped phase observable as this require less bandwidth than the normal phase observable.

Small By small we mean something that can be held in one hand. As the hardware is to only be powered by solar energy, the size directly influences the amount of power the hardware can use.

Low power In our case low power means taking steps to reduce the power requirements of the hardware. Being low power, means designing an algorithm that uses a singleband receiver, as each additional band uses more power. This means an algorithm that can deal with only singleband observations is needed.

Low power means the GPS receiver may not have sufficient power to be on continuously and instead needs to be duty cycled to reduce its average power to be less than what can be produced by the solar panels. This prevents the GPS receiver from continuously tracking the phase observable and the algorithm may need to deal with a considerable number of cycle slips and relatively short periods of time where the receiver can track the satellite resulting in an increase in the number of ambiguities. As described in Appendix B, the wrapped double difference is the positive fractional component of the double differenced phase observable. For this purpose using the wrapped double difference makes sense as many cycle slips will have to be dealt with anyway.

2.2 General idea of solving

The main section of the CFFS algorithm is divided into four stages, the code, float, fix and the sidereal stages. Each stage is designed to produce a position solution more accurate than the last. The float, fix and the sidereal stages require the previous stage position solution as their input. All four stages produce solutions, these solutions are called the code solution, float solution, the fix solution, and the sidereal solution.

The code stage solution of the algorithm is obtained by subtracting the course autonomous position solutions to produce an initial estimate of the final solution.

In the float stage, a real number N is given to each unique pair of satellites. *Least squares* (LS) solves for all N s, the position solution, and the covariance information of both. For this the double differences are unwrapped using the using an approximate position estimate. If the code solution estimate is reasonably accurate (less than say 20 m) the wrapped double differences can be unwrapped fairly easily as the wrapped double differences when corrected using the approximate position estimate don't change too rapidly even when the GPS receivers are separated by less than a few kilometers. The float stage is an iterative one.

The fix stage then takes the real valued N s and their covariance and performs *Integer Least Squares* (ILS) using the LAMBDA algorithm to fix the N s to

integers. Using LS, the fixed N s are used to produce a fixed solution. The double differences are then rounded to the nearest solution plane and another solution obtained. A solution plane as explained in Appendix B are the position solutions given one double difference for a particular integer ambiguity N .

For the sidereal stage, sidereal filtering is then used to improve daily solution precision. Absolute wrapped residual stacking with respect to a biased location is used in addition to fix solution stacking. Arithmetic averaging of both when possible results in daily solution precision improvements due to at least in part the random nature with which observations are obtained.

Due to the random nature of obtaining observations in our scenario, sidereal filtering can be used to improve the precision of daily solutions. From simulations and experiments this improvement seems to at least be partly due to the random and intermittent nature of the observations as well as an inability to obtain high rate observations continuously.

The algorithm does not need code pseudoranges and the integer component of the phase observable to be sent over the radio link, as a result a considerable reduction of bandwidth can be achieved by removing these items before sending these things over the radio link. This results in smaller packets which in turn results in lower epoch losses thus mitigating the effect of a lossy channel.

Navigation data is still required and need not be of high accuracy. Broadcast navigation is suitable and can be obtained from the satellites directly or downloaded from the Internet.

2.3 Algorithm

The CFFS algorithm is a distributed algorithm consisting of two sections the node section and the main section. The main section consists of five main parts, namely a creation and storage part, in addition to a code, float, fix and the sidereal parts.

2.3.1 Required data

The required data for the algorithm are phase and code observations along with time of reception as given by the receiver, and broadcast navigation data. These are standard measurements as can be seen in appendix A, generally GPS receivers must obtain this data as a bare minimum.

2.3.2 Data flow

As one aspect of the algorithm is to not use too much bandwidth, this means some processing of this data has to be done before being sent over the network. Therefore, the algorithm as a whole, from obtaining the raw data to obtaining a final position solution, must be able to be split between multiple hardware devices reducing the data transmitted from one device to another. The following figure shows the distribution of the algorithm between multiple devices and the data input to the algorithm.

Figure 2.1: Algorithm distribution and data in flow

To find a relative positioning solution for two receivers A and B, figure 2.1 shows the distribution of the algorithm and where the data enters the algorithm. The receivers can be regarded as data sources supplying the algorithm with its data. This data contains code, phase, time, and navigation information. The node sections take the raw data, process it, and then send this processed data to the main section. Each node contains different phase, code and time data while navigation data is common to all.

CHAPTER 2. THE CODE FLOAT FIX SIDEREAL ALGORITHM

A more detailed view of the CFFS algorithm is shown in the Figures 2.2 and 2.3 below.

Figure 2.2: Part one of the CFFS algorithm

Figure 2.3: Part two of the CFFS algorithm

2.3.3 Node section

To keep the data flow between the node sections and the main section of the algorithm to a minimum, navigation data and code observations are not passed between the nodes and the main section. What is passed are wrapped phase observations extrapolated to a common second based epoch $\hat{\phi}_B^S(t)$, the satellite number S , the receiver number B and the second that these observation estimates are for t . In addition, occasionally averaged autonomous code based solutions $\bar{\mathbf{X}}_B$ are passed to the main section along with the receiving number B as depicted in Figure 2.4.

Figure 2.4: Data passed from node to main section of the algorithm

Code based solutions are required to be passed only once and hence contribute an insignificant fraction of data that flows between the node and the main sections.

Generally GPS receivers try to take their observations on the second and it appears to be somewhat of a de facto standard for measurements. For example the Ublox LEA-6T takes its observations on the second $\pm 0.5 ms$ (as measured by us). While *Receiver Independent Exchange Format* (RINEX) observation files are typically aligned to the second as well. Because of this we design our algorithm to work on observations that are approximately aligned to the second. The following algorithm describes the node section of the CFFS algorithm where $\{\}$ represents the sawtooth function.

Algorithm 2.1 Node algorithm

1. Wait for new data from the receiver to arrive
 2. Use an algorithm such as A.1 to obtain an autonomous position solution $\mathbf{X}_B$ and clock bias $\Delta\hat{T}_B$ for a given reception time T_B as given by the receiver.
 3. Perform a running average of the autonomous position solution $\mathbf{X}_B$ and call it $\bar{\mathbf{X}}_B$
 4. Calculate an estimate for reception time $t_B = T_B - \Delta\hat{T}_B$
 5. Calculate satellite range at transmission time to reception time $\rho_B^S(t_B) = \|\mathbf{S}(t_B - \rho_B^S(t_B)/c) - \mathbf{X}_B\|$ using iteration
 6. Calculate satellite transmission time $t_B^{tx} = t_B - \rho_B^S(t_B)/c$
 7. Calculate satellite's radial velocity $\left. \frac{d\rho_B^S}{dt} \right|_{t_B} \approx \frac{\mathbf{S}(t_B^{tx}) - \mathbf{X}_B}{\rho_B^S(t_B)} \cdot \left. \frac{d\mathbf{S}}{dt} \right|_{t_B^{tx}}$ (See formula B.6 in appendix B. Alternatively use Doppler)
 8. Let t equal the second closest to t_B
 9. If $|t - t_B|$ is too big then fail this second and goto step 1
 10. If signal strength of satellite S is too weak, unhealthy or phase is not being tracked goto 1
 11. Calculate the extrapolated wrapped phase observable for the common epoch t $C_f \hat{\phi}_B^S(t) = \left\{ C_f \Phi_B^S(t_B) + \frac{C_f}{\lambda} \left. \frac{d\rho_B^S}{dt} \right|_{t_B} (t - t_B) \right\}$
 12. Send $\hat{\phi}_B^S(t)$, t , B , and S regularly to the main section of the algorithm and occasionally send $\bar{\mathbf{X}}_B$ to main section of algorithm.
 13. Goto step 1
-

2.3.4 Main section

The main section of the algorithm takes the data that is given by the node ($\hat{\phi}_B^S(t)$, t , B , S , and $\bar{\mathbf{X}}_B$) sections and calculates a relative position solution.

2.3.4.1 Wrapped double difference creation algorithm

The following algorithm is used to create the wrapped double difference.

Algorithm 2.2 Wrapped double difference creation algorithm

1. Extrapolate wrapped phase observations $\hat{\phi}_B^{S2}(t)$, $\hat{\phi}_A^{S2}(t)$, $\hat{\phi}_B^{S1}(t)$ and $\hat{\phi}_A^{S1}(t)$ to a common epoch if needed
 2. Form all possible wrapped double difference $\hat{\phi}_{BA}^{S2S1}(t)$ from wrapped phase observations
 3. Calculate delta norm $\Delta \mathbf{n}_{S2S1}$ from navigation data.
 4. Store wrapped double difference, delta norm, and common epoch. $\hat{\phi}_{BA}^{S2S1}(t)$, $\Delta \mathbf{n}_{S2S1}$, t .
-

2.3.4.2 Code

The code section calculates the first relative position solution and is simply the difference of the two autonomous position solutions calculated in the previous node section.

Algorithm 2.3 Code algorithm

1. let $\mathbf{P}_{IBA} = \bar{\mathbf{X}}_B - \bar{\mathbf{X}}_A$
-

2.3.4.3 Float

Here we briefly describe the float algorithm followed by the float algorithm itself. The aim of the float algorithm is to obtain a rough solution with an accuracy of a few centimeters along with the ambiguity N estimates and their covariance information for the fix stage.

Equation B.14 is our model of a double difference. We have the wrapped double difference as given by the wrapped double difference creation algorithm, which we can unwrap to re-create double difference estimates (see B.3.3 on page 213). This unwrapping tries to produce an integer N that is fixed for all observations to these two satellites for all times. Therefore we assume N_{BA}^{S2S1} is fixed for all

observations to these two satellites for all epochs t_k . Therefore we can write equation B.14 as follows for our unwrapped double differences given a position estimate $\hat{\mathbf{P}}_{BA}$ where $\lambda' = \lambda/C_f$.

$$\lambda \hat{\Phi}_{BA}^{S2S1}(t_k, \hat{\mathbf{P}}_{BA}) = \Delta \mathbf{n}_{S2S1}(t_k) \cdot \hat{\mathbf{P}}_{BA} + \lambda' N_{BA}^{S2S1} + (M_{BA}^{S2S1} + e_{BA}^{S2S1}) \quad (2.1)$$

In matrix form with many observations to many satellites we obtain the following where $\mathbf{r}_{BA}$ is the residual vector.

$$\begin{bmatrix} \lambda \hat{\Phi}_{BA}^{S2S1}(t_0, \hat{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \hat{\Phi}_{BA}^{SvSu}(t_k, \hat{\mathbf{P}}_{BA}) \end{bmatrix} = \begin{bmatrix} (\Delta \mathbf{n}_{S2S1}(t_0))^T & \lambda' & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ (\Delta \mathbf{n}_{SvSu}(t_k))^T & 0 & \cdots & \lambda' \end{bmatrix} \begin{bmatrix} \mathbf{P} \\ N_{BA}^{S2S1} \\ \vdots \\ N_{BA}^{SvSu} \end{bmatrix} + \mathbf{r}_{BA} \quad (2.2)$$

Linear set of equations for float algorithm in matrix form

This set of equations has just one unknown integer N for each pair of satellites along with three unknown position variables. The number of equations can be much greater than the number of unknowns hence this form is suitable for being solved using LS which minimises the mean square error of the residual. Being a linear method the integer N solutions end up being “floating” numbers hence the algorithm name. The unwrapping algorithm requires all possible double differences while the set of double differences required for the LS is a linearly independent set. The independent set for the LS is created using a reference satellite, where one satellite for each epoch is chosen as the reference satellite and all double differences within that epoch must contain this reference satellite. The reference satellite is selected as the satellite with the highest elevation on an epoch by epoch basis. The satellite with the highest elevation is assumed to have the least noise and hence brings to the LS solution the least amount of error. This removes linear dependence amongst double differences but however does not remove stochastic dependence amongst the double differences in each epoch. We ignore the stochastic dependence.

Equation 2.2 has the form $\mathbf{y} = [\mathbf{N} \mathbf{A}] \begin{bmatrix} \mathbf{P} \\ \mathbf{a} \end{bmatrix} + \mathbf{r}$. A LS solution would result in estimates $\begin{bmatrix} \hat{\mathbf{P}} \\ \hat{\mathbf{a}} \end{bmatrix}$ with covariance $\begin{bmatrix} \mathbf{Q}_{\hat{\mathbf{P}}} & \mathbf{Q}_{\hat{\mathbf{P}}\hat{\mathbf{a}}} \\ \mathbf{Q}_{\hat{\mathbf{a}}\hat{\mathbf{P}}} & \mathbf{Q}_{\hat{\mathbf{a}}} \end{bmatrix}$

Solving for the solution using LS depends on the initial position estimate of $\hat{\mathbf{P}}_{BA}$ for unwrapping. If we assume that this procedure of first unwrapping the data given an approximate solution then solving for the solution produces a more accurate solution, then we can unwrap the data again given our better approximation of the solution. We can repeat this process until our solution does not change much. Therefore, an iterative method is used feeding the output solution back into the initial position estimate to produce a new solution. This iteration is continued until convergence or divergence. The solution is assumed to be divergent when the iteration count becomes too large or clearly distinct solutions are repeating. Convergence is assumed when the input solution estimate and the output solution are very similar to one another. To help avoid convergence to incorrect solutions, the float algorithm is run twice. The first time it is iterated until convergence or divergence with the code solution as the initial solution, the second time it is iterated only once and yesterday's best solution is used as the initial solution for the float algorithm. The solution with the least mean square error of the wrapped residuals is chosen as the correct solution.

Algorithm 2.4 Float algorithm

1. If not enough wrapped double differences then fail
 2. If initial solution $\mathbf{P}_{IBA}$ does not exist then fail
 3. Let $\hat{\mathbf{P}}_{BA} = \mathbf{P}_{IBA}$
 4. Use unwrapping algorithm B.1 to unwrap wrapped all double differences given solution estimate $\hat{\mathbf{P}}_{BA}$ to obtain double difference estimates $\hat{\Phi}_{BA}^{SvSu}(t_k, \hat{\mathbf{P}}_{BA})$ for all pairs of satellites u, v
 5. Form the independent set of double differences such that for each epoch the satellite with the highest elevation is selected as the reference satellite
 6. Form
$$\begin{bmatrix} \lambda \hat{\Phi}_{BA}^{S2S1}(t_0, \hat{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \hat{\Phi}_{BA}^{SvSu}(t_k, \hat{\mathbf{P}}_{BA}) \end{bmatrix} = \begin{bmatrix} (\Delta \mathbf{n}_{S2S1}(t_0))^T & \lambda' & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ (\Delta \mathbf{n}_{SvSu}(t_k))^T & 0 & \cdots & \lambda' \end{bmatrix} \begin{bmatrix} \mathbf{P} \\ N_{BA}^{S2S1} \\ \vdots \\ N_{BA}^{SvSu} \end{bmatrix} +$$

 $\mathbf{r}_{BA}$ using only the independent set of double differences formed in the previous step
 7. Use LS to find an estimate for the unknowns along with the covariance estimate. Define the resulting LS solution estimate as $\begin{bmatrix} \hat{\mathbf{P}}_{BA} \\ \hat{\mathbf{a}} \end{bmatrix}$ and covariance matrix as $\begin{bmatrix} \mathbf{Q}_{\hat{P}} & \mathbf{Q}_{\hat{P}\hat{a}} \\ \mathbf{Q}_{\hat{a}\hat{P}} & \mathbf{Q}_{\hat{a}} \end{bmatrix}$
 8. If solution has not converged or diverged then goto 4
 9. Using yesterday's best solution estimate as the initial solution for the float algorithm perform the float algorithm again with just one iteration. If the mean square error of the wrapped residuals of this solution is less than that of the iterative float solution use this solution instead.
 - 10.
-

Float example We took four hours of observational data for two receivers separated by $2.13m$. This produced 39,766 independent double differences. We took $\mathbf{P}_{IBA} = [10, 10, 10]$ as the initial code solution position vector. This was a solution chosen to be close to the real solution. The baseline of the solution was calculated and plotted in Figure 2.5.

Figure 2.5: Float algorithm convergence example

The final solution baseline was calculated as 2.13 m matching the real baseline. The solutions appeared to converge by about the ninth iteration.

2.3.4.4 Fix

Here we describe the fix algorithm. The aim of the fix algorithm is to improve the accuracy of the float solution using the the position estimate, the ambiguity estimate, and the covariance of the ambiguity estimate from the float section.

The fix solution fixes the integer N s, also referred to as fixing the integer ambiguities, or just fixing the ambiguities. The float solution method is a linear one and hence the N s when solved by the LS method are not integers as they should be. If we can fix the N s to the correct integers then we reduce the number of unknowns from being the position vector plus all the N s to just the position vector. This has the effect of producing more accurate solutions in theory if we set the N s to the correct values.

For fixing the ambiguities, the covariance as well as the ambiguity estimates themselves obtained from the float solution ($\hat{\mathbf{a}}$ and $\mathbf{Q}_{\hat{\mathbf{a}}}$) are fixed to integers

via ILS using the LAMBDA algorithm as described by [53] and implemented with software written by [58]. The fixed ambiguities $\check{\mathbf{a}}$ are then used to correct the unwrapped double differences and hence remove the integer ambiguities from equation 2.2 to produce the following.

$$\begin{aligned}
 \begin{bmatrix} \lambda \check{\Phi}_{BA}^{S2S1}(t_0, \hat{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \check{\Phi}_{BA}^{SvSu}(t_k, \hat{\mathbf{P}}_{BA}) \end{bmatrix} &= \begin{bmatrix} \lambda \hat{\Phi}_{BA}^{S2S1}(t_0, \hat{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \hat{\Phi}_{BA}^{SvSu}(t_k, \hat{\mathbf{P}}_{BA}) \end{bmatrix} - \lambda' \check{\mathbf{a}} \\
 \begin{bmatrix} \lambda \check{\Phi}_{BA}^{S2S1}(t_0, \hat{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \check{\Phi}_{BA}^{SvSu}(t_k, \hat{\mathbf{P}}_{BA}) \end{bmatrix} &= \begin{bmatrix} (\Delta \mathbf{n}_{S2S1}(t_0))^T \\ \vdots \\ (\Delta \mathbf{n}_{SvSu}(t_k))^T \end{bmatrix} \mathbf{P} + \mathbf{r}_{BA} \\
 \begin{bmatrix} \lambda \check{\Phi}_{BA}^{S2S1}(t_0, \hat{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \check{\Phi}_{BA}^{SvSu}(t_k, \hat{\mathbf{P}}_{BA}) \end{bmatrix} &= \mathbf{NP} + \mathbf{r}_{BA} \tag{2.3}
 \end{aligned}$$

Linear set of equations for fix algorithm in matrix form used to obtain $\dot{\mathbf{P}}_{BA}$

LS can then be used on equation 2.3 to obtain a fix solution estimate $\dot{\mathbf{P}}_{BA}$. The solution $\dot{\mathbf{P}}_{BA}$ is then used to obtain a another fix solution by rounding to the nearest solution plane and obtaining another LS estimate. In this step an integer ambiguity is given to each double difference and rounded to the nearest solution plane given the solution $\dot{\mathbf{P}}_{BA}$ as follows where $f()$ is the sawtooth difference function (B.3.3).

$$\begin{aligned}
 \lambda \hat{\Phi}_{BA}^{S2S1}(t_k, \dot{\mathbf{P}}_{BA}) &= \Delta \mathbf{n}_{S2S1}(t_k) \cdot \dot{\mathbf{P}}_{BA} \\
 &+ \lambda' f \left(C_f \hat{\Phi}_{BA}^{S2S1}(t_k) - \frac{C_f}{\lambda} \Delta \mathbf{n}_{S2S1}(t_k) \cdot \dot{\mathbf{P}}_{BA} \right) \tag{2.4}
 \end{aligned}$$

From which LS can be used to obtain a solution $\check{\mathbf{P}}_{BA}$ using equation (2.5)

$$\begin{bmatrix} \lambda \dot{\Phi}_{BA}^{S2S1}(t_0, \dot{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \dot{\Phi}_{BA}^{SvSu}(t_k, \dot{\mathbf{P}}_{BA}) \end{bmatrix} = \mathbf{NP} + \mathbf{r}_{BA} \quad (2.5)$$

Linear set of equations for fix algorithm in matrix form used to obtain $\check{\mathbf{P}}_{BA}$

This solution is also a fixed solution but in doing this rounding, any cycle slips introduced whilst unwrapping are removed as this rounding is invariant to cycle slips.

One would expect that the mean square error of the residuals given the actual true solution would result in the lowest value for all given solutions. Because of this we include one final step in the fix algorithm whereby the mean square error of the residuals for today's best estimate is compared with yesterday's best estimate. If the mean square error of the residuals using yesterday's best estimate results in a lower value than using today's, this means it is likely that for whatever reason that yesterday's best estimate is a better estimate for today than the current estimate we have for today; in this case we round the double differences to the nearest solution planes using yesterday's best estimate and then use least squares to find a new estimate for today.

The fix algorithm is described as follows.

Algorithm 2.5 Fix algorithm

1. If $\hat{\mathbf{P}}_{BA}$, $\mathbf{Q}_{\hat{\mathbf{a}}}$ or $\hat{\mathbf{a}}$ do not exist then fail
 2. Use ILS to find integer estimates $\check{\mathbf{a}}$ of $\hat{\mathbf{a}}$ given ambiguity covariance $\mathbf{Q}_{\hat{\mathbf{a}}}$
 3. Correct double differences based on integer ambiguity estimates $\check{\mathbf{a}}$ as

$$\begin{bmatrix} \lambda \check{\Phi}_{BA}^{S2S1}(t_0, \hat{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \check{\Phi}_{BA}^{SvSu}(t_k, \hat{\mathbf{P}}_{BA}) \end{bmatrix} = \begin{bmatrix} \lambda \hat{\Phi}_{BA}^{S2S1}(t_0, \hat{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \hat{\Phi}_{BA}^{SvSu}(t_k, \hat{\mathbf{P}}_{BA}) \end{bmatrix} - \lambda' \check{\mathbf{a}}$$
 4. Use LS to obtain $\dot{\mathbf{P}}_{BA}$ a solution estimate to

$$\begin{bmatrix} \lambda \check{\Phi}_{BA}^{S2S1}(t_0, \hat{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \check{\Phi}_{BA}^{SvSu}(t_k, \hat{\mathbf{P}}_{BA}) \end{bmatrix} = \mathbf{NP} + \mathbf{r}_{BA}$$
 5. Round double differences to nearest solution planes given solution $\dot{\mathbf{P}}_{BA}$ via $\lambda \check{\Phi}_{BA}^{S2S1}(t_k, \dot{\mathbf{P}}_{BA}) = \Delta \mathbf{n}_{S2S1}(t_k) \cdot \dot{\mathbf{P}}_{BA} + \lambda' f \left(C_f \hat{\phi}_{BA}^{S2S1}(t_k) - \frac{C_f}{\lambda} \Delta \mathbf{n}_{S2S1}(t_k) \cdot \dot{\mathbf{P}}_{BA} \right)$
 6. Use LS to obtain $\check{\mathbf{P}}_{BA}$ a solution estimate to

$$\begin{bmatrix} \lambda \check{\Phi}_{BA}^{S2S1}(t_0, \dot{\mathbf{P}}_{BA}) \\ \vdots \\ \lambda \check{\Phi}_{BA}^{SvSu}(t_k, \dot{\mathbf{P}}_{BA}) \end{bmatrix} = \mathbf{NP} + \mathbf{r}_{BA}$$
 7. Compare the mean square error of the wrapped residuals using yesterday's best solution estimate $\mathbf{P}_{yesterday}$ with the mean square error of the wrapped residuals using $\check{\mathbf{P}}_{BA}$. If the mean square error is less using yesterday's solution then let $\dot{\mathbf{P}}_{BA} = \mathbf{P}_{yesterday}$ and round the double differences to the nearest solution plane as described in step 5 to obtain a new estimate for $\check{\mathbf{P}}_{BA}$ as described in step 6.
-

2.3.4.5 Sidereal

The daily sidereal filtered equation derived in Chapter 3 is given as follows, where the subscript n represents day n , the bar notation is average over days of matching double differences, $\bar{\mathbf{P}}_{M,n}$ is the average multipath bias error, $\bar{\bar{\mathbf{P}}}_n$ is the average of a priori known solutions and r_n is the double difference residual; all notation for intraday time (time within the same day), satellites and receiver nomenclature have been dropped for clarity.

$$\lambda(r_n - \bar{r}_n) + \Delta \mathbf{n} \cdot \bar{\check{\mathbf{P}}}_n = \Delta \mathbf{n} \cdot \mathbf{P}_n + \Delta \mathbf{n} \cdot \bar{\mathbf{P}}_{M,n} + e_n \quad (2.6)$$

Sidereal filtering equation

Equation (2.6) is derived and explored more fully in Chapter 3 . For the meantime we take it as given that for our scenario it improves daily solution estimate precision. The main idea behind sidereal filtering is that the satellite constellation approximately repeats every 24h -246 seconds also erroneously referred to as the GPS sidereal day. It is erroneous because the truth sidereal day length is approximately 24h -236 seconds on Earth. Double differences that belong to a pair of satellites and a pair of receivers are considered to match if they are separated in time by an integer number of nominal GPS satellite orbital repeat periods and are on different sidereal days. As multipath is a function of the GPS receiver's surroundings and satellite geometry, as long as the GPS receiver's surroundings do not change with respect to the GPS receiver then the multipath error should be near identical from one sidereal day to the next.

The bar notation is the arithmetic mean calculated when possible including the current day n . For r_k at day n this is $\bar{r}_n = \text{mean}(\Omega \cup \{r_n\})$ where Ω contains a set of matching residuals (the residuals of matching double differences); in this particular instance $\{r_n\}$ represents the set containing only r_n and ensures $\bar{r}_n$ is sensibly defined. This means there may be m days to average over but Ω may contain far fewer entries.

For calculating the sidereal filtered solution there is the need to calculate $\bar{\check{\mathbf{P}}}_n$, $\bar{r}_n$, r_n and $(r_n - \bar{r}_n)$. Incorrectly estimating the double difference integer ambiguity introduces an integer ambiguity offset of the double difference residual. Therefore, instead of calculating $(r_n - \bar{r}_n)$ directly we use an integer ambiguity invariant method using absolute wrapped residual stacking with respect to a biased location in addition to expected observation stacking using fix solutions from m consecutive days of the past including day n but excluding days k where $\|\check{\mathbf{P}}_k - \check{\mathbf{P}}_n\|$ is excessive so as to prevent outliers from affecting the sidereal solution. $\bar{\check{\mathbf{P}}}_n$ values are calculated from solutions obtained from the fix section. $\bar{r}_n$ are calculated considering only the absolute value of wrapped residuals, while r_n only the wrapped residual is considered. To calculate $(r_n - \bar{r}_n)$

the sign of $\bar{r}_n$ is chosen such to minimise the absolute magnitude of $(r_n - \bar{r}_n)$. In this way calculation of $(r_n - \bar{r}_n)$ is invariant with respect to the estimated double difference integer ambiguities. Equation (2.6) represents one double difference, many equations are needed to use LS therefore we use matrix notation and describe the implementation of the algorithm as follows where a_p is the approximate satellite constellation repeat period (approximately 24h -246s as already stated).

Algorithm 2.6 Sidereal algorithm

- Initialise $\mathbf{P}_B$ by selecting it as the first ever fix solution
 - Shift wrapped residuals $\mathbf{r}_k$ temporarily by an integer multiple of a_p to align them with today, day n
 - Calculate and save today's biased wrapped residuals $\mathbf{r}_n = f\left(\frac{C_f}{\lambda}(\lambda\Phi_n - \mathbf{N} \cdot \mathbf{P}_B)\right) / C_f$
 - Excluding all days k from the m days where $\|\check{\mathbf{P}}_k - \check{\mathbf{P}}_n\|$ is excessive to prevent outliers affecting the sidereal solution
 - Average all $|\mathbf{r}_k|$ over all matching double differences from the m days and produce $\overline{|\mathbf{r}_n|}$
 - Average all $\mathbf{N} \cdot \check{\mathbf{P}}_k$ over all matching double differences from the m days and produce $\overline{\mathbf{N} \cdot \check{\mathbf{P}}_n}$
 - Let $\mathring{\mathbf{r}}_n = \text{magmin}\left(\mathbf{r}_n - \overline{|\mathbf{r}_n|}, \mathbf{r}_n + \overline{|\mathbf{r}_n|}\right)$
 - Use LS to solve $\left(\lambda\mathring{\mathbf{r}}_n + \overline{\mathbf{N} \cdot \check{\mathbf{P}}_n}\right) = \mathbf{N} \cdot (\mathbf{P}_n + \bar{\mathbf{P}}_{M,n}) + \mathbf{e}_n$ and hence obtain an estimate of $\mathbf{P}_n$. Call this estimate $\mathring{\mathbf{P}}_n$
-

The function $\text{magmin}(x, y)$ selects the value to return of the argument with the least absolute magnitude. While we are aware that each satellite repeat period differs from one satellite to another we chose a_p to be fixed value for simplicity. We chose a_p to be 24h -246s. The effect that this sidereal filtering algorithm makes on a static node can be seen in figure 3.17.

2.4 Independent LS and reference satellite selection

For both the float and the fix stages, sets of linear equations are solved using LS. As each double difference can be created as the addition or subtraction of two other double differences (eg. $\Phi_{BA}^{S3S1} = \Phi_{BA}^{S3S2} + \Phi_{BA}^{S2S1}$) if all double differences were used in the equations, biased solutions could result as each single difference contributes a different amount of noise. Therefore an independent set can be used to avoid this. To decide which double differences stay and which ones are removed we use the standard idea of a reference satellite. A reference satellite is the satellite in each epoch that must appear in each double difference. This produces a star pattern of double differences where each double difference has been taken with respect to this reference satellite. Which reference satellite is used may make a large difference to the position solution; this is because the reference satellite brings its noise into each equation in the set of equations. Therefore selecting the reference satellite to be the one with the lowest noise should result in the most accurate solution. A typical method of selecting the reference satellite is to choose the one with the highest elevation due to the assumption that satellites with high elevation have low noise and low multipath [59].

The selection of the reference satellite can alter the accuracy of the final solution and is critical for precise difference based solutions [59]. Both the fix and float stages of the algorithm use the idea of a reference satellite and use the same set of independent double differences. As an example we used two different methods when choosing the reference satellite and compared the results. The first method was randomly choosing a reference satellite on an epoch basis while the other method was to choose the satellite with the highest elevation. One month of data was obtained and the two methods implemented. The final solutions were then removed of any linear trend. Choosing the satellite with the highest elevation as the reference satellite resulted in a smaller variance in all directions than when randomly choosing a reference satellite. The measured variance improvement over the random method was 5% 34% 16% in the north east and up directions respectively. The Figure 2.6 shows a two dimensional view of the solutions of both of the methods as an example. As these two sets of solutions are not the same this means what reference satellite is chosen can

make a difference to the final solution. In fact choosing the satellite with the highest elevation produces superior solutions to that of randomly choosing a reference satellite in this experiment.

Figure 2.6: Random and highest elevation reference satellite selection method comparison

Due to its very simple implementation, that it is a commonly used method of reference satellite selection and that our experiment over 30 days showed that using the satellite with the highest elevation as the reference satellite outperformed that of using a random satellite, we used this method to create independent sets of wrapped double differences.

Chapter 3

Multipath and sidereal filtering

An epoch based solution is one where a solution is obtained using observations from one epoch at the time. A daily based solution is one where a solution is obtained using all available observations from one day at a time. We primarily focus on daily static based solutions in this thesis.

The sidereal part of the algorithm implements sidereal filtering using a stacking technique of both fix solutions and absolute wrapped residuals stacking with respect to a biased location; this is designed to improve solution accuracy and precision for our particular scenario where observations are intermittent. In this chapter we start by investigating multipath which turns out to be the largest source of error in our double differences. From this, assuming ambiguity resolution is a solved problem, we rewrite our double difference model to include only multipath and additive white Gaussian noise and determine the distribution of our solutions we obtain when intermittently taking observations. We then derive our sidereal filtering formula and show that both in simulations and in real life it does indeed improve our solutions. We also show that due to the random intermittent nature that we obtain observations, we obtain a greater improvement in solution precision than if we were using continuously powered GPS receivers outputting constant high rate observations.

The multipath error of the phase observable is due to the superposition of the carrier wave reaching the receiver via different paths. This has the effect of appearing to delay the carrier wave reaching the receiver. For a single reflected signal in addition to the direct line of sight signal the multipath error in the

phase observable can be modeled as follows where α is a damping factor, λ is the nominal wavelength, and ϕ is the phase difference in cycles between the two signals [60].

$$e_{\phi B}^S = \frac{\lambda}{2\pi} \arctan \frac{\alpha \sin(2\pi\phi)}{1 + \alpha \cos(2\pi\phi)}$$

This produces an absolute maximum error of $|e_{\phi B}^S| = \lambda/4$ when the damping factor is equal to one. Therefore the theoretical maximum double difference error can be one cycle or approximately 19 cm as four such phase observables are used to produce a double difference. Double differencing removes most systematic errors but leaves errors that are related to the geometry of the observed satellites such as multipath and PCV. Figure 3.1 shows double difference residuals for all pairs of satellites obtained using our hardware at the LuShan testbed site in Taiwan for one day.

Figure 3.1: All double difference residuals for one day of observations at the LuShan site for one pair of receivers

PCV appears very similar to multipath except for a dependence on antenna rotation. Because of this we lump PCV together with multipath errors. As can be seen from Figure 3.1 there are deviations of up to about one cycle which is the maximum value we would expect from the multipath error. However, the unwrapping algorithm can fail and introduce cycle slips which may also be a cause of some of the extremely high readings obtained in the previous figure,

Figure 3.1. In processing our data we used a code factor of two so cycle slips in Figure 3.1 would results in an offset of 0.5 of a cycle. We shall see that multipath turns out to be the most major component of the variance in double difference residuals. This also supported by references such as [55] claiming that “multipath is the dominant error source in differential carrier phase and pseudorange measurements” and [45] saying “multipath is often considered the most limiting factor in precise GPS positioning”. The double differences from Figure 3.1 are not the ones used to obtain the solution. Instead, the independent set of double differences using the satellite with the highest elevation as the reference satellite is used; the residuals of these double differences can be seen in Figure 3.2.

Figure 3.2: Independent set of double difference residuals for one day of observations at the LuShan site for one pair of receivers using the highest elevation satellite as the reference satellite

From Figure 3.2 it is clear that the variance is considerably less but as we shall see is still a great deal larger than the average measurement noise.

From the residuals in Figure 3.1 there appears to a large component that varies slowly over a relatively long time of a few hours. In addition to a slow component, a fast component that appears periodic and has a period of typically around 5 to 10 minutes can also be seen and has somewhat of triangular nature. The fast periodic component can be seen more clearly if we

magnify just a small amount of one of the traces from the figure; this can be seen in Figure 3.3.

Figure 3.3: Fast periodic component of residuals

This highly time correlated multipath is called specular multipath and results from reflections of smooth surfaces (presumably the ground) and leads to positioning errors both when solutions are calculated every epoch and also as in our case when one solution is calculated for an entire day [41, 20]. In high multipath environments solution errors can be substantial and be in the order of a few centimeters [20].

3.1 Satellite and multipath periodicity

Due to the GPS satellite orbital period, the satellites repeat their locations approximately every 24 hours -246 seconds with respect to a ground based user, also erroneously referred to as the GPS sidereal period (erroneously, as a sidereal day is not 24 hours -246 seconds but instead 24 hours -236 seconds). While the satellites have an orbit period of slightly less than 12 hours, due to the rotation of the earth the satellites appear approximately in the same place in the sky with respect to an earth based observer approximately every 24 hours rather than 12 hours. The GPS satellites perform manoeuvres occasionally to ensure they approximately maintain fixed ground tracks with respect to the earth. According to literature the earth oblateness causes a westward drift in the ascending node which is compensated for by decreasing the GPS orbital period by approximately four seconds [19]. There are two orbital periods in

a day which causes approximately eight seconds difference from the sidereal day. This value is only approximate and the satellite repeat times have been seen to vary by at least 9 seconds [24]. As the satellite repeat period is only approximate we simply use one fixed repeat time of 24 hours -246 seconds which is 1 second difference from the mean repeat time found by for a set of observations by [24] and matches the mean repeat period given by [34] in a set of data. In our discussion we refer to the GPS satellite repeat time of 24 hours -246 seconds erroneously as the sidereal period and take it as given that it refers to our given nominal satellite repeat period and not the actual sidereal day length, we do this as this filtering process is commonly called sidereal filtering not GPS orbital repeat period filtering.

Multipath is a function of the GPS receiver's surroundings and satellite geometry. Therefore, as long as the GPS receiver's surroundings do not change with respect to the GPS receiver then the multipath error should be similar from one sidereal day to the next. This allows a way of determining whether or not the large errors we see in Figure 3.1 are due to multipath or some other form of error. In the two Figures 3.4 and 3.5 that follow we plot the double difference residuals for one pair of satellites for two distinct sidereal days for fixed GPS receivers obtained from our hardware; one for the whole time the satellite pair were visible in a day and the other one for a small close up portion of them.

Figure 3.4: Double difference residuals for one SV pair over two different days; full view

Figure 3.5: Double difference residuals for one SV pair over two different days; close up view

As can be seen there is an extraordinary close match between days even for the small fast periodic triangular components. Therefore we conclude that most of the errors visible in Figure 3.1 are due to multipath and PCV.

3.2 Remaining errors

Assuming integer ambiguity resolution has been successful, we can rewrite our double difference model as follows more simply where all subscripts, superscripts and indexing have been removed.

$$\lambda\Phi = \Delta\mathbf{n} \cdot \mathbf{P} + M + e \quad (3.1)$$

We have divided the remaining errors into M consisting of errors that are due to geometry (Satellite location, receiver surroundings, and PCV) which we treat as multipath, and miscellaneous errors e . The miscellaneous errors will consist of measurement noise and other errors. As we only use the independent set of double differences where the satellite with the maximum elevation is used as the reference satellite for a solution, we are only concerned with the remaining errors in only these independent double differences not the others. The double

differences we're interested other ones as seen in Figure 3.2. We only use these double differences when estimating the remaining errors.

3.2.1 Measurement noise

To obtain an approximate value of the measurement noise, we took the one second time differenced double difference residuals ($r(t) - r(t-1)$) for 60 pairs of satellites given an accurate position estimate, then measured the average variance. As the satellites appear to move relatively slowly from and earth-bound observer and assuming other errors change relatively slowly, the one second difference will consist mainly of measurement noise. Figure 3.6 below shows the one second time differenced double difference residuals for one of the pairs of satellites used to obtain the estimate.

Figure 3.6: One second time differenced double difference residuals for one pair of satellites using our receivers

As this is a difference of additive white gaussian noise, for a double difference this needs to be divided by the square root of two. From this we obtain the average standard deviation of the double difference due to measurement noise of 0.013 cycles or a variance of $var(\text{noise}) = 0.000169 \text{ cycles}^2$. As measurement error depends on signal strength, an error as low as 0.009 cycles was obtained for one pair of satellites while a maximum of 0.019 cycles was obtained for

another pair of satellites. As a double difference consists of two doublings of error, to obtain an estimate of the measurement error of one of our GPS receivers these values have to be divided by the square root of four. Therefore, on average our GPS receivers should have a 95% confidence that it will correctly measure the phase observable to better than 0.013 cycles of the true value or about 1% a cycle.

3.2.2 Multipath error

The multipath error is highly dependent on location hence no exact value can be obtained for the variance of it. An upper bound as we have already seen is one cycle for a double difference and quarter of a cycle for one receiver itself. In addition, measuring the multipath from actual data can also be tricky, as a sidereal differencing of double difference residuals does not entirely cancel multipath from one day to the next. This is because double differences needs two satellites and as the repeat times of these two satellites will not be identical, strictly speaking there is no exact time when the satellites will be in the exact same locations with respect to one another. Others have realised the same problem and techniques to transform double differences into single differences and then align each satellite with a much more exact repeat time then transform these back into a double difference have been designed [62]. However, we do not use this method and is beyond the scope of this thesis; therefore we make the approximation that the alignment of the satellites are exactly the same from one nominal satellite repeat period to the next. With this approximation using sidereal differencing we can obtain an approximate idea of the magnitude of multipath error and other errors that we cannot correct with our particular method of sidereal filtering.

Figure 3.7 shows the sidereal difference of two consecutive days along with the two days residuals.

Figure 3.7: Sidereal differenced double difference residuals for one SV pair over two distinct days

Performing this sidereal differencing calculation for all pairs of satellites used for the final daily solution for one day at the LuShan testbed site, an average value of the sidereal variance was determined to be 0.0014 while that of day two was 0.0096. The sidereal variance can be written as

$$\text{var}(\text{sidereal difference}) = 2 \times \text{var}(\text{noise}) + 2 \times \text{var}(\text{other})$$

while the variance of day 2 can be written as

$$\text{var}(\text{day 2}) = \text{var}(\text{noise}) + \text{var}(\text{other}) + \text{var}(\text{multipath that we can correct})$$

Which in turn implies

$$\text{var}(\text{multipath that we can correct}) = \text{var}(\text{day 2}) - \frac{\text{var}(\text{sidereal difference})}{2}$$

and

$$\text{var}(\text{other}) = \text{var}(\text{day 2}) - \text{var}(\text{noise}) - \text{var}(\text{multipath that we can correct})$$

Hence

$$\text{var}(\text{multipath that we can correct}) = 0.0089 \text{ cycles}^2 \text{ and } \text{var}(\text{other}) = 0.000531 \text{ cycles}^2.$$

Making the approximation that the alignment of the satellites are exactly the same from one nominal satellite repeat period to the next, we say that the

average multipath in the signal at the LuShan testbed for two receivers used had a standard deviation of 0.094 cycles, while that of other errors had a standard deviation 0.023 cycles.

So the standard deviation of 0.094 cycles for the multipath is a lower bound for this particular instance while 1 cycle is an upper bound. Therefore it appears that measurement noise is at least an order of magnitude less than multipath error hence fairly insignificant. The other errors undoubtedly have some component of multipath but are hard to determine. we combine the other errors with the measurement noise which we call the miscellaneous errors, while that of the multipath errors that we can correct, we call the multipath errors. Hence in this particular circumstance we obtain the following standard deviations for these two errors.

$$\sigma_e = 0.026 \text{ Cycles}$$

$$\sigma_M = 0.094 \text{ Cycles}$$

Qualitative standard deviation values

While of course this is dependent on the GPS receivers surroundings it gives an idea of the magnitude of both multipath errors and ones that are not multipath. The multipath error here is a lower bound so is conservative. Even so, the repeating nature of the errors we have examined here which we lump together into multipath are still many times greater than that of the rest of the transient errors combined. Therefore some form of sidereal filtering seems advantageous to use.

3.3 Interday sidereal filtering

An interday sidereal filtered solution (an interday solution) is a daily solution using all available observations in other days in addition to the all observations available in the day in question itself; the other days may not necessarily be consecutive days.

The satellite repeating nature allows us to treat double differences obtained on different sidereal days as if they were almost the same. When the satellites

location are the same, the delta norm and the multipath M are also the same if both of the receiver locations and their surroundings are unchanged. In this case if the double differences used today to obtain a position solution are also the same as the double differences obtained tomorrow to obtain a solution, then the two solutions should be very similar.

We define subscripts k and n to denote the sidereal day. For simplicity we do not give a notation that represents time within a day. For two double difference to match one another we define that they must be on different sidereal days separated by no more than a second with respect to an integer multiple of a_p , and the double differences belong to the same pair of satellites and receivers, where a_p is nominal GPS satellite orbital repeat period. When double differences match one another we make the approximation that the satellite geometry is exactly the same for both double differences. We define $\mathbf{N}$ as containing every possible delta norm for the day, while $\mathbf{N}_k$ contains only the delta norms used on day k , similar notation is used for multipath, $\mathbf{M}$ for every possible multipath during a day and $\mathbf{M}_k$ for multipath used on day k .

We can factorize multipath both for all double differences in the day and for just the ones sampled on day k ; namely $\mathbf{M} = \mathbf{N}\mathbf{P}_M + \mathbf{M}_{MNL}$, and $\mathbf{M}_k = \mathbf{N}_k\mathbf{P}_{M,k} + \mathbf{M}_{MNL,k}$. This allows us to rewrite equation 3.1 as follows which means our solution is biased due to multipath.

$$\lambda\Phi_k = \Delta\mathbf{n}_k \cdot (\mathbf{P}_k + \mathbf{P}_{M,k}) + M_{MNL,k} + e_k \quad (3.2)$$

Therefore we can say for day k the solutions take the distribution as follows assuming multipath is normally distributed with zero mean, where α_k is the percentage of daily double differences used for day k .

$$\check{\mathbf{P}}_k \sim N\left(\mathbf{P}_k + \mathbf{P}_{M,k}, \frac{1}{\alpha_k}\sigma_e^2(\mathbf{N}^T\mathbf{N})^{-1}\right) \quad (3.3)$$

Daily solution distribution for day k

If the same double differences are used every day then $\mathbf{P}_{M,k}$ becomes fixed and the variance of the solution is due to solely the number of double differences

used and the miscellaneous errors. Whilst the solution would be a biased solution, for land deformation monitoring where relative movements is the merit of interest, this fixed biased would be irrelevant hence the solution would be as good as one could hope.

If we assume we cannot obtain the same double differences every day and instead we randomly sample α percent of all the daily double differences with a uniform distribution, then we have only a probability that given a double difference obtained today there will exist a matching one tomorrow, then our solutions should be distributed as follows given enough days such that the average value of $\mathbf{P}_{M,k}$ more closely resembles that of $\mathbf{P}_M$ as we expect $\mathbf{P}_{M,k} \sim N\left(\mathbf{P}_M, (1 - \alpha_k) \sigma_M^2 \frac{1}{\alpha_k} (\mathbf{N}^T \mathbf{N})^{-1}\right)$.

$$\check{\mathbf{P}}_k \sim N\left(\mathbf{P}_k + \mathbf{P}_M, \left((1 - \alpha) \sigma_M^2 + \sigma_e^2\right) \frac{1}{\alpha} (\mathbf{N}^T \mathbf{N})^{-1}\right) \quad (3.4)$$

Daily solution distribution given uniformly random double differences

In this case over enough days we expect a fixed bias but with a variance larger than before. We are now faced with two problems the first is obviously the increase in variance, while the other is how the daily multipath bias $\mathbf{P}_{M,k}$ changes from one day to another. The GPS satellite repeat time is about four minutes lagged every day with respect to 24 hours, therefore if we could obtain observations throughout all daylight hours we would expect a slow change in the daily multipath bias. The solutions in this case could appear similar as if the actual location of the node was moving. As an example we performed a simulation where we generated normally distributed multipath, delta norms, and miscellaneous errors where the miscellaneous error was significantly smaller than the multipath error. We set a fixed node location of $[1, 1, 1]$, calculated the observable Φ for all epochs, used 50% of consecutive epochs everyday and shifted this by 0.3% every day, thus simulating obtaining observations only during daylight hours. Figure 3.8 below shows the plot for the x coordinate of the solutions using our simulation. “Phat” is for $\check{\mathbf{P}}_k$, “Pm offset” for $\mathbf{P}_M + \mathbf{P}_k$ and “Actual” for $\mathbf{P}_k$. As can be seen there is a slow movement where the deviations from $\mathbf{P}_M$ can be great. The distribution of the solutions matched the distribution as given in equation 3.4. Visually it is hard to tell whether or not the node is stationary or moving. Also, as can be seen from the graph the

solutions have a repeat period of approximately one year which is due to the four minute shift every day.

Figure 3.8: GPS model simulation of solutions for 50% of day observations and 4 minute daily shift

In comparison the same simulation was run again with the same parameters except exactly the same double differences were used everyday instead. The x coordinate of the solutions are plotted in Figure 3.9. As can be seen the remaining variance is due to miscellaneous errors as given in equation 3.3 when $\mathbf{P}_{M,k}$ is fixed.

Figure 3.9: GPS model simulation 50% of observations and the same matching double differences used everyday

Therefore our scenario the day to day the distribution may follow equation 3.3 but in the longer term the distribution may follow equation 3.4.

Therefore to reduce solution variance over the short and long terms, observations should be taken once every second 24 hours a day and data processed using the same double differences everyday. With permanently powered GPS receivers it is very easy to obtain the same double differences each day and obtain one every second 24 hours a day as there is enough power to do this. However, in our scenario we cannot simply turn our GPS receiver on in the middle of the night just because there is a double difference there that we need to match the double difference we used yesterday to calculate a solution. Therefore we would expect for our scenario a rather small α value causing an increase in the solution variance, while that of permanently powered high rate GPS receivers sampling at once a second would have a large α value producing solutions with less variance. We have a biased estimator with a large amount of variance. We now take advantage of the fact that over the long term $\mathbf{P}_{M,k}$ is a random variable.

Using our model from equation 3.1 we see that the residual in cycles for sidereal day k , evaluated at $\mathbf{P}_B$ with an actual location $\mathbf{P}_k$, can be written as follows..

$$r_k = \frac{1}{\lambda} (\Delta \mathbf{n} \cdot (\mathbf{P}_k - \mathbf{P}_B) + M + e_k) \quad (3.5)$$

For sidereal differencing using double differences, today's residual minus yesterday's residual results in the displacement since yesterday's observation; this is written as follows.

$$(r_n - r_{n-1}) = \frac{1}{\lambda} (\Delta \mathbf{n} \cdot (\Delta \mathbf{P}_n) + 2e) \quad (3.6)$$

The additive noise is double, the multipath is removed, and using LS results in $\Delta \hat{\mathbf{P}}_n$ the estimate of the displacement between yesterday and today. It's a very precise solution but the solution is for displacement and not for position, hence very good for precise epoch based solutions but gives nothing for daily solution improvements. Therefore, sidereal differencing removes the position information. If one wished to obtain a position solution one would have to add yesterday's position to the displacement which would once again add the multipath error back into the equations.

$$\begin{aligned} \lambda(r_n - r_{n-1}) + \Delta \mathbf{n} \cdot \check{\mathbf{P}}_{n-1} &= \Delta \mathbf{n} \cdot (\check{\mathbf{P}}_{n-1} + \Delta \mathbf{P}_n) + 2e \\ &= \Delta \mathbf{n} \cdot (\check{\mathbf{P}}_{n-1} + \Delta \mathbf{P}_n) + e_n - e_{n-1} \\ &= \Delta \mathbf{n} \cdot (\mathbf{P}_{n-1} + \mathbf{P}_{M,n-1}) + \Delta \mathbf{n} \cdot (\Delta \mathbf{P}_n) + e_n \\ &= \Delta \mathbf{n} \cdot \mathbf{P}_n + \Delta \mathbf{n} \cdot (\mathbf{P}_{M,n-1}) + e_n \end{aligned} \quad (3.7)$$

Averaging over many such days we obtain the following.

$$\begin{aligned} \text{ave} \left(\lambda(r_n - r_k) + \Delta \mathbf{n} \cdot \check{\mathbf{P}}_k \right) &= \text{ave} (\Delta \mathbf{n} \cdot \mathbf{P}_n + \Delta \mathbf{n} \cdot (\mathbf{P}_{M,k})) + e_n \\ &= \Delta \mathbf{n} \cdot \mathbf{P}_n + \Delta \mathbf{n} \cdot \text{ave} (\mathbf{P}_{M,k}) + e_n \end{aligned} \quad (3.8)$$

Over enough days we expect $\text{ave}^\infty (\mathbf{P}_{M,k}) = \mathbf{P}_M$ then

$$\text{ave}^\infty \left(\lambda(r_n - r_k) + \Delta \mathbf{n} \cdot \check{\mathbf{P}}_k \right) = \Delta \mathbf{n} \cdot \mathbf{P}_n + \Delta \mathbf{n} \cdot \mathbf{P}_M + e_n \quad (3.9)$$

CHAPTER 3. MULTIPATH AND SIDEREAL FILTERING

Therefore using LS to obtain an estimate $\bar{\mathbf{P}}_n^\infty$ for $\mathbf{P}_n$ using this equation we would then expect following distribution.

$$\bar{\mathbf{P}}_n^\infty \sim N\left(\mathbf{P}_n + \mathbf{P}_M, \sigma_e^2 \frac{1}{\alpha} (\mathbf{N}^T \mathbf{N})^{-1}\right) \quad (3.10)$$

Which is identical to the distribution when we had used the same double differences every day. This of course is an upper bound to accuracy and precision. If we consider arithmetic mean averaging over many such days things become more complicated as there is only a probability that we can form the residual difference $r_n - r_k$ in the first place. Because of this, we define the bar notation as the arithmetic mean calculated when possible including the current day n . For r_k at day n this is $\bar{r}_n = \text{mean}(\Omega \cup \{r_n\})$ where Ω contains a set of matching residuals; in this particular instance $\{r_n\}$ represents the set containing only r_n and ensures $\bar{r}_n$ is sensibly defined. Whilst the bar notation is valid for nonconsecutive days we only consider m consecutive days less than or equal to day n here for simplicity; this we take as read from here on. Therefore, more generally we write the following.

$$\lambda(r_n - \bar{r}_n) + \Delta \mathbf{n} \cdot \bar{\mathbf{P}}_n = \Delta \mathbf{n} \cdot \mathbf{P}_n + \Delta \mathbf{n} \cdot \bar{\mathbf{P}}_{M,n} + e_n \quad (3.11)$$

Sidereal filtering equation

The distribution of $\bar{\mathbf{P}}_{M,n}$ will be the same as $\mathbf{P}_{M,k}$ but with a scaled variance due to the days included in the average. For small values of m , $\bar{\mathbf{P}}_{M,n}$ can approximately be written as follows.

$$\bar{\mathbf{P}}_{M,n} \sim N\left(\mathbf{P}_M, (1 - \alpha)^m \sigma_M^2 \frac{1}{\alpha} (\mathbf{N}^T \mathbf{N})^{-1}\right) \quad (3.12)$$

Therefore, the LS estimate $\dot{\mathbf{P}}_n$ for $\mathbf{P}_n + \bar{\mathbf{P}}_{M,n}$ using equation 3.11 has a distribution as follows.

$$\dot{\mathbf{P}}_n \sim N\left(\mathbf{P}_n + \mathbf{P}_M, \left((1 - \alpha)^m \sigma_M^2 + \sigma_e^2\right) \frac{1}{\alpha} (\mathbf{N}^T \mathbf{N})^{-1}\right) \quad (3.13)$$

In our scenario we cannot randomly sample α observations throughout the day as the nodes are only able to operate during daylight hours. If we define the percentage of daylight hours with respect to an entire day as β , then we are only able to randomly sample $\alpha\beta$ percent of the observations that are within daylight hours (i.e we obtain $\alpha\beta$ percent of all the possible observations each day and all of these are in daylight hours). Every day the observations that β refer to are shifted by four minutes, therefore the probability of being able to form a residual falls the further away in time day k is away from day n . Hence the more days the average is run over the smaller the increase in the number of items in $\bar{r}_n$ until given enough days there is no overlap and there is zero probability that adding that day to the average will make any difference whatsoever. Also, we note that $\bar{r}_n$ is the average of fewer things than if there was no four minute shift every day no matter how many days we averaged over. In this case we have some function $g(\alpha, \beta, m, \delta)$ where we can write the distribution of the solutions as follows.

$$\hat{\mathbf{P}}_n \sim N\left(\mathbf{P}_n + \mathbf{P}_M, \left(g(\alpha, \beta, m, \delta) \sigma_M^2 + \sigma_e^2\right) \frac{1}{\alpha\beta} (\mathbf{N}^T \mathbf{N})^{-1}\right) \quad (3.14)$$

We do not obtain an explicit formula for $g(\alpha, \beta, m, \delta)$ instead we simply note that it is positively correlated with $\alpha\beta$ and m .

During daylight hours the nodes obtain observations but do not obtain them uniformly during daylight hours. Instead they obtain many more around the middle of the day and fewer in the morning and evenings. However, to keep things simple we do not consider this extra complication and instead continue to assume uniform acquisition of observations during daylight hours. In the Paekakariki testbed in New Zealand over a two week period we measured the average percentage of epochs obtained during daylight hours as 30%. Therefore, for a simulation we use $\alpha = 0.3$, $\beta = 0.5$ and $\delta = 0.003$. In addition we move the node by $[0.05 \ 0.05 \ 0.05]$ halfway through the simulation. We then use 15 days to perform the sidereal averaging ($m = 15$). Using the same simulation settings as before we obtain the following simulation result in Figure 3.10.

Figure 3.10: Simulation of sidereal filtering using 15 days with intermittent daylight observations ($\alpha = 0.3$)

As can be seen on average there is a slight solution bias and the sidereal filtered solution perform significantly better. As a comparison we performed the same simulation again but this time using $\alpha = 1$ signifying that all double difference during daylight hours were used.

Figure 3.11: Simulation of sidereal filtering using 15 days with continuous daylight observations ($\alpha = 1, \beta = 0.5$)

From Figures 3.10 and 3.11 we see a low frequency noise component and a

high frequency noise component. The low frequency component is clearly the same, only the high frequency component has been reduced when using interday sidereal filtering. The low frequency component appears as a slow moving solution bias that repeats every year and is due to the four minute daily shift. The low frequency component is problematic because it looks like a GPS receiver node slowly moving in time. From these figures we also see that in the long term interday sidereal filtering is only effective when the high frequency noise component has a magnitude significantly greater than the magnitude of the low-frequency component. This is equivalent to saying interday sidereal filtering is only really effective at improving daily solution accuracy when the magnitude of the usual solution precision is significantly greater than the magnitude of the slow moving bias that repeats every year.

To examine interday sidereal filtering on the the high frequency noise we look at just the noise obtained by using one day differencing. To be explicit $\dot{\mathbf{P}}_k - \dot{\mathbf{P}}_{k-1}$ and $\check{\mathbf{P}}_k - \check{\mathbf{P}}_{k-1}$. The the difference between consecutive days produces approximately double the variance of the high frequency noise that exists in the solutions and eliminates the low frequency component. From this we can then see the reduction of high frequency noise and hence the improvement in the short term solution accuracy. This time we run the simulation without the addition of the movement halfway through the simulation to avoid an outlier. Figure 3.12 shows simulations of $\dot{\mathbf{P}}_k - \dot{\mathbf{P}}_{k-1}$ and $\check{\mathbf{P}}_k - \check{\mathbf{P}}_{k-1}$ for an alpha of 1 and 0.3 while $\beta = 0.5$ and $\delta = 0.003$ and $m = 15$ as before; these correspond to the settings of Figures 3.11 and 3.10 respectively.

Figure 3.12: Day differenced solutions for $\alpha = 1.0$ and $\alpha = 0.3$ while $\beta = 0.5$ and $\delta = 0.003$

Figure 3.12 shows only the high frequency noise component once the low frequency noise component has been removed. As can be seen even when all double differences during daylight hours are used ($\alpha = 1.0$) there is a reduction in the high frequency noise component when using sidereal filtering. However, the improvement in the high frequency noise component is more significant when only using a fraction (30% in this case) of the double differences during daylight hours. In this case, using all double differences during daylight hours caused a reduction in the high frequency noise component of approximately 50%, while when using only 30% of double differences during daylight hours produced a reduction in the high frequency noise component of approximately 80%. Therefore, interday sidereal filtering is effective at reducing the high frequency noise component when using any fraction of observations during daylight hours. A reduction in the high frequency component is of course an improvement in solution precision.

We finally examine the effect interday sidereal filtering has on solutions where every double difference in a day is used to compute the solutions; in this case $\alpha = 1$ $\beta = 1$ and δ is irrelevant. Running this through the simulation we obtain Figure 3.13.

Figure 3.13: Simulation of sidereal filtering using all double differences in a day ($\alpha = 1, \beta = 1$)

As expected the solutions before and after sidereal filtering are identical and there is no improvement in accuracy or precision. In addition there is no slow moving solution bias that repeats every year.

It appears that interday sidereal filtering improves precision when obtaining observations solely through daylight hours. The improvement in precision is most marked when only a few observations are taken during daylight hours. Interday sidereal filtering may or may not significantly improve accuracy depending on the magnitude of the slow moving bias that repeats every year compared to the magnitude of the usual solution precision. The magnitude of the usual solution precision of course at least in part depends on the number of observations obtained during daylight hours.

In summary, we see added benefit from using sidereal filtering for our unique intermittent observation scenario that would otherwise not be available to us had it not be intermittent. In addition because our nodes are duty cycled even during daylight hours we obtain more benefit from interday sidereal filtering than we would if we were continuously obtaining observations throughout daylight hours. However, we also observe a slow moving solution bias that repeats once a year when using only daylight hour observations that cannot be reduced using our form of interday sidereal filtering. This slow moving bias affects accuracy while for short periods of time it appears as a fixed bias.

We now proceed to verify that interday sidereal filtering does indeed improve solutions when intermittently obtaining observations with the use of real life observations.

3.3.1 Simulation validation

We have seen for simulations based on our model of multipath, that for our particular unique hardware where observations are only available during the day, and even then only intermittently, that we predict our solutions should take the form which is seen in Figure 3.10. Firstly we predict the sidereal filter should respond instantly to a change in node location and the solutions should match that of the physical displacement performed. Secondly we predict that we do obtain an improvement in daily solution precision using the interday sidereal filtering described in this chapter when intermittently obtaining observations and that no improvement can be got when using continuous high rate observations. Thirdly, that there should be slow change in solution bias that will repeat once every year that our interday sidereal filtering cannot reduce. The third prediction we could not verify as it would require more than

a year of observations of a static node. Fourthly that any improvement is more pronounced when using fewer daylight observations than many daylight observations. The fourth prediction we do not validate here.

The first two predictions of response and improvement we validate as follows.

3.3.1.1 Sidereal response

An experiment was run with a node as described in the hardware chapter 5 but modified to be run off a continuous power supply. The antennas were placed on a roof and the receivers were mounted beneath the eaves away from direct sunlight and the elements, the baseline was approximately 10 m. With continuous power it is possible to obtain the highest possible accuracy and to determine whether or not the sidereal solution moves instantaneously when the underlying movement of the node changes, in addition, that the change of the sidereal solution is consistent with the underlying movement. The node's antenna was moved in a NNW direction and slightly down; firstly by $20 \pm 2mm$ and then by $5 \pm 1mm$. Using every epoch of the day, the primary principal component of the solutions were plotted with respect to the day the solution was calculated for. The Figure 3.14 shows this plot where the sidereal filter had a moving window period of 15 days.

Figure 3.14: Primary principal component of daily solutions given near continuous 24-hour observations

The node's antenna was moved approximately at midday on the 357th day; this can be seen in Figure 3.14 above, as a point that is neither at -0.01 nor 0.01 as some double differences were obtained while at the former location and others at the latter location causing the final solution to be neither one nor the other. The 5 mm movement was performed at 7 p.m. on the 368th day. As can be seen the sidereal filter responds instantaneously to movements as expected, but unlike simulations, for larger movements like the 2 cm movement, the sidereal solution underestimates the movement and slowly adjusts to the new location. This makes sense because multipath is a function of location, hence, the previous days used for the sidereal solution have slightly incorrect multipath information. For simulation we assumed that multipath was independent of location, hence why we do not see the slow linear rise of the sidereal solution that we see in Figure 3.14 in Figure 3.10. For the 5 mm movement the sidereal filter did not have sufficient time to discard all the observations before the 2 cm movement, 4 days we used before the 2 cm movement was performed when calculating the first sidereal solution after the 5 mm movement. From Figure 3.14 we estimate the magnitude of the movements of the sidereal solutions by examining the sidereal solution before the movements were performed and obtaining an estimates for the positions before the movements were performed; these values were $-9.9 \pm 0.2mm$, $8.0 \pm 0.4mm$ and $12.6 \pm 0.2mm$ in chronological order with a 95% confidence. From this we estimate the first movement was $17.8 \pm 0.6mm$ while the second movement was $4.6 \pm 0.6mm$ both with 95% confidence; therefore, both estimates are consistent with the estimates obtained from using a ruler. This shows for relatively small movements over the short timeframes at this particular test site at least, that sidereal solutions as far as we can determine are consistent with the underlying movement of of the GPS antennas themselves.

3.3.1.2 Sidereal solution improvement

All our static real life observational tests were relatively short lasting no more than a couple of months. This means we were able to see the high frequencies of the solutions but not the low frequencies. This means we could only observe solution precision improvement not accuracy. This of course we predict is where interday sidereal filtering will show a clear improvement.

According to our multipath model the solutions obtained when using the same

epochs every day to obtain solutions such as when using a permanently powered GPS receiver taking high rate observations interday sidereal filtering should have no effect. However, from Figure 3.14 this appears to not be so. If we consider just days before the 2 cm movement and after the 5 mm movement we see an improvement in 2DRMS of approximately 30%. We theorise that this improvement that we did not predict in our simulations maybe due to the fact that the satellite repeat period is not exactly 24 hours -246 seconds but instead varies for each satellite causing the multipath of the double differences not to entirely repeated each day. Plotting the first two principal components of the solutions when using all observations from the permanently powered receiver Figure 3.15 was obtained where this improvement can be seen more clearly.

Figure 3.15: First two principal components of daily solutions given all observations from a permanently powered receiver

To investigate the improvement when observations are not continuous we used the observations obtained from the continuously powered node and selected only ones that had a time in common with a solar powered GPS node placed nearby. Therefore, the distribution of activity of the permanently powered observations would match that of the observations obtained from the solar powered GPS node itself. In this way we eliminate the possible dependence of location on improvement. Once again, considering just days before the 2 cm movement and after the 5 mm movement we obtain an improvement

in 2DRMS of approximately 60% in this case. This improvement is double that of the unexpected improvement obtained when obtaining observations continuously once a second. Plotting the first two principal components of the solutions when only using observations from the permanently powered receiver when the solar powered GPS receiver node obtains an observation we obtain the following plot where this improvement can be seen.

Figure 3.16: First two principal components of daily solutions given only observation times that match a solar powered GPS node

As can be seen from both of these figures (3.15 and 3.16) using sidereal filtering produces solutions reasonably similar to one another. The 5 mm step without using sidereal filtering in the second figure (3.16) cannot be easily determined while it is easy to determine when using sidereal filtering. As desired the improvement when using intermittent observations during daylight hours is considerably greater than that obtained when using a permanently powered GPS receiver.

Finally we compare sidereal solutions with non-sidereal solutions for an actual solar powered GPS receiver placed at the Paekakariki testbed and left for 22 days. This GPS receiver node is one as described in Chapter Chapter 5. A birds eye view of the solutions can be seen in Figure Figure 3.17 on page 68.

Figure 3.17: Effect of interday sidereal filtering from observations obtained from intermittent solar powered hardware on solutions (one square centimeter)

As can be seen sidereal filtering has a significant effect on solution precision. An improvement of approximately 50% in 2DRMS from 5.6 mm to 2.8 mm was obtained. This improvement percentage in 2DRMS is comparable to the improvement we saw when using the permanently powered GPS receiver using only observations taken when a solar powered GPS receiver node was active.

3.4 Intraday sidereal filtering

An intraday sidereal filtered solution (an intraday solution) is a epoch based solution using observations in other days in addition to the observations in the second in question itself; the other days may not necessarily be consecutive days.

Intraday sidereal filtering is not a primary concern of ours as our system is not competing with real time systems such as RTK. In addition, the decision was made to concentrate on long term land deformation measurements using daily solutions and doing this well. However, there is no reason why our system can't be extended to epoch based solutions but more work would be needed to investigate this type of solution. In the section we briefly examine intraday solutions using past information and the present epoch to calculate a solution.

Calculating the multipath estimate for the intraday algorithm is performed by using previous days where the location of the node is unchanged to which we have interday sidereal solutions. These interday solutions are very accurate with the residuals mainly consisting of multipath and hence we can calculate the multipath for those previous days. This multipath information we have then gets combined using simple geometric averaging to create a multipath map to which we can correct any double difference for the present day. Rather than correcting the double differences themselves, we turn the double differences into residuals using a biased location, correct them, then convert these back into double differences to which least squares can be used. This is done so as to avoid using unwrapped double differences as was done in the interday sidereal filtering algorithm. Assuming we have an estimate to the true multipath M as M' this procedure is summarised as follows initially for unwrapped double differences.

- $r_n = \frac{1}{\lambda} (\lambda \Phi_n - \Delta \mathbf{n} \cdot \mathbf{P}_B)$ definition of residual for location $\mathbf{P}_B$ for day n
- $\lambda r_n - M'_{n-1} = \lambda \Phi_n - \Delta \mathbf{n} \cdot \mathbf{P}_B - M'_{n-1}$ use multipath map to correct residuals
- $\lambda r_n - M'_{n-1} = \Delta \mathbf{n} \cdot \mathbf{P}_n + M_n + e - \Delta \mathbf{n} \cdot \mathbf{P}_B - M'_{n-1}$ substituting model for Φ
- $\lambda r_n - M'_{n-1} + \Delta \mathbf{n} \cdot \mathbf{P}_B = \Delta \mathbf{n} \cdot \mathbf{P}_n + e + \Delta M_n$ defining ΔM_n as multipath map error between today's actual and yesterday's estimate of the multipath
- $\lambda r_n - M'_{n-1} + \Delta \mathbf{n} \cdot \mathbf{P}_B \approx \Delta \mathbf{n} \cdot \mathbf{P}_n$ approximation if good multipath map and noise is small

To obtain the multipath estimate M' geometric averaging was used due to the simple implementation and is summarised as follows.

- $m_n = \frac{1}{\lambda} (\lambda \Phi_n - \Delta \mathbf{n} \cdot \mathring{\mathbf{P}}_n)$ definition of residual for location $\mathring{\mathbf{P}}_n$. This residual is close to multipath if the receiver did not move on day n .
- $M'_n = (1 - \alpha) M'_{n-1} + \lambda m_n$ geometric averaging for predicting the multipath of tomorrow.

CHAPTER 3. MULTIPATH AND SIDEREAL FILTERING

Therefore, two equations are used for our intraday sidereal filtering; a corrected equation based on yesterday's multipath estimate and an update equation of the multipath estimate, these two equations are equations (3.15) and (3.16) is not based.

$$\lambda r_n - M'_{n-1} + \Delta \mathbf{n} \cdot \mathbf{P}_B \approx \Delta \mathbf{n} \cdot \mathbf{P} \quad (3.15)$$

$$M'_n = (1 - \alpha) M'_{n-1} + (\lambda \Phi_n - \Delta \mathbf{n} \cdot \mathring{\mathbf{P}}_n) \quad (3.16)$$

Intraday sidereal filtering equations

Rather than using unwrapped residuals we chose instead to use wrapped residuals so as to avoid cycle slips as we did in the interday sidereal filtering. Upon using wrapped residuals our algorithm can be described as listed in the following algorithm.

Algorithm 3.1 Intraday sidereal algorithm

Correct

- Shift multipath map $\mathbf{m}$ temporarily by an integer multiple of a_p to align them with today, day n
- Calculate today's wrapped residuals for epoch z , $\mathbf{r}_{n,z} = f\left(\frac{C_f}{\lambda} \left(\lambda \Phi_{n,z} - \mathbf{N}_z \cdot \dot{\mathbf{P}}_{n-1}\right)\right) / C_f$ based on yesterday's interday sidereal solution $\dot{\mathbf{P}}_{n-1}$
- Let $\tilde{\mathbf{r}}_{n,z} = \text{magmin}(\mathbf{r}_{n,z} - \mathbf{m}_{n-1,z}, \mathbf{r}_{n,z} + \mathbf{m}_{n-1,z})$ be the corrected residuals for this epoch

Solve

- Use LS to solve $\lambda \tilde{\mathbf{r}}_{n,z} + \mathbf{N}_z \cdot \dot{\mathbf{P}}_{n-1} \approx \mathbf{N}_z \cdot \mathbf{P}_{n,z}$ if sufficient independent equations and PDOP is low enough, hence obtain an estimate of $\mathbf{P}_{n,z}$
- Use Kalman smoothing on the LS solution where the model is stationary spherical and the measurement covariance is taken from the LS solution as $(\mathbf{N}_z^T \mathbf{N}_z)^{-1}$
- Call the solution after smoothing $\dot{\mathbf{P}}_{n,z}$

Update

- At the end of the day update the multipath map $\mathbf{m}$ as follows if no movement has happened throughout the day
 - Calculate today's wrapped residuals $\mathbf{r}_n = f\left(\frac{C_f}{\lambda} \left(\lambda \Phi_n - \mathbf{N} \cdot \dot{\mathbf{P}}_n\right)\right) / C_f$ based on today's interday sidereal solution $\dot{\mathbf{P}}_n$
 - Update map $\mathbf{m}_n = (1 - \alpha) \mathbf{m}_{n-1} + \alpha |\mathbf{r}_n|$
-

3.4.1 Intraday sidereal filtering improvement

We compare epoch based solutions, with and without intraday sidereal filtering. Figure 3.18 below shows epoch based solutions without intraday sidereal filtering on observations obtained from the LuShan testbed by our wireless solar powered GPS receiver nodes as described in the hardware chapter Hardware. Epoch's with insufficient equations and/or excessive *Position Dilution Of Precision* (PDOP) were not processed. In addition Kalman smoothing was

performed on the solutions where the model was stationary spherical and the measurement covariance taken from the LS solution as $(\mathbf{N}_z^T \mathbf{N}_z)^{-1}$. As the hardware was intermittent only a fraction of solutions during the day could be calculated producing many seconds where no solution could be calculated. Because of this these the x axis only represents the epochs where solutions could be calculated, Figure show the results 3.18.

Figure 3.18: Epoch based North solutions without intraday sidereal filtering

Figure 3.19 below shows the same observations as was used for Figure 3.18 but with intraday sidereal filtering instead. The algorithm had 10 days beforehand to adjust its multipath map.

Figure 3.19: Epoch based North solutions using intraday sidereal filtering

As can be seen the effect the multipath has been considerably reduced on a per second basis. For the solutions themselves an improvement of approximately 60% was observed in 2DRMS accuracy, while an improvement of approximately 80% was observed once the solutions themselves had been run through Kalman filter. For this test the 2DRMS precision measurements are tabulated in the following table, Table 3.1.

	2DRMS (<i>mm</i>)
Without intraday sidereal filtering	51
With intraday sidereal filtering	20
Kalman without intraday sidereal filtering	28
Kalman with intraday sidereal filtering	6

Table 3.1: Epoch based accuracies with and without intraday sidereal filtering

3.4.2 Intraday sidereal filtering response

3.4.2.1 5 mm permanent power

The response of the intraday sidereal filtering with respect to movement was first examined using the 5 mm movement of the permanently powered GPS receiver as seen in section 3.3.1.1. Firstly the primary horizontal component of the movement without intraday sidereal filtering over four days was plotted in Figure 3.20 with a one hour moving average filter applied to the solutions.

Figure 3.20: 5 mm movement no intraday sidereal filtering with a permanently powered GPS receiver

The same data was processed once again identically but with the addition of intraday sidereal filtering with an α of 0.25; the results can be seen in the Figure 3.21 with the same scale as the previous figure, Figure 3.20.

Figure 3.21: 5 mm movement intraday sidereal filtering with a permanently powered GPS receiver $\alpha = 0.25$

As can be seen, without intraday sidereal filtering it is very difficult to determine that movement has taken place, while, with intraday sidereal filtering, the movement can be detected more easily. The time of the movement was determine by Figure 3.21 as being between 6 p.m. and 8 p.m. on the 3rd of January while that of the actual movement was at 7:10 p.m. on the 3rd of January.

Performing *Kernel Density Estimation* (KDE) on Figure 3.21 Figure 3.22 was obtained.

Figure 3.22: KDE plot of the 5 mm movement using intraday sidereal filtering with the permanently powered GPS receiver

Figure 3.22 shows that the movement was approximately 3.2 mm. The slope of the roof as measured by an inclinometer was 15° therefore the total movement should have been 3.3 mm which is 1.7 mm out from the expected value of 5 mm.

Finally the intraday sidereal filtering was performed again but this time using an α value of one. The results of this can be seen in Figure 3.23.

Figure 3.23: 5 mm movement intraday sidereal filtering with a permanently powered GPS receiver $\alpha = 1$

The difference between Figures 3.21 and 3.23 is the impulse response. It can be seen in Figure 3.23 that there is a sharp dip exactly one day after the the movement that was not noticeable in Figure 3.21. This is expected as when $\alpha = 1$ only the previous day is used for obtaining a multipath estimate and the requirement when obtaining the multipath estimate is that the node stationery for that day which is clearly not the case when the node moves. When $\alpha = 0.25$ less of the multipath estimate is based on the previous day and hence why we do not notice any dip one day after the actual movement in Figure 3.21.

3.4.2.2 2 cm solar power

A step response test using a solar powered node over four days was performed at the Paekakariki testbed in New Zealand. The first day was to record a multipath map, the second day to make sure the multipath mitigation was successful and to obtain a control, a third day where the node was moved, and a fourth day to monitor the response of the intraday sidereal filter. An α of

1 was used for the intraday sidereal filtering algorithm which meant a finite impulse response and only one day after the step would be needed to see any type of response.

The movement of the node was in approximately a north direction and the movement as measured by a ruler was approximately 2 cm. The movement was done at 11:06 AM on the 5th of December 2014. Using the intraday sidereal filtering algorithm, a linear movement was seen starting at approximately 11 AM and finishing at 12 PM but it was difficult to determine as the solutions were only intermittent. As the motion was instantaneous, we could say that the first sign of the linear movement should be at or after the actual instantaneous motion as caused by us was performed. Therefore we determined from the intraday sidereal filtering algorithm that the instantaneous motion happened at approximately 11 AM which is consistent with the actual movement as performed by us.

The Figure 3.24 shows the density of the solutions over the second to the fourth days inclusive. As can be seen there is a clear movement in almost a due north direction; the displacement between the two locations as measured from Figure 3.24 below is approximately 1.8 cm. This displacement is slightly less than the 2 cm as measured using the ruler.

Figure 3.24: Birds eye view of the density of the epoch based solutions with intraday sidereal filtering

Figure 3.25 below shows the primary principal component of the horizontal

solutions around the time when the movement was performed.

Figure 3.25: Epoch based intraday sidereal filtering around time of the movement of a solar powered GPS receiver node

As can be seen the intermittent nature of using solar powered GPS receivers means if a movement happens when the GPS receiver is not on or actively obtaining useful observations no movement will be measured until after the GPS receiver becomes active again and obtains a sufficient number of observations.

Figure 3.26 below shows the primary principal component of the horizontal solutions for all four days.

Figure 3.26: Epoch based intraday sidereal filtering over four days of a solar powered GPS receiver node

As a comparison, without using the intraday sidereal filtering, the density of the solutions of days 2 to 4 inclusive were plotted as before in Figure 3.24 but without intraday sidereal filtering; this can be seen in Figure 3.27 where the scale of the plot is the same as in Figure 3.24. As can be seen in this case it is not possible to determine the two locations.

Figure 3.27: Birds eye view of the density of the epoch based solutions without intraday sidereal filtering

Chapter 4

Algorithm comparison

4.1 Reduction of network bandwidth comparison

GPS receivers for precision applications output the following; L1 pseudorange observable, L1 phase observable, L1 Doppler observable, L1 signal strength observable, loss of lock indicator, quality indicator, satellite number, and the observation time as given by the receiver's clock. Usually a great deal more is given out. While not all are needed for high accuracy solutions, the pseudorange, phase, satellite number and time would be regarded as the minimum needed for some sort of high accuracy solution.

A typical breakdown of data size would be, the pseudorange, phase and time are all eight byte values, Doppler is four bytes, signal strength one byte, loss of lock indicator one byte, quality indicator one byte, and satellite number one byte.

For an epoch with n many observations to different satellites usually the time would be only given out once. Table 4.1 shows a breakdown of the number of bytes sent in this situation.

CHAPTER 4. ALGORITHM COMPARISON

Data name	Bytes	Bytes per epoch	Total bytes per epoch
Time	8	8	
pseudorange	8	$8n$	
phase	8	$8n$	
doppler	4	$4n$	
signal strength	1	n	
loss of lock indicator	1	n	
quality indicator	1	n	
SV	1	n	
			$8 + 24n$

Table 4.1: Typical breakdown of the minimum number of bytes sent per epoch from the receiver to the user for standard algorithms.

While n could be anywhere from 0 to 32 for one epoch a typical value would be between 4 and 10. Assuming an upper bound of 10 this would produce up to 248 bytes a second per node.

Low power wireless network solutions have relatively slow data rates restricting channel capacity. This is partly due to the fact that slower data rates mean larger transmission ranges for the same amount power and the same method of modulation. In addition, less bandwidth requirement for nodes results in lower epoch data loss thus mitigating the effect of a lossy channel. Because of channel capacity, power usage, and epoch loss, minimizing the amount of data that the nodes actually sends is advantageous. Another advantage of less bandwidth requirement is more nodes can occupy a channel increasing the number of nodes in a network on a given channel.

To reduce the amount of data from the GPS receivers themselves, some of the CFFS algorithm is implemented on the nodes themselves thus changing the data sent over the RF link. Algorithm 2.1 is the algorithm run on the nodes.

The extrapolated wrapped phase observable, satellite number and the second aligned time are the main variables returned. In addition, occasionally an approximate code based solution is returned; this can be done as infrequently as once or twice a day. This is so infrequent that effectively it contributes nothing to the needed bandwidth and can be ignored.

The main part of the algorithm has an approximate knowledge of the time via the Internet or some other mechanism and the node section has already

aligned the time to on the second. If we assume that the main part of the algorithm knows the time to within one minute and the latency between the node section and the main section of the algorithm is also less than a minute, then it is possible for the main part of the algorithm to calculate the time of an observation if the node section sends just a modulo 256 version of the time. As these timing requirements are very easy to achieve, only one byte per epoch is needed to be sent between the node and the main section of the algorithm.

The wrapped phase observable removes the integer part of the phase observable and leaves a number between zero and one. This transforms a number that was in the order of 20,000 needing an accuracy of a fraction of one, to a number in the order of one needing a fraction of one; this means we only need a couple of bytes to encode the remaining information just as accurately and for our purposes 2 bytes should be sufficient. 2 bytes means we have the phase resolution of about one in 65,000.

The following table shows a breakdown of the bare minimum number of bytes that a node could send for the CFFS algorithm.

Data name	Bytes	Bytes per epoch	Total bytes per epoch
Aligned time	1	1	
Corrected phase	2	$2n$	
SV	1	n	
			$1 + 3n$

Table 4.2: Breakdown of the minimum number of bytes a node could send per epoch for the CFFS algorithm

As before, assuming an upper bound of 10 satellites in one epoch this would produce up to 31 bytes a second per node. This equates to an 87.5% reduction in required bandwidth for the same given epoch rate.

4.1.1 Real life reduction of bandwidth comparison

Low cost GPS receivers nodes powered by small solar panels and batteryless using low power radios were designed and implemented. The radio links had no packet loss correction, the packets were sent using a broadcast method but if they did not arrive they would not be sent again. The GPS receivers

CHAPTER 4. ALGORITHM COMPARISON

themselves were singleband Ublox LEA6T while the antennas used were 25 mm patch and 20 mm helical antennas. The receivers were placed on the ground outside with a high amount of surrounding trees and buildings in winter. The nodes received no direct sunlight due to the time of year and the tall trees. The Sun’s maximum elevation was 27° at noon and the day length was 9.5 hours. During these 9.5 hours of the day the nodes on average were able to have their GPS receivers on approximately 20% of the time (about 2 hours a day). A receiver was placed approximately 10 m away from the nodes and data gathered. A permanently powered reference receiver was used to maximize overlapping epochs of the nodes; this was placed on a roof.

Two experiments were performed, the first one where all the typical data as in Table 4.1 was returned (“typical data”) for approximately one week and another one for approximately two weeks where the output from the node algorithm as in Table 4.2 in addition to signal strength and some debugging information was returned (“node algorithm data”). The epoch loss as well as the bytes per epoch sent over the radio link where compared when the nodes returned the “node algorithm data” and when the nodes returned “typical data”; the results of which are summarized in the table below.

	Average bytes per epoch	Average epoch loss
“typical data”	196	26%
“node algorithm data”	50	14%

Table 4.3: Comparison of channel capacity requirement and data loss for typical algorithm compared to CFFS algorithm

The reason for a reduction of only 74% of data compared to the estimated 87.5% is due to the added signal strength and debugging information added to the “node algorithm data”.

There is approximately a 74% reduction in bandwidth and a 46% reduction in epoch loss when using the “node algorithm data” compared to what is normally outputted. As the “node algorithm data” is from observations that have been partly processed through the CFFS algorithm, this data wouldn’t be able to be used by typical GNSS packages. The “node algorithm data” is intermediate data designed for the CFFS algorithm, it’s an integral part of the algorithm itself.

4.2 Solution comparisons

Accuracy and precision can be confusing. In particular accuracy is difficult to measure and a subtle subject made worse by common vague usage of the word in everyday speech. The figure below shows the distinction between precision and accuracy for a set of solutions colored blue with respect to a true solution colored red. The black circle signifies the level of precision for the set of measurements while the distance between the center of the black circle and the center of the red dot represents the accuracy of the set of measurements.

Figure 4.1: Precision and accuracy (red dot is true location)

The precision can be easily estimated from a set of solutions, the accuracy is an entirely different matter. As pointed out in [48], it is not uncommon for accuracy to be calculated internally where solutions are obtained over a long period such that they take on a normally distributed form and it is assumed that the mean of this distribution matches the true solution. This assumption is not necessarily valid. As we have seen in the multipath chapter 3, multipath causes a fixed solution bias that is present even if observations are taken continuously throughout the day and such a bias does not change at least in simulations from one day to the next. Unavoidably systematic errors

causing biases creep into solutions. For many applications this would not be a problem as a fixed bias that does not change no matter what just means you're consistently measuring to a different point of your apparatus. However, multipath causes a location dependent bias, if the GPS receiver gets moved this bias changes with respect to the mean location of the solutions, hence invalidating the assumption that the mean of normally distributed solutions taken over a large enough period matches the true solution. To measure the true location requires calculating accuracy externally which means using some other device with a better accuracy. This is beyond the scope of this thesis so we only consider approximate internal accuracy assessment. As said by [48] "Keep in mind that any given GPS measurement can be represented by the following equation: $\text{measurement} = \text{exact value} + \text{bias} + \text{random error}$. The random error component presents roughly the same problem for both internal and external assessments. The bias however, requires external truth for detection. There is no easy way to detect a constant shift from truth in a dataset by studying only the shifted dataset."

While probably difficult to measure, the bias caused due to multipath using 24 hours of observations in a low multipath environment with a high quality GPS antenna most likely produces a small bias partly due to the environment and equipment, and also partly due to the 24 hours observations obtaining an average of the instantaneous multipath bias. In this case obtaining accuracy internally is probably reasonably close to the actual accuracy. However, for our application we have poor uncalibrated antennas with no choke rings, a high multipath environment and a predicted yearly cycle of a bias caused due to multipath and only being able to obtain daylight our observations. This means reliably calculating accuracy internally is far more difficult and potentially more inaccurate. The yearly solution bias cycle as predicted in Chapter 3 particularly makes an accuracy estimate very difficult as at least a year worth of observations would be needed to take into account the yearly solution bias cycle.

Accuracy and precision are meaningless without some type of measure. Accuracy and precision measures are generally the probability that a solution will be within some shape of some given size. Generally the shapes used are circles or spheres and the probabilities go from 50% and up. 2D horizontal (northing easting) seem to be the most commonly given probably because GPS solutions tend to be more accurate in this plane compared to the up direction and are

CHAPTER 4. ALGORITHM COMPARISON

reasonably symmetrical in this plane. The following table defines three very common measures where CEP DRMS and 2DRMS are for the 2D horizontal plane and SEP is for all three dimensions. For the one dimensional up direction RMS is usually used.

Name	Description
CEP	50% probability circle, half of the solutions will fall inside a circle with radius CEP
DRMS	63 to 68% probability circle, 63 to 68% of the solutions will fall inside a circle with radius RMS
2DRMS	95 to 98% probability circle, 95 to 98% of the solutions will fall inside a circle with radius 2DRMS
SEP	50% probability sphere, half of the solutions will fall inside a sphere of radius SEP
RMS	68% line, 68% of solutions will fall on the a line of radius RMS

Table 4.4: Measures of accuracy and precision

The following table shows the approximate calculations we used to calculate the these measures where σ_N , σ_E and σ_U are standard deviations of the solutions in the North East and up directions respectively. These formulas were obtained from [26].

Name	Approximate calculation
CEP	$0.589 (\sigma_N + \sigma_E)$
DRMS	$\sqrt{\sigma_N^2 + \sigma_E^2}$
2DRMS	$2\sqrt{\sigma_N^2 + \sigma_E^2}$
SEP	$0.513 (\sigma_N + \sigma_E + \sigma_U)$
RMS	σ_U (for up direction)

Table 4.5: Approximate internal accuracy and precision calculations from a set of solutions

We only consider internal accuracies in this thesis and when we talk about the accuracy of a solution, we mean that we have obtained an estimate for the standard deviations of the population from a sample of solutions in sufficiently large quantity that the mean of these solutions we hope is the same as the true solution and have calculated the accuracies as in the previous table. For solution precision as we typically deal with daily solutions and only able to calculate one solution a day, the precision of the solution means the precision

as measured from the previous table using a set of solutions taken over a period of time no more than a few months or over longer periods of time when the low frequency components of the solution have been removed leaving just the high frequency components.

4.2.1 Comparison methodology

We designed two programs called JAC-MM and JAC. JAC-MM implemented the CFFS algorithm while JAC implemented the CFFS algorithm bar the sidereal filtering stage. If we assume a remote GNSS receiver setup isolated from the power grid and connected to the user via a wireless link, then, if we reduce the cost and power requirements of the hardware and deployment, the user will most likely obtain fewer types of observations, fewer and more intermittent observations, increased multipath, increased PCV, and generally poorer quality data. The question then remains is it possible to still obtain accurate/precise static solutions and if so how accurate/precise will they likely be. One would not expect current GNSS software to do a particularly good job in this respect as the people designing the software maybe assuming continuous high quality data is available. Therefore, we ran different types of modified observations through JAC and JAC-MM and compared these static relative solutions with the same modified observations run through the open source GNSS packages RTKLib [11] and GPSTk [8]. RTKLib and GPSTk were used because these were the only two open source GNSS packages freely available that supported singleband phase based relative position solutions that we found. Static solutions were obtained on a day by day basis.

RTKLib performs static solutions in addition to a myriad of other solution types such as RTK, *Precise Point Positioning* (PPP), *Differential GPS* (DGPS), Single, Fixed, and so on. Both singleband and dual band receivers are supported. For static solutions on short base lines less than 10 km it uses some form of an extended kalman filter and double differences [52] (pg 161).

GPSTk has in it an application called DDBase that is capable of calculating static solutions using only singleband receivers and also uses double differences [35].

JAC implements the CFFS algorithm but without sidereal filtering. Therefore, all daily solutions that JAC calculates is solely based on observations obtained

for that day just as RTKLib and GPSTk are. This allows for meaningful performance comparisons between the three programs.

Sidereal filtering uses information from previous days to calculate solutions and hence is not a fair comparison with JAC GPSTk and RTKLib. Sidereal filtering algorithms could be appended to anyone of these three programs to obtain solutions of higher accuracies/precision. However, we do append an implementation of sidereal filtering to the JAC program so as to see the effect that sidereal filtering has compared to the other three programs; this JAC program with the appended sidereal filtering is called JAC-MM and is a full implementation of our CFFS algorithm as described in Chapter 2.

4.2.2 Continuous high quality data

The purpose of this subsection is to show that our implementation of the CFFS algorithm produces solutions that are comparable to pre-existing solution processing programs under good quality conditions. For initial performance testing we used two months of high quality data obtained from GPS receivers, the first located at Te Papa museum and the second at Wellington Airport. This data was obtained from the RINEX data archive of Land and information New Zealand (LINZ) [5]. The data rate was one epoch every 30 seconds. The baseline between these two receivers was around 4 km. Both receivers were Trimble NetRS [12] while the antennas were different, one being an LEIAT504 and the other a TRM41249.00. The data was supplied to us by the GeoNet project. These types of receivers and antennas are high-end dualband. The static solution accuracy of such a receiver as given by the data sheet is $5\text{ mm} + 0.5\text{ ppm DRMS}$ for the horizontal, and the up direction is $5\text{ mm} + 1\text{ ppm RMS}$, where *ppm* refers to the baseline length. This means the datasheet implies that the static solutions as calculated by the receivers themselves would be $14\text{ mm } 2\text{DRMS}$ for the horizontal. Power consumption is between 3 and 4 W.

For a fair comparison, as our algorithm is intended to only deal with singleband GPS receivers, observations were modified first by stripping of precise P codes as well as L2 band observations, so as to be left with just C1, D1, S1, and L1 observation types and broadcast navigation data; these types of observations are typical for low cost low power singleband GPS receivers.

DDBase failed on six days due to not being able to read the navigation files correctly despite these files being obtained from the IGS products [27]. Both GPSTk and JAC obtained solutions for all days processed. Figure 4.2 shows a bird's eye view of the solutions while table 4.6 shows some statistics of the solutions where NEU is an acronym for *North East Up*. The JAC-MM solutions are also included.

Figure 4.2: Birds eye view using high quality data

CHAPTER 4. ALGORITHM COMPARISON

	Mean NEU (m,m,m)	Standard deviation NEU (mm,mm,mm)
JAC	(3666.689,-2035.341,15.529)	(1.94,2.48,4.79)
DDBase	(3666.674,-2035.352,15.521)	(2.05,2.29,5.03)
RTKLib	(3666.675,-2035.352,15.537)	(3.49,3.01,5.39)
JAC-MM	(3666.689,-2035.341,15.529)	(1.32,2.05,3.72)

Table 4.6: Statistics using high quality data

As can be seen all programs produced solutions with comparable precision. The standard deviations of all solutions are similar. In the horizontal plane DDBase and RTKLib agree with each other on average to within a millimeter, in the vertical RTKLib is 1.6 cm higher than DDBase. JAC agrees with DDBase in the vertical direction to within 8mm while in the horizontal plane JAC differs from both DDBase and RTKLib by approximately 1.8 cm. The JAC to DDBase difference is 20mm, JAC to RTKLib difference is 20mm and the DDBase to RTKLib difference is 16mm. Three measures of accuracy are summarized in the table below.

	CEP (<i>mm</i>)	2DRMS (<i>mm</i>)	SEP (<i>mm</i>)
JAC	2.6	6.3	4.7
DDBase	2.6	6.1	4.8
RTKLib	3.8	9.2	6.1
JAC-MM	2.0	4.9	3.6

Table 4.7: Solution precision using high quality data

As already mentioned, JAC-MM computes solutions using data from previous days. Therefore the results JAC-MM's solutions are unfair when directly compared to the other three programs solutions. However, as expected JAC-MM's solutions are more precise than the other three programs. It's also interesting to note that sidereal filtering improves precision despite the observations only having an epoch rate of once every 30 seconds. This is interesting because the sidereal shift required means there is no matching observation of a pair of satellites today as there was yesterday, as $246/30$ is not an integer. In fact it takes five days of separation for a matching observation of a pair of satellites to occur. Examining the numbers, JAC-MM is about 20% more precise than JAC for this data.

All programs had significant solution biases when compared with one another ranging from 16 to 20 mm. For our purposes we are not concerned with solution biases that do not change with location or over time. Whether or not these biases caused by using the three different programs change or not with location or extended time we do not know. The figure below shows the mean solutions with 95% confidence spheres over these two months worth of processing. As can be seen none of them are consistent with one another. JAC-MM's sphere was not plotted as this was within JAC's sphere. It's interesting to note that all programs used the same observations but all programs produced different results. All programs produced very precise results.

Figure 4.3: Mean solutions calculated over two months

4.2.3 Simulated intermittent high quality data

Simulated intermittent high quality data was created by using the same high quality Te Papa Wellington Airport data as before but with randomly removing 50% of satellite observations. As before the data was run through JAC, DDBase and RTKLib. DDBase and RTKLib failed to return any solutions. Figure 4.4 shows the bird's eye view of the solutions while Table 4.8 shows

some statistics of the solutions and Table 4.9 shows measures of precision of the solutions.

Figure 4.4: Birds eye view using high quality data with 50% random removal of satellite observations

	Mean NEU (m,m,m)	Standard deviation NEU (mm,mm,mm)
JAC	(3666.685,-2035.340,15.530)	(2.51,2.38,4.61)
JAC-MM	(3666.685,-2035.340,15.530)	(2.21,2.14,4.01)

Table 4.8: Statistics of simulated intermittent high quality data

	CEP (<i>mm</i>)	2DRMS (<i>mm</i>)	SEP (<i>mm</i>)
JAC	2.9	6.9	4.9
JAC-MM	2.6	6.1	4.3

Table 4.9: Solution precision using simulated intermittent high quality data

JAC’s solutions are almost identical to the situation before where no satellite observations were removed; the mean location is 4 mm difference between the two. As can be seen the effective randomly removing 50% of the satellite observations makes little difference to the precision of JAC’s solutions.

4.2.4 Algorithm performance using real life intermittent poor quality data

The purpose of this subsection is to determine whether or not any of the three programs (JAC, DDBase and RTKLib) could obtain solutions of reasonable precision using real life intermittent poor quality data obtained from the GPS WSN nodes that we designed. The same two experiments used in subsection 4.1.1 used to obtain epoch loss were also used to obtain real life intermittent poor quality data.

The CFFS algorithm is designed to be a distributed algorithm where some of the algorithm is run on the nodes themselves. This has the advantage of a reduction in epoch losses as seen in subsection 4.1.1. It would be unfair to compare the CFFS algorithm with DDBase and RTKLib without considering this. If the nodes just return the “typical data” and the node section of the CFFS algorithm was to be run in a non-distributed manner on the base station, the CFFS algorithm could conceivably be incurring a 26% loss of epochs compared to a 14% loss of the epochs as seen in Table 4.3. Therefore, two node setups were configured. The first set up where the “typical data” was returned by the nodes, and another set up where the “node algorithm data” was returned by the nodes. The first set up was designed to supply data to DDBase and RTKLib, whilst the second set up was designed to supply data to the main part of the CFFS algorithm.

Approximately 1 week of “typical data” was obtained for DDBase and RTKLib, while approximately 2 weeks of “node algorithm data” was obtained for the

main section of the CFFS algorithm to process. The observations were not taken simultaneously, and due to the need to reprogram the nodes to output “node algorithm data” rather than “typical data”, the location of the nodes where unavoidably shifted slightly during the two data gathering periods. Such movements were minimal and do not change our results. The following figure shows a birds eye view of the solution results. JAC calculated solutions for all days. DDBase could calculate solutions for all days but they were inaccurate, RTKLib did not successfully calculate any solutions.

Figure 4.5: Bird’s eye view using intermittent poor quality data. “typical data” for DDBase and RTKLib while “node algorithm data” for JAC. (One DDBase solution not plotted due to large variation)

To compare the effect on the ability to obtain solutions due to the reduction in epoch loss when transmitting “node algorithm data” rather than “typical data” between the nodes in the base station when using JAC, the node algorithm was implemented on the base station and the one week worth of data that DDBase and RTKLib used to calculate solutions above was run through JAC. JAC calculated solutions for all days; the birds eye view of the obtained solution are shown in the following Figure.

Figure 4.6: Bird's eye view using intermittent poor quality data. "typical data" for DDBase and RTKLib and JAC. (One DDBase solution not plotted due to large variation)

As can be seen despite the increase in loss of epochs, JAC performs very similar to when transmitting "node algorithm data" between the nodes and the base station. The following two tables summarize some statistics of the solutions.

	Mean NEU (m,m,m)	Standard deviation NEU (mm,mm,mm)
DDBase	(18.898,-1.405,-2.499)	(370,2671,205)
JAC "node algorithm data"	(18.718,-2.384,-2.593)	(5.47,2.91,10.82)
JAC-MM "node algorithm data"	(18.718,-2.385,-2.596)	(3.25,2.29,7.04)
JAC "typical data"	(18.727,-2.372,-2.580)	(4.18,3.13,5.96)
JAC-MM "typical data"	(18.727,-2.372,-2.580)	(2.92,2.60,3.72)

Table 4.10: Statistics of intermittent poor quality data

	CEP (<i>mm</i>)	2DRMS (<i>mm</i>)	SEP (<i>mm</i>)
JAC (average)	5	11	8
JAC-MM (average)	3	8	6
DDBase	1800	5400	1700

Table 4.11: Solution precision using intermittent poor quality data

As can be seen only JAC was able to obtain solutions of any real value. JAC's measures of precision are approximately double of that of the high quality Te Papa Wellington Airport data. Due to the small sample sizes the sample standard deviations of JAC when transmitting "typical data" compared to "node algorithm data" are different than one would expect. The higher epoch loss incurred when "typical data" was transmitted between the node and the base station had no noticeable effect on being able to obtain valid solutions than when "node algorithm data" was transmitted.

4.2.5 Solution comparison summary

A qualitative summary is pictured in the following figure.

Figure 4.7: Qualitative GNSS software solution results

CHAPTER 4. ALGORITHM COMPARISON

All three programs obtained comparable precisions when high quality GPS data over lossless radio links that were permanently powered were used, while, at the same time produced significantly different biases with respect to each other ranging from 16 mm to 20 mm. Both DDBase and RTKLib were extremely sensitive to missing satellite observations while JAC was insensitive to such events. Due to JAC's insensitivity to missing satellite observations and small sample sizes, no noticeable improvement in precision was detected when transmitting "node algorithm data" compared to transmitting "typical data". As expected JAC's solutions when using our low end hardware in a high multipath environment were worse than when using high end equipment in a low multipath environment. Only JAC and JAC-MM could produce solutions of any meaningful accuracy when using our GPS WSN nodes. Therefore, as JAC-MM is an implementation of our CFFS algorithm, the CFFS algorithm is suited for our application of land deformation monitoring using our hardware.

While accuracy may not be entirely known, it is possible to measure small sudden movements of just a few millimeters due to the high precision of our system.

Chapter 5

Hardware

Figure 5.1 below shows a block diagram of the connected hardware components needed for monitoring a site.

Figure 5.1: Hardware components and connections

As can be seen from the figure there are many components required for monitoring a site. Both the base station and the main computer connects to a

public server on the Internet so as to allow both the base station and the main computer to not have public Internet addresses while still being able to communicate with one another. The task of the base station is to gather the broadcast “node algorithm data” from the WSN in addition to creating its own “node algorithm data” from a physically attached GPS receiver. This data is then removed of the time ambiguities and passed on to the server so as to allow the main computer access to this data. The task of the main computer is to take this data along with downloading of broadcast navigation data and implement the CFFS algorithm to calculate position solutions. The tasks of the nodes are to harvest solar energy, control their energy usage so as to allow as many satellite observations as possible, implement the node algorithm, and broadcast their “node algorithm data”. The task of the repeaters is to rebroadcast broadcast data so as to allow larger WSNs and WSNs with poor line of sight characteristics to be formed.

5.1 Node

The node hardware consists of two sections, a power supply section and a host section. This can be seen in Figure 5.2.

Figure 5.2: Node block diagram

The *Power Management Block* (PMB) is responsible for storing solar energy in a super capacitor and delivering it when needed to the host. The PMB

CHAPTER 5. HARDWARE

makes sure the super capacitor does not overcharge, manages current needs by information it is supplied by the host, ensures there is enough energy before supplying any power to the host and informs the host how much energy there is at any one time.

The host consists of three sections a *MicroProcessor Unit* (MPU), an 802.15.4 radio (the radio), and a GPS receiver. The host section provides the brains of the operation as well as the communication to the outside world.

The MPU is partly responsible for energy management along with the power management block. The MPU has the ability to turn off and on the radio as well as the GPS receiver to conserve energy.

The GPS receiver passes raw and processed data from satellite observations to the microprocessor so as to be able to implement the node algorithm on the node itself.

The radio and the GPS receiver were off the shelf modules while we designed the PMB specifically for our requirements. The GPS receivers we used were Ublox LEA6T modules while the radios we used were MRF24J40MB. One node used a passive helical antenna while the rest used 25 mm x 25 mm active patch antennas. The approximate cost of a node was \$100 USD.

The following four Figures 5.3, 5.4, 5.5 and 5.6 detail the *Printed Circuit Board* (PCB) layouts and the schematics for the node while Table 5.1 lists the node's *Direct Current* (DC) operating characteristics.

Figure 5.3: Host PCB (NB. radio marked as MRF24J40MC on PCB but MRF24J40MB used instead for node)

(a) Top side

(b) Top side

Figure 5.4: PMB PCB

CHAPTER 5. HARDWARE

Figure 5.5: Host schematic

Figure 5.6: PMB schematic

CHAPTER 5. HARDWARE

DC specifications		$T_A = -25^{\circ}\text{C}$ to $+70^{\circ}\text{C}$				
Parameter	Sym.	Min.	Typ.	Max.	Units	Conditions
Power						
Quiescent power	P_Q		130		μW	Preventing powerlines from failing at 2V
Active power	P_A		240		mW	At 2V
Peak current use	P_M			350	mA	
Startup Voltage	V_1	1.6			V	
Nominal operating voltage	V_2	0.9		2.3	V	
Absolute maximum voltage	V_3			2.3	V	
Maximum current clamping	I_3		400	550	mA	Requires MPU to be programmed correctly above 400 mA

Table 5.1: DC node specifications

The following three figures show populated versions of the printed circuit boards.

Figure 5.7: Populated PMB v2.0

Figure 5.8: Populated host PCB under test

Figure 5.9: Populated host PCB being inserted into a node enclosure

5.1.1 Node hardware design requirements

The requirements for node were that it be low cost, no battery, solar powered, compact enough to be held in one hand, and operate even on dark cloudy days. Due to the small form factor, energy consumption and power supply, along with cost, were the driving factors to influence design.

The 802.15.4 radio was chosen because of being relatively long range while being cheaper, lower power and simpler than other standards such as Wifi. The MPU was a nanopower item from microchip and was chosen due to its low power consumption and cheap cost. The GPS receiver is the most expensive item and uses the most power of any component in the node (in the order of 100 mW); this kind of energy consumption is typical of small devices containing GPS receivers. To a first order approximation the GPS receiver can be thought of as the only device that uses energy in the node and the sole energy cost of the node. Because of the desire that the nodes could operate on dark cloudy days meant the GPS receiver had to be able to be put in some kind of off state or extremely low power state while still retaining ephemerides data or at least be able to be injected with ephemerides data from the MPU when switched back on again. As mentioned before in the algorithm chapter 2, the low power requirement and the cost ment a singleband GPS receiver was the obvious choice. In addition the GPS receivers had to allow for access to raw observation data. While initially experimenting with SirfIV, we found that SirfIV chips would not reliably switch between various power states and we were unsuccessful with injecting them with ephemerides. Therefore we settled on the Ublox LEA6T and these proved to be a user friendly singleband GPS receiver to use allowing a variety of power states, ephemerides retention, and being able to injected with ephemerides. After the size of the housing was settled on, we chose polycrystalline solar panels to fill as much of the top of the housing as we could. Polycrystalline and/or monocrystalline solar panels were the only sensible decisions to make when deciding what type of solar panels to use due to cost and efficiency with usual daylight conditions. Amorphous solar panels although more efficient at low light levels, would never manage to power our GPS receiver on such a small surface area. The PMB was then designed to link solar panels and the host section together.

5.1.2 PMB

A simplified version of the block diagram of the PMB is pictured in the figure below. The initial goal of the PMB design was to minimize quiescent power for times of low power demands so as to be able to store charge on the super capacitor even at times of low light levels.

Figure 5.10: PMB block diagram

The PMB has no *Maximum Power Point Tracking* (MPPT), the energy from the solar panel is simply fed into the super capacitor via clamping circuitry to avoid the voltage over the capacitor going too high. Upon the initial application of power to the super capacitor the PMB is in an “off” state. The boost converter is off and two voltage monitors are connected to the solar power. One voltage monitor set to 2.3 V to avoid overvoltage on the super capacitor in addition to enabling the boost converter, while another voltage monitor is set to 1.7 V which when triggered enables the boost converter. The 1.7 V voltage monitor once triggered will enable the boost converter and filling of the the local reservoir is started. This filling of the local reservoir causes the 1.7 V voltage monitor to be disconnected, the capacitor voltage line will be connected to the super capacitor so as to allow the host to monitor the super capacitor voltage and vdd will be 3.3 V.

When the host initially boots up it will be due to the 1.7 V voltage monitor which shortly get disconnected due to vdd going high thus turning off the boost converter. This means that the host has to take immediate action as it is not using the super capacitor reservoir and instead is just using the local

reservoir which is tiny in comparison and will not be entirely full. Either, the host must enter an extremely low power state waking up occasionally to enable the boost converter to replenish the local reservoir, else it must enable the boost converter to keep the local reservoir full. Either way it has to take the responsibility of keeping its vdd powerline from collapsing.

During normal times of operation when the host is sleeping and the boost converter is disabled quiescent power is extremely low. This allows the solar panel to deposit charge on the super capacitor. When the host wishes to turn on either the GPS or the radio it must keep the boost enable line active. Generally the host should maintain a super capacitor voltage of between 1.4 and 2.2V.

5.1.3 Node quiescent power

The quiescent power of the node we define as the average power used by the node at the nominal voltage of 2 V to only keep the power lines from failing while the node tries to use as little energy as possible. This means the host periodically refills the local reservoir but nothing else. The host needs to remain in a sleeping state occasionally waking itself up to refill the local reservoir while all it's peripherals that it can control remain switched off.

To measure the quiescent power of the node the solar panel was removed and the voltage over it was measured over time. The host was programmed to only keep the local reservoir full enough so the power lines did not cause a brown out to the host. Due to the hardware design the node was unable to remove the backup power supply to the GPS receiver which used 70 μ W according to the data sheet and 35 μ W according to measurements we performed while it was sleeping. The node was left for 34740 seconds (about nine hours) and a 33 F capacitor was used. The voltage initially was measured at 1.994 V and finally at 1.924 V, this equates to a quiescent power of 130 μ W. The same test was repeated for a 10 F capacitor from a voltage of 1.993 V to 1.926 V and a value of 90 μ W was measured. In both cases the super capacitor was left for one hour before the test was started. Due to capacitor properties such as self healing, temperature dependence, and memory effects these values can only be approximate.

Making the assumption that capacitor self discharge is proportional to capacitor size, we see that there is a 73 μW base power while the rest of the quiescent power is used for the capacitor self discharge. Therefore the quiescent power with respect to super capacitor size can be written as follows, where C_S is super capacitor size in farads and P_Q is quiescent power in microwatts.

$$P_Q = 1.739C_S + 73$$

So, approximately half the quiescent power goes to the super capacitor self discharge when using the 33 F capacitor while for the 10 F capacitor approximately 20% goes to the self discharge of the super capacitor. We see a slight effect due to the different capacitors but both 90 μW or 130 μW are acceptably small that either capacitor could be used.

5.1.4 Peripheral power use

The peripherals are the GPS receiver and the radio. As the boost regulator must be on when using the peripherals, the voltage supplied to the peripherals is 3.3 V regulated. The power usage of the two peripherals as obtained from the data sheets are given in the following tables.

	Power (mW)
RX mode	83
TX mode	495
Sleep	0.02

(a) Radio

	Power (mW)
Active (typ)	123
Backup power	0.07

(b) GPS receiver

Table 5.2: Peripheral power usage

As energy consumption is power times time, the radio can use a insignificant amount of energy compared to that of the GPS receiver. Generally the nodes

do not go into the RX mode and are entirely disconnected altogether using near zero power. The period of time that a node turns on its radio to transmit an epoch packet is in the 2 or 3 ms. A large part of this time is not in the TX mode but rather in some other mode while the oscillators of the radio stabilize. For a worse case scenario of 3 ms for packet using 495 mW sent once a second equates to 1.5 mW on average. At the same time the GPS receiver needs to be on continuously to supply the radio epoch packet and uses around 123 mW of power; two orders of magnitude greater than the power required by the receiver. Therefore, to a first approximation only the GPS receiver uses energy when the receiver is active and we can ignore the radio's energy consumption.

When the GPS receiver is inactive the radio is disconnected and the GPS receiver is in a backup power mode using 70 μ W. However, measurements generally showed a lower value around 30 μ W. For some tests we loaded the GPS receiver with ephemerides and time directly avoiding the need for the GPS receiver to obtain such items from the satellite; this procedure is called injection. For tests where ephemeris injections were used the hardware was modified such that the GPS receiver could be entirely disconnected or put into the backup power mode.

5.1.5 Active node power use

The node's purpose is to obtain observations and send them back to the base station. Therefore we define the active node power use as the approximate average power at the nominal voltage of 2 V used by the node during times of taking observations and sending them every second. This is energy used by all devices of the node, including things such as the boost converter in the PMB, capacitor leakage, MPU, radio, GPS receivers, and GPS *Low Noise Amplifier* (LNA) to name but a few.

To measure active power we removed the solar panel and connected a 2.03 V power supply to the PMB. The node was kept inside and had no clear view of the sky. The GPS receiver had no almanac and no ephemerides. The active node power use was measured at 240 mW; approximately double that as given for the GPS receiver in the data sheet.

5.1.6 Solar power supply

As the goal of the node is to have the GPS receiver on for as much time as possible taking observations and returning them to the base station, we have a situation where most likely a mismatched load is attached to a solar panel wanting to use around 240 mW. The load would be matched if the *Maximum Power Point* (MPP) voltage of the solar panel was around 2 V as this is the voltage nominally on the super capacitor. Therefore, if we want to choose a solar panel that is reasonably effective at powering the node at low light levels it makes sense to choose a solar panel such that the MPP voltage is close to 2 V when light levels are low. We now wish to determine what we should classify as a low light level and what is the irradiance of this low light level .

Solar panel typical voltages are typically given for direct sunlight conditions and an irradiance of 1000 Wm^{-2} [9]. The MPP voltage of the solar panel decreases when irradiance decreases, therefore, a solar panel with a rated typical voltage of more than 2 V makes sense. While 1000 Wm^{-2} may be obtained in summer, even on clear days in winter depending on the location irradiance can be more than half of this. Even at times of year where the theoretical maximum light levels can be large, from day to day light levels can alter dramatically. The following figure shows measurements taken using a Fine Offset WS1080 *Personal Weather Station* (PWS) with Cumulus for our site at Paekakariki New Zealand in late spring. The pink plot shows the theoretical maximum solar irradiance, while the yellow plot shows the actual measured solar irradiance. As can be seen, on the 17th the irradiance was a fraction of the theoretical maximum; this was a very dark day.

Figure 5.11: Variability of irradiance

From this figure and being conservative we take 20 Wm^{-2} as the minimum irradiance that we consider a node needs to work at.

A solar panel can basically be modeled as a constant current source over a diode; this is also how the program Simulink models it. The following equation shows the ubiquitous way of modeling the current obtained from a solar panel using just one diode and consists of the superposition of the photocurrent and the dark current.

$$I = I_{PH} - I_0 [\exp(\lambda(V + IR_S)) - 1] - \frac{V + IR_S}{R_{SH}} \quad (5.1)$$

$$\lambda = \frac{q}{nNkT} \quad (5.2)$$

To a first approximation assuming the series resistance of the solar panel is small the short circuit current is approximately equal to the photo current. The photo current in turn is proportional to the irradiance. Therefore the short circuit current is approximately proportional to the irradiance $I_{SC} \propto I_r$. This allows an easy way of measuring irradiance once the proportionality constant is determined using measurements like the ones in the previous figure. In fact due to the flat shape of the IV curve (current versus voltage curve) for low voltages, this proportionality somewhat holds for voltage is less than the

MPP voltage which is where we plan to generally operate the solar panel at. Generally the super capacitor is about 2 V which in turn means we expect to obtain somewhat of a linear relationship between power obtained from the solar panel and irradiance. In addition at an irradiance of 20 Wm^{-2} we would then expect that we could only obtain 2% of the power we could have obtained at an irradiance of 1000 Wm^{-2} .

The solar panel modeling equation as given is an implicit function that can be made explicit using the Lambert W function but generally it is just left in its implicit form. Due to the hassle of dealing with such a function it easier to use a program like Simulink to deal with it.

A simple Simulink solar panel simulation was put together to measure the effect of a mismatched load of the PMB on solar panels. The following figure depicts the schematic of the simulation in Simulink.

Figure 5.12: Simulink solar panel simulation

Two polycrystalline solar panels connected in parallel that had a rating of 5 V were investigated. The short circuit current, the open circuit voltage, and irradiance were measured. These numbers along with the number of solar cells connected in series and diode quality factor were inputted into the Simulink

simulation. The voltage of the MPP voltage was then plotted with respect to irradiance; the result of which is shown in the following figure.

Figure 5.13: MPP voltage simulation for 5V solar panel

As can be seen the voltage of the maximum power point falls with a reduction of irradiance. This figure shows that at 20 Wm^{-2} the simulation says that the maximum power point voltage should occur at 3.28 V. This is still considerably higher than the 2 V that we would be wanting. Therefore, one would expect a 3.5 V or 4 V solar panel to be more suited.

The solar panel to PMB efficiency is defined as the amount of power being delivered to the super capacitor at 2 V compared to the power delivered to a load matched to the solar panel. This efficiency was then plotted with respect to irradiance and is pictured in the following figure. This figure essentially shows the amount of power going on to the super capacitor that could possibly be extracted from the solar panel .

Figure 5.14: Solar panel to PMB efficiency for 5 V solar panel simulation

As can be seen at low light levels the matching of the super capacitor to the solar panel improves as wanted to offset the reduction of power due to low light levels. However, from the plot we do not see it go through a local maximum as it should have if we had used a 3.5 or 4 V solar panel. However, due to already having 5 V solar panels that almost completely covered our housings, we decided to use the 5 V ones we already had and accept the loss of efficiency.

We compared the energy obtainable from the solar panels with respect to irradiance using the simulation. Figure 5.15 that follows shows the simulation of MPP power and the power being delivered to the PMB; in addition line fits have been added for visualization. Due to the lack of a battery, power to the PMB greater than the 240 mW active node power is not particularly useful as it can't be used. Therefore in the figure we only show power levels lower than 240 mW.

Figure 5.15: Power from 5V solar panel simulation

As can be seen, according to the simulation the solar panel is large enough that at an irradiance above $150 Wm^{-2}$ there should be enough power that the solar panel can keep the node continuously active gathering GPS observations if the PMB was perfectly matched to the solar panel. However, due to the mismatch of the solar panel and the super capacitor, the simulation says that an irradiance of $275 Wm^{-2}$ is needed to keep the node continuously active gathering GPS observations. Levels below this irradiance the GPS receiver must be duty cycled appropriately, where at an irradiance of $20 Wm^{-2}$ a duty cycle of 7% is needed. To do away with duty cycling altogether above $20 Wm^{-2}$ larger solar panels could be used, however, this would require around 13 times more solar panel area which would be unacceptable.

Finally, on a overcast afternoon when the irradiance fluctuated approximately around 50 to $150 Wm^{-2}$ a node with a 33 F capacitor was tested to determine its duty cycle and hence the accuracy both the simulated plot in the previous figure and the active node power use in real life. One cycle of activity was measured from discharging the super capacitor from 2.2 V to 1.4 V while GPS measurements were taken, and then a period of quiescence while the super capacitor voltage once again rose to 2.2 V. In all this took 800 seconds while the node was able to obtain GPS measurements for 216 seconds of these, thus

equating to a duty cycle of approximately 30%. As we have already seen the measured approximate active node power is 240 mW , in comparison the quiescent power of the node can be ignored. As the node is only active for 30% of the time this means the node must have been using approximately 70 mW . If we then look at the previous figure we see that 70 mW should equate to an irradiance of approximately 80 Wm^{-2} which is consistent with the the irradiance measurements we've made.

5.1.6.1 Dark current and blocking diode

Dark current I_{Dark} is the current that flows through the the solar panel due to an applied external voltage when no light is applied to the solar panel.

$$I_{Dark} = I_0 [\exp(\lambda(V + IR_S)) - 1] + \frac{V + IR_S}{R_{SH}} \quad (5.3)$$

Generally a blocking diode such as the ubiquitous 1N4148 would be added to the solar panel to reduce this current to near zero. Such a blocking diode reduces the MPP voltage by approximately 0.6 V which would only bring our nominal voltage of 2 V closer to that of the MPP voltage thus improving our choice of solar panel. However, as our 2 V is considerably lower than the MPP voltage even at very low light levels, we have the possibility of not using the blocking diode at all. According to Simulink at zero irradiance and 2 V, only $260\text{ }\mu\text{W}$ should flow into the 5V solar panel, while measuments taken showed $200\text{ }\mu\text{W}$ flowed into the solar panel which would equate to around 66 hours before a 33 F capacitor would drop from 2.2 V to 1.4 V ignoring self discharge.

For applications that require the nodes to function continuously at night and hosts that use very little energy it makes sense to use a blocking diode as this would considerably reduce the quiescent power used by the node and the solar panel at night. For applications that do not require the nodes to function at night or ones where the host uses considerable energy it makes sense not to use a blocking diode as this would increase the power delivered to the node from the solar panel while the $200\text{ }\mu\text{W}$ caused by the dark current of the solar panel would be insignificant compared to that of the energy used by the node. Due to our GPS receiver peripheral of the host our application falls into the latter and hence we choose not to use a blocking diode.

5.1.6.2 Solar panel supply summary

Ideally a 3.5 or 4 V solar panel may have proved to be a better decision due to better matching between the solar panel and the PMB but a 5 V one was chosen instead. The advantages of the 5 V one was availability, physical size matching the housing, guarantee that the solar to PMB power mismatch would improve with a reduction of irradiance and no need for a blocking diode.

Ideally a solar panel large enough that duty cycling would not be needed even at an irradiance of 20 Wm^{-2} would have been preferred but this would have required 13 times more solar panels than we deemed to be acceptable. Instead smaller solar panels were used requiring duty cycling to start at 275 Wm^{-2} .

5.1.7 Super capacitor sizing

Super capacitor sizing depends on the application. If the node is to function at night then it makes sense for the super capacitors be as large as possible to facilitate as many observations at night as possible while there is no sun. However, if the node is to function only during daylight hours and the role of the super capacitor is solely to allow duty cycling during the daylight hours then it's size is less critical as a smaller one is possible.

Super capacitors up to 5000 F can currently be easily obtained at a cost of a couple hundred US dollars (digkey ESHSP-5000C0-002R7) and are currently about as large as they come. A GPS receiver using 240 mW would drop the capacitor voltage from 2.2 V to 1.4 V within about eight hours or say approximately the length of the night and is therefore approximately the minimum capacitor size needed for continuous night operation. If continuous night and continuous day operation was required, approximately 500 mW on average would be required every day during daylight hours. As even a 5000 F super capacitor is unable to store enough charge for more than one nights continuous operation, even one day where the average daily power during daylight hours is not greater than 500 mW would require duty cycling. If this 500 mW was to be obtained even on days of light levels as low as 20 Wm^{-2} this would require a monocrystalline/polycrystalline solar panel roughly half a meter by half a meter. This is approximately 26 times the area of our node hardware housing and unacceptable. Therefore, duty cycling algorithms would be inevitable.

CHAPTER 5. HARDWARE

These algorithms would need to carefully decide who gets the limited energy from the solar panel and how should the energy that has been stored on the super capacitor be used such that the voltage on it does not drop below 1.4 V all such that the observations are as evenly distributed as possible throughout the 24-hour day period. Due to the added complexity and cost we chose to focus on using the super capacitor as a method of allowing duty cycling during daylight hours rather than obtaining observations at night thus making the super capacitor size less critical.

No matter how GPS receivers are initialized there is always some sort of initialization time or *Time To First Fix* (TTFF) after they are turned on before useful observations are obtained. The GPS receiver has to acquire and track the satellites and then resolve the code observable ambiguity. For the Ublox LEA6T they quote what they call acquisition time of between 1 and 26 seconds depending on many factors such as whether or not ephemerides have been retained, what timing information the GPS receiver has, how long has it been off for, and so forth. We assume that this value of between 1 and 26 seconds excludes the time required before the code observable ambiguity has been resolved as this can be as much as 6 seconds as shown in the code observable subsection A.4.1. From our experiments using various GPS receivers it is even possible for many minutes to pass before an observation is returned. This initialization energy is energy spent not usefully obtaining observations. Therefore duty cycles that have long periods can help mitigate the total energy going towards continuously initializing the GPS receiver every duty cycle period. Duty cycle periods can be increased using larger super capacitors. The approximate wasted percentage of on time can be calculated as follows where t_{init} is TTFF, P_{GPS} power of the GPS receiver, D Duty cycle, C capacitance, and v_i and v_f the initial and final voltages over the super capacitor respectively.

$$W_{on} = \frac{t_{init}P_{GPS}(1-D)}{0.5C(v_i^2 - v_f^2)} \quad (5.4)$$

This wasted percentage of on time is plotted in the following figure for both the 10 F and 33 F capacitors with respect to duty cycle for an initialization time of 42 seconds. A value 42 seconds was used as this was the value as measured by us as the TTFF for the GPS receiver without backup power (see Appendix C.2).

Figure 5.16: Wasted percentage of on time due to 42 seconds of GPS receiver initialization with respect to duty cycle

As can be seen using larger capacitors reduce this wasted time and the waste becomes greater when using low duty cycles. Using larger capacitors causes the observations to become more clumpy as well as a general trend in greater self discharge of the super capacitors neither of which we want. However, compared to GPS power consumption the quiescent power was insignificant for both the 10 F capacitor and 33 F capacitor measurements and both quiescent powers were reasonably similar. The effect of observation clumpyness on position solutions was not investigated so we ignore this the effect that clumsiness may have on solutions.

TTFF for an autonomous GPS receiver can be somewhat unpredictable; there is no guarantee that the GPS receiver will ever acquire the satellite. From our experiments keeping the backup mode of the Ublox LEA6T active continuously generally causes TTFF to become more consistent and shorter while using minimal power; values of 10 seconds are typical (see Appendix C.3). As the wasted percentage of on time is also proportional to TTFF it makes sense to at least want a very short TTFF that is consistent. A 10 second TTFF produces a wasted percentage of on time approximately 4 times less than what is seen in the previous figure thus reducing the benefits of using a larger capacitor

CHAPTER 5. HARDWARE

From our experiments using ephemeris and time injections it is possible to produce initialization times that are short and consistent, using this method of initialization, 5 seconds is a very typical time before useful observations are obtained C.4. A 5 second TTFF produces a wasted percentage of on time approximately 8 times less than what is seen in the previous figure thus reducing the benefits of using a larger capacitor. However, ephemeris and time injections add to the complexity of a system.

So weighing all things up a 33 F capacitor was used, this allowed a reasonable amount of flexibility when testing types systems. It allowed for a reasonably small wasted percentage of on time due to initialization even without the use of ephemeris and time injections by permanently supplying GPS backup power instead; generally less than 5% compared to less than 20% for a 10 F capacitor. While its purpose was mainly for duty cycling during daylight hours using a simple system where the GPS receiver retained ephemerides and time using the GPS backup power, it also allowed the ability to investigate creating algorithms to take observations at night, as well as investigating ephemeris and time injections to reduce TTFF.

5.1.8 Precise solutions at 20 Wm^{-2}

From simulations, the minimum target irradiance of 20 Wm^{-2} requires 7% duty cycling of the GPS receiver. This duty cycling in turn causes approximately a 5% wasted time due to GPS initialisation when using a 33 F capacitor. In winter, say the daylight length might be 7.5 hours when the irradiance is 20 Wm^{-2} for the 7.5 hours, and 0 Wm^{-2} for the rest of the 24 hours. This would equate to a receiver obtaining observations for approximately half an hour a day in total. Depending how this is distributed, compared to normal permanently deployed GPS receivers this most likely would be an extremely small period of time to obtain a precise location. This in turn puts a great pressure on the solution algorithm. Can a GPS receiver even obtain a solution in such a situation? There is only any need to power a GPS receiver if it can obtain enough observations that are distributed in such a way that a solution can be calculated, if it can't there is no point in turning it on in the first place. However, we will see in Chapter 6 that solutions with a precision of 8 mm 2DRMS can still be calculated in high multipath environments when days

with as little as half an hour of GPS on time per day happen. This means that solutions are still obtainable even when an irradiance of 20 Wm^{-2} is the average irradiance in a day where the daylight length is 7.5 hours.

5.1.9 GPS antenna

All but one of the nodes created used 25 mm x 25 mm active patch antennas of the model VTGPSIA-3 from V.TECH [2], the remaining one used a 20 mm passive helical antenna. Initially we bought one helical antenna but the supplier then stopped selling them so we switched to the active patch antennas when we needed to create more nodes. In retrospect using active patch antennas may not have been a good idea as the waterproofing of the housing became trickier and we suspect that node failures may have been water leaking into the patch antenna LNA circuitry.

5.1.10 Housing

The housing of the nodes was accomplished using 190 mm x 110 mm x 61 mm enclosures. Various methods of waterproofing were tried with varying success. Spraying the boards with protective PCB lacquer was not found successful and PCBs corroded. Encapsulating the PC boards in aroldite was more successful and we did not see any corrosion of boards using this method. Due to various ways of mounting the housings and the somewhat sealed nature of the enclosures, water could build up inside the housings filling up the enclosure until short circuiting of exposed programming pins occurred and node failure would happen. The top of the enclosures were either coated with a clear protective coating or *Room Temperature Vulcanization silicone* (RTV) would be applied around objects to try and prevent water entering the enclosure. Of the various enclosure waterproofing methods we attempted, none were successful at keeping the water out. When using the active patch antennas, the patch antennas were placed on top of the enclosures and sealants would be used to attempt to stop the water entering the LNA section that sits just below the antenna itself. Due to our difficulty in the sealing the entire enclosure we suspect that water could still mostly likely get into the LNA section of the antenna. The passive

helical patch antenna was mounted through a hole in the enclosure such that water could drain away from the antenna.

Figure 5.17: Node with helical antenna in housing at the Paekakariki New Zealand testbed.

5.2 Base station

The base station consisted of a Raspberry Pi Computer, a WSN gateway of our own design, a Ublox LEA6T with a USB interface, a TL-MR3020 router by TP-LINK, a 3G USB modem, and control relay to reset the 3G modem upon internet connection failure. Two 190 mm x 110 mm x 61 mm enclosures were used, one for the WSN gateway and the GPS receiver, and the other one for the rest of the electronics excluding the 3G modem. A small fan was used to keep the electronics cool that housed the Raspberry Pi.

Figure 5.18: Base station excluding GPS and 802.15.4 hardware box before being shipped to Taiwan

No consideration was made for power usage as the base station was designed to use a permanent power source.

Figure 5.19: GPS and 802.15.4 hardware box

The WSN Gateway of our own design is pictured in the following picture and is housed in the GPS and 802.15.4 enclosure as seen in the previous Figure 5.19. While it was our own design, any 802.15.4 radio that could be connected to a computer could be used; we just so happened to have this board already made. It's design is trivial so we do not detail its design. Figure 5.21 shows the Ublox-LEA-6T that we used for the GPS and is also housed in the previous Figure 5.19. The other aspects of the hardware design of the base station were also trivial as all these products can simply be bought and plugged together; no soldering required. To see how the components are connected together within the base station enclosure see Figure 5.1 (the base station computer is the Raspberry Pi).

Figure 5.20: WSN Gateway

Figure 5.21: GPS receiver module used in the base station

5.3 Repeater

The repeater contained similar hardware to that of the node and used the same PCB as the node. No GPS receiver or GPS receiver antenna were included. A MRF24J40MC was used instead of a MRF24J40MB which are identical except the MRF24J40MC requires an external antenna. An external whip antenna for the MRF24J40MC was placed where the GPS receiver antenna was in the node hardware.

Figure 5.22: Repeater in housing in Taiwan before being deployed at the LuShan in Taiwan

2.4 GHz radio frequencies as used by 802.15.4 radios do not travel well without line of sight links. When testing the range of the MRF24J40MB radios as used in the nodes we were able to get 1.76km using a receiver with a Yagi even without a perfect line of sight link.

Figure 5.23: 1.76km test using MRF24J40MB radio and receiver with Yagi

However, we discovered from the LuShan testbed that due to very poor line of sight and dense vegetation, a communication range of even 100 m was difficult at a genuine site of interest such as the LuShan testbed. As the LuShan testbed was approximately 1 km² this required designing a method of increasing range using a multihop network architecture. As time was a pressing issue we modified the node hardware to create repeaters that would function as dedicated repeaters for the nodes.

5.4 Remaining hardware

The remaining hardware such as depicted in figure 5.1 were standard computers. The server that logged and distributed the observations from the nodes was an embedded Panda board, while the computer that calculated solutions was a standard desktop PC. The panda board Internet connection was a ca-

CHAPTER 5. HARDWARE

ble broadband link. All these components were permanently powered via the mains.

Chapter 6

Testbed results and analysis

6.1 Paekakariki testbed

The Paekakariki testbed shown in the figure below. Nodes were placed relatively close to one another as depicted by the mauve circle. In addition permanently powered GPS receiver nodes were placed on the roof of the building at the bottom right hand corner of the figure. The left of the figure is the north direction.

Figure 6.1: Paekakariki testbed site

As can be seen there is poor visibility of the sky the Paekakariki testbed for the nodes on the ground while the receivers on the roof have somewhat better visibility. Observations obtained from this testbed and results using them are used throughout this thesis.

First we look into the number of hours the GPS receivers tend to be active for in a day and how this varies. The figure below shows the number of hours node 272's GPS was active for per day and successfully returned epochs when placed at the Paekakariki test site for approximately two weeks in summer and then in winter.

Figure 6.2: Number of hours a day GPS a receiver is active for

In winter we observed a minimum of less than 0.5 hours, while in summer we observed a maximum of more than 8 hours. On average in winter there was 1.7 hours of activity while in summer there was 6.2 hours of activity; this means 3.6 times as many epoch are used for processing summer solutions than winter ones. The horizontal solutions using JAC-MM to implement the CFFS algorithm are pictured in the following figure.

Figure 6.3: Qualitative summer winter solution precision comparison

As can be seen there appears to be better precision during summer than winter as one would expect as there are 3.6 times as many epochs used for the summer solutions than the winter ones.

	2DRMS (mm)
JAC-MM (Summer)	3
JAC-MM (Winter)	8

Table 6.1: Solution precision for summer winter comparison test

The picture below shows a deployment at the Paekakariki testbed site run in summer for one month where north is in the up direction of the picture and the nodes are placed directly on the ground; these three nodes can just be seen in figure 6.1. As can be seen in this figure a shadow of a tree is passing the nodes.

Figure 6.4: Three nodes at the testbed site

The measurements to the approximate physical horizontal center of each of the antennas were measured using a tape measure with an accuracy of approximately 1 cm. The results of these measurements is pictured in the following figure.

Measured with tape measure
±1cm accuracy

Figure 6.5: Tape measure measurements of the nodes to the approximate physical center of the top of the antennas

After one month the daily solutions were processed, and the horizontal solutions all plotted in the figure below; the mean range between each node is also

pictured in the following figure with 95% uncertainties as derived from the horizontal components of the CFFS solutions.

Figure 6.6: Birds eye view of CFFS solutions for one month Paekakariki static test

As the phase center of each of the three antennas were not known as they were uncalibrated antennas, the difference between the relative distance measurements using the tape measure between the approximate approximate physical centers of the antennas and the difference between the mean values of the horizontal CFFS solutions, one should expect these to only be approximate. However, despite this, the difference in these range values are still consistent with the ones measured using the tape measure.

The average 2DRMS precision calculated for the three nodes for this one month static test was 5 mm and appears typical for solution precision for high multipath environments.

6.2 LuShan testbed

The LuShan site is a known high risk landslide site on a hill above the town of Lushan in Taiwan's mountainous Nantou region. It is a site studied using

CHAPTER 6. TESTBED RESULTS AND ANALYSIS

satellite imagery, aerial imagery, laser survey and GPS. The active movement zone approximately covers 1 km² and resembles the shape of an upsidedown V starting at the top of the hill.

A testbed of 4 small batteryless GPS wireless sensor nodes were deployed at the site. The nodes were deployed alongside retro reflectors used for laser based land deformation monitoring using a total station by *National Chi Nan University* (NCNU).

Due to dense and tall vegetation, both network coverage and adequate visibility of GPS satellites were problematic. For adequate network coverage two batteryless solar powered radio repeaters were used. One placed on the hill of interest while the other on the opposite hill. For adequate visibility of GPS satellites, node sites were found by visual inspection and deciding whether or not there was enough of a clear view of the sky. Node 274 was placed at a knowingly stationary location at the bottom of the hill as a point of reference. The other three nodes or placed on active movement sites.

A permanently powered base station (also known as the router) that functioned as a sink for the data obtained by the nodes, a router to route the data via a 3G modem to us via the Internet for solution processing, and a GPS node itself to obtain satellite observations was placed on the hill called node 1. The base station was mounted in a large pre-existing weatherproof metal cabinet box with mains outlet sockets inside.

The figure below shows the locations of the nodes, repeaters and the base station; the size of the testbed is approximately 300m by 700m.

Figure 6.7: LuShan testbed site (Courtesy of NCNU)

The direction of land slippage on the hill of interest is in a direction approximately southeast. It is uncertain whether repeater 104 was situated on stationary land or whether or not it too is sliding, as repeater 104 was not on the hill of interest. Node 274 was the only known stationary location. Node 276 and 277 were on the very west side of the sliding area while node 275 was on the very east of the sliding area. The base station which was permanently powered also functioned as a GPS receiver node called node 1. Node 1 was also on the westside of the sliding area.

L1 GPS phase observables were the principal observations used to calculate relative node positions. These phase observables along with autonomous solutions calculated by the GPS receiver itself were combined to modify the phase

CHAPTER 6. TESTBED RESULTS AND ANALYSIS

observables on the nodes before sending them back to the sink using 2.4Ghz 802.15.4 radios either directly or via a series of repeaters. The sink then passed this data onto us via a 3G Internet connection for processing. Despite the lack of a battery the nodes were designed to operate in extremely low light conditions as expected on dark days during heavy rain, the nodes however would not work at night. Because of this the nodes only returned data during the day.

Temperature measurements using a mock node installation at NCNU near the test site in summer were performed and a maximum temperature of 58.1°C was measured within the node. The minimum rated value of any component inside the node was 70°C so the maximum recorded temperature was within component tolerances.

The first node was installed on 4/7/2014 while the last node installed on the 15/7/2014. The mounting method of the nodes can be seen in the following figure where a metal flange is bolted to the side of one of the walls of the road. The node or repeater is then glued to the top using RTV silicone.

Figure 6.8: Node and repeater mounting method at LuShan

Taiwan is prone to typhoons with a season lasting approximately from June to October. It was hoped that heavy rain due to tropical cyclones might cause land deformation that we could measure. Only two tropical cyclones hit Taiwan during the 2014 typhoon season; a category 2 typhoon called Matmo approximately on 23/7/2014 (day 204) and a tropical storm called Fung-wong approximately on 21/9/2014 (day 264).

Figure 6.9: 2014 tropical cyclones for Taiwan

Node failure was a significant issue. Node 274 ultimately failed on the 11/8/2014, node 275 on 13/8/2014, and node 277 on 21/12/2014. Node 274 and 275 performed normally until they ultimately failed. Node 277 failed during typhoon Matmo and returned no signal for two days after that. Node 277 became increasingly intermittent ultimately the last signal being heard on 20/12/2014 while the second to last signal was heard on the 16/11/2014. Node 276 functioned almost flawlessly and only on three days during tropical cyclone Fung-wong did it fail to transmit any signals.

Both the base station hardware and the repeaters performed flawlessly and not one day was observed where either of them malfunctioned.

The reason for the node failures was presumed to be caused by water entering electronic circuitry and causing corrosion and short circuits. However as we were based in New Zealand we could not easily go back and examine the reason for these node failures. The figure below shows node 275 some time after it failed. Originally the GPS antenna was a shiny silver colour while clearly in the figure below it is been tarnished. The antenna is under RTV silicone which was designed to protect it against the elements. A small amount of brown colour at the bottom right corner of the antenna is also presumably corrosion. Therefore it seems likely that this particular node failed due to water damage.

Figure 6.10: Suspected water related corrosion in node 275 (Circa 28/8/2014, Courtesy of NCNU)

Another suspected issue with node failure was vegetative overgrowth. Below is a picture of node 277 during time when it was intermittently failing.

Figure 6.11: Vegetative overgrowth on node 277 (Circa 28/8/2014, Courtesy of NCNU)

Previous attempts by ourselves to create nodes impervious to water damage proved difficult. At the Paekakariki testbed after six months outside the following picture was obtained upon opening up the GPS receiver node. As can be seen considerable amount of water has entered the housing and has also managed to get in past the conformal coating that covers the PCB. In fact, because of the conformal coating on the PCB the water can't drain away making the situation worse.

Figure 6.12: Corrosion at the Paekakariki testbed

At the LuShan testbed at the beginning of 2014, we sent initial GPS receiver nodes to be tested on the hill. By the time we arrived in late June, upon opening one GPS receiver node up, the housing contained hot water that would not drain away as can be seen in the following figure, figure 6.13. The housing of the new nodes we put on the hill in July had been redesigned and the PC boards inside the nodes themselves were entirely covered in epoxy resin except for the programming pins. The GPS antenna was covered in epoxy. This we hoped would prevent node failure due to water damage. However, we did not have enough time to rigorously do testing beforehand.

Figure 6.13: Water inside an initial testing node at the LuShan site

We were wanting to use node 274 as a fixed reference so as to be able to measure the relative locations to the other nodes. However, as node 274 failed approximately a month after installation this could not be done for most of the time testbed was running. This left for the majority of the time only node 1 and 276 usable. Node 1 was mounted on a metal I-beam that once upon

a time presumably was vertical but due to land deformation was on a slight south tilted slope. Both these nodes were on the active sliding zone meaning over the long term all we could hope to measure as the long term relationship between a wall along a road and the top of an I-beam.

6.2.1 Movements relative to node 274

In the figures that follow we show the NEU time series of the wireless solar powered GPS receivers with respect to node 274. This was done by taking the difference of the final solutions between each node with respect to node 274. When taking the difference between solutions additive noise increases. However, it seems that there is a considerable stochastic dependence between solutions obtained from two distinct nodes with respect to a common reference station, presumably as the similar sunlight conditions causes observations to be taken reasonably similarly between the two nodes. Therefore the precision of the solutions obtain this way is not as bad as one might first anticipate. This method of calculating solutions has the advantage of being able to use a permanently powered reference station that is on an active sliding zone as a reference station such to maximise the probability that there exists an epoch for an observation on a given solar powered GPS receiver, hence more double differences can be used in the final solution. This contrasts to the method by using the two wireless GPS receiver nodes themselves to obtain double differences and hence solve for position; in this case the two GPS receiver nodes must be on at the same time to obtain a double difference which does not necessarily happen. In addition it is far easier subtracting final solutions than creating double difference is between the two GPS receivers themselves.

For the figures that follow, we plot the NEU solutions produced with JAC-MM which implements the CFFS algorithm as described in chapter 2. The time series of all the wireless GPS receiver nodes with respect to node 274 were plotted as we were told that no 274 was on a non-sliding zone. As node 274 failed shortly after installation only approximately one month of solutions are plotted on each of the figures. A black vertical line just before the 204th day as to signify the approximate time observations would be affected by typhoon Matmo. In the 277-274 NEU timeseries plot two outliers were removed by hand as they were significantly inconsistent with the rest of the solutions.

CHAPTER 6. TESTBED RESULTS AND ANALYSIS

Ignoring the first 14 solutions to allow the sidereal filtering to stabilise the average 2DRMS precision was 3.3 mm using this indirect way of obtaining relative solutions relative to node 274.

Figure 6.14: 275-274 NEU timeseries

Figure 6.15: 276-274 NEU timeseries

Figure 6.16: 276-274 NEU timeseries

As can be seen there is no noticeable movement caused due to typhoon Matmo. Any solution aberrations around the few days after the typhoon may be due to the lack of double differences because of cloudy days or swelling of the soil or some other temporary phenomena. All nodes had a southward trend between 0.12 mm/day to 0.22 mm/day. Node 277 had a slightly eastward component while 275 and 276 had very slightly westward components all nodes had an up component of between 0.3 mm per day to 0.7 mm per day. There is no compelling evidence that this very slight trend over such a short period of time is related to any land deformation and is entirely conceivable that it is instead the slow bias variation predicted in simulations in chapter 3 caused due to the four minute sidereal shift with respect to the solar day or the nodes settling over time.

6.2.2 277 and 276 relative to node 1

After day 222 node 274 was never heard of again. Node 277 was heard of less and less, while node 276 still continues today as of writing. Therefore at this point we compare solutions relative to node 1 and in particular concentrate on 276 relative to node 1 as this pair was most consistently active.

CHAPTER 6. TESTBED RESULTS AND ANALYSIS

The figure below shows the NEU timeseries solutions for node 277 relative to node 1 (277-1) . The second black vertical line centered just before day 264 is to signify the time tropical storm Fung-wong might affect solutions.

Figure 6.17: 277-1 NEU timeseries

As can be seen a significant anomaly occurs centered around date 230 and extends at least 10 days to either side of the date. There is a drop in the north direction of about 5 mm while there is a rise in the east direction of approximately the same magnitude at the same time; what this as we do not know. As can be seen there is no significant change around the day tropical storm Fung-wong hit. As with all the solutions shown so far for LuShan, there is a slow approximately linear increase in the up direction; in this case about 0.03 mm/day to 0.05 mm/day.

The following figure shows north east up time series solutions for 276-1 obtained using JAC-MM.

Figure 6.18: 276-1 NEU timeseries. 2DRMS precision 3.6mm. 2DRMS internal accuracy between 3.6 to 9.12mm

Between approximately days 340 to 360 there was an unusual increase in noise, this was during the month of December. We believe this noise to be as a result of roadworks, as during this time some of the asphalt of road was being replaced, the heavy machinery passing by the sensors presumably vibrating them. However, apart from the month with the roadworks, generally the solutions were very precise and slowly changed over the seven months they were obtained. As this was our test that lasted the greatest period of time, we considered seven months close enough to 12 months that we could at least obtain an idea of the expected accuracy a system such as ours would have in the typical setting. If node 276 did not move and/or its surroundings did not change then ignoring the first 14 days to allow the sidereal filtering to settle the internal solution accuracy was 9.12 mm 2DRMS; this of course is only valid if there was no movement or the surroundings did not change. This can also be considered approximately as a worst case scenario for internal accuracy. If on the other hand the slow change of the solutions were caused by physical movement of the node itself, then using 10 degree polynomial fits to remove the

slow change of the solutions, a 2DRMS internal accuracy of approximately 3.6 mm was obtained; this can be considered the best case for what the solution accuracy was and also considered as the solution precision.

First off we note that the up direction unlike node 277 is not linear, the east solutions undulate smoothly while the north solutions are fairly flat until day 310 where they rapidly drop while at the same time the up direction rapidly increases. The most significant thing we noticed was that the up solutions obtained just prior to tropical storm Fung-wong were significantly higher than the ones obtained just after the storm. There was also a reasonably significant quick change in the north solutions at the same time. In addition the rate of change of the solutions changed abruptly before and after the storm. These rapid changes can be seen better if we obtain the primary principal component direction vector between days 242 and 287 inclusive and plot these solutions as can be seen in the following figure.

Figure 6.19: Primary principal component of 276-1 during tropical cyclone Fung-Wong

This figure shows approximately 20 days before the cyclone and 20 days after the cyclone. The green and the mauve lines are linear fits for the time before

CHAPTER 6. TESTBED RESULTS AND ANALYSIS

and after the cyclone respectively. The solutions are changing at a different rate before the cyclone and after the cyclone. In addition the two linear fits mismatch at the time of the cyclone by 7.45 mm. The solutions also appear to be more or less normally distributed around their linear fits. We then estimate the accuracy of this displacement by removing the linear trends from these solutions to obtain the following figure with standard deviation of dotted lines.

Figure 6.20: Primary principal component of 276-1 during tropical cyclone Fung-Wong with linear trends removed

As can be seen it appears to be reasonably normally distributed with a standard deviation of 1.758 mm. Therefore we should obtain a 95% confidence that the movement measured was $\pm 4 \times 1.758 / \sqrt{20} \text{ mm} = 1.6 \text{ mm}$ around our estimate. In other words $7.5 \pm 1.6 \text{ mm}$ is our estimate for the change in the primary principal component before and after the cyclone. From chapter 3, the slow solution bias change due to the four minute daily shift, we did not see that in our simulations there should be a discontinuity of the solution derivative or a discontinuity in the solutions themselves. As we expect from simulations that the solution bias change should have a repeating period of one year we expect over a short period such as 20 days that the solution bias should not alter significantly over such a short period of time; this justifies what we have done by using linear fits on both sides of the cyclone. In addition it seems that a linear fit does indeed fit very well as can be seen in the previous figure. While it appears that there are other discontinuities of the solution derivative in figure 6.18 that may or may not be related to movement, that this disconti-

nuity at day 264 appears exactly when cyclone Fung-Wong hit, it seems more than probable that this discontinuity was caused either due to one of the nodes moving or the surroundings of the nodes changing such as trees falling down changing the multipath environment. Therefore, ignoring the possibility that the nodes surroundings changed we say that it's quite possible that one of the nodes did indeed move by $7.5 \pm 1.6 \text{ mm}$. Node 276 relative to node 1 moved in a direction of travel near vertical with an elevation of approximately -75° and a small component in the horizontal plane with an azimuth of approximately 280° ; that is to say either node 276 moved down or node 1 moved up.

The discontinuities in the solution derivative are not entirely understood by us as they did not appear in our simulations. One significant discontinuity in figure 6.18 is in the up direction around day 315 which is also accompanied by a slow drop on the north direction. This just happens to coincide with a small shallow earthquake of 3.6 magnitude and 4.9 km in depth very close by but could just be coincidence.

6.2.3 Comparison with retro reflector observations

The following figure shows the locations of the retro reflectors at the site. The green coloured retro reflectors are ones that are outside the active slipping zone and are used for the relative positioning of the reflectors in the slipping zone.

Figure 6.21: Retro reflector locations (Courtesy of NCNU)

As can be seen there is no retro reflector fitted at the site of our base station's GPS receiver node 1. Retro reflector 8 was situated next to our node 276 while retro reflector 9 was next to our node 274. The total station used by NCNU was a Trimble S8; according to NCNU, the report they received from Trimble said that range accuracy was $2mm \pm 1ppm$ and an angle accuracy of $0.5''$, though we were not informed of what type of accuracies these were. From the Trimble S8 datasheet [56] there are many different accuracies given for different situations; where retroreflectors were used all accuracies were around the same order and are given as RMSE for range and standard deviation for angle. For performance under "DR HP" a range accuracy $1mm + 1ppm$ RMSE and an angle accuracy of $0.5''$ standard deviation. Therefore we assume a range accuracy $2mm + 1ppm$ RMSE and an angle accuracy of $0.5''$ standard deviation which approximately matches the "DR HP" performance as given in the datasheet and the numbers are very similar to the ones given to us by NCNU. The total station was placed approximately a kilometer due south of the site; the angle accuracy and the distance accuracy along with orientation

can be seen in the following figure.

Figure 6.22: Trimble S8 orientation and accuracies

To obtain an idea of what 3D accuracy should be expected from a total station, a very approximate estimate was derived. For the east and up accuracies we used the small angle approximation assuming rotational symmetry around the north axis, while that of the north accuracy simply to be the distance accuracy of the total station. Doing this means the standard deviation in the east and up directions is about $1000 \frac{\pi}{180} \left(\frac{0.5}{60^2} \right) \approx 2mm$ while that of the north direction is about $2 \times 10^{-3} + \frac{1}{1000000} 1000 = 3mm$. Therefore, a 95% confidence interval produces east or up $\pm 4mm$ and north $\pm 6mm$ and a 2DRMS accuracy of $7mm$.

The following three figures show the total displacement since 8/Nov/2013 of the retro reflectors with respect to retro reflector 2 and 9 to the date as indicated by the colour. For example, between 2/7/2014 and 28/8/2014 retro reflector 6 moved 10 mm south in 57 days.

We are interested in retro reflector 8 as this corresponds to our node 276, and we are interested in retroreflectors 3 and 4 as these are close to our node 1 and hence we would expect that movements of the land are more likely to be similar than other retro reflectors.

CHAPTER 6. TESTBED RESULTS AND ANALYSIS

Figure 6.23: North total station measurements (Courtesy of NCNU)

Figure 6.24: East total station measurements (Courtesy of NCNU)

Figure 6.25: Up total station measurements (Courtesy of NCNU)

As can be seen there are some considerable changes in the solutions obtained using total station; both third up solutions and the fourth east solutions are considerably out from what one would expect. There also appears to be considerable correlation of the solutions for each time the total station was used to obtain measurements as can be seen when one solution moves south all the other solutions tend to move at the same time. The sparseness of the measurements are problematic as this makes it hard to know how reliable the measurements are. Of all the solutions the north solutions seem to be the most consistent and shows a slow general trend south as expected. Both the east and up solutions do not seem to say a great deal even after removing the measurements performed in July and August. Clearly there are other errors introduced into these solutions that are considerably greater than the ones we estimated. Due to these other errors how the actual movement of the hill relates to total station measurements is unknown. While unsatisfactory we really can't compare the GPS solutions with the total station solutions.

The total station measurements are time sparse and variable making conclusions difficult. The GPS solutions for instantaneous movements (ie. ones that are not slow and continuous) are clearly very precise as we saw for the solution change due to tropical storm Fung-Wong. In contrast it is difficult to make conclusions about the long-term movement of the GPS receivers over

the six months from the GPS solutions. This is because from Multipath and sidereal filtering we predict a solution component due to the multipath bias that is predicted to change over the year making it difficult to distinguish true movement and merely a slowly changing solution bias. In addition to GPS movements can only be compared for for the month of July before node 274 failed. However, in this month the GPS receivers all showed a slight southward trend relative to node 274 which is given as being stationary, this matches the general trend of that of the north solutions obtained from the total station and what one would expect.

Chapter 7

Conclusions

In this thesis we designed, implemented and analysed an accurate land deformation monitoring system where we concentrated primarily on daily static solutions where all observations in a day are used to calculate one solution.

A novel small batteryless low power solar powered GPS wireless sensor node was designed using low end uncalibrated low cost components costing approximately a \$100 USD a node. To the best of our knowledge no one had designed such a node for use in a positioning system before us. Novel GNSS software was designed and created for continuous long-term monitoring applications and attempted to improve solution precision and number of nodes in the network in the presence of intermittent random observations, limited power supply, high multipath environments, and limited radio bandwidth.

The mere fact of not using a battery turned out to be significant for both the design and the form that the solutions took. While this may have not been initially obvious it turned out that no battery in turn meant only observations during daylight hours were practical to obtain, and even then due to the extremely high power demands of even the most low power GPS receiver meant that duty cycling of the GPS receiver itself needed to be implemented in the design for the node to be able to function at low light levels and to be kept small. This duty cycling in turn limited the period of time that the GPS receiver could continuously track the phase observable which meant usual processing of the phase observable would have required a re-initialisation of all the integer ambiguities for every duty cycle of the GPS receiver. We also

CHAPTER 7. CONCLUSIONS

considered the RF link with which the observations are transferred from the node to the final processing software. For our implementation it was a simple broadcast method and with such a simple RF link epochs did get lost. The intermittency caused by both the duty cycling of the GPS receiver and the epochs that were lost over the RF link fractured what standard processing software programs expect causing them to typically fail, as can be seen in Chapter 4 when both the standard processing software programs DDBase and RTKLib failed to produce solutions all were imprecise, rendering these programs unsuitable for our application. Therefore, this required the design of an entirely unique solution methodology where there was no expectation that the phase observables would be continuously tracked and data could be fractured. This clearly justifies the effort that we spent on designing a solution algorithm for this application.

The design of our solution algorithm was a distributed algorithm where some aspects of the algorithm were performed on the nodes themselves thus significantly reducing the amount of data needed to be sent over the RF link. In tests we performed that can be found in Chapter 4, we measured a 74% reduction in data required to be sent over the RF link and a reduction of 46% in lost epochs. This in turn allows a far greater number of wireless sensor nodes to occupy a site, thus increasing node density and granular resolution of the solutions of the site of interest.

When using high quality lossless non-intermittent observations as obtained from high quality Trimble devices from New Zealand's LINZ Rinex data Archive, our algorithm solution precision was comparable if not better to other solution processing programs such as DDBase and RTKLib. However, where our algorithm really excelled at was for intermittent data. Even on dark days of winter when less than 0.5 hours of daily activity were observed on some days our algorithm performed surprisingly well. For a two week test in winter where on average a node was actively returning data for only approximately 1.5 hour a day on average, we still calculated a 2DRMS precision of 8 mm; this compared with the same test performed in summer when a 2DRMS precision of 3 mm was obtained (Chapter 6).

As the GPS receiver antenna of the nodes was a cheap uncalibrated antenna with no choke ring to prevent multipath and our nodes were expected to function in a high multipath environment such as being attached to the side of a

CHAPTER 7. CONCLUSIONS

wall with poor visibility of the sky, sidereal filtering was used in our algorithm to minimise the effect that multipath had on our solutions. Sidereal filtering made sense as our application was one of longtime monitoring of a location that would not change particularly fast over time. Due to the intermittent nature of our observations, we were in the unique situation to have the ability to improve daily solution precision more than we would have otherwise according to simulations. Actual real life measurements showed that using our interday sidereal filtering, daily 2DRMS solution precision could be improved by 60% (Chapter 3). However, contrary to our simulations a precision improvement of 30% could still be obtained when using near continuous observations 24 hours a day (Chapter 3). We could only presume that this unexpected precision improvement was due to the satellite geometry not entirely repeating every nominal satellite repeat period. While unexpected it still shows that because of our intermittent nature of obtaining observations, interday sidereal filtering can be used to have a greater effect on solution precision improvement than would otherwise be possible if we used permanently powered GPS receiver to obtain observations 24 hours a day.

In Chapter 3 it was seen that simulations predicted an undesirable solution bias that would slowly change and repeat once a year. This bias was caused due to only being able to take observations during daylight hours. There is approximately a four minute shift between the diurnal period and the satellite repeat period every day, causing a change in multipath between daylight hours of consecutive days, which sidereal filtering could not remove. If the observations were intermittent but uniformly distributed throughout the 24-hour day this problem would not have manifested. It is only because our observations are not uniform throughout the 24-hour and are in fact only during daylight hours. We did not have the time to create a static test bed that lasted between one and two years to verify this phenomenon, and more importantly to estimate the magnitude one might expect for this solution bias for a high multipath environment and hence obtain an estimate for long term accuracy over many months. However, from the LuShan testbed site which we considered to be a moderate to high multipath environment, node 276 was deployed for seven months, and while it was not a static test and physical movement of the node was extremely likely to have happend, we obtained a tight lower bound and a soft upper bound for accuracy. The lower bound was calculated by removing low frequencies from the solution which may have been actual

CHAPTER 7. CONCLUSIONS

slow physical movements, this resulted in a 2DRMS accuracy of 3.6 mm. The upper bound was calculated by assuming the the low frequencies were due to the undesirable phenomenon and that any physical movement during the seven months was minimal; in this case a 2DRMS accuracy of 9.1 mm was obtained. Even without knowing the magnitude of the slow changing bias that repeats every year, we still could calculate the precision for the LuShan site as 3.6 mm 2DRMS.

Despite the shortcomings of not being able to determine a more exact typical accuracy value better than 9 mm 2DRMS, precision values were relatively easy to calculate and values between 3 to 8 mm 2DRMS were all observed. One exciting aspect of the solutions was to confidently be able to determine that the second tropical cyclone that hit Taiwan for the year 2014, Fung-wong, did indeed have an effect on the solutions obtained at LuShan. Using a short term window of 40 days we were able to determine that 7.5 ± 1.6 mm of movement had occurred between node 276 and node 1 at LuShan almost overnight assuming the surroundings had not changed. While another tropical cyclone hit Taiwan during the 2014 season no significant change in solutions were observed at LuShan for that cyclone. Due to not knowing the magnitude of the slow changing bias of solutions caused due to the four minute shift every day, we could not say that any movement apart from the one caused by tropical cyclone Fung-wong happened at LuShan for 276-1.

Hardware wise the node was designed to work during daylight hours down to 20 Wm^{-2} of solar irradiance which equates to an extremely dark day allowing observations to be taken even during storms when one might be interested in obtaining observations due to land deformation caused by water. A solar powered repeater network structure was created to extend the system to a size that would be able to cover sites of a few kilometers without the need for mains power. The solar powered repeaters were of the same form factor as that of the GPS nodes and also did not use any batteries. The transmitted data came back to a mains powered base station that collected the data to be sent back to us for processing. Both the solar powered repeaters and the base station worked flawlessly and one set up required no intervention from us whatsoever. Of all our problems the waterproofing of the GPS wireless sensor nodes was our greatest. Whilst seemingly trivial we could not consistently waterproof these devices. The solar powered repeater nodes however were far easier to waterproof and none of them failed due to water leakage. The GPS

CHAPTER 7. CONCLUSIONS

wireless sensor nodes were more difficult due to the antennas we used which were active patch and it was suspected that water was entering the electronics of the patch antenna; this of course the repeater nodes did not have.

The failure of our nodes due to suspected water leakage caused three of the four wireless GPS receiver nodes at LuShan to ultimately fail. Most importantly our only fixed GPS receiver node failed. This meant we were unable to obtain any absolute movement that the hill experienced and only relative between the last surviving node 276 and the base station node 1, both of which were in the sliding zone.

Observations obtained from a total station whose solutions were relative to points outside the sliding zone were examined in the hope that these could be compared with the GPS solutions. The variation in displacement for the total station solutions were great, in addition to the solutions being time sparse where there were months between some solutions. Both of these aspects brought into question the accuracy of these solutions so little could be done to compare these solutions with the GPS solutions. For the GPS solutions only the node combination 276-1 produced solutions for any extended period of time, these solutions appeared to be as good as they could have been and typically one solution a day was obtained, but both nodes 271 and 1 were in the sliding zone.

We showed that intraday solutions are possible but due to the intermittent nature of such observations solutions would only be intermittent. Looking at intraday solutions it was possible to more accurately tell when a specific instantaneous event happened (Chapter 3). However, this form of solutions were not investigated in great depth as we concentrated on daily solutions instead.

Our system showed promise for land deformation monitoring applications. For the work that we did to be practical, more robust housing would be needed to protect the fragile electronics from the extreme conditions that the GPS receiver nodes experience; in particular the waterproofing of the nodes is arguably the most pressing aspect. Long term monitoring tests would also be needed, ones that last continuously for many years to ascertain the system's long term reliability and unforeseen problems to be identified. Long term tests would also allow better estimates of accuracy to be calculated.

CHAPTER 7. CONCLUSIONS

There exists a great deal of future work that could be done to improve our system both at a hardware level and software level. Some aspects that could be worked on include the following.

- Improving electronic design
- Improved housing and ruggedization
- Improving the quality of the observations
- Number of observations
- More even distribution of the observations
- Alternative solution processing techniques such as networked solutions
- Alternative multipath mitigation techniques, etc.

While accuracy is desirable, multipath and an unknown PCV were clearly the biggest factors limiting solution accuracy. There are many other methods and ways that could be used to reduce the multipath/PCV problem either at a hardware level or at a software level. Both methods could be explored to increase accuracy. Networked solutions where many fixed nodes are used to calculate the location of one unknown node could also increase solution accuracy.

When people look to GPS as a solution for deformation monitoring their first thought might be kinematic solutions with the best possible accuracy. For some applications this might be appropriate but is likely to be expensive and not always the best solution for a problem. For some applications such as long term monitoring of land deformation this may not be needed; one solution a day with good precision and/or accuracy at an affordable cost may provide a better solution. Our system has a very cheap initial cost outlay compared to typical high end GPS receivers and is easy to deploy, without any need to keep replacing batteries the total cost of the system is kept to a minimum allows a greater density of nodes to be deployed thus increasing solution granularity. In addition the solution precision is very good.

Our system shares some of the advantages that a GPS receiver based land deformation monitoring system has; it does not require human intervention

to obtain observations, works in most weather conditions, and provides very good accuracy. Beyond that our system has a small form factor, no batteries, easy to deploy and is cheap as desired.

7.1 Conclusion summary

A unique permanently deployed land deformation monitoring system using small, cheap, batteryless, solar powered singleband GPS wireless sensor nodes were designed, implemented and validated. Development consisted of hardware, software, and algorithm design. A unique distributed relative static positioning algorithm design was shown to be successful at obtaining precise solutions using intermittent poor quality phase observable measurements in high multipath sites with minimal radio channel bandwidth requirements. Precise solutions were obtained with as little as 0.5 hours of GPS on time per day and light levels as little as 20 Wm^{-2} of solar irradiance during daylight hours, this equates to an extremely dark day. All other GNSS software packages we tried were either unable to return any solution or ones that were highly inaccurate and used higher radio channel bandwidths.

As the nodes lacked batteries, only diurnal observations could be obtained in sufficient quantity, simulations suggested this would result in a slowly changing solution bias that would repeat every year. Despite the changing solution bias, rapid movements lasting far less than compared to one year could be precisely determined and a clear change in solutions were observed after tropical storm Fung-Wong. Using data obtained from the LuShan testbed, typical solution precision was 4 mm 2DRMS while solution accuracy was unknown but appears to be less than 10 mm 2DRMS for daily based solutions.

All nodes in the LuShan testbed ultimately failed bar one, this was suspected to be due to water entering the housing and corroding the electronics but we were unable to confirm this due to being based in New Zealand.

The design and implementation was successful demonstrating such a system is feasible.

CHAPTER 7. CONCLUSIONS

Nomenclature

2D two-dimensional

2DRMS 2 times Distance Root Mean Square

AFM Ambiguity Function Method

AGPS Assisted GPS

AR Ambiguity Resolution

Baseline Distance between two GPS receivers

BPSK Binary Phase Shift Keying

C/A Coarse acquisition code

CDMA Code Division Multiple Access

CFFS Code Float Fix Sidereal

Chip One bit of a PRN code

CORS Continuously Operating Reference Station

DC Direct Current

DGPS Differential GPS

DLL Delay Locked Loop

EQC The Earthquake Commission, New Zealand

FFT Fast Fourier Transform

FLL Frequency Locked Loop

Nomenclature

GLONASS GLObalnaya NAvigatsionnaya Sputnikovaya Sistema

GNSS Global Navigation Satellite System

GPS Global positioning system

HOW Hand Over Word

ID IDentification

IGS International GNSS Service

ILS Integer Least Squares

IMU Inertial Measurement Unit

InSAR Interferometric Synthetic Aperture Radar

KDE Kernel Density Estimation

LAMBDA Least-squares AMBiguity Decorrelation Adjustment

LNA Low Noise Amplifier

LS Least Squares

MMPT Maximum Power Point Tracking

MPP Maximum Power Point

MPU MicroProcessor Unit

NAV Navigation Data

NCNU National Chi Nan University

NEU North East Up

NTRIP Networked Transport of RTCM via Internet Protocol

P Precise unencrypted code

PAN Personal Area Network

PC Personal Computer

PCB Printed Circuit Board

Nomenclature

PCO	Phase Center Offset
PCV	Phase Center Variation
PDOP	Position Dilution of Precision
PLL	Phase Locked Loop
PMB	Power Management Block
PPP	Precise Point Positioning
PRN	Pseudo Random Number
PWS	Personal Weather Station
RF	Radio frequency
RINEX	Receiver Independent Exchange Format
RTCC	Real Time Calendar Clock
RTK	Real time kinematic
RTV silicone	Room temperature vulcanization silicone
TEC	Total Electron Count
TTFF	Time To First Fix
UV	UltraViolet
WSN	Wireless sensor network
Y	Precise encrypted code

Nomenclature

Appendices

Appendix A

GPS basics

In this appendix a general introduction is given to GPS operation and in particular, aspects that relate to acquiring observations for solving position solutions using singleband receivers.

A.1 GPS signal

As of writing GPS is currently undergoing a modernization to improve both civilian and military use. Between 1990 and 2004 legacy satellites were launched while from 2005 modernized satellites have been launched. According to the National Coordination Office for Space-Based Positioning, Navigation, and Timing [16] this is in an effort to upgrade the features and performance of GPS. Currently GPS transmits on three different RF links from the satellites to end-users. The RF links are called L1, L2 and L5 and are named after the bands that they transmit in. *Code Division Multiple Access* (CDMA) is used as the channel access method so all satellites used the same carrier frequencies. L1 has a nominal frequency of 1575.42Mhz as seen from Earth, L2 1227.60Mhz and L5 1176.45Mhz. These nominal frequencies are modulated with various signals to aid navigation. The current GPS modernization consists of generally improving the hardware as well as adding more signals that are sent over the RF links. The GPS modernization currently underway will take many years and satellites producing signals such as L1C on L1 are not expected to be launched until 2016 with 24 satellites expected by around 2026 [31]. As we are

APPENDIX A. GPS BASICS

interested in only singleband receivers because they are cheaper than multi-band receivers and that no new modernized signals are currently in operation on L1, we restrict our investigation here to that of legacy L1 signals.

Figure A.1: Legacy L1 signal generation block diagram

GPS legacy signals on L1 consist of a coarse acquisition *Pseudo Random Number* (PRN) code called C/A sent at 1.023Mb/s, a precise PRN code P sent at 10.23Mb/s which is called Y if encrypted, and navigation data called NAV sent at 50b/s. The two PRN codes are unique to each satellite and each spread the navigation signal. Bits of the PRN codes are also called chips. L1 consists of in-phase and quadrature components. Each component is separately modulated using *Binary Phase Shift Keying* (BPSK) as their modulation technique. One modulator is supplied with a bit train from modulo-2 addition of P(Y) and NAV while the other is supplied with a bit train from modulo-2 addition of C/A and NAV. The signals and the carrier frequencies are derived from a single time source and synchronized with one another thus causing the signals to be bit phased with one another in addition to the signals being synchronized with one another at a higher level. Figure A.1 depicts a block diagram of the generation of such signals. The instantaneous L1 wavefront a satellite sends $W_{TX}(t)$ at a time t , can be written as follows assuming the satellite clock keeps perfect frequency.

$$W_{TX}(t) = A_{TX} \Re \left\{ S(t + \Delta T_{TX}(t)) e^{2\pi i(t f_{TX} + \phi_0^{TX})} \right\} \quad (\text{A.1})$$

APPENDIX A. GPS BASICS

Where $S(t) = NAV(t) (P(Y)(t) e^{i0} + C(t) e^{i\pi/2})$ is the composite signal, $C(t)$ C/A code, $P(Y)(t)$ P(Y)-code, $NAV(t)$ navigation data, ϕ_0^{TX} satellite oscillator phase at time zero, A_{TX} transmission amplitude, ϕ_0^{TX} satellite oscillator phase at time zero, f_{TX} nominal satellite oscillator frequency as seen from Earth so as to account for general relativistic effects, f_0 1575.42Mhz as seen from Earth, and τ_0^{TX} is the satellite composite signal offset at time zero. $C, P(Y), NAV \in \{-1, 1\}$ and are functions so as to produce the correct composite signal where a mapping of $0 \rightarrow 1$ and $1 \rightarrow -1$ has been applied. At the receiver the instantaneous L1 wavefront a receiver receives $W_{RX}(t)$ at a time t , can then be written as follows assuming no hindrance by the atmosphere.

$$W_{RX}(t) = A_{RX} \Re \left\{ S(t + \Delta T_{TX}(t) - \Delta t(t)) e^{2\pi i((t - \Delta t(t))f_{TX} + \phi_0^{TX})} \right\} \quad (\text{A.2})$$

Where A_{RX} is reception amplitude and $\Delta t(t)$ is the transmission flight time from the satellite at transmission time to the receiver at reception time t . It's important to realize that $\Delta t(t)$ is not the flight time between the satellite and receiver at time t , but rather the flight time based as the receiver sees it. It is similar to when a airplane passes by and one hears the sound of the plane lagging where the plane actually is. The flight time of the sound from the plane as determined by the listener is different from the flight time one would get from calculating where the plane actually is to the user with respect to the same reception time.

A.2 What the GPS receiver does to the received $W_{RX}(t)$ signal

Figure A.2: Simplified version of a possible GPS satellite acquisition and tracking scheme

Figure A.2 shows a simplified version of what a GPS satellite receiver can do to acquire the satellite and track the signals that it produces. Upon reception the receiver tries to separate the signals from L1 and track the C/A code and the carrier phase. This can effectively be performed by mixing $W_{RX}(t)$ with a complex local oscillator $LO(t) = e^{2\pi i(t f_{RX} + \phi_0^{RX})}$ where f_{RX} is the frequency of the oscillator and ϕ_0^{RX} the phase of the oscillator at time zero, filtering using a low pass filter LPF , then phase and/or frequency tracking to stop rotation along with correct C/A timing to match the C/A code as sent by the satellite so as to acquire access to the data being sent by the satellite.

A.2.0.1 Obtaining the baseband signal $R(t)$

After mixing with the local oscillator and filtering using the low pass filter, the receiver obtains the following baseband, where, $\Delta f = f_{RX} - f_{TX}$ and $\Delta\phi_0 = \phi_0^{RX} - \phi_0^{TX}$.

$$\begin{aligned} R(t) &= \frac{2}{A_{RX}} LPF \{W_{TX}(t) LO(t)\} \\ &= S(t + \Delta T_{TX}(t) - \Delta t(t)) e^{2\pi i(t\Delta f + \Delta t(t)f_{TX} + \Delta\phi_0)} \end{aligned} \quad (\text{A.3})$$

We define $t\Delta f + \Delta t(t) f_{TX} + \Delta\phi_0$ as the received beat carrier phase (carrier phase).

A.2.0.2 Stopping rotation

We see that this is a constellation of four points that rotates due to the frequency difference between the receiver's local oscillator and the satellite's oscillator, and also rotates due to the radial motion of the satellite itself with respect to the receiver.

If we let $\Phi(t) = t\Delta f + \Delta t(t) f_{TX} + \Delta\phi_0$ then we can correct for rotation by multiplying $R(t)$ as follows.

$$\begin{aligned} R(t) e^{-2\pi i(\Phi(t))} &= S(t + \Delta T_{TX}(t) - \Delta t(t)) e^{2\pi i(\Phi(t))} e^{-2\pi i(\Phi(t))} \\ &= S(t + \Delta T_{TX}(t) - \Delta t(t)) \end{aligned} \quad (\text{A.4})$$

This stops the constellation from rotating and removes any constant constellation rotation offset. This then resolves the composite signal. Hence we define $\Phi(t)$ as the estimated received beat carrier phase by the receiver (estimated carrier phase). The constellation's phase (or equivalently the constellation's rotation offset) is defined as the difference between the carrier phase and the estimated carrier phase $t\Delta f + \Delta t(t) f_{TX} + \Delta\phi_0 - \Phi(t)$. More generally as long as the estimated carrier phase is in phase with the carrier phase the constellation stops rotating and there is no constant constellation rotation offset.

A.2.0.3 C/A Code alignment

After the constellation rotation has been stopped by letting the estimated carrier phase be in phase with the carrier phase, a local replica of the C/A code has to be mixed with the composite signal and phase shifted in time until the local replica of the C/A code is in phase with the one that is in the received composite signal. We then define $\tau(t)$ as the C/A code alignment offset and is how much the incoming C/A is misaligned with the local replica. We define the local replica of the C/A code as $LCA(t) = C(t + \Delta T_{RX}(t))$

where $\Delta T_{RX}(t)$ is the receiver's clock offset. First we notice that if we offset the local C/A replica by $\tau(t)$ and let $\tau(t) = \Delta t(t) + \Delta T_{RX}(t) - \Delta T_{TX}(t)$ and multiply this offseted C/A with $R(t)$ we expect the following for a random time t .

$$\begin{aligned}
 & E \left[LCA(t - \tau(t)) R(t) e^{-2\pi i(\Phi(t))} \right] \\
 &= E \left[LCA(t - \tau(t)) S(t + \Delta T_{TX}(t) - \Delta t(t)) \right] \\
 &= E \left[C(t + \Delta T_{TX}(t) - \Delta t(t)) S(t + \Delta T_{TX}(t) - \Delta t(t)) \right] \\
 &\approx iE \left[NAV(t + \Delta T_{TX}(t) - \Delta t(t)) \right] \tag{A.5}
 \end{aligned}$$

This is due to the fact that $P(Y)$ and C are not well cross-correlated, while of course C is perfectly correlated with itself.

$LCA(t - \tau(t)) R(t) e^{-2\pi i(\Phi(t))}$ is the input to the acquisition and tracking block in figure A.2. Visually $LCA(t - \tau(t)) R(t) e^{-2\pi i(\Phi(t))}$ is a constellation without any rotation of four points. The two points lying on the imaginary axis move slowly at no more than 50 times a second and contain the navigation data, while, the two points that lie on the real axis move very rapidly at up to 10.23 million times a second in a seemingly random way with a mean value of zero. More generally when this happens the incoming C/A code is aligned with the local replica and the carrier phase is in phase with the estimated carrier phase.

A.3 Acquisition, tracking and NAV data extraction

A.3.0.1 Extraction of NAV data using filtering

When the incoming C/A code is aligned with the local replica and the carrier phase is in phase with the estimated carrier phase, navigation data can easily be obtained simply by using a low pass filter

$$NAV(t + \Delta T_{TX}(t) - \Delta t(t)) = -iLPF \left[LCA(t - \tau(t)) R(t) e^{-2\pi i(\Phi(t))} \right] \quad (\text{A.6})$$

This allows the GPS receiver access to the satellite's navigation data which includes a wealth of data including the satellite's estimate of $\Delta T_{TX}(t)$ and time of transmission of specific navigation data transitions. Along with data to calculate the satellites positions.

A.3.0.2 A metric for C/A code alignment and rotation

We restrict t to be within a small period of time 2δ which is less than NAV 's period while still ensuring a long enough period such that $P(Y)$ and C are still not well correlated over that period. To find such a time period is possible as NAV 's period is 204600 times longer than $P(Y)$'s period. We assume the C/A codes are perfectly aligned and the carrier phase is in phase with the estimated carrier phase. We then expect the following.

$$E \left[LCA(t - \tau(t)) R(t) e^{-2\pi i(\Phi(t))} \right] \approx iNAV(t + \Delta T_{TX}(t) - \Delta t(t)) \quad (\text{A.7})$$

$$\begin{aligned} \therefore NAV(t + \Delta T_{TX}(t) - \Delta t(t)) \\ \approx \frac{-i}{2\delta} \int_{k=t-\delta}^{k=t+\delta} LCA(k - \tau(k)) R(k) e^{-2\pi i(\Phi(k))} dk \end{aligned} \quad (\text{A.8})$$

As the navigation data has a constant magnitude of 1, any imperfections in the correlation between $LCA(t - \tau(t))$ and $C(t + \Delta T_{TX}(t) - \Delta t(t))$ due to incorrectly estimating $\tau(t)$ will affect the magnitude of the acquired navigation data. In addition, incorrectly estimating $\Phi(k)$ causing a rotating constellation motion, will, after integration also negatively affect the magnitude of the acquired navigation data. Therefore, we can say the following where γ is defined as the correlation coefficient between a local time shifted replica of the C/A code and the one being received that may be rotating.

APPENDIX A. GPS BASICS

$$\gamma = \frac{-1}{4\delta^2} \left(\int_{k=t-\delta}^{k=t+\delta} LCA(k - \tau(k)) R(k) e^{-2\pi i(\Phi(k))} dk \right)^2 \quad (\text{A.9})$$

This coefficient more generally can be calculated more accurately with averaging; therefore a GPS receiver could calculate it as follows.

$$\gamma = \frac{-1}{4n\delta^2} \sum_{q=1} \left(\int_{k=t_q-\delta}^{k=t_q+\delta} LCA(k - \tau(k)) R(k) e^{-2\pi i(\Phi(k))} dk \right)^2 \quad (\text{A.10})$$

Squaring eliminates the sign of the BPSK. The magnitude of γ is related to how well the C/A codes are aligned and how well the rotating motion of the constellation has been stopped, while the angle of γ is related to the constellation's constant rotation offset with a half cycle ambiguity. A constant constellation rotation offset does not affect the magnitude of γ as a constant rotation offset is just a constant that can be taken out of the integral. Therefore, gamma will be maximized when the C/A codes are perfectly aligned and the constellation is not rotating while it is invariant for constant rotation offset of the constellation.

- $|\gamma|$ is maximized when C/A codes are aligned and the constellation is not rotating.

Treating Φ and τ as variables a GPS receiver can vary Φ and τ to maximize γ . If γ is above a certain threshold the GPS receiver can assume that the satellite is acquired and to commence tracking τ , Φ , and decoding NAV data.

We are interested in maximizing γ because when it is a maximum, with the addition of some ambiguity both Φ and τ are good estimates for $t\Delta f + \Delta t(t) f_{TX} + \Delta\phi_0$ and $\Delta t(t) + \Delta T_{RX}(t) - \Delta T_{TX}(t)$ respectively which turns out to be useful in finding positions solutions. In addition when γ is maximized we are able to obtain navigation data which is also useful for finding position solutions.

A.3.0.3 First-order linear approximations of unknown functions τ and Φ

We wish to maximize γ , the correlation coefficient. Due to the surface shape of the magnitude of γ with respect to Φ and τ initial estimates for both Φ and τ are required. Without good initial estimates, γ is dominated by noise making standard tracking schemes such as *Phase Locked Loop* (PLL), *Frequency Locked Loop* (FLL), *Delay Locked Loop* (DLL) and early/late time useless. It's like trying to track an ant crawling in long grass; you have to find it first before you can track it as the grass makes it difficult to see the ant from afar.

First we create a first order linear approximation model of how Φ and τ change. We have already seen that when $\Phi(t) = t\Delta f + \Delta t(t)f_{TX} + \Delta\phi_0$ and $\tau(t) = \Delta t(t) + \Delta T_{RX}(t) - \Delta T_{TX}(t)$ we are able to stop rotation and align our local C/A code replica with the incoming one. Therefore, these are the Φ and τ that we are looking for. Linear approximations of these two equations are written below where $F(t_m) = (\Delta f + \Delta f_{TX}(t_m))$, $\Delta f_{TX}(t_m)$ is the change of frequency of f_{TX} due to Doppler at time t_m , where a positive value is for the satellite moving away from the receiver and $\theta(t_m)$ and $\Xi(t_m)$ are some constants. Proofs can be found in A.5.1 and A.5.2.

$$\Phi(t) \approx tF(t_m) + \theta(t_m) \quad (\text{A.11})$$

$$\tau(t) \approx (t - t_m)F(t_m) \frac{-1}{f_0} + \Xi(t_m) \quad (\text{A.12})$$

These approximations are only valid if $F(t_m)$ does not change too rapidly around time t_m . The maximum rate at which velocity will change is about 0.1178 ms^{-2} and is when the satellite is directly overhead (A.5.4). On the L1 band this implies that the constellation rotation speed will change by less than about 0.9 Hz s^{-1} if $F(t_m)$ is left unchanged in equation A.13 (see A.5.3 and A.5.4). Now compare this to the range of $F(t_m)$. $\Delta f_{TX}(t_m)$ can be as large as about $\pm 5 \text{ kHz}$ (A.5.4), and depending on the receiver clock accuracy Δf could be out by another 5 kHz if we assume a receiver clock accuracy of 3.5 ppm . This means the range of $F(t_m)$ is in the order of 20 kHz . If we restrict our time of interest to 1 ms , then, $F(t_m)$ will change by less than 0.0009 Hz which

APPENDIX A. GPS BASICS

is far less than the range of the 20 kHz of $F(t_m)$. Therefore, for short periods of time this is a valid approximation.

Equations A.11 and A.12 form an approximate model of how Φ and τ will change in the short term.

A.3.0.4 Further simplifications to the first-order linear approximations of unknown functions τ and Φ in regard to acquisition

In figure A.2 the input of the acquisition and tracking block as we have already seen is written as in equation A.13.

$$LCA(t - \tau(t)) R(t) e^{-2\pi i(\Phi(t))} \quad (\text{A.13})$$

To acquire Φ and τ initially we can make further simplifications to equations A.11 and A.12. $\theta(t_m)$ in equation A.11 when placed in equation A.13 has no effect on changing constellation rotation with respect to time. A.13 will still be a non-rotating four-point constellation but just with a constant rotation offset of $\theta(t_m)$ cycles. As we have already mentioned in equation A.10, any constant rotation offset has no effect on the magnitude of the correlation coefficient. Therefore we can ignore $\theta(t_m)$ when initially acquiring Φ .

The carrier wave frequency is 1540 times greater than the bit rate of the C/A code. As frequency times time is phase, if $F = 10,000\text{ Hz}$ then it takes 0.025 ms for Φ to change by a quarter of the cycle, while it takes 38.5 ms for the C/A code to change by a quarter of a chip. If we then assume a digitalization of R at the rate of 4.092 Mb/s and sampling 1 ms worth of R , more often than not we couldn't even detect the difference between $F = 0$ and $F = 10,000$ in the C/A code directly while it would be easy to detect in the carrier wave.

Due to these two points we make the following two approximations when considering initial acquisition of Φ and τ . Here we acknowledge that the constellation will have an arbitrary constant rotation offset, are only valid for periods of time of a few milliseconds, and baseband sampling rate is no more than a few times per chip.

$$\Phi(t) \approx tF(t_m) \quad (\text{A.14})$$

$$\tau(t) \approx \Xi(t_m) \quad (\text{A.15})$$

A.3.0.5 Acquisition

With these two approximations A.14 and A.15 the receiver can do a two-dimensional search, F with a frequency dimension and the other Ξ with a time dimension to find the point that maximizes the correlation coefficient in equation A.10. Assuming a receiver clock accuracy of 3.5 ppm , the receiver would have to search from -10 kHz to 10 kHz . As the C/A code is a periodic function with a period of 1 ms , the receiver would have to search from 0 ms to 1 ms .

Estimating F and Ξ by trying to maximize the correlation between the local C/A code replica with the incoming one turns out to be computationally demanding using more energy than tracking, and is a major concern for GPS receivers. Because of this much research has been directed towards this problem to reduce the computational effort to estimate these two parameters [63]. A parallelized 2D search by using *Fast Fourier Transform* (FFT) is a conventional method currently used in software defined receivers [63].

As an example we cross-correlated a local C/A code replica with an incoming one on L1. We used a sample rate of 4.092 Mb/s being four times the nominal frequency of the chip rate and searched by varying the frequency term by $\pm 10 \text{ kHz}$ in steps of 125 Hz and then using cross correlation varying the time term by no more than $\pm 511.5 \text{ chips}$ ($\pm 511.5 \text{ chips}$ covers the entire 1 ms). Navigation data was simulated by using random data, while no P(Y) data was added. An offset of 5000 Hz was applied to the 1.57542 GHz carrier frequency and a phase offset of 250.15 chips was applied to the C/A code of the incoming signal. Using 1 ms worth of sequential incoming data, figure A.3 was obtained.

Figure A.3: Correlation coefficient versus phase and frequency offsets. 1 ms of sequential data of C/A PRN 16 cross-correlation. 4.092Mb/s sampling rate. 16 kb of data, 658,651 evaluation points.

As can be seen there is a clear peak in the graph representing the estimates of F and Ξ . The estimated phase offset using interpolation around the highest peak was 250.18 *chips* while that of the frequency was 5000 *Hz*; this matches well with the exact values. From this figure it is clear to see why tracking will not work without good initial estimations of τ and Φ . It's interesting to note in passing that the 0.03 chips that the interpolation value was out equates to the time light takes to travel 9m. This is roughly the correct order of accuracy cheap consumer grade GPS receivers have.

The number of points needed to be evaluated in figure A.3 was 658,651. While less points could have been used and still be able to obtain reasonable estimates for F and Ξ , by the 2-D search method there are inherently always going to be a large number of points needed to be evaluated. While other methods such as the parallelized 2D search by using FFT exist and are less computationally demanding and hence less energy demanding, acquisition to the best of our knowledge still uses more energy than tracking. As an example the Ublox NEO-7N which is a modern consumer grade GPS receiver uses 23% more energy during its acquisition state than it's tracking state [17].

A.3.0.6 Tracking

Once the acquisition has been performed and the point of the maximum correlation coefficient has been found, the receiver can then track τ and Φ using standard techniques such as PLL, FLL, DLL and early/late tracking methods. The linear approximations do not stop the maximum correlation coefficient point from moving but it slows it down sufficiently that one can treat it as a stationary point until it has been found then one can simply track it as it moves.

From acquisition, estimates for F and Ξ are obtained, which via equation A.12 gives an estimate for $\tau(t)$. To track $\tau(t)$ then early/late tracking can be used. Such a method usually consists of three locally produced C/A replicas, one slightly ahead of what is expected from the satellite, one as expected from the satellite, and one slightly behind what is expected from the satellite.

$$\text{Early } LCA(t - \tau(t) + \xi)$$

$$\text{Prompt } LCA(t - \tau(t))$$

$$\text{Late } LCA(t - \tau(t) - \xi)$$

These three C/A replicas are then each correlated with $R(t)e^{-2\pi i(\Phi(t))}$ to produce three correlation coefficients ($\gamma_E, \gamma_P, \gamma_L$) and using interpolation a new estimate for $\tau(t)$ can be obtained to keep the code aligned. Keeping the code aligned is the one of the two requirements for maximizing γ .

$LCA(t - \tau(t))R(t)$ has the effect of removing the C/A code from R . This removal is called wiping the code. Once it is removed a carrier tracking scheme such as a PLL can be used on $LCA(t - \tau(t))R(t)$ because in one direction it appears as a standard BPSK signal. A costas PLL could be performed on $LCA(t - \tau(t))R(t)$ to estimate $\Phi(t)$ as it is invariant to the navigation transitions. Assume that we designed the costas loop to align on the imaginary axis. Then, the costas loop will align NAV's BPSK signal along the imaginary axis with an ambiguity as to which way around it is aligned. The costas

APPENDIX A. GPS BASICS

loop will also stop the constellation from rotating which is one of the two requirements for maximizing γ .

So, using code tracking and carrier tracking simultaneously the maximum point of correlation can be continuously tracked. It is not sufficient for a receiver solely to track only one of $\tau(t)$ or $\Phi(t)$; both need to be tracked simultaneously.

We have already seen that in A.3.0.2 $\tau(t) = \Delta t(t) + \Delta T_{RX}(t) - \Delta T_{TX}(t)$ implies maximum correlation. Due to the 1 ms C/A ambiguity the converse is not true. Therefore we can say that maximum correlation implies $\tau(t) = \Delta t(t) + \Delta T_{RX}(t) - \Delta T_{TX}(t) + M/1000$ where M is some fixed integer.

Likewise $\Phi(t) = t\Delta f + \Delta t(t)f_{TX} + \Delta\phi_0$ implies maximum correlation but due to carrier phase cycle ambiguity and that gamma is maximized for any constant rotation offset the converse is not true. The costas loop removes the constant rotation offset with an ambiguity of half a cycle, and if the PLL accumulates its phase offset rather than resetting it as it passes through an angle of zero, then maximum correlation implies $\Phi(t) = t\Delta f + \Delta t(t)f_{TX} + \Delta\phi_0 + N/2$ for some fixed integer N .

τ and Φ when these ambiguities are considered become the two observables used by almost all low end consumer grade GPS receivers for position solution calculations.

A.4 Observables

Observables are measurements taken by the GPS receiver of quantities that the GPS receiver can directly measure. Observables do not directly tell you where the GPS receiver is situated but with using various techniques will allow you to calculate position solutions that do tell you where the GPS is situated. The two observables we consider are the code observable and the phase observable.

We have seen by tracking the maximum point of γ using early late timing and a costas PLL that we have found $\tau(t) = \Delta t(t) + \Delta T_{RX}(t) - \Delta T_{TX}(t) + M/1000$ for some fixed integer M and $\Phi(t) = t\Delta f + \Delta t(t)f_{TX} + \Delta\phi_0 + N/2$ for some fixed integer N . These are the code observable and the phase observable respectively so far. However, there are some added complications and the form they take can differ.

A.4.1 The code observable

Once γ is tracked the receiver has access to the navigation data. The navigation data is sent as 30 bits per word. There are 10 words in a subframe taking 6 seconds to transmit. Each subframe contains a *Hand Over Word* (HOW) word that indicates the exact time when the leading-edge of the first bit of the navigation data was transmitted from the satellite. In the satellite the first bit of every NAV data transition is aligned to first chip of the the C/A code. This is possible as they are derived from the same oscillator (see A.1). Figure A.4 shows the C/A NAV timing relationship.

Figure A.4: C/A NAV timing relationship

Because of this unique time stamp every 6 seconds and that the receiver is continuously tracking the C/A code of the satellite, each chip of a C/A code can be uniquely identified with an exact time of transmission. Therefore, the receiver can resolve the ambiguity M in the code observable and hence can estimate $\tau(t)$ such that $\tau(t) = \Delta t(t) + \Delta T_{RX}(t) - \Delta T_{TX}(t)$.

Usually the code observable is in units of meters and is called the pseudorange. Converting $\tau(t)$ into meters by multiplying by the speed of light c results in the following pseudorange equation where the variable time has been removed for brevity and ρ is the range from the transmitter at transmission time to receive at reception time.

$$p = \rho + c(\Delta T_{RX} - \Delta T_{TX}) \quad (\text{A.16})$$

A.4.1.1 Calculating code based solutions

The satellite's current clock bias ΔT_{TX} is transmitted in the navigation data and therefore is a known value. The unknown values are therefore the receiver's position and clock bias; together these are $\mathbf{P} = [x, y, z, \Delta T_{RX}]$. This means the pseudorange is a function of these unknown variables. Obtaining one such pseudoranges for a satellite results in the following nonlinear equation.

$$p_n(\mathbf{P}) = \rho_n + c(\Delta T_{RX} - \Delta T_{TX}^n) \quad (\text{A.17})$$

Because of the ease of solving linear equations, linearization of this equation using a first order Taylor expansion is sensible for deriving a generalized method of solving sets of pseudoranges of arbitrary sizes. We let an estimated solution be $\hat{\mathbf{P}} = [\hat{x}, \hat{y}, \hat{z}, \Delta \hat{T}_{RX}]$ for time of reception. Then, a first order Taylor expansion for p_n around $\hat{\mathbf{P}}$ is as follows.

$$\begin{aligned} p_n(\mathbf{P}) \approx p_n(\hat{\mathbf{P}}) &+ (x - \hat{x}) \left. \frac{\partial}{\partial x} p_n \right|_{\hat{\mathbf{P}}} + (y - \hat{y}) \left. \frac{\partial}{\partial y} p_n \right|_{\hat{\mathbf{P}}} \\ &+ (z - \hat{z}) \left. \frac{\partial}{\partial z} p_n \right|_{\hat{\mathbf{P}}} + (\Delta T_{RX} - \Delta \hat{T}_{RX}) \left. \frac{\partial}{\partial \Delta T_{RX}} p_n \right|_{\hat{\mathbf{P}}} \end{aligned} \quad (\text{A.18})$$

Given a satellite's position $\mathbf{S}_n = [x_n, y_n, z_n]$ at time of transmission for a pseudorange p_n , then, the partial derivatives can be calculated given $\rho_n(x, y, z) = \sqrt{(x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2}$. Therefore, upon evaluation, equation A.18 can be written as follows.

$$\begin{aligned} p_n(\mathbf{P}) - p_n(\hat{\mathbf{P}}) \approx (x - \hat{x}) \frac{(\hat{x} - x_n)}{\hat{\rho}_n} \\ + (y - \hat{y}) \frac{(\hat{y} - y_n)}{\hat{\rho}_n} + (z - \hat{z}) \frac{(\hat{z} - z_n)}{\hat{\rho}_n} + (\Delta T_{RX} - \Delta \hat{T}_{RX}) c \end{aligned} \quad (\text{A.19})$$

APPENDIX A. GPS BASICS

Given a good estimate $\hat{\mathbf{P}}$, this equation has four independent unknowns, hence at least four equation just like it are needed to solve the unknowns. These four equations require the receiver's clock bias to be the same for all equations and the receiver's position to be the same for all equations. Therefore, the receiver has to obtain four pseudoranges simultaneously. A set of m such pseudoranges for different satellites obtained simultaneously are given below.

$$\begin{aligned} p_1(\mathbf{P}) &= \rho_1 + c(\Delta T_{RX} - \Delta T_{TX}^1) \\ &\vdots \\ p_m(\mathbf{P}) &= \rho_m + c(\Delta T_{RX} - \Delta T_{TX}^m) \end{aligned} \tag{A.20}$$

Because equation A.19 is linear this set of pseudo ranges can be written in matrix form as follows.

$$\begin{bmatrix} \frac{(\hat{x}-x_1)}{\hat{\rho}_1} & \frac{(\hat{y}-y_1)}{\hat{\rho}_1} & \frac{(\hat{z}-z_1)}{\hat{\rho}_1} & c \\ \vdots & \vdots & \vdots & \vdots \\ \frac{(\hat{x}-x_m)}{\hat{\rho}_m} & \frac{(\hat{y}-y_m)}{\hat{\rho}_m} & \frac{(\hat{z}-z_m)}{\hat{\rho}_m} & c \end{bmatrix} \begin{bmatrix} (x - \hat{x}) \\ (y - \hat{y}) \\ (z - \hat{z}) \\ (\Delta T_{RX} - \Delta \hat{T}_{RX}) \end{bmatrix} \approx \begin{bmatrix} p_1(\mathbf{P}) - p_1(\hat{\mathbf{P}}) \\ \vdots \\ p_m(\mathbf{P}) - p_m(\hat{\mathbf{P}}) \end{bmatrix} \tag{A.21}$$

Using bold type notation vectors or matrices, this has the form of $\mathbf{A}(\mathbf{P} - \hat{\mathbf{P}}) \approx \mathbf{b}$ where $\mathbf{A}$, $\mathbf{b}$, and $\hat{\mathbf{P}}$ are known. Using LS and rearranging for the unknown yields $\mathbf{P} \approx (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} + \hat{\mathbf{P}}$. The right-hand side of this equation when calculated yields only an approximation of the desired solution $\mathbf{P}$. Therefore $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} + \hat{\mathbf{P}}$ is another solution estimate and we denote it as $\hat{\mathbf{P}}_n = [\hat{x}_n, \hat{y}_n, \hat{z}_n, \Delta \hat{T}_{RX_n}]$. So we have seen a way of obtaining a new solution estimate from an old solution estimate. This process can be reiterated as the following algorithm describes.

Algorithm A.1 Iterative LS solution using code observable

1. $\hat{\mathbf{P}}_0 = [0, 0, 0, 0]$, $n = 0$
 2. increment n
 3. estimate reception time $\hat{t}_{rx} = T_{RX} - \Delta\hat{T}_{RX}$
 4. calculate $\mathbf{S}_m$ at time $\hat{t}_{rx}$
 5. calculate $\mathbf{S}_m$ at time $\hat{t}_{rx} - \|\mathbf{S}_m - [\hat{x}_n, \hat{y}_n, \hat{z}_n]\| / c$ and reiterate a few times
 6. if $\nabla(\mathbf{A}^T \mathbf{A})^{-1}$ stop with error
 7. $\hat{\mathbf{P}}_n = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} + \hat{\mathbf{P}}_{n-1}$
 8. if ($\|[\hat{x}_n, \hat{y}_n, \hat{z}_n] - [\hat{x}_{n-1}, \hat{y}_{n-1}, \hat{z}_{n-1}]\| < \text{less than desired error}$) stop
 9. if ($n > \text{to big}$) stop with error
 10. goto 2
-

The correction made to the time of reception as believed by the receiver T_{RX} to produce an estimate of the reception time in step 3, for most receivers would be small in the order of less than a millisecond and could be conceivably ignored. The correction needed for calculating an estimate of the time of transmission as performed in step 5 is generally comparatively large, and is in the order of around 60 ms being the approximate flight time from the satellite to the receiver; this step can't be ignored.

Figure A.5 shows an example of algorithm A.1 converging for a set of six satellites and code observations taken of them. As can be seen using the center of the Earth as the initial estimate within six iterations the algorithm has converged.

Figure A.5: Example of convergence of algorithm A.1

The output of this algorithm for the example in figure A.5 resulted in a receiver clock bias estimate of $-116.644 \mu s$. This compares to an estimate as computed by the receiver itself of $-116.738 \mu s$ with a spatial solution discrepancy of $32 m$ between the two.

A.4.1.2 Final code observable model

One reason why the discrepancy of $32 m$ between our solution and the solution as calculated by the GPS receiver itself is we have neglected some things in our modeling of equation A.16. The model in equation A.16 can be extended by adding tropospheric delays T , ionospheric delays I , multipath M_r and miscellaneous errors e_r . Therefore, a more exact model of the phase observable can be written as in the following equation.

$$p = \rho + c(\Delta T_{RX} - \Delta T_{TX}) + T + I + M_r + e_r \quad (\text{A.22})$$

Of the sources of error ionospheric are typically the greatest. The ionosphere stretches from 50 km to 1000 km above the Earth consisting mainly of charged particles, charged atoms and charged molecules. A large part of the ionization is caused by *UltraViolet* (UV) rays from the sun and hence there is a large

diurnal change in the *Total Electron Count* (TEC) which in turn effects the ionospheric correction term I . The ionosphere can produce a satellite range error as little as 1 m to as much as 100 m [13]. The ionospheric correction term is frequency dependent and with dual band receivers ionospheric free combinations of observables are possible. For singleband receivers no such combination is possible, instead Klobuchar ionospheric model is used for singleband receivers GPS. Klobuchar coefficients are transmitted in the navigation message so that the receiver can then estimate ionospheric correction terms. The Klobuchar algorithm corrects about 50% of the ionospheric errors [40].

The important thing for us in this section is that we are able to obtain an approximate solution, the spatial component with some sort of accuracy less than 100 m, and a time accuracy of some sort less than a $1\mu s$. This is all we are concerned about regarding the code observable.

A.4.2 The phase observable

The phase observable is a measured quantity taken by the GPS receiver for a particular satellite for a particular time. Phase observables allows higher accuracy GPS measurements to be made than compared to that of ones solely using code observables. This is due to the much shorter wavelength of GPS carrier than compared to the chip length of the code observable. The wavelength of L1 is approximately 20 cm compared to approximately 300 m length for the code chip of the C/A signal and can result in a correspondingly large increase in accuracy. The measurement comes from monitoring the phase difference between the received satellite carrier and a reference oscillator on the GPS receiver. The receiver accumulates this instantaneous phase difference by tracking and outputs this to the user as the phase observable. Figure A.6 shows a block diagram of what the GPS receiver is doing when observing a satellite for the phase observable neglecting all signals sent on the carrier such as C/A, navigation and P(Y) code.

Figure A.6: Simplified block diagram of phase measurement

For GPS, phase is customary in units of cycles rather than radians or degrees for GPS work. Phase is the argument inside a trigonometric function that accepts units of cycles. Φ^S is the phase of the carrier of the satellite while Φ^U is the receiver's reference phase. Both Φ^S and Φ^U can become arbitrarily large.

Inherent in accumulation of phase is an ambiguity N that depends on when you started accumulating phase. In addition to the ambiguity there is the possibility of missing some rotations. Counting the number of times a car tire rotates depends on when you started counting its rotations and also depends on whether you missed any rotations. While GPS receivers try to continuously track the phase, this is not always possible. Due to noise, loss of signal or turning the GPS receiver off and on again, the tracking of the phase can be lost resulting in an integer change in the value of N . This produces what is called a "cycle slip". So, ideally this phase ambiguity should be fixed while the satellite is being tracked but due to cycle slips occasionally it will change.

As we have seen satellites don't send out continuous waves, they are modulated with two BPSK signals, one in the quadrature phase and the other in the in-phase. Code tracking has the effect of wiping the C/A code from one of the BPSK signals but still leaves the navigation adding a level of complexity when trying to track it. When the BPSK data is not used to regenerate the original carrier wave a half cycle ambiguity in the carrier phase is introduced into the

APPENDIX A. GPS BASICS

phase observable and it is said that the phase observable has a code factor of two $C_f = 2$. When the BPSK data is used to regenerate the original carrier wave, the original carrier wave can be fully regenerated with an ambiguity of one cycle and it is said that the phase observable has a code factor of two $C_f = 1$. Thus the ambiguity of the phase observable can be reduced when a code factor of one is used.

A.4.2.1 Final phase observable model

As we have seen tracking the carrier phase maximizes γ which in turn means the accumulated phase while tracking is $\Phi = t\Delta f + \Delta t f_{TX} + \Delta\phi_0 + N/2$ for some fixed integer N when using a costas PLL. When considering the code factor this can be written as follows.

$$\Phi = t\Delta f + \Delta t f_{TX} + \Delta\phi_0 + N/C_f \quad (\text{A.23})$$

Assuming the receiver's clock is based around its local oscillator and its frequency keeps perfect time, true GPS time can be converted into the time as determined by the receiver as $T_{RX} = (tf_{RX} + \phi_0^{RX})/f_0$. By definition true time plus clock bias is also the time as determined by the receiver $T_{RX} = t + \Delta T_{RX}$. Equating the two and rearranging yields the clock bias in term of true GPS time. Likewise this can be done for the satellite's clock.

$$\Delta T_{RX} = (tf_{RX} + \phi_0^{RX})/f_0 - t \quad (\text{A.24})$$

$$\Delta T_{TX} = (tf_{TX} + \phi_0^{TX})/f_0 - t \quad (\text{A.25})$$

Subtracting the two and multiplying by f_0 results in the following.

$$f_0 (\Delta T_{RX} - \Delta T_{TX}) = t\Delta f + \Delta\phi_0 \quad (\text{A.26})$$

Equating this with equation A.23 we see that we can write equation A.23 as follows

APPENDIX A. GPS BASICS

$$\Phi = f_0 (\Delta T_{RX} - \Delta T_{TX}) + \Delta t f_{TX} + N/C_f$$

$$\Rightarrow \Phi = f_0 (\Delta T_{RX} - \Delta T_{TX}) + \frac{1}{\lambda_0} \rho f_{TX}/f_0 + N/C_f \quad (\text{A.27})$$

f_{TX} is the oscillator of the GPS transmitter and is an atomic clock being extremely close to f_0 . Therefore $f_{TX}/f_0 = 1$ for our purposes and we can rewrite equation A.27 as follows.

$$\Phi = f_0 (\Delta T_{RX} - \Delta T_{TX}) + \rho/\lambda_0 + N/C_f \quad (\text{A.28})$$

This model in equation A.28 can be extended by adding tropospheric delays T , ionospheric delays I , multipath M_ϕ and miscellaneous errors e_ϕ as was done with the code observable. However, the ionospheric correction for the phase while being of the same magnitude of that of the code observable is of the opposite sign. Therefore, a more exact model of the phase observable can be written as in the following equation.

$$\Phi = f_0 (\Delta T_{RX} - \Delta T_{TX}) + (\rho + T - I + M_\phi + e_\phi)/\lambda_0 + N/C_f \quad (\text{A.29})$$

The range term ρ in equation A.29 is for the receiver at reception time t_{RX} and the satellite at transmission time of t_{TX} . So the distance ρ is a measure of where you are to where the satellite was a short period of time ago because t_{TX} is an earlier time than the current time of t_{RX} ; the difference between these two values is typically around the 60 ms mark and a satellite can move a few hundred meters in this time.

The phase observable was measured at GPS time t_{RX} , this variable itself has to be solved for, as you are not going to know exactly what time the measurement was performed; you know you performed a measurement now, but you don't know when now is. This can be obtained using the code observable as previously shown.

The receiver's clock bias is at t_{RX} while the satellite's clock bias is at t_{TX} , but these aren't so critical as these don't change rapidly over 60 ms and can safely be assumed to be constant over the short time periods.

APPENDIX A. GPS BASICS

Equation A.29 is our final model for the phase observable. The left-hand side is what the receiver gives us, while the right-hand side is what we interpret it as. Multiplying it by the satellite's nominal wavelength is still classified as the single difference but rather than units of cycles for the phase observable, the phase observables units become meters.

A.5 Selected proofs

A.5.1 Received phase using flight time approximation.

Taylor expansion of flight time

$$\Delta t(t) = \sum_{k=0}^{\infty} \frac{\Delta t^{(k)}(t_m)}{k!} (t - t_m)^k$$

$$\Delta t(t) = \Delta t(t_m) + (t - t_m) \frac{v(t_m)}{c} + (t - t_m)^2 \frac{1}{2} \frac{a(t_m)}{c} + \dots$$

First order linear approximation

$$\Delta t(t) \approx \Delta t(t_m) + (t - t_m) \frac{v(t_m)}{c}$$

Define received phase

$$\Phi(t) = t\Delta f - \Delta t(t) f_{TX} + \Delta\phi_0$$

$$\rightarrow \Phi(t) \approx t\Delta f - \left(\Delta t(t_m) + (t - t_m) \frac{v(t_m)}{c} \right) f_{TX} + \Delta\phi_0$$

$$\rightarrow \Phi(t) \approx t \left(\Delta f - \frac{v(t_m)}{c} f_{TX} \right) + t_m \frac{v(t_m)}{c} f_{TX} - \Delta t(t_m) f_{TX} + \Delta\phi_0$$

$$\rightarrow \Phi(t) \approx t \left(\Delta f - \frac{v(t_m)}{c} f_{TX} \right) + \theta(t_m)$$

$$\rightarrow \Phi(t) \approx t(\Delta f - \Delta f_{TX}) + \theta(t_m) \quad \square$$

A.5.2 τ First-order linear approximation

$$\Delta t(t) \approx \frac{\Delta f_{TX}(t_m)}{f_{TX}} (t - t_m) + \text{constant}_1$$

$$\Delta T_{TX}(t) \approx \frac{f_{TX}(t_m) - f_0}{f_0} (t - t_m) + \text{constant}_2$$

$$\Delta T_{RX}(t) \approx \frac{f_{RX}(t_m) - f_0}{f_0} (t - t_m) + \text{constant}_3$$

$$\rightarrow \tau \approx (t - t_m) (\Delta f - \Delta f_{TX}(t_m)) \frac{-1}{f_0} + \Xi(t_m) \text{ as } f_{TX} \approx f_0$$

A.5.3 Radial velocity with constant radial velocity offset ϵ

What happens with constant velocity offset.

If radial velocity is out by ϵ at all times then

$$\lim_{t \rightarrow t_m} \Phi(t) = t\Delta f - \left(\Delta t(t_m) + (t - t_m) \frac{v(t_m)}{c} \right) f_{TX} + t_c^\epsilon f_{TX} + \delta + \Delta\phi_0$$

$$\therefore \lim_{t \rightarrow t_m} \Phi(t) = t(\Delta f - \Delta f_{TX}) + \theta'(t_m) + t_c^\epsilon f_{TX}$$

Where $\theta'(t_m)$ and δ are some constants

A.5.4 Maximum radial velocity and acceleration of the satellite with respect to the receiver

Figure A.7 is a simplified model of satellite orbiting the Earth while transmitting to a receiver. No relativistic effects are considered and it is assumed that the satellite's orbit is perfectly circular with constant tangential velocity and when the satellite is closest to the receiver the satellite is directly overhead.

Figure A.7: Simplified model of a satellite transmitting to a receiver while orbiting

From figure A.7 we obtain the following equations to describe the system.

$$\begin{aligned}
 x(t) &= r_s \cos\left(\frac{2\pi t}{T}\right) \\
 y(t) &= r_s \sin\left(\frac{2\pi t}{T}\right) - r_e \\
 \rho(t) &= \sqrt{x^2(t) + y^2(t)}
 \end{aligned} \tag{A.30}$$

The following approximate generally recognized values for GPS satellite orbital period T , radius of the earth and radius of GPS satellite orbits were used. By differentiating the range equation A.30 figure A.8 was obtained.

APPENDIX A. GPS BASICS

$$\begin{aligned}
 r_e &= 6371000 \text{ m} \\
 r_s &= 26600000 \text{ m} \\
 T &= 43080 \text{ s}
 \end{aligned}
 \tag{A.31}$$

Figure A.8: Modeled radial velocity and acceleration

The maximum radial velocity derived from the model was 929 m s^{-1} at satellite rise time and set time. Maximum acceleration was 0.1178 m s^{-2} and was when the satellite was directly overhead. [57] (pg 91) states that GPS satellites can have a radial velocity of up to 800 m s^{-1} with respect to a stationary receiver on earth. This is consistent with our simplified model. On the L1 band an acceleration of 0.1178 m s^{-2} is approximately a 0.9 Hz s^{-1} doppler shift rate while 929 m s^{-1} on the L1 band is approximately a 5 kHz doppler shift, therefore $f_0 \pm 5 \text{ kHz}$ must be searched for the carrier frequency.

Appendix B

Differencing the phase observable

Using multiple phase observations and taking the difference of them produces various differences that can be used finding position solutions. In this appendix we look at these differences and what form the position solutions take.

B.1 Single difference

A single difference is the difference between two phase observables at the same epoch for two different GPS receivers (two different users). Performing a single difference eliminates systematic errors when observing one satellite. Such errors include the satellite clock bias and to a large extent atmospheric errors.

We now use two GPS receivers, A and B to observe satellite S. The phase observables are measured at approximately the same time by both receivers. User A reports a time of T_A as reported by the system clock on the receiver itself, while user B reports that the clock on it's GPS receiver when taking it's measurement was T_B . The relationship between actual GPS time and times according to the clocks on the receivers themselves are as follows.

$$t_A = T_A - \Delta T_A \tag{B.1}$$

APPENDIX B. DIFFERENCING THE PHASE OBSERVABLE

$$t_B = T_B - \Delta T_B \quad (\text{B.2})$$

Using the pseudo-range observables it is possible to obtain the receivers clock biases ΔT_A and ΔT_B . A good estimate of the clock bias of less than $1 \mu s$ is assumed as given from now on.

Two observable models can be written down as follows where it is understood that the multipath errors and miscellaneous errors refer to the phase observable and not code.

$$\begin{aligned} \lambda \Phi_B^S(t_B) &= \rho_B^S(t_B) + c(\Delta T_B - \Delta T_S) + T - I + M_B^S + e_B^S + \lambda N_B^S / C_f \\ \lambda \Phi_A^S(t_A) &= \rho_A^S(t_A) + c(\Delta T_A - \Delta T_S) + T - I + M_A^S + e_A^S + \lambda N_A^S / C_f \end{aligned}$$

Two receivers one satellite near simultaneous observations (B.3)

Where ΔT_S is the satellite's clock bias, T the tropospheric correction, I the ionospheric correction, C_f the code factor, N_B^S and N_A^S integer ambiguities, M_B^S and M_A^S multipath correction, e_B^S and e_A^S miscellaneous errors, c speed of light, $\rho_B^S(t_B)$ and $\rho_A^S(t_A)$ receiver to satellite ranges, $\Phi_B^S(t_B)$ and $\Phi_A^S(t_A)$ receiver phase measurements, and λ is the satellite's nominally wavelength. The subscript on Φ indicates which receiver has performed the measurement and the superscript indicates to which satellite. $\rho_A^S(t)$ is the range from the satellite S at transmission time to the receiver A at reception time of t . N_A^S indicates that the unknown integer N is for receiver A monitoring satellite S. M_A^S is the phase multipath affecting the signal from satellite S to receiver A and e_A^S are the miscellaneous errors affecting the phase observable that the receiver measures for satellite S that are not modeled. These two phase observations ($\lambda \Phi_B^S$ and $\lambda \Phi_A^S$) are given in meters rather than cycles by multiplying by the satellite's nominal wavelength.

While the tropospheric and ionosphere corrections are not exactly the same for both receivers we make an assumption that they are. Also as measurements were not entirely done simultaneously, the satellite clock bias will be different when the two receivers take measurements, but here we assume the satellite clock bias ΔT_S is the same for both receivers.

APPENDIX B. DIFFERENCING THE PHASE OBSERVABLE

As the observations were not simultaneous the two users see the satellite in a different place; the following figure describes the situation.

Figure B.1: One satellite as seen from two different receivers when the observations are not simultaneous

We therefore extrapolate to a common epoch on the second t . We now have to adjust the phase observables as if we had actually taken the measurement at this common epoch. To do this we can use a first order approximation using a Taylor expansion on the two range terms and rearrange which results in the following.

$$\rho_B^S(t_B) \approx \rho_B^S(t) - \left. \frac{d\rho_B^S}{dt} \right|_{t_B} (t - t_B) \quad (\text{B.4})$$

$\rho_B^S(t_B)$ is the range of the satellite calculated from where the satellite was at transmission time to where the receiver is at reception time given a reception time of t_B . This can be written as $\rho_B^S(t_B) = \|\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)\|$ where $\mathbf{S}$ is the satellite's position, $\mathbf{B}$ is the receiver's position, t_B is reception time and $t_B^{tx} = t_B - \Delta t$ is the transmission time.

Substituting this back into the observable model in equation B.3 we get the following.

APPENDIX B. DIFFERENCING THE PHASE OBSERVABLE

$$\lambda\Phi_B^S(t_B) + \left. \frac{d\rho_B^S}{dt} \right|_{t_B} (t - t_B) \approx \rho_B^S(t) + c(\Delta T_B - \Delta T_S) + T - I + M_B^S + e_B^S + \lambda N_B^S / C_f \quad (\text{B.5})$$

The range rate $\frac{d\rho_B^S}{dt}$ can be calculated either by using an approximate receiver's position and using navigation data, or, by measuring the Doppler. To find the range rate using the receiver's approximate position, differentiating $\rho_B^S(t)$ using the chain rule produces the following (see B.4.1).

$$\left. \frac{d\rho_B^S}{dt} \right|_{t_B} = \frac{(\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)) \cdot \left(\left. \frac{d\mathbf{S}}{dt} \right|_{t_B^{tx}} - \left. \frac{d\mathbf{B}}{dt} \right|_{t_B} \right)}{\rho_B^S(t_B) + \frac{1}{c} \left. \frac{d\mathbf{S}}{dt} \right|_{t_B^{tx}} \cdot (\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B))} \quad (\text{B.6})$$

Range rate calculation

If we assume the receiver is more or less stationary and the range rate is much less than the speed of light then we obtain the following.

$$\left. \frac{d\rho_B^S}{dt} \right|_{t_B} \approx \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \left. \frac{d\mathbf{S}}{dt} \right|_{t_B^{tx}} \quad (\text{B.7})$$

Approximate range rate calculation

The maximum range rate as we have already seen is less than 800 ms^{-1} which is much less than the speed of light so it's a valid assumption. Generally $t - t_B$ will be much less than 1 ms which would produce a correction error of less than $10 \mu\text{m}$ in equation B.5 which is an acceptable error.

The terms on the left-hand side of equation B.5 are all obtainable. Φ is the measured phase observable at the common epoch t we have chosen, the actual time of the measurement t_B is known and the range rate we also can estimate. The left-hand side is a corrected version of the phase observable to account for the fact that the phase observables were not taken at the same time. We therefore rewrite the given phase observables as follows.

APPENDIX B. DIFFERENCING THE PHASE OBSERVABLE

$$\begin{aligned}\lambda\hat{\Phi}_B^S(t) &\approx \lambda\Phi_B^S(t_B) + \left.\frac{d\rho_B^S}{dt}\right|_{t_B} (t - t_B) \\ \lambda\hat{\Phi}_A^S(t) &\approx \lambda\Phi_A^S(t_A) + \left.\frac{d\rho_A^S}{dt}\right|_{t_A} (t - t_A)\end{aligned}\tag{B.8}$$

The hat on top of the Φ symbol is to show that we have not measured the phase at this time, but instead it is an estimate of it at that time. The time difference between say t and t_A is likely to be less than a millisecond which is a relatively short period of time for what we're talking about here and hence is an acceptable approximation. Upon substituting equations B.8 into equations B.3 we get the following phase observables extrapolated to the common epoch t .

$$\begin{aligned}\lambda\hat{\Phi}_B^S(t) &= \rho_B^S(t) + c(\Delta T_B - \Delta T_S) + T - I + M_B^S + e_B^S + \lambda N_B^S/C_f \\ \lambda\hat{\Phi}_A^S(t) &= \rho_A^S(t) + c(\Delta T_A - \Delta T_S) + T - I + M_A^S + e_A^S + \lambda N_A^S/C_f\end{aligned}$$

Two receivers one satellite extrapolated to a common epoch (B.9)

Taking the difference between the two observables in equation B.9 we get the following where $N_{BA}^S = N_B^S - N_A^S$, $\hat{\Phi}_{BA}^S(t) = \hat{\Phi}_B^S(t) - \hat{\Phi}_A^S(t)$, $\Delta\rho_{BA}^S(t) = \rho_B^S(t) - \rho_A^S(t)$, $M_{BA}^S = M_B^S - M_A^S$, and $e_{BA}^S = e_B^S - e_A^S$.

$$\lambda\hat{\Phi}_{BA}^S(t) = \Delta\rho_{BA}^S(t) + c(\Delta T_B - \Delta T_A) + M_{BA}^S + e_{BA}^S + \lambda N_{BA}^S/C_f$$

Single difference model extrapolated to a common epoch (B.10)

This single difference has the effect of removing errors that are common to both receivers. These errors include atmospheric ones and ones that relate to the satellite itself. It assumes that the two receivers are close enough such that atmospheric conditions are identical for both users and the time between these two observations to be as small as possible. However it also doesn't account for multi-path and miscellaneous errors such as noise.

B.1.1 Single difference interpretation

We now make the approximation that the satellite is much further away than the two receivers are to one another. This means the satellite appears almost at infinity to the receivers meaning that angle SAB is equal to the angle ABS, as is depicted in the following figure.

Figure B.2: Satellite almost appears at infinity for two receivers close to one another

GPS satellites are at least around 20,000 km away so for any baseline of a few tens of kilometers this is a valid assumption.

This means that there is a common vector pointing towards the satellite for both receivers. This we call the unit normal vector $\hat{\mathbf{n}}_S$. Below is a figure that depicts the current situation. We can see that there is a vector between the two receivers which we now call the position vector $\mathbf{P} = \mathbf{B} - \mathbf{A}$ and we also are able to visualize the range difference term in the right-hand side of equation B.10.

Figure B.3: Unit norm in direction of satellite at infinity

Taking the dot product of the position vector (displacement vector) with the unit normal vector we can rewrite the range difference term as follows.

$$\Delta\rho_{BA}^S(t) = \hat{\mathbf{n}}_S(t) \cdot \mathbf{P} \quad (\text{B.11})$$

This allows us to rewrite equation B.10 as follows.

$$\lambda\hat{\Phi}_{BA}^S(t) = \hat{\mathbf{n}}_S(t) \cdot \mathbf{P} + c(\Delta T_B - \Delta T_A) + M_{BA}^S + e_{BA}^S + \lambda N_{BA}^S / C_f$$

Model of a single difference extrapolated to a common epoch (B.12)

The position vector is what we want to solve for. The position vector tells how user B's position is related to user A.

Using the code observable it is possible to obtain autonomous code based solutions which also consist of the receiver clock bias. However, these are not accurate enough to allow an improvement in position accuracy using Equation B.12 compared the autonomous code based solutions themselves. The Ublox

LEA-6T which is a singleband receiver designed for precision timing applications claims “an accuracy of up to 15 ns” [15] which equates to theintroducing an error in equation B.12 of around 4.5 m. This gives us motivation for performing a double difference which eliminates systematic errors of the receivers themselves performing simultaneous observation

B.2 Double difference

Double differences are the differences of single differences. A double difference eliminates systematic errors of the receivers themselves when performing simultaneous observations. Such errors include receiver clock biases.

We take two single differences to two satellites S2 and S1 still using receiver A and B. we also require that these single differences are taken simultaneously. This simultaneity is possible as GPS uses CDMA which means the GPS receiver sees every satellites in the sky simultaneously. Equation B.13 shows the pair of single differences.

$$\begin{aligned}\lambda\hat{\Phi}_{BA}^{S2}(t) &= \hat{\mathbf{n}}_{S2}(t) \cdot \mathbf{P} + c(\Delta T_B - \Delta T_A) + M_{BA}^{S2} + e_{BA}^{S2} + \lambda N_{BA}^{S2}/C_f \\ \lambda\hat{\Phi}_{BA}^{S1}(t) &= \hat{\mathbf{n}}_{S1}(t) \cdot \mathbf{P} + c(\Delta T_B - \Delta T_A) + M_{BA}^{S1} + e_{BA}^{S1} + \lambda N_{BA}^{S1}/C_f\end{aligned}\quad (\text{B.13})$$

Taking the difference of the two we get the following where $\Delta\mathbf{n}_{S2S1} = \hat{\mathbf{n}}_{S2} - \hat{\mathbf{n}}_{S1}$, $M_{BA}^{S2S1} = M_{BA}^{S2} - M_{BA}^{S1}$, $e_{BA}^{S2S1} = e_{BA}^{S2} - e_{BA}^{S1}$, $N_{BA}^{S2S1} = N_{BA}^{S2} - N_{BA}^{S1}$ and $\hat{\Phi}_{BA}^{S2S1} = \hat{\Phi}_{BA}^{S2} - \hat{\Phi}_{BA}^{S1}$.

$$\lambda\hat{\Phi}_{BA}^{S2S1}(t) = \Delta\mathbf{n}_{S2S1}(t) \cdot \mathbf{P} + M_{BA}^{S2S1} + e_{BA}^{S2S1} + \lambda N_{BA}^{S2S1}/C_f$$

Model of double difference extrapolated to a common epoch (B.14)

Phase multipath can be as high as a quarter of a cycle for an observation and is not well correlated from one site to another [28] therefore the multipath component in the double difference can be as much as one cycle. The miscellaneous

error component depends on many things but if we assume it is solely due to the ability to accurately measure the phase and we assume we can measure the phase to 1% of a cycle we would expect the miscellaneous error component to be 4% of the cycle. Indeed because of this multipath can be the dominant source of positioning error when using double differences.

B.2.1 Double difference interpretation

$\Delta \mathbf{n}_{S_2 S_1}$ we call the delta norm, while any vector that is dotted with a position vector is called a direction vector. The delta norm is therefore also a direction vector. We notice that the direction vector for the double difference is no longer a unit vector. The interpretation of this can be visualized with the following diagram.

Figure B.4: Visual interpretation of the Delta norm

For brevity let $m = \hat{\Phi}_{BA}^{S_2 S_1}(t) C_f - C_f (M_{BA}^{S_2 S_1} + e_{BA}^{S_2 S_1}) / \lambda - N_{BA}^{S_2 S_1}$ and call it a corrected measurement, the double difference can be modeled as follows using our model in equation B.14.

$$\frac{C_f}{\lambda} \Delta \mathbf{n}_{S_2 S_1}(t) \cdot \mathbf{P} = m$$

Simplified model of the double difference (B.15)

APPENDIX B. DIFFERENCING THE PHASE OBSERVABLE

$\mathbf{P}$ is the unknown we wish to solve for so we treat it as a variable. As a variable many values of $\mathbf{P}$ exists that satisfy equation B.15, not just the one we're looking for of $\mathbf{P} = \mathbf{B} - \mathbf{A}$ as hence we can now no longer say $\mathbf{P} = \mathbf{B} - \mathbf{A}$. We now look for what form the solutions to equation B.15 take thus finding the solutions to our double difference model. Equating equation B.15 when $\mathbf{P} = \mathbf{B} - \mathbf{A} = \vec{\mathbf{AB}}$ which we know satisfies equation B.15 and is the actual answer, with the general case when $\mathbf{P}$ satisfies equation B.21 but need not equal $\vec{\mathbf{AB}}$ we obtain the following

$$\Delta \mathbf{n}_{S2S1} \cdot \mathbf{P} = \Delta \mathbf{n}_{S2S1} \cdot \vec{\mathbf{AB}} \quad (\text{B.16})$$

Upon rearranging we see

$$\Delta \mathbf{n}_{S2S1} \cdot (\mathbf{P} - \vec{\mathbf{AB}}) = 0 \quad (\text{B.17})$$

If we treat $\mathbf{P}$ and $\vec{\mathbf{AB}}$ as point vectors then this has the form of a plane. The end of the $\vec{\mathbf{AB}}$ vector is a point on this plane and the plane is normal to the delta norm. This can be visualized in the following figure.

Figure B.5: Double difference solution plane

This means the solution to equation B.14 forms a plane sitting on receiver B. We say that this plane is the solution plane of the double difference.

B.2.2 Double difference cycle slips

Ideally the integer $N_{BA}^{S_2S_1}$ in the double difference in equation B.14 would be a constant, but as we have already mentioned in subsection A.4.2 on page 188 the GPS receiver can lose lock causing what is called a cycle slip. Figure B.6 shows an example of how these cycle slips effect double differences.

Figure B.6: Cycle slips in double differences

As can be seen both half cycle and full cycle ambiguities have been detected therefore the receiver (a Ublox LEA-6T) is not using the navigation data to reconstruct the carrier wave when this plot was produced. In this case from the plot is easy to see where the cycle slips have occurred and what is needed to correct for them but this is not always so especially when the receivers are widely separated. Various ways have been devised to detect and/or correct double difference cycle slips. Uncorrected and/or undetected cycle slips severely affect position solution accuracy [46].

B.2.3 Double difference residual

A double difference residual (or simply a residual) is the difference between the double difference measured and the one expected given a position $\mathbf{P}$; in

cycles it is written as follows.

$$r = \frac{1}{\lambda} (\lambda\Phi - \Delta\mathbf{n} \cdot \mathbf{P}) \quad (\text{B.18})$$

Residuals are useful as a way of obtaining an idea of how correct a solution is or how much error is in a double difference.

B.3 Wrapped double difference

We define the following notation for the positive fractional component of a double difference with respect to satellite S1 and S2 and receivers A and B and call it the wrapped double difference where $\{\}$ is the sawtooth function.

$$C_f \hat{\phi}_{BA}^{S2S1}(t) = \{C_f \hat{\Phi}_{BA}^{S2S1}(t)\} \quad (\text{B.19})$$

Wrapped double difference

Equating this with our double difference model we see the following.

$$C_f \hat{\phi}_{BA}^{S2S1}(t) = \left\{ \frac{C_f}{\lambda} \Delta\mathbf{n}_{S2S1}(t) \cdot \mathbf{P} + C_f \left(M_{BA}^{S2S1} + e_{BA}^{S2S1} \right) / \lambda \right\} \quad (\text{B.20})$$

Model of wrapped double difference

This is our model for the wrapped double difference. The wrapped double difference removes the unknown integer N removing cycle slips.

B.3.1 Wrapped double difference interpretation

The wrapped double difference removes the unknown integer N by wrapping the double difference. When the wrapped double differences from one pair of satellites is plotted against time we get something that looks like the following figure. The data appears wrapped, hence its name.

Figure B.7: Wrapped double difference data for one pair of satellites using two GPS receivers spaced a few centimeters apart.

Figure B.7 is for two receivers placed only a few centimeters apart. For receivers that are at distances greater than this the plot one gets when plotting wrapped double differences becomes visually meaningless and looks like nothing but a homogeneous mass of points.

For brevity let $m_w = \{C_f \hat{\phi}_{BA}^{S2S1}(t) - C_f (M_{BA}^{S2S1} + e_{BA}^{S2S1}) / \lambda\}$ and call it a corrected wrapped measurement then our model for the wrapped double difference can be written as follows.

$$\left\{ \frac{C_f}{\lambda} \Delta \mathbf{n}_{S2S1}(t) \cdot \mathbf{P} \right\} = m_w$$

Simplified model of the wrapped double difference (B.21)

The form of the the solutions to our model of the wrapped double difference can be found in a similar way as was done for the double difference model in subsection B.2.1. We look for what form the solutions to equation B.21 take. Equating equation B.21 when $\mathbf{P} = \mathbf{B} - \mathbf{A} = \vec{\mathbf{AB}}$ which we know satisfies equation B.21 and as the actual answer, with the general case when $\mathbf{P}$ satisfies equation B.21 but need not equal $\vec{\mathbf{AB}}$ we obtain the following.

$$\begin{aligned}\left\{\frac{C_f}{\lambda}\Delta\mathbf{n}_{S2S1}(t) \cdot \mathbf{P}\right\} &= m_w \\ \left\{\frac{C_f}{\lambda}\Delta\mathbf{n}_{S2S1}(t) \cdot \vec{\mathbf{AB}}\right\} &= m_w\end{aligned}\tag{B.22}$$

Taking the difference of the two.

$$\begin{aligned}\left\{\frac{C_f}{\lambda}\Delta\mathbf{n}_{S2S1}(t) \cdot \mathbf{P}\right\} - \left\{\frac{C_f}{\lambda}\Delta\mathbf{n}_{S2S1}(t) \cdot \vec{\mathbf{AB}}\right\} &= 0 \\ \left\{\frac{C_f}{\lambda}\Delta\mathbf{n}_{S2S1}(t) \cdot \mathbf{P} - \frac{C_f}{\lambda}\Delta\mathbf{n}_{S2S1}(t) \cdot \vec{\mathbf{AB}}\right\} &= 0 \\ \left\{\frac{C_f}{\lambda}\Delta\mathbf{n}_{S2S1} \cdot \left(\mathbf{P} - \vec{\mathbf{AB}}\right)\right\} &= 0\end{aligned}\tag{B.23}$$

This is similar to the form the solutions take for the double differences except this time due to the sawtooth function there are an infinite number of planes. To remove the sawtooth function we note that there exists a $k \in \mathbb{Z}$ such that

$$\frac{C_f}{\lambda}\Delta\mathbf{n}_{S2S1} \cdot \left(\mathbf{P} - \vec{\mathbf{AB}}\right) + k = 0\tag{B.24}$$

Which means

$$\Delta\mathbf{n}_{S2S1} \cdot \left(\mathbf{P} - \vec{\mathbf{AB}}\right) + k\frac{\lambda}{C_f} = 0\tag{B.25}$$

This can be rewritten as follows where $\mu_{S2S1} = \lambda / (C_f \|\Delta\mathbf{n}_{S2S1}\|)$.

$$\begin{aligned}\Delta\mathbf{n}_{S2S1} \cdot \left(\left(\mathbf{P} + k\mu_{S2S1}\frac{\Delta\mathbf{n}_{S2S1}}{\|\Delta\mathbf{n}_{S2S1}\|}\right) - \vec{\mathbf{AB}}\right) &= 0 \\ \text{Wrapped double difference solution form} &\end{aligned}\tag{B.26}$$

Thus the solutions form an infinite set of planes. One plane is the same as the solution plane for the double difference and the others are parallel to this

plane where each plane has a fixed spacing of $\mu_{S_2S_1}$ meters between each other. Therefore we call $\mu_{S_2S_1}$ the plane spacing. The figure below visualizes three of the set of infinite solution planes.

Figure B.8: Three of the infinite set of wrapped double difference solution planes

B.3.2 Wrapped phase observable

We defined the wrapped phase observable as follows.

$$C_f \phi_B^S(t_B) = \{C_f \Phi_B^S(t_B)\}$$

Wrapped phase observable (B.27)

This can be extrapolated to common epoch t as follows using equation B.8.

$$\begin{aligned}
 & \left\{ C_f \phi_B^S(t_B) + \frac{C_f}{\lambda} \frac{d\rho_B^S}{dt} \Big|_{t_B} (t - t_B) \right\} \\
 = & \left\{ \left\{ C_f \Phi_B^S(t_B) \right\} + \frac{C_f}{\lambda} \frac{d\rho_B^S}{dt} \Big|_{t_B} (t - t_B) \right\} \\
 = & \left\{ C_f \Phi_B^S(t_B) + \frac{C_f}{\lambda} \frac{d\rho_B^S}{dt} \Big|_{t_B} (t - t_B) \right\} \\
 = & \left\{ C_f \hat{\Phi}_B^S(t) \right\} \text{ by equation B.8} \\
 = & C_f \hat{\phi}_B^S(t) \text{ by definition}
 \end{aligned} \tag{B.28}$$

A single difference can then be formed given two receivers A and B that we call a wrapped single difference $\hat{\phi}_{BA}^S(t)$.

$$\begin{aligned}
 & \left\{ C_f \hat{\phi}_B^S(t) - C_f \hat{\phi}_A^S(t) \right\} \\
 = & \left\{ C_f \hat{\Phi}_B^S(t) - C_f \hat{\Phi}_A^S(t) \right\} \\
 = & \left\{ C_f \hat{\Phi}_{BA}^S(t) \right\} \\
 = & C_f \hat{\phi}_{BA}^S(t)
 \end{aligned} \tag{B.29}$$

And finally given two wrapped single differences to two different satellites S1 and S2 a wrapped double difference can be formed.

$$\begin{aligned}
 & \left\{ C_f \hat{\phi}_{BA}^{S2}(t) - C_f \hat{\phi}_{BA}^{S1}(t) \right\} \\
 = & \left\{ C_f \hat{\Phi}_{BA}^{S2}(t) - C_f \hat{\Phi}_{BA}^{S1}(t) \right\} \\
 = & \left\{ C_f \hat{\Phi}_{BA}^{S2S1}(t) \right\} \\
 = & C_f \hat{\phi}_{BA}^{S2S1}(t)
 \end{aligned} \tag{B.30}$$

Thus it is possible to form wrapped double differences extrapolated to a com-

mon epoch given wrapped phase observables. The advantages of doing it this way is that fewer bits are needed to represent a wrapped phase observable compared to a normal phase observable for a given accuracy; this saves network bandwidth.

Due to the properties of the sawtooth function, when creating a wrapped double difference from four wrapped observations, only one sawtooth operation is needed; this can be written as follows.

$$C_f \hat{\phi}_{BA}^{S2S1}(t) = \left\{ \left(C_f \hat{\phi}_B^{S2}(t) - C_f \hat{\phi}_A^{S2}(t) \right) - \left(C_f \hat{\phi}_B^{S1}(t) - C_f \hat{\phi}_A^{S1}(t) \right) \right\} \quad (\text{B.31})$$

B.3.3 Unwrapping the wrapped double difference

Unwrapping is a method of re-creating the double difference from the wrapped double difference.

First off we define the sawtooth difference function as follows.

$$f(x) = \{x + 0.5\} - 0.5 \quad (\text{B.32})$$

Here we define an algorithm that does unwrapping and is used in the CFF algorithm. $\alpha \in [0, 1]$ and determines how sensitive the algorithm is to sudden changes in input.

Algorithm B.1 Unwrapping algorithm

1. Let $k = 0$
 2. $y_0 = f(x_0)$
 3. $x_k = C_f \hat{\phi}_{BA}^{S2S1}(t_k) - \frac{C_f}{\lambda} \Delta \mathbf{n}_{S2S1}(t_k) \cdot \hat{\mathbf{P}}$
 4. Increment k
 5. $y_k = f(x_k - a_{k-1}) + a_{k-1}$
 6. Let $a_k = \alpha y_k + (1 - \alpha) a_{k-1}$
 7. If not end of wrapped double differences goto 4
 8. $C_f \hat{\Phi}_{BA}^{S2S1}(t_k, \hat{\mathbf{P}}) = y_k + \frac{C_f}{\lambda} \Delta \mathbf{n}_{S2S1}(t_k) \cdot \hat{\mathbf{P}}$
-

This unwrapping algorithm needs as an input an approximate position estimate $\hat{\mathbf{P}}$. This is because the phase of the double difference for a pair of satellites can change very rapidly with respect to time, using a position estimate in effect slows the wrapping effect so as to increase the likelihood that the unwrapping algorithm will be successful and not introduce cycle slips. The figure below shows an example of this algorithm on one pair of satellites with a position estimate half a meter different from reality for the residuals before and after the unwrapping has been performed.

Figure B.9: Unwrapping algorithm given a position estimate half a meter out from reality

The unwrapping algorithm can make mistakes by introducing cycle slips, particularly when the data is intermittent as is often the case for our scenario. However, the unwrapping algorithm, as we shall see, performs surprisingly well for static scenarios such as ours.

B.3.4 Wrapped double difference residual

A wrapped double difference residual (or simply a wrapped residual) is the sawtooth difference between the wrapped or unwrapped double difference measured and the one expected given a position $\mathbf{P}$; in cycles it is written as follows.

$$r = f \left(\frac{C_f}{\lambda} (\lambda\Phi - \Delta\mathbf{n} \cdot \mathbf{P}) \right) / C_f \quad (\text{B.33})$$

Wrapped residuals in contrast to double difference residuals are cycle slip invariant. Wrapped residuals like double difference residuals express the error between what is measured in what is expected.

B.4 Selected proofs

B.4.1 Range rate calculation

Given $\rho_B^S(t_B) = \|\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)\|$ and $t_B^{tx} = t_B - \rho_B^S(t_B)/c$ then derive for 2D.

$$\begin{aligned}
 \frac{d\rho_B^S(t_B)}{dt} &= \frac{d}{dt} \left(\sqrt{(S_x(t_B^{tx}) - B_x(t_B))^2 + (S_y(t_B^{tx}) - B_y(t_B))^2} \right) \\
 &= \frac{1}{2} \left((S_x(t_B^{tx}) - B_x(t_B))^2 + (S_y(t_B^{tx}) - B_y(t_B))^2 \right)^{-1/2} \\
 &\quad \times \frac{d}{dt} \left((S_x(t_B^{tx}) - B_x(t_B))^2 + (S_y(t_B^{tx}) - B_y(t_B))^2 \right) \\
 &= \frac{1}{2} \left(1/\sqrt{(S_x(t_B^{tx}) - B_x(t_B))^2 + (S_y(t_B^{tx}) - B_y(t_B))^2} \right) \\
 &\quad \times \frac{d}{dt} \left((S_x(t_B^{tx}) - B_x(t_B))^2 + (S_y(t_B^{tx}) - B_y(t_B))^2 \right) \\
 &= \frac{1}{2\rho_B^S(t_B)} \frac{d}{dt} \left((S_x(t_B^{tx}) - B_x(t_B))^2 + (S_y(t_B^{tx}) - B_y(t_B))^2 \right) \\
 &= \frac{1}{2\rho_B^S(t_B)} \left(2(S_x(t_B^{tx}) - B_x(t_B)) \frac{d}{dt} (S_x(t_B^{tx}) - B_x(t_B)) \right. \\
 &\quad \left. + 2(S_y(t_B^{tx}) - B_y(t_B)) \frac{d}{dt} (S_y(t_B^{tx}) - B_y(t_B)) \right) \\
 &= \frac{1}{\rho_B^S(t_B)} \left((S_x(t_B^{tx}) - B_x(t_B)) \frac{d}{dt} (S_x(t_B^{tx}) - B_x(t_B)) \right. \\
 &\quad \left. + (S_y(t_B^{tx}) - B_y(t_B)) \frac{d}{dt} (S_y(t_B^{tx}) - B_y(t_B)) \right)
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \left[\frac{d}{dt} (S_x(t_B^{tx}) - B_x(t_B)), \frac{d}{dt} (S_y(t_B^{tx}) - B_y(t_B)) \right] \\
 &= \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \frac{d(\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B))}{dt} \\
 &= \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \left(\frac{d\mathbf{S}(t_B^{tx})}{dt} - \frac{d\mathbf{B}(t_B)}{dt} \right) \\
 &= \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \left(\mathbf{S}'(t_B^{tx}) \frac{dt_B^{tx}}{dt} - \mathbf{B}'(t_B) \right) \\
 &= \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \left(\mathbf{S}'(t_B^{tx}) \frac{d}{dt} (t_B - \rho_B^S(t_B)/c) - \mathbf{B}'(t_B) \right) \\
 &= \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \left(\mathbf{S}'(t_B^{tx}) \left(1 - \frac{1}{c} \frac{d\rho_B^S(t_B)}{dt} \right) - \mathbf{B}'(t_B) \right) \\
 &= \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \left((\mathbf{S}'(t_B^{tx}) - \mathbf{B}'(t_B)) - \mathbf{S}'(t_B^{tx}) \frac{1}{c} \frac{d\rho_B^S(t_B)}{dt} \right) \\
 &= \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot (\mathbf{S}'(t_B^{tx}) - \mathbf{B}'(t_B)) \\
 &\quad - \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \mathbf{S}'(t_B^{tx}) \frac{1}{c} \frac{d\rho_B^S(t_B)}{dt} \\
 &\rightarrow \frac{d\rho_B^S(t_B)}{dt} + \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \mathbf{S}'(t_B^{tx}) \frac{1}{c} \frac{d\rho_B^S(t_B)}{dt} \\
 &\quad = \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot (\mathbf{S}'(t_B^{tx}) - \mathbf{B}'(t_B))
 \end{aligned}$$

$$\begin{aligned}
 & \rightarrow \frac{d\rho_B^S(t_B)}{dt} \left(1 + \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \mathbf{S}'(t_B^{tx}) \frac{1}{c} \right) \\
 & = \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot (\mathbf{S}'(t_B^{tx}) - \mathbf{B}'(t_B))
 \end{aligned}$$

$$\begin{aligned}
 & \rightarrow \frac{d\rho_B^S(t_B)}{dt} = \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot (\mathbf{S}'(t_B^{tx}) - \mathbf{B}'(t_B)) \\
 & \quad / \left(1 + \frac{\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)}{\rho_B^S(t_B)} \cdot \mathbf{S}'(t_B^{tx}) \frac{1}{c} \right)
 \end{aligned}$$

$$\begin{aligned}
 & = \frac{(\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B))}{1} \cdot (\mathbf{S}'(t_B^{tx}) - \mathbf{B}'(t_B)) \\
 & \quad / \left(\rho_B^S(t_B) + \frac{1}{c} \frac{(\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B))}{1} \cdot \mathbf{S}'(t_B^{tx}) \right)
 \end{aligned}$$

$$= \frac{(\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)) \cdot (\mathbf{S}'(t_B^{tx}) - \mathbf{B}'(t_B))}{\rho_B^S(t_B) + \frac{1}{c} \mathbf{S}'(t_B^{tx}) \cdot (\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B))}$$

$$\Rightarrow \left. \frac{d\rho_B^S}{dt} \right|_{t_B} = \frac{(\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)) \cdot \left(\left. \frac{d\mathbf{S}}{dt} \right|_{t_B^{tx}} - \left. \frac{d\mathbf{B}}{dt} \right|_{t_B} \right)}{\rho_B^S(t_B) + \frac{1}{c} \left. \frac{d\mathbf{S}}{dt} \right|_{t_B^{tx}} \cdot (\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B))}$$

For 3D the extension is obvious and results in same formula.

$$\left. \frac{d\rho_B^S}{dt} \right|_{t_B} = \frac{(\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B)) \cdot \left(\left. \frac{d\mathbf{S}}{dt} \right|_{t_B^{tx}} - \left. \frac{d\mathbf{B}}{dt} \right|_{t_B} \right)}{\rho_B^S(t_B) + \frac{1}{c} \left. \frac{d\mathbf{S}}{dt} \right|_{t_B^{tx}} \cdot (\mathbf{S}(t_B^{tx}) - \mathbf{B}(t_B))}$$

Appendix C

Node functionality at night

As a proof of concept we designed and implemented an energy distribution algorithm based on the circadian rhythm in addition to using an *Assisted Global Positioning System* (AGPS) system that injected timing, location and ephemerides into the GPS receivers. A 33 F capacitor was used for this test.

C.1 Assisted GPS

To spread the observations obtained reasonably uniformly at night using such an extremely small energy storage devices as a 33 F capacitor, an extremely low duty cycle with an extremely low on time was needed (Chapter 5 of [44]); this motivated us to design a AGPS system. The AGPS system supplied the ephemerides, the almanac, and an approximate time and position to the GPS module via an alternative link rather than the normal 50baud communication channel from the satellites. This allowed the GPS receivers to save time not having to listen to the satellites just to obtain the ephemerides and almanac data.

Our implementation of assisted GPS comprised of obtaining the ephemerides, the almanac, and approximate time from the Assistnow server provided by Ublox. This data was saved on a local PC and synchronized periodically with the Assistnow server. This PC continuously sent out this data at a rate of approximately 250kb/s on a dedicated channel with a distinct *Personal Area Network* (PAN) *IDentification* (ID) (the AGPS channel) using an 802.15.4

APPENDIX C. NODE FUNCTIONALITY AT NIGHT

radio. Another channel with a different PAN ID (the main channel) was used for retrieving the data from the GPS node and sending it back to the *Personal Computer* (PC). Figure C.1 shows this communication implementation.

Figure C.1: AGPS network structure

Acknowledgments were required for the main channel due to cross talk from the AGPS channel to avoid packet losses on the main channel.

The amount of assisted GPS data at any particular time is not constant but is around the 3 kB amount. The GPS receiver could be initialized with GPS data within around 300 ms this compares to around 15 minutes to download this data directly from satellites.

We performed 10 trials with a clear view of the sky with and without our implementation of assisted GPS data without backup power being supplied to the GPS receiver. The TTFF was recorded for each trial and the results plotted in the figure below.

Figure C.2: Time to first fix without GPS backup power

As can be seen the TTFF when using the AGPS system is considerably lower than when not using it. The average time when not using AGPS was 42 seconds compared to 18 seconds when using the AGPS system. This TTFF test was then performed again supplying the GPS receiver with backup power using approximately 30 μ W according measurements; a period of 60 seconds was used between each trial. The results are pictured in the following figure.

Figure C.3: Time to first fix with GPS backup power

The reason the first GPS trial TTFF is much greater than the remaining trials is due to the fact that the GPS backup power had not been given power until that trial. Excluding the first trial our AGPS implementation produced a TTFF of 4 seconds compared to 10 seconds when not using it.

Both with and without GPS backup power TTFF is approximately half in both cases.

The period of time between trials is important when using GPS backup power as whatever information the GPS receiver holds when the backup power is applied will age; given a long enough period of time whatever information it holds will become useless. For example ephemerides are generally not used beyond four hours of age. Therefore a longer interval of 30 minutes was performed when using the AGPS and supplying GPS backup power. Seven sequential trials of this type was taken and the results are plotted in the figure below.

Figure C.4: Time to first fix for AGPS with GPS backup power and 30 minutes between trials

This test, excluding the first trial, the average TTFF was 5 seconds. This shows that periods of node inactivity of up to 30 minutes affects the TTFF when using AGPS and GPS backup power minimally.

The duty cycle at night can be calculated as follows

$$D = \frac{0.5C(v_i^2 - v_f^2)}{T_{night}P_{GPS}} \quad (C.1)$$

As can be seen the duty cycle is a function of night length. Therefore assuming a night length of 12 hours we see the duty cycle approximately 0.5% which needless to say is an extremely low duty cycle.

The wasted percentage of on time is given as follows where t_{on} is the on time for each active part of the duty cycle.

$$W_{on} = \frac{t_{init}}{t_{on}} \quad (C.2)$$

While the number of active periods throughout the night is given as follows.

$$n = \frac{E_C}{t_{on}P_{GPS}} \quad (C.3)$$

Therefore we take a wasted percentage of on time of 50% which implies, given a TTFF of 5 seconds using our AGPS, an on time period of 10 seconds. This in turn implies approximately 20 active periods throughout the night. Each active period lasts 10 seconds and half of this energy used for each of these active periods is wasted due to initialization. While the duty cycle is 0.5%, as half of this percentage is wasted due to initialization we expect only 0.25% of the night to obtain observations; this we call effective duty cycle.

C.2 Circadian energy distribution algorithm

To perform measurements at night energy stored on a capacitor has to be distributed throughout the night to perform this task. In addition to this there must be enough energy reserves throughout the night to run other such things like the microprocessor. To allow this to happen we designed and implemented an energy distribution algorithm based on the idea of the circadian rhythm. The following describes the algorithm.

The first assumption is that it is always possible to fully charge the capacitor during one day of daylight hours. If not, then we say the capacitor is too big for the solar panel or the solar panel was not big enough. This assumption has to be checked for every solar panel and capacitor combination.

The second assumption is that all periods of time used to obtain observations take the same period of time and energy. Each one of these periods of times is called an event.

There are two types of events triggered and spaced.

Triggered events can happen at any time. They happen when the voltage of the capacitor exceeds the fixed voltage of 2.2 V.

The other type of event is the spaced event. These events happen evenly spaced between the last triggered event and the first trigger event as pictured in the following figure.

APPENDIX C. NODE FUNCTIONALITY AT NIGHT

Figure C.5: Circadian distribution of stored energy

The sum of the energy of spaced events between the last and the first triggered events during the spaced event phase is roughly equal to the usable energy of the capacitor.

The first spaced event happens $\alpha/(n+1)$ after the last triggered event and happens approximately every $\alpha/(n+1)$ until the first trigger event. n is how many events a fully charged capacitor can do without going flat and without getting any new energy input, while α is the period of time between the last triggered event and the first triggered event during the spaced event phase.

When the node starts doing spaced events it will not do triggered events until at least a third of the spaced events have occurred. Under voltage is monitored during the spaced events phase and if encountered will stop doing spaced events. An event will be extended if the voltage is still above a fixed voltage say 2.2 V. The spacing of events is updated every time a spaced event happens. The number of spaced events is calculated by taking into consideration the current capacitor voltage after an event, the amount of energy each spaced event uses and how long away the next triggered event could possibly be. The node keeps a *Real Time Calendar Clock* (RTCC) active at all times. The RTCC is not set any particular time in the world it only cycles around every 24 hours. It is not necessary for it to know what the time is but it is necessary for it to have an idea on how long a rotation of the earth takes around its own axis. It performs the same function as our circadian rhythm.

C.3 Circadian energy distribution algorithm and AGPS system implementation results

The circadian energy distribution algorithm along with the AGPS system as described was then implemented on hardware and was tested. The following describes the results obtained from such a system.

At 5:00 PM a node was placed outside for approximately 18 hours and it's super capacitor voltage was recorded; figure C.6 shows the results of these measurements; as can be seen this graph looks similar to the one in figure C.5.

Figure C.6: Voltage over super capacitor at night

The short sharp drops throughout the night that can be seen in figure C.6 are caused by the GPS going active and taking measurements. The reason that there are only approximately 14 measurements throughout the night rather than closer to the 20 measurements as expected throughout the entire period between the last triggered event and the first triggered event is because the nodes circadian rhythm is not synchronized until a full 24 hours has past.

Performing a test lasting approximately 75 hours the autonomous code based position solutions of GPS module were recorded, the mean of this set of data was calculated and then for each position solution the offset from this mean was calculated and plotted in figure C.7.

Figure C.7: Autonomous code based solution offset from the mean code based solution with respect to time

The lower density places in figure C.7 are the position solutions obtained at night, the heavy density places are the daytime solutions. For this test one standard deviation equated 2.9 m. Approximately 7.7 position solutions were calculated for each spaced event. Nighttime approximately lasted 13 hours at the particular time of year at that location meaning a solution density at night of 0.003 position solutions a second, or an effective duty cycle of 0.3%; this compares very well with the approximate expected effective duty cycle of 0.25%.

During the day, to all intents and purposes we can say the duty cycle was 100% by comparison, therefore 99.7% of the solutions were obtained during the day. However, the nighttime data was relatively evenly spaced throughout the night which was the goal of the energy distribution. Therefore our energy distribution algorithm is effective at operating the GPS receiver at night in such a way as not to cluster the observations it gets to greatly. A possible advantage of obtaining observations at night might be as a way of mitigating solution biases. In addition the GPS receiver is able to operate at times at night and at least obtain autonomous code based solution for those times of night increasing temporal information.

APPENDIX C. NODE FUNCTIONALITY AT NIGHT

Due to time constraint, added complexity and that we were primarily interested in static solutions where many observations usually from a 24 hour period form one solution, distributing observations throughout the night was not taken further. However we have shown that it is indeed possible to obtain solutions at night using extremely low duty cycles and or very large capacitors. Clearly many other new algorithms could be created with greater complexity to allow such techniques as sidereal filtering and obtaining epoch based solutions.

Appendix D

Network

Data has to transfer between the nodes and the main computer that performs final processing. We divided this into two sections; the WSN section and the Internet section. The Internet section consists of the data flow from when it arrives at base station after any processing has been done such as time ambiguity resolution, duplicate packet removal, non-sequential packet removal, etc. The WSN section consists of the flow of data and processing between the processed data of the nodes until it is handed over to the Internet section.

D.1 Internet section

The Internet structure of the network is a castor, server, client structure. This is the same idea that Networked Transport of *RTCM via Internet Protocol* (NTRIP) uses for accessing GPS receivers over the Internet. Our own implementation of a custom castor, server, client structure was implemented that we called JAC packet. The following figure shows the structure for two testbeds.

APPENDIX D. NETWORK

Figure D.1: Caster server client structure over the Internet

The device server acts as a driver for both the WSN gateway devices and physically connected GPS receivers as well as a supplier of data to the JAC packet caster. As such some of its tasks are for the Internet section while others are for the WSN section. In the previous figure the dotted line on the device server represents the junction between its two tasks. The device server acts as a server that the JAC packet caster can connect to to obtain data wanting to be logged at the JAC packet server. The JAC packet caster then attaches to the JAC packet server via a TCP connection that is kept open and the JAC packet server logs the incoming data from each test bed. JAC packet clients can then connect to the JAC packet server and either obtain the logged data from the test a bed or a live stream from one of the test beds. For a JAC packet caster to connect to a JAC packet server authentication is via a username and password; no authentication requirement was implemented for JAC packet clients. No logging of data was performed on the base station

meaning if the Internet connection went down data would be lost.

D.2 The WSN section

During the design phase initially a single hop network was envisaged as the process of obtaining solutions was more important than creating difficult multihop networks. However, in a real life testbed in LuShan Taiwan the node transmission range were found not to be sufficient and a way to extend the range was needed. Due to both limited time and limited resources on the nodes, a dedicated repeater node was designed and implemented to extend the range.

The following picture shows the nodes, repeaters and the part of the base station that is dedicated to creating the data that is handed over to the Internet section.

Figure D.2: WSN and interface

D.2.1 Repeater

To use the repeaters two extra bytes are added to each data packet. One byte is a sequence number while the other byte is 8 flags where each flag corresponds to the modulo 8 of the repeater node number. The data packet sequence number is not the same as the sequence number in the header of each 802.15.4 packet.

APPENDIX D. NETWORK

Every time a packet is transmitted by the node the data packet sequence number will increase. At the same time the flag byte transmitted will be set to zero. Therefore anyone listening to data packets will be able to tell the order of such packets, have any of them been missed, in addition to knowing that it comes from a node and not a repeater as the flag byte has been set to zero.

A repeater contains a small list of the packets it has recently sent by storing the 802.15.4 header packet sequence number and source node number. When a repeater hears a packet if the repeater finds that it has already sent the packet it will not send it again; this helps prevent infinite transmission loops from occurring and repeating data the repeater has already sent. In addition when a repeater hears a packet, if the repeater finds that its bit of the repeater flag byte is already set it will not send this packet again; while maybe not strictly necessary this totally prevents infinite transmission loops from occurring.

When a repeater sends a packet it spoofs the 802.15.4 header so as to match the originating node. This spoofed information means both the source MAC address and the sequence number in the 802.15.4 header match that of the originating node. The repeater sets the bit of the repeater flag byte that corresponds to the repeater's node number. In addition it updates the list of transmitted packets using node number and the 802.15.4 header packet sequence number. The reason for using the 802.15.4 header packet sequence number rather than the data packet sequence number is to allow the nodes the opportunity to send duplicate packets to reduce packet loss.

We are aware that the possibility still remains that two repeaters transmit simultaneously due to hearing the same packet from a node. However, if the repeaters can hear one another then the csma/ca algorithm should at least try to mitigate this effect so that the two resulting packets from the repeaters are sent sequentially.

D.2.2 Device server

The other half of the device server's role is to resolve time ambiguity from the "node algorithm data" that it hears, remove duplicate packets, and remove non sequential packets.

First the the device server resolves the time ambiguity. To remove duplicate packets and non-sequential packets the device server does the following. It keeps track of the last 10 unique data packets received from each node in a cycle buffer. If the packet is found in the cycle buffer it is taken as a duplicate and dropped. If it's not a duplicate, the packet is put into the cycle buffer. If the GPS time of the packet is older than any of the other 10 packets in the cycle buffer the packet is dropped. Else, it is passed to the Internet section. This prevents both duplicate packets and non-sequential order of the packets being passed to the Internet section.

D.2.3 Phasing

The repeaters simply repeats packets they hear; they have no knowledge as to when the packet will arrive. Because of this for the repeaters to function as repeaters, their radios have to be continuously on, which uses approximately 70% of the power that having the GPS receiver on uses. This is considerable power meaning duty cycling is required.

While there may not be enough irradiance to keep any one node active continuously, it is possible for there to always be one node active during daylight hours if the nodes all take a turn being active thus forcing the repeater to be on continuously. Of course if there is not enough irradiance, one repeater cannot be in an active state continuously. If the nodes and the repeaters were all somehow synchronized to become active at the same time and both nodes and repeaters were obtaining approximately the same amount of irradiance, the nodes would then all become inactive before the repeater had to, as the nodes use somewhat more power than the repeater; thus the repeater would be fully serving the nodes needs.

Another problem arises when a repeater is active and all nodes are inactive. In such a case when a node finally does become active the repeater may have to become inactive to save energy unable to function as a repeater for the active node. Therefore it makes sense for a repeater to become inactive when node activity ceases.

From these two problems the idea of phasing all nodes and repeaters in the WSN network was formed. The idea was to make the repeaters and nodes

become active at the same time and become inactive at the same time; such as to be in phase with one another.

As double differences can only be formed from observations from two GPS receivers that are on at the same time, phasing the nodes to become active at the same time means more double differences can be formed from nodes themselves than if the nodes were not in phase with one another. While an interesting observation, due to time constraints this effect was not investigated.

D.2.3.1 Phasing algorithm

We designed and implemented a phasing algorithm on our nodes and repeaters. The main assumption of this algorithm is that all nodes and repeaters obtain the same irradiance.

With the assumption that the irradiance is the same for all nodes and repeaters, all nodes can have the same duty cycle and on time, at the same time as the repeaters use slightly less power when active than the nodes, the repeaters can at least match the duty cycle and on time of the nodes.

Whether or not a node is obtaining useful observations it will send data packets to inform both nodes and repeaters that the node has its GPS receiver on; these will be empty when no useful observations are obtainable else they will contain the usual “node algorithm data”. If they are empty they are dropped at the device server.

When a node’s super capacitor reaches 2.2 V it is considered to be full and the node will start a phasing period that times out after 10 minutes if no GPS packet are heard. The phasing algorithm as run on the node is pictured in the following figure.

Figure D.3: Phasing algorithm on node

In our implementation during the node phasing a comparatively small power is required as the node only turns on its radio for 2 seconds and then off again for 18 seconds using approximately 10 mW or 4% of what the GPS receiver would use if switched on. In our implementation this phasing period may last up to 10 minutes. During this phasing period if any data packets are heard that signify other nodes have their GPS receivers on, the node will exit the phasing period and commence usual gathering of observational GPS data. If

APPENDIX D. NETWORK

on the other hand, after 10 minutes still no data packets are heard that signify other nodes have their GPS receivers on, the node will exit the phasing period and commence usual gathering of observational GPS data.

The phasing algorithm as run on the repeaters is depicted in the following figure.

APPENDIX D. NETWORK

Figure D.4: Phasing algorithm on repeater

During times of no node activity the repeater runs a phasing period, the repeater will turn its radio on for 2 seconds to listen for data packets and then turn off again. If data packets are heard that signify other nodes have their GPS receivers on, the repeater then becomes active and a timeout commences

APPENDIX D. NETWORK

that will get triggered if no new data packets are heard that signify other nodes have their GPS receivers on. When the timeout gets triggered the repeater goes back into the phasing period once again. Wherever the repeater is in its algorithm, if voltage level of the capacitor becomes too low, the radio will get switched off and the repeater will wait for its super capacitor voltage to rise once again to 2.2 V.

Bibliography

- [1] Cansel <http://www.cansel.ca/en/products/survey-instruments/survey-gps/promark-700>.
- [2] Datasheet VTGPSIA-3 GPS Antenna by V.TORCH Electronics Ltd. <http://www.sparkfun.com/datasheets/GPS/embeddeddsma.pdf>.
- [3] *Solar Bluetooth GPS Receiver SolarGPS Users Manual* http://www.proporta.com/F03/pdf/GpsBT_UserManualV2_Eng.pdf.
- [4] Sourced from GeoNet. <https://magma.geonet.org.nz/delta/app?service=external/ViewMark&sp=SPA&EK> and licensed by GeoNet for re-use under the creative commons attribution 3.0 New Zealand licence (<http://creativecommons.org/licenses/by/3.0/nz/>).
- [5] Sourced from the RINEX data archive of LINZ <http://www.linz.govt.nz/data/geodetic-services/positionz/rinex-data-archive> and licensed by LINZ for re-use under the creative commons attribution 3.0 New Zealand licence (<http://creativecommons.org/licenses/by/3.0/nz/>).
- [6] Tropical storm fung-wong 2014. by national aeronautics and space administration, lance-modis [public domain], via wikimedia commons.
- [7] Typhoon matmo 2014. by nasa image from the aqua satellite, darkened 20% by meow (<http://1.usa.gov/1n8zrFN>) [public domain], via wikimedia commons.
- [8] <http://www.gpstk.org/>.
- [9] http://www.ldksolar.com/uploadfiles/down/LDK_DS_290P-24_professional_EN_V3_120817.pdf.

BIBLIOGRAPHY

- [10] <http://www.retrievor.com>.
- [11] <http://www.rtklib.com/>.
- [12] *Trimble NetRS Datasheet*, 10 10/2006.
- [13] Klobuchar, j.a. ionospheric effects on gps. *gps world*, April 1991. Vol. 2, No. 4, pp. 48-51.
- [14] Gps guidebook. standards and guidelines for land surveying using global positioning system methods. survey advisory board and the public land survey office for the state of washington department of natural resources, November 2004.
- [15] Lea-6t product summary sheet, 2013.
- [16] Gps.gov: Gps modernization <http://www.gps.gov/systems/gps/modernization/>. Webpage, 2014. Modified: Monday, 15 September 2014 12:53:11 p.m.
- [17] Neo-7,u-blox 7, gnss modules data sheet. [http://www.u-blox.com/images/downloads/Product_Docs/NEO-7_DataSheet_\(GPS.G7-HW-11004\).pdf](http://www.u-blox.com/images/downloads/Product_Docs/NEO-7_DataSheet_(GPS.G7-HW-11004).pdf), May 2014.
- [18] ANDERSEN, H.-E., CLARKIN, T., WINTERBERGER, K., AND STRUNK, J. An accuracy assessment of positions obtained using survey-and recreational-grade global positioning system receivers across a range of forest conditions within the tanana valley of interior alaska. *Western Journal of Applied Forestry* 24, 3 (2009), 128–136.
- [19] AXELRAD, P., AND LARSON, K. Is it true that satellite geometry repeats every day shifted by 4 minutes? *Inside GNSS (July/August 2006)* (2006), 16–17.
- [20] BILICH, A., LARSON, K. M., AND AXELRAD, P. Modeling gps phase multipath with snr: Case study from the salar de uyuni, boliva. *Journal of Geophysical Research: Solid Earth (1978–2012)* 113, B4 (2008).
- [21] BLEWITT, G. Basics of the gps technique: observation equations. *Geodetic applications of GPS* (1997), 10–54.
- [22] BROWN, A. K., AND BROWN, P. Ultra low-power gps recorder (track-tag®). In *Proceedings of ION 61st Annual Meeting* (2005).

BIBLIOGRAPHY

- [23] BUCHLI, B., SUTTON, F., AND BEUTEL, J. Gps-equipped wireless sensor network node for high-accuracy positioning applications. In *Wireless Sensor Networks*. Springer, 2012, pp. 179–195.
- [24] CHOI, K., BILICH, A., LARSON, K. M., AND AXELRAD, P. Modified sidereal filtering: Implications for high-rate gps positioning. *Geophysical research letters* 31, 22 (2004).
- [25] CINA, A., PIRAS, M., AND BENDEA, H. Monitoring of landslides with mass market gps: an alternative low cost solution. ISPRS.
- [26] D’AMERICA. ARMY. CORPS OF ENGINEERS, E. U. *NAVSTAR global positioning system surveying[: engineer manual]*. U.S. Department of the Army. Army Corps of Engineers, 2003.
- [27] DOW, J. M., NEILAN, R., AND RIZOS, C. The international gnss service in a changing landscape of global navigation satellite systems. *Journal of Geodesy* 83, 3-4 (2009), 191–198.
- [28] FAN, K., DING, X., ET AL. Estimation of gps carrier phase multipath signals based on site environment. *Positioning* 1, 10 (2006).
- [29] GRANGER, R., AND SIMPSON, S. An analysis of multipath mitigation techniques suitable for geodetic antennas. *ION GNSS Proc* (2008).
- [30] GROVES, P. D., JIANG, Z., SKELTON, B., CROSS, P. A., LAU, L., ADANE, Y., AND KALE, I. Novel multipath mitigation methods using a dual-polarization antenna.
- [31] GRUBER, C. B. Gps modernization and program update. In *Munich Satellite Navigation Summit, Munich, Germany* (2011).
- [32] HEDGECOCK II, R. W. *Precise real-time relative localization using single-frequency GPS*. PhD thesis, Vanderbilt University, 2014.
- [33] HOFMANN-WELLENHOF, B., LICHTENEGGER, H., AND WASLE, E. *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.
- [34] HSIEHA, C.-H., AND WU, J. Multipath reduction on repetition in time series from the permanent gps phase residuals. *The International Archives*

BIBLIOGRAPHY

of the Photogrammetry, Remote Sensing and Spatial Information Sciences 37 (2008), 911–916.

- [35] HUGHES, J. C., BANKS, J. A., KERKHOFF, A. J., TOLMAN, B. W., WYANT, J. R., AND ELLISON, R. E. "sub-millimeter precision gps survey system at the holloman high speed test track". In *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)* (2001).
- [36] JIN, W., ZHAI, C., WANG, L., ZHANG, Y., AND ZHAN, X. Ambiguity function method scheme for aircraft attitude sensor utilising gps/glonass carrier phase measurement. *Defence Science Journal* 59, 5 (2009), 466–470.
- [37] JOHNSON, J., BRAUN, J., ROCKEN, C., AND VANHOVE, T. The role of multipath in antenna height tests at table mountain. *UNAVCO Home Page* (1995).
- [38] KIM, D., AND LANGLEY, R. B. Gps ambiguity resolution and validation: methodologies, trends and issues. In *Proceedings of the 7th GNSS Workshop–International Symposium on GPS/GNSS, Seoul, Korea* (2000), vol. 30.
- [39] KIM, D., AND LANGLEY, R. B. Instantaneous real-time cycle-slip correction of dual frequency gps data. In *Proceedings of the international symposium on kinematic systems in geodesy, geomatics and navigation, Banff, Alberta, Canada* (2001), pp. 5–8.
- [40] KLOBUCHAR, J. Ionospheric time-delay algorithm for single-frequency gps users. *Aerospace and Electronic Systems, IEEE Transactions on AES-* 23, 3 (May 1987), 325–331.
- [41] LARSON, K. M., BILICH, A., AND AXELRAD, P. Improving the precision of high-rate gps. *Journal of Geophysical Research: Solid Earth (1978–2012)* 112, B5 (2007).
- [42] LIMPACH, P., AND GRIMM, D. Rock glacier monitoring with low-cost gps receivers. In *7th Swiss Geoscience Meeting* (2009), pp. 247–248.
- [43] MCCLUSKY, S., AND TREGONING, P. Background paper on subsidence monitoring and measurement with a focus on coal seam gas (csg) activities, June 2013.

BIBLIOGRAPHY

- [44] OLDS, J. P. S. Small batteryless wireless solar powered gps receivers for subcentimeter land deformation monitoring. Master's thesis, Victoria University of Wellington, 2015.
- [45] RAGHEB, A., CLARKE, P., AND EDWARDS, S. Gps sidereal filtering: coordinate-and carrier-phase-level strategies. *Journal of Geodesy* 81, 5 (2007), 325–335.
- [46] REN, Z., LI, L., ZHONG, J., ZHAO, M., AND SHEN, Y. A real-time cycle-slip detection and repair method for single frequency gps receiver. *International Proceedings of Computer Science and Information Technology (IPCSIT)* 17 (2011), 224–230.
- [47] RIZOS, C., HAN, S., ROBERTS, C., HAN, X., ABIDIN, H. Z., SUGANDA, O. K., AND WIRAKUSUMAH, A. D. Continuously operating gps-based volcano deformation monitoring in indonesia: The technical and logistical challenges. In *Geodesy Beyond 2000*. Springer, 2000, pp. 361–366.
- [48] RUTLEDGE, D. Innovation-accuracy versus precision-a primer on gps truth-true to its word origins, accuracy demands careful and thoughtful work. this article provides a close look at the differences between the precision and accuracy of gps-determined positions, and should alleviate the confusion between the termsmaking abuse of the truth perhaps less likely in the business of gps positioning. *GPS World* 21, 5 (2010), 42.
- [49] SATIRAPOD, C., AND RIZOS, C. Multipath mitigation by wavelet analysis for gps base station applications. *Survey Review* 38, 295 (2005), 2–10.
- [50] SEYNAT, C., HOOPER, G., ROBERTS, C., AND RIZOS, C. Low-cost deformation measurement system for volcano monitoring. In *Proceedings of The 2004 International Symposium on GNSS/GPS* (2004).
- [51] SPECTRA PRECISION DIVISION. *ProMarkTM 700 Product Brochure* <http://www.spectraprecision.com/>, 2013.
- [52] TAKASU, T. Rtklib ver. 2.4. 2 manual.
- [53] TEUNISSEN, P. Least-squares estimation of the integer gps ambiguities. In *Invited lecture, section IV theory and methodology, IAG general meeting, Beijing, China* (1993).

BIBLIOGRAPHY

- [54] TEUNISSEN, P. A canonical theory for short gps baselines. part i: The baseline precision. *Journal of Geodesy* 71, 6 (1997), 320–336.
- [55] TOWNSEND, B., FENTON, P., VAN DIERENDONCK, K., AND VAN NEE, R. L1 carrier phase multipath error reduction using medll technology. In *PROCEEDINGS OF ION GPS. Vol. 8. INSTITUTE OF NAVIGATION* (1995).
- [56] TRIMBLE. *Trimble total station S8* http://trl.trimble.com/dscgi/ds.py/Get/File-390412/022543-410H_TrimbleS8_DS_0115_LR_Trimble, ph 022543-410h ed. Trimble, 1 2015.
- [57] VAN DIGGELEN, F. *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House Gnss Technology and Applications Library. Artech House, 2009.
- [58] VERHAGEN, S., AND LI, B. Lambda - matlab implementation, version 3.0. delft university of technology and curtin university, 2012.
- [59] WANG, B., WANG, S., MIAO, L., AND SHEN, J. An Improved Satellite Selection Method in Attitude Determination Using Global Positioning System (GPS). *Recent Patents on Space Technology* 1 (2009), 6–11.
- [60] WANNINGER, L., AND MAY, M. Carrier-phase multipath calibration of gps reference stations. *NAVIGATION, Journal of The Institute of Navigation, Vol. 48, No. 2, Summer 2001, pp. 113-124.* (2001).
- [61] WING, M. G., AND EKLUND, A. Performance comparison of a low-cost mapping grade global positioning systems (gps) receiver and consumer grade gps receiver under dense forest canopy. *Journal of Forestry* 105, 1 (2007), 9–14.
- [62] ZHONG, P., DING, X., YUAN, L., XU, Y., KWOK, K., AND CHEN, Y. Sidereal filtering based on single differences for mitigating gps multipath effects on short baselines. *Journal of Geodesy* 84, 2 (2010), 145–158.
- [63] ZHOU, Y. Dsp in a satellite navigation receiver with a perspective of computational complexity. Internet, Nov 2013.