**Institutionen för systemteknik**
Department of Electrical Engineering

Examensarbete

# RTK Compression for DARC

Peter Ekberg

Reg nr: LiTH-ISY-EX-2055
1999-04-22

**TEKNISKA HÖGSKOLAN**
**LINKÖPINGS UNIVERSITET**

Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Institutionen för systemteknik
581 83 Linköping

# RTK Compression for DARC

Thesis project done at Data Transmission,
Linköping University
by

Peter Ekberg

Reg nr: LiTH-ISY-EX-2055

Examiner: Ulf Henriksson

Supervisor: Michael Bertilsson

Linköping, April 22, 1999

**Avdelning, institution**
Division, department

Department of Electrical Engineering
Data Transmission

**Datum**
Date

1999-04-22

**Språk**
Language

- ○ Svenska/Swedish
- ⦿ Engelska/English
- ○ ______________

**Rapporttyp**
Report category

- ○ Licentiatavhandling
- ⦿ Examensarbete
- ○ C-uppsats
- ○ D-uppsats
- ○ Övrig rapport
- ○ ______________

**ISBN**

**ISRN**

**Serietitel och serienummer**
Title of series, numbering

**ISSN**

LiTH-ISY-EX-2055

**URL för elektronisk version**
URL for electronic version

http://www.lysator.liu.se/~peda/EX-2055/

**Titel** Komprimering av RTK för DARC
Title RTK Compression for DARC

**Författare** Peter Ekberg
Author

**Sammanfattning**
Abstract

RTK (Real-Time Kinematic) is a special kind of differential GPS (Global Positioning System) that requires a broadcasting radio link between two GPS receivers. DARC (Data Radio Channel) provides such a link. However, DARC is a multiplexed channel; if the data needed for the RTK service can be compressed, the freed up bandwidth can be used for something else.

This thesis presents three different methods of compressing RTK data. All three exploit both temporal and spatial redundancies. The main result of the work is the transmission protocols, as no implementations have been made as a part of this thesis.

The three methods reduce the required bandwidth to the estimated levels of 16%, 18.5% and 22% of the original requirement. Lower bandwidth has been achieved partly at the expense of RTK service availability when messages are lost on the radio link.

Since the difference in bandwidth requirement between the three methods is relatively minor, the third and most expensive method is recommended since this method is most resilient to lost messages. This method has been named Interleaved Absolute Relativity (IAR) in the thesis.

**Nyckelord**
Keywords

GPS, RTK, carrier phase, compression, DARC, error control

# Abstract

RTK (Real-Time Kinematic) is a special kind of differential GPS (Global Positioning System) that requires a broadcasting radio link between two GPS receivers. DARC (Data Radio Channel) provides such a link. However, DARC is a multiplexed channel; if the data needed for the RTK service can be compressed, the freed up bandwidth can be used for something else.

This thesis presents three different methods of compressing RTK data. All three exploit both temporal and spatial redundancies. The main result of the work is the transmission protocols, as no implementations have been made as a part of this thesis.

The three methods reduce the required bandwidth to the estimated levels of 16%, 18.5% and 22% of the original requirement. Lower bandwidth has been achieved partly at the expense of RTK service availability when messages are lost on the radio link.

Since the difference in bandwidth requirement between the three methods is relatively minor, the third and most expensive method is recommended since this method is most resilient to lost messages. This method has been named Interleaved Absolute Relativity (IAR) in the thesis.

# Table of Contents

**APPENDICES**

# 1 Introduction

This document was written as the report of a Master of Science thesis in Computer Science and Engineering at Linköping Institute of Technology. The task was performed at Sectra Communications AB – Data Broadcasting Systems.

## 1.1 Background

Differential GPS (Global Positioning System) can be used to improve the accuracy of ordinary GPS. With differential GPS, there is typically one fixed receiver at a known position called the base and one roving receiver at an unknown position called the rover. Corrections or measurements have to be transmitted from the base to the rover in order to get the position of the rover in real-time. One type of differential GPS is called RTK GPS (real-time kinematic GPS).

The transmission of corrections and measurements from the base to the rover used in RTK is standardized by the Radio Technical Commission for Maritime Services (RTCM), Special Committee no. 104. This standard typically uses 4 kbit/s to transmit measurements and corrections (updates once every second), with peaks of about 5-6 kbit/s.

The capacity of the transmission medium used in this thesis – DARC (Data Radio Channel) – is approximately 8 kbit/s. RTK seems to fit nicely in DARC, but there is always the desire to use DARC for something else in parallel to RTK. The freed up bandwidth can also be used to transmit RTK from two or more base receivers or to increase the update rate.

Generally, there is a desire to compress the RTK data to free up bandwidth for other purposes.

## 1.2 Previous Work

As far as I know, the only previous work is that of Yuki Hatanaka who has developed a data format and some tools [13] for compressing the Receiver Independent Exchange (RINEX) format [12]. RINEX is a text-based format used to store GPS observations for post-processing. The compressed RINEX format is also text-based and it is assumed that the resulting file is compressed further using some generic compression tool such as compress or gzip. Compressed RINEX uses the triple difference of the observations, which removes most of the temporal redundancies. However, no spatial redundancies are removed.

## 1.3 Scope

Several – sometimes conflicting – goals exist in this thesis. The primary objective is to identify and remove spatial and temporal redundancies from the RTK data. Removing redundancies reduces the used bandwidth.

However, in parallel to the primary objective, the availability of the RTK service cannot be drastically affected by the measures taken to reduce the redundancies.

It is also interesting to keep the calculations simple and straightforward, so that primarily the decoder complexity is kept low.

## 1.4 Limitations

The following things should be considered when using the compression described in this thesis:

- The resulting transmission protocol only supports epoch intervals of 0.5, 1 and 2 seconds.
- Minor alterations to the RTCM messages are made. The link established by the parity algorithm and the sequence number in the header are recreated.
- There is no support for GLONASS or RTCM messages 20/21.

## 1.5 Disposition

Chapter 2 – "DARC" – introduces DARC and explains some of its properties. Emphasis is on the protocol layers and error control.

Chapter 3 – "RTK Messages in RTCM" – provides some insight in the RTK messages in the RTCM standard. This chapter also introduces some notations and relations that are used in the following chapters.

Chapter 4 – "RTK Compression" – is the main chapter of the thesis. Several methods for saving bandwidth are developed. The different methods have different availability characteristics.

Chapter 5 – "Transmission Protocol" – outlines the content of the messages that are going to be transmitted instead of the RTCM messages.

Chapter 6 – "Bandwidth Estimates" – provides some expected bandwidth requirements of the protocols presented in chapter 5. These requirements are compared to the bandwidth requirements of standard RTCM messages.

Chapter 7 – "Conclusions and Further Studies" – presents conclusions that can be drawn from chapter 6 and the different availability characteristics of the methods. Some areas that have not been covered by this thesis are also mentioned.

## 1.6 Acknowledgements

I would like to thank the following people who helped me during the work.

- Josef Semler at ORF (Österreichischen Rundfunk) who provided RTCM data from a Trimble 4000 SSi dual frequency GPS base station. He has also helped with testing and has provided a lot of general input on DGPS services.
- Björn Ågårdh at Ag Com AB who graciously lent me an Ashtech Z-Surveyor dual frequency GPS base station receiver and a matching antenna.
- Henrik Sundberg at Teracom AB, Gunnar Hedling at the National Land Survey of Sweden, Tor Egil Melgård at Kværner Marine Automation AS and the other members of the MERCATOR group for various input and suggestions.
- Michael Bertilsson at Sectra Communications AB for interesting theoretical discussions and guidance in the day to day work.

# 2 DARC

In this chapter some aspects of DARC (Data Radio Channel) is introduced and explained. This chapter is focused on the use of DARC for real-time applications. Error control is also covered. Readers familiar with DARC can safely skip this chapter.

## 2.1 Introduction

DARC – which is standardized by ETSI [1] – is a signal transmitted as a subcarrier on normal FM transmissions. The signal is designed to not disturb the normal FM transmission. DARC is a digital packet oriented channel. The raw bandwidth is 16 kbps and about half is used for framing and error control. DARC is in many ways similar to RDS (Radio Data System), but has a higher bandwidth.

Currently there exist two nation wide DARC networks in Sweden and one in Austria. Germany, France, USA, Norway etc have transmitters fitted with DARC encoders. Japan was the first country to have a network, but their network does not fully adhere to the ETSI specification. The Japanese specification has small differences in the upper layers.

## 2.2 The DARC Layers

The DARC stack is divided into several layers, which are described in the following sections. Layer 1 (physical signal) and Layer 5 (data group) have been excluded since they are not relevant for this thesis.

### 2.2.1 The Frame (Layer 2)

Above the physical signal layer, data is organized into frames. A frame consists of 272 rows of data. Each row contains 288 bits. Several types of frames exist, namely A0, A1, B and C. Each row, called a block, takes 0.018 seconds to transmit. I.e. it takes close to 5 seconds to transmit a frame. The frame is filled with two kinds of blocks, information blocks and vertical parity blocks. The vertical parity is used to correct columns in the frame.

Layer 2 also provides a CRC and horizontal parity for the upper layers. These bits are used to protect individual information blocks from corruption. The CRC is 14 bits and the horizontal parity is 82 bits. In addition, each row in the frame is marked with a Block Identification Code (BIC) of 16 bits, which is used by the receivers to synchronize to the frame. This leaves 176 bits per row that can be used by Layer 3 and up. Frame type A0 is shown in Figure 1, which displays how a frame is constructed.

<table>
<tr><th>16 bits</th><th>176 bits</th><th>14 bits</th><th>82 bits</th></tr>
<tr><td>BIC3</td><td>Information (60 blocks)</td><td rowspan="3">C<br>R<br>C</td><td rowspan="3">Horizontal parity</td></tr>
<tr><td>BIC2</td><td>Information (70 blocks)</td></tr>
<tr><td>BIC1</td><td>Information (60 blocks)</td></tr>
<tr><td>BIC4</td><td colspan="3">Vertical parity (82 blocks)</td></tr>
</table>

*Figure 1: Frame type A0.*

The vertical parity blocks are used to detect and correct errors in the information blocks. However, when dealing with real-time data, this is not a working scheme since the information frequently is too old when it is possible to correct it using the vertical parity. For this reason, it is possible to mark information blocks as real-time. Real-time blocks are excluded from the vertical parity calculations.

Frame type A1 differs from A0 in that it contains three groups of four real-time blocks, which are inserted among the group of vertical parity blocks. Since these blocks are real-time blocks they are not part of the vertical parity calculations and can thus be filled with information even if the calculation of the vertical parity has been performed in the DARC encoder.

Frame type B differs from A0 in that the parity blocks are distributed throughout the whole frame and frame type C contains no vertical parity blocks at all.

### 2.2.2 The Block (Layer 3)

Layer 3 establishes three logical channels. The short and the long message channel and the service channel. The channels are frequently abbreviated SMCh, LMCh and SeCh. The L3 header starts with 4 bits that indicate what logical channel the block belongs to. The L3 header is 16 bits for SMCh and LMCh and 24 bits for SeCh. I.e. the message channels leave 160 bits (20 bytes) to the upper layers.

One L3 block is transmitted in one row in the L2 frame.

An important insight is that the smallest amount of data that can be transmitted is a block. If one bit of information needs to be transmitted every second, it will still take one whole block to transmit that one bit.

### 2.2.3 The Packet (Layer 4)

There are two kinds of layer 4 packets, long and short. Long packets can include up to 255 bytes of data and the short packets up to 127 bytes. The packets also include a header. The L4 headers in the SMCh are between 3 and 6 bytes and the L4 headers in the LMCh are between 4 and 7 bytes.

The packets (header and content) are divided into blocks. Therefore, it is necessary to think twice if the bits are to be used as efficiently as possible. Optimally, packets should fit evenly into blocks. If they don’t, the last block will be padded with zeros. The packets fit evenly into blocks when the size of the sum of the L4 header and the L4 content is divisible by 20 bytes.

The service identifier (SID), which is the final piece of the address, is transmitted in the packet header. The other pieces of the address – the country identifier, the extended country code and the network identifier – are transmitted in the service channel and are common for all data on a transmitter. The receivers use the address to distinguish between several services provided over DARC.

## 2.3 Using DARC for RTK

When dealing with RTK, only two types of frames are interesting; A1 or C. A0 is not interesting since it transmits all the 82 vertical parity blocks in the frame in conjunction, which introduces a delay close to 1.5 seconds. This is an unnecessary and not always acceptable delay. Frame type B introduces a delay of one frame at the DARC encoder, which of course is worse than the 1.5 seconds for frame type A0 are.

Frame type A1 is useful since it is possible to transmit real-time blocks inserted into the block of vertical parity blocks. The longest delay introduced by the vertical parity is thus reduced to less than 0.4 seconds if four or fewer information blocks are needed. If up to eight blocks are needed the longest delay introduced by the vertical parity is less than 0.8. The longest delay when up to 12 blocks are needed is 1.2 seconds. If more than 12 blocks are needed the vertical parity has the same impact as in the case with frame type A0.

Frame type C is optimal if only real-time blocks are transmitted, since no vertical parity blocks are transmitted. However, the channel is often shared between several services and there is more than one need to fulfill.

## 2.4 Error Control

This section has been included in order to enable a comparison of the error control provided by DARC and by RTCM.

As mentioned above, the block is protected by a 14 bit CRC and 82 bits of horizontal parity. The L3 blocks contain a 6 bit CRC on its header (not for the SeCh, but for both LMCh and SMCh). The L4 packets contain a CRC of 8 bits for the SMCh and 6 bits for the LMCh on their respective headers.

If another frame than C is used, blocks that are not marked as real-time get extra protection from the vertical parity blocks. This is ignored in this context, since only real-time blocks are of interest.

When an error is detected that cannot be corrected, the block is thrown away. If one block that is part of a packet is thrown away, the whole packet is thrown away. In other words, the channel is packet oriented. Either an L4 packet gets through correctly or it doesn’t get through at all.

## 2.5 DARC Receivers

It is of interest to implement the compression decoder in the DARC receiver or in the GPS receiver. For Sectra, it is of interest to implement the decoder in the DARC receiver, since Sectra produces DARC receivers. This section has been included to state some restrictions that apply to the decoder if it is to be implemented in an existing DARC receiver.

The receiver is going to be used in the field, typically powered by some kind of battery. Power consumption is an issue and the devices are trimmed of all unnecessary features. One such feature is processing power in general and floating point arithmetic in particular. Other limitations exist, such as memory size.

The processor of the target receiver is only capable of 32 bit arithmetic on the instruction level. This means that it is preferred that all arithmetic can be performed with 32 bit addition, subtraction, multiplication and division. A floating-point library in software is not possible, due to memory size and probably processing power. Software emulation of 64 bit integer arithmetic probably cost too much processing power, and should be avoided if possible.

# 3 RTK Messages in RTCM

In this chapter some of the RTK messages of the RTCM standard [2] is introduced and explained. Readers familiar with these RTK messages can safely skip this chapter. Some issues of RTK are mentioned in this chapter that may require a more general RTK understanding. Chapter 8 of [7] and/or chapter 10 of [9] are recommended reading.

## 3.1 Introduction

RTK is a differential method of using GPS (or some other satellite navigation system). Two GPS receivers are involved when making a measurement. The first, called the base or the reference station, is installed at a know position. The other, called the rover, may move about in the vicinity. The maximum distance between the base and the rover, or the maximum *baseline*, is about 10 to 50 km depending on the required accuracy and atmospheric conditions. Some kind of data link is used to transmit data from the base to the rover.

The RTCM Special Committee no. 104 has standardized a data format [2] so that base stations and rovers from different manufacturers may interoperate.

As the name of the method implies, the transmission from base to rover must occur in real-time. In this case, ‘real-time’ translates into a couple of seconds. I.e. the data link may not introduce a delay longer than a couple of seconds. In addition, a shorter delay is directly increasing the precision of the calculated position at the rover [4].

There are two different methods to implement RTK. The first uses uncorrected measurements and the second uses corrections. Uncorrected measurements mean that the base is not using the ephemerides information from the GPS satellites. When uncorrected measurements are used, the base generates raw measurements of the carrier phase and the pseudorange that are delivered to the rover. For corrections, the base instead delivers the deviation between expected values and the corrected measurements of the carrier phase and the pseudorange. For the remaining of this document, only uncorrected measurements are discussed for a couple of reasons:

- The correction messages do not provide the same level of accuracy as the uncorrected measurements [4].
- The correction messages are not yet fixed in the RTCM standard [2] and are thus subject to change.
- I do not have access to recorded correction messages or equipment that can generate them.

## 3.2 GPS Error Sources

This section has been included to get a feel for what error sources exist for a GPS user. This section also covers the magnitude of the errors and what error sources cancel out when using differential GPS. Table 1 presents some background on this subject.

*Table 1: The following error sources exist for GPS users of different categories.*

| | SPS GPS with SA | SPS GPS no SA | PPS GPS | DGPS |
|---|---|---|---|---|
| **Satellite clocks** | 3.0 | 3.0 | 3.0 | 0 |
| **Orbit errors** | 4.2 | 4.2 | 4.2 | 0 |
| **Ionosphere** | 5.0 | 5.0 | 2.3 | 0 |
| **Troposphere** | 1.5 | 1.5 | 2.0 | 0 |
| **Receiver noise** | 1.5 | 1.5 | 1.5 | 2.1 |
| **Multipath** | 2.5 | 2.5 | 1.2 | 2.5 |
| **SA** | 32.3 | 0 | 0 | 0 |
| **Other** | 1.5 | 1.5 | 1.5 | 0.5 |
| **Total (rss)** | 33.3 | 8.0 | 6.6 | 3.3 |

All numbers in this section are taken from [7]. They are the $1s$ error in meters. The total is the root sum squared (rss) of the other values[1]. SPS GPS is the Standard Positioning Service, SA is Selective Availability and PPS GPS is the Precise Positioning Service. No column for RTK GPS is provided, but the errors are much smaller than for ordinary DGPS.

## 3.3 Data Content

In this section, the content of the data sent from the base to the rover is presented.

### 3.3.1 RTCM Type 3 - GPS Reference Station Parameters

The rover needs to know the position of the base. The base transmits its coordinates in earth centered, earth fixed (ECEF) coordinates. The coordinates have an accuracy of 1 cm. One value of 32 bits is transmitted for each of the x-, y- and z-axes. See section 4.3.3 of [2] for more details.

Another message (RTCM Type 22 – Extended Reference Station Parameters) may be used if even greater resolution is required. This new message is still not fixed and will not be discussed further. Interested readers may look in section 4.3.23 of [2] for more details.

### 3.3.2 RTCM Type 18 - RTK Uncorrected Carrier Phases

For the remaining of the document, the shorter name “carrier phase” is used instead of the longer “uncorrected carrier phase”.

In RTK the carrier wave of the GPS signal is used. To be more exact, what is used is the difference between the received carrier wave and a carrier wave reference in the GPS receiver. If a satellite is moving towards the receiver, the Doppler effect makes the received carrier wave appear to have a higher frequency than the

[1] The total is not correct for SPS GPS with SA and for PPS GPS. I do not know the reason for this; the values have just been copied as stated in the source.

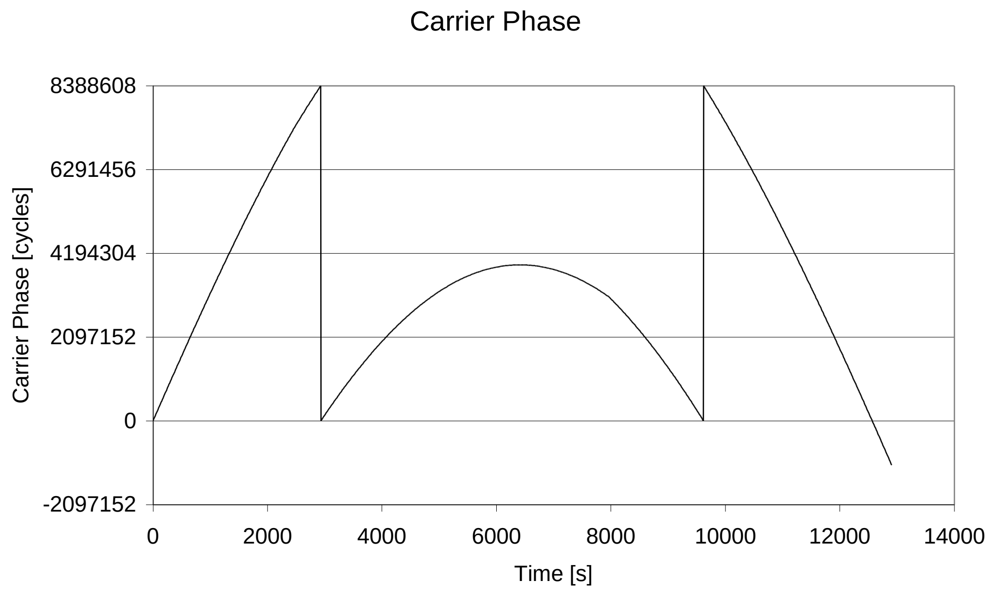



*Figure 2. Example of a sequence of carrier phase values generated for the L1 frequency.*

reference carrier wave. Vice versa when the satellite is moving away. The receiver clock will have an impact on the measurement, as the reference carrier wave is directly derived from it. The carrier phases that are present in the message are the accumulated Doppler difference between the received carrier wave and the reference carrier wave. I.e. what is called carrier phase in the GPS community is not the phase of the received carrier wave; it is the accumulated Doppler difference. The GPS receiver starts the accumulation when it starts to track the satellite.

GPS satellites transmit information on two separate frequencies called L1 and L2. The L1 frequency is 1.57542 GHz and the L2 frequency is 1.2276 GHz (section 3.3.1.1 of [5]). This gives wavelengths of approximately 190 mm and 244 mm respectively. The precision in the type 18 message for carrier phases is 1/256 wavelength. I.e. the carrier phase can be expressed with an accuracy better than one mm for both frequencies. The message contains carrier phase values for all satellites the base is currently tracking.

In addition to the carrier phase values, the message also includes a time stamp with a precision of 1 µs and with the range of one hour. The value is separated into two parts, one that is common for all RTCM messages and another part that is specific for RTK messages that increase the precision. The common part has a precision of 0.6 seconds and a range of 1 hour. Thirteen bits is used for this part. The RTK specific part specifies the microsecond within the 0.6 second interval. Twenty bits are used for this part. I.e. 33 bits are used to timestamp an RTK message.

The distance between a typical base station and a GPS satellite is about 20000 to 25000 km. When a satellite has moved 5000 km towards the base receiver, the accumulated Doppler is $5 \cdot 10^6 / 0.190 \cong 26 \cdot 10^6$ cycles. With 256 intervals per cycle, this amounts to vary large values, which made the RTCM SC-104 decide on another scheme than to transmit the whole number. Instead, the committee decided that only 32 bits should be transmitted.

When the base starts to track a satellite, it starts to transmit a carrier phase value of something close to zero since the accumulated Doppler is small. The assigned 32 bits are not enough to hold the whole value. When the carrier phase value gets larger than or equal to $2^{31}$, $2^{31}$ is subtracted. The base remembers how many of these subtractions it has made and adds $2^{31}$ every time the carrier phase value decreases past 0 again. When all jumps have been canceled, the carrier phase value is allowed to decrease past zero. The same scheme is used for large negative values.

To explain the above description, Figure 2 has been included.

The base started to track the satellite at time 0, presumably when it was at the horizon in some direction. The satellite was then moving across the sky. At 2931 seconds, the satellite had moved so much closer that the carrier phase value got greater than $2^{31}$ (or 8388608 = $2^{23}$ full cycles) and the value jumped down towards zero again. At 6415 seconds, the satellite reached the point where it was closest to the base and started to move away. At 9618 seconds, the jump was canceled and at 12567 seconds the satellite had moved to a point where it was further away from the base than when the base started to track the satellite. The base kept tracking the satellite for some more minutes and the carrier phase value dropped below zero. Finally the base lost track of the satellite and no more values were generated for it.

The reason to transmit carrier phase values for both L1 and L2 is that the two signals are affected in different ways by different error sources. By providing both L1 and L2, the rover can eliminate some of these error sources. If no errors are present, the following formula relates the two:

(1) $$f_{L2}(t) - f_{L2}(t-n) \cong \left(f_{L1}(t) - f_{L1}(t-n)\right) \times \frac{f_{L2}}{f_{L1}}$$

$f_{L1}(t)$ is the carrier phase for the L1 frequency at time *t*, $f_{L2}(t)$ is the carrier phase for the L2 frequency, $f_{L1}$ is the L1 carrier frequency and $f_{L2}$ is the L2 carrier frequency. The reason for this not to be an exact equality is that the carrier phase values are integers and rounding may have an impact.

The two differences in equation (1) are going to be used a lot later in the text, hence they are given their own names:

(2) $$\begin{cases} \Delta_n f_{L1}(t) = f_{L1}(t) - f_{L1}(t-n) \\ \Delta_n f_{L2}(t) = f_{L2}(t) - f_{L2}(t-n) \end{cases}$$

Using the notation from equation (2) and excluding the $(t)$ from the deltas, equation (1) can be rewritten as:

(3) $$\Delta_n f_{L2} \cong \Delta_n f_{L1} \times \frac{f_{L2}}{f_{L1}}$$

Along with each carrier phase value, two other values are transmitted. One is a data quality indicator and the other is a value called the cumulative loss of continuity indicator. The data quality indicator is a three bit number that indicates the estimated one sigma phase measurement error. The cumulative loss of continuity indicator is a five bit number that is incremented (modulo 32) by the base each time it loses track of the satellite. The base may also increment the cumulative loss of continuity indicator whenever the base knows that the rover

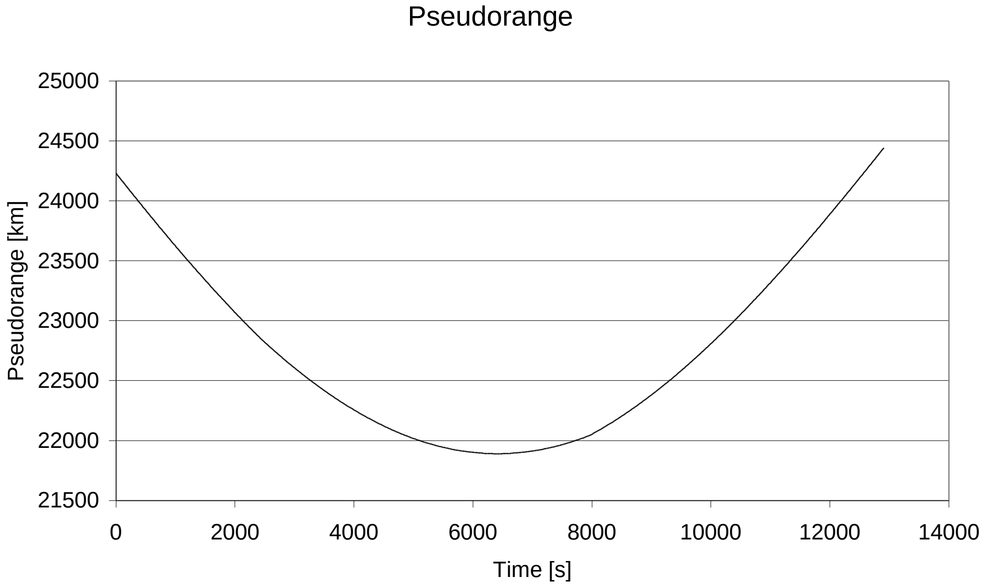



*Figure 3. Example of a sequence of pseudorange measurements generated for the L1 frequency.*

will need to recalculate the full cycle ambiguity. For a more complete description of these indicators, see section 4.3.19 of [2].

### 3.3.3 RTCM Type 19 - RTK Uncorrected Pseudoranges

For the remaining of the document, the shorter name "pseudorange" is used instead of the longer "uncorrected pseudorange".

The pseudorange is the measured travel time for the signal (from satellite to receiver), converted to a distance by multiplying with the speed of light. The value has an accuracy of 2 cm. 32 bits are allocated to transmit the value, which makes the maximum distance a little more than 85000 km. As is the case for the carrier phase, the base may calculate and transmit pseudoranges for both L1 and L2. The message contains pseudorange values for all satellites the base is currently tracking. The message also includes the same kind of timestamp as the type 18 message.

Since the whole value fits in the assigned bits, no special encoding of the values are needed, as was the case for the carrier phase values. This makes it a lot easier as is seen in Figure 3.

If no error sources were present, the following equation would relate the carrier phase value to the pseudorange value:

$$r_{Lx}(t) - r_{Lx}(t-n) \cong \left(f_{Lx}(t-n) - f_{Lx}(t)\right) \times \frac{c}{f_{Lx}} \qquad (4)$$

$r_{Lx}(t)$ is the pseudorange in meters and $f_{Lx}(t)$ is the carrier phase in cycles for time $t$ and frequency $Lx$, where $x$ is either 1 or 2. $c$ is the speed of light in meters per second and $f_{Lx}$ is the carrier frequency in Hertz for frequency $Lx$. Again, the reason for this not to be an exact equality is that the carrier phase values and the pseudorange values are integers and rounding may have an impact.

As was the case with the carrier phase, the subtractions in equation (4) are given their own names, one for L1 and one for L2:

(5)
$$\begin{cases}\Delta_n r_{L1}(t) = r_{L1}(t) - r_{L1}(t-n) \\ \Delta_n r_{L2}(t) = r_{L2}(t) - r_{L2}(t-n)\end{cases}$$

Substituting equations (2) and (5) into equation (4) and again dropping $(t)$ yields the following two equations:

(6)
$$\Delta_n r_{L1} \cong -\Delta_n f_{L1} \times \frac{c}{f_{L1}}$$

(7)
$$\Delta_n r_{L2} \cong -\Delta_n f_{L2} \times \frac{c}{f_{L2}}$$

If no error sources were present, the pseudoranges for L1 and L2 would be equal epoch for epoch. Therefore, $\Delta_n r_{L1}$ and $\Delta_n r_{L2}$ would be equal as well, as can be verified by combining equations (3), (6) and (7).

Along with each pseudorange value, two other values are transmitted. One is a data quality indicator and the other is a multipath error indicator. The data quality indicator is a four bit number that indicates the estimated one sigma pseudorange measurement error. The multipath error indicator is a four bit number that indicates – well – the estimated multipath error. A special value of 15 in this field indicates that the base did not estimate the multipath error. For a more complete description of these indicators, see section 4.3.20 of [2].

## 3.4 RTCM Units

Equations (3), (6) and (7) are valid for pseudorange measurements in meters and carrier phase measurements in cycles. As stated above, the RTCM messages represent these measurements differently. Equation (3) is not affected since the adjustment is canceled in the quotient. Equations (6) and (7) are adjusted according to equations (8) and (9).

(8)
$$\Delta_n r_{L1} \cong -\Delta_n f_{L1} \times \frac{c}{f_{L1}} \times \frac{1/0.02}{256}$$

(9)
$$\Delta_n r_{L2} \cong -\Delta_n f_{L2} \times \frac{c}{f_{L2}} \times \frac{1/0.02}{256}$$

From equation (8) to the end of this document, the unit of $f$ is $1/256$ cycles, not whole cycles. Likewise, the unit of $r$ is 0.02 meters, not whole meters.

## 3.5 Error Control

An RTCM message is packed into something called a frame. The frames consist of a variable number of words. The words consist of 24 bits of information and 6 bits of parity.

The parity bits are used solely to detect errors. No error correction is possible.

In addition to this, RTCM segments the word into 5 groups. Each group is six bits in size. To the front of each group the two bits “01” is augmented. Each group now forms a complete byte. This augmentation makes the RTCM stream printable on most ASCII terminals since each of these bytes is in the interval 64-127. This interval consists mostly of ordinary letters.

| | RTCM word | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RTCM stream | **01** | **–info–** | **01** | **–info–** | **01** | **–info–** | **01** | **–info–** | **01** | **parity** | RTCM stream |
| | Byte 1 | | Byte 2 | | Byte 3 | | Byte 4 | | Byte 5 | | |

*Figure 4: One RTCM word in the RTCM stream.*

One RTCM word (shown in Figure 4) is packed into 40 bits, 24 bits are information, 6 bits are parity and 10 bits are used to make the data stream reasonably printable.

An odd thing about the parity bits is that two of the parity bits of the previous RTCM word are used when calculating the parity bits of the current word. This establishes a link between RTCM words.

## 3.6 A Typical RTK Service

Since the base is not moving, the GPS Reference Station Parameters message is not needed very often. However, there is a need for rovers that have just been switch on to start functioning without too much delay. For this reason, the GPS Reference Station Parameters are typically transmitted once or twice a minute.

The base can be either a single frequency receiver or a dual frequency receiver. If the base is a single frequency receiver, each epoch will consist of one carrier phase message and one pseudorange message. For a dual frequency receiver, each epoch will consist of two messages of each type, one pair for L1 and one for L2. A new epoch is typically transmitted every other second, once a second or twice a second [2]. An epoch is a set of corrections and measurements generated by the base station.

All the messages of an epoch must have the exact same timestamp. In addition, the base must synchronize the output so that one epoch is generated *on* the second. The allowed deviation is approximately 1.1 ms as shown by the gray section in Figure 5.

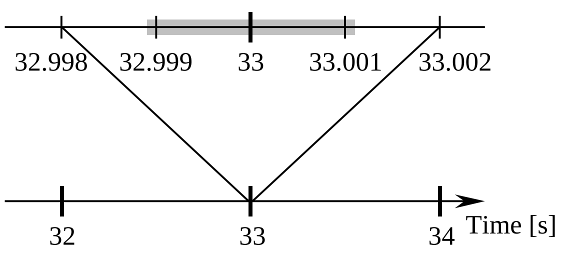



*Figure 5. The gray section indicates the possible timestamps that can be generated by the base when it is generating an epoch for the 33rd second.*

# 4 RTK Compression

In this chapter, the knowledge from the two previous chapters is combined with some statistical background from a couple of real RTK installations. The result is a more efficient way of transmitting the information from the base to the rover.

## 4.1 Introduction

A number of different compression techniques exist. They are mainly divided into two groups, lossy and lossless compression. Lossy compression introduces some kind of distortion while lossless compression does not.

In this case, only lossless compression is of interest. In order to be successful with lossy compression, you need a lot of knowledge about what distortions may be introduced without rendering the service unusable. Since I do not have this knowledge and since the task of acquiring it is huge, I wanted to limit myself to lossless compression. However, as the reader will notice, some trivial distortions are introduced since this simplifies a great deal and have no impact on the service.

## 4.2 Stripping

The RTK messages contain error control, as do all RTCM messages. If it can be shown that the error protection provided by DARC is stronger than the one provided by RTCM, the RTCM parity bits can be removed before transmission and later recreated in the receiver.

This is no proof; it is just an argument that it is likely that the DARC error protection is strong enough. DARC uses 82 bits of parity and 14 bits of CRC to protect 176 bits of information, as described in section 2.2.1. It follows that $82 + 14 = 96$ bits are used to protect 176 bits of information. To express it in another way, the code rate is:

$$\frac{96}{176 + 96} \cong 0.353 \tag{10}$$

In RTCM, 6 bits of parity are used to protect 24 bits of information. I.e. the code rate is:

$$\frac{6}{24 + 6} = 0.2 \tag{11}$$

A point to make is that the parity bits in DARC can be used for both error correction and for error detection while RTCM parity bits can only be used for error detection. This indicates that the information is treated more carefully in the RTCM case. However, DARC provides significantly more error protection bits. In addition, DARC uses larger blocks – 272 bits compared to 30 – which should provide a *much* stronger protection given the same code rate [3].

As stated, this is no proof and this issue should be investigated further. For the time being, it is assumed that the error control of DARC is good enough and that the error control provided by RTCM can be thrown out.

Throwing out the parity bits introduces one of the trivial distortions mentioned in section 4.1. The parity bits of RTCM are used to link the RTCM words of an RTCM frame. The parity bits of one RTCM word depend on two parity bits of the previous RTCM word as well as the information bits of the current RTCM word.

These two bits result in four different sets of parity bits for the same content of a word. However, whether the link is rebuilt in the exact same manner or not is not of any consequence, as long as the link exists.

To conclude this section, of the 40 bits in the RTCM word (see section 3.5), only 24 bits needs to be transmitted over DARC. I call this stripping and the reduction in bit rate is 40%.

## 4.3 Temporal Compression

Given that lossless compression should be used, this still leaves many options. There exist a number of techniques to compress data but the real-time nature of the data is a limiting factor. With a dual frequency receiver as base, four measurements are generated for every epoch – $f_{L1}(t)$, $f_{L2}(t)$, $r_{L1}(t)$ and $r_{L2}(t)$. If no error sources were present only one of these four would be necessary, along with information about how the carrier phase values should be encoded (see section 3.3.2). The four values carry mutual information and should be compressible; the error sources introduce the new information that has to be transmitted. Further, a satellite is not moving in a random fashion, which introduces a high temporal redundancy in the four values.

However, the instant a new epoch of four values arrives, the encoder must output enough information for the decoder to be able to reconstruct the epoch.

The real-time aspect renders the temporal redundancy untouchable for a block code. The block code encoder collects a block of values, compresses the block and outputs the result. The collection of a block introduces an unacceptable delay that rules out all block codes, including transform coding, subband coding and fractal coding. These codes are also quite complex which is another reason not to use them in this application.

The only compression technique known to me that satisfies the real-time requirements and is able to address the temporal redundancies is predictive coding.

The compressed data is going to be transmitted over a broadcast medium (DARC) where some pieces of the data stream may be lost due to bad reception at the receiver. In addition, new users may tune in to the service at any time. These facts must be accounted for when the code is designed.

The idea is to transmit Initialization Data Sets (IDS) from time to time. The IDS would allow the decoder to synchronize to the transmission. Between the Initialization Data Sets, Update Data Sets (UDS) are transmitted. There are two ways to do this. Either the UDS is relative to the last transmitted data set, or the UDS are always relative to the IDS. Let us name the former method Incremental Relativity (IR) and the latter Absolute Relativity (AR).

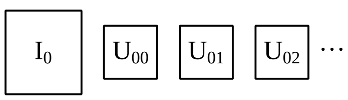



*Figure 6. Initialization Data Set and Update Data Sets.*

In Figure 6, IR can be described as "$U_{00}$ depends on $I_0$, $U_{01}$ depends on $U_{00}$ and $U_{02}$ depends on $U_{01}$", while AR can be described as "$U_{00}$, $U_{01}$ and $U_{02}$ all depend only on $I_0$".

Which of the two should be selected? For IR, all the Update Data Sets are representing the same thing – the difference between two adjacent epochs. This fact might make it easier to design the code.

AR is the more robust with respect to decoder resynchronization – if an Update Data Set is lost, only that epoch is lost. Not all epochs, until the next Initialization Data Set, are lost. The availability of the service is better with AR.

A variant of AR, which I have named Interleaved Absolute Relativity (IAR), is to send the Initialization Data Set for different satellites at different times. This protects the user from loss of reception when the Initialization Data Set should be received; data for one or a few satellites are lost, not for all satellites. The major drawback is loss of bandwidth. Much of the temporal redundancies exist in the values around the four central measurements. These redundancies can not be exploited as effectively with IAR.

Some things are common to IR, AR and IAR. They all seek to predict a future epoch given a base epoch (the IDS). This is done since it is anticipated that the difference between the two epochs (the UDS) is more effective to transmit. Additional features about the content of this difference ($\Delta_n f_{L1}$, $\Delta_n f_{L2}$, $\Delta_n r_{L1}$ and $\Delta_n r_{L2}$) are addressed in section 4.4, which makes this approach even more interesting.

## 4.4 Spatial Compression

As equation (3) shows, $\Delta_n f_{L1}$ can be used to estimate $\Delta_n f_{L2}$ (and vice versa). Similarly, as equations (8) and (9) shows, $\Delta_n f_{L1}$ and $\Delta_n f_{L2}$ can be used to estimate $\Delta_n r_{L1}$ and $\Delta_n r_{L2}$ (and vice versa).

As previously stated, $\Delta_n r_{L1}$ and $\Delta_n r_{L2}$ are essentially the same quantity measured in different ways. The measured quantity is the time it takes for electromagnetic waves to travel from the satellite to the GPS receiver. This time is multiplied by the speed of light to get a pseudorange measurement. GPS error sources affect the measurements differently depending on how the measurements have been performed but the two measurements still contain mutual information. The histograms in appendix A, sections A.1, A.2 and A.6, show that $\Delta_n r_{L1}$ and $\Delta_n r_{L2}$ have more mutual information than $r_{L1}$ and $r_{L2}$, at least as long as *n* is small ($\leq 10$ seconds). I.e. $\Delta_n r_{L1}$ can be used to estimate $\Delta_n r_{L2}$ (and vice versa) and this will be more effective than estimating $r_{L2}$ with $r_{L1}$.

### 4.4.1 Definitions

The following notation are used from here on:

- $\tilde{\Delta}_n f_{L2}(t)$ is the estimate of $\Delta_n f_{L2}(t)$ given $\Delta_n f_{L1}(t)$
- $\tilde{\Delta}_n r_{L1}(t)$ is the estimate of $\Delta_n r_{L1}(t)$ given $\Delta_n f_{L1}(t)$
- $\tilde{\Delta}_n r_{L2}(t)$ is the estimate of $\Delta_n r_{L2}(t)$ given $\Delta_n r_{L1}(t)$
- $\tilde{\Delta}'_n r_{L2}(t)$ is the estimate of $\Delta_n r_{L2}(t)$ given $\Delta_n f_{L2}(t)$
- $C_{\Delta_n f_{L2}}(t)$ is the correction to $\tilde{\Delta}_n f_{L2}(t)$, $\Delta_n f_{L2} = \tilde{\Delta}_n f_{L2} + C_{\Delta_n f_{L2}}$
- $C_{\Delta_n r_{L1}}(t)$ is the correction to $\tilde{\Delta}_n r_{L1}(t)$, $\Delta_n r_{L1} = \tilde{\Delta}_n r_{L1} + C_{\Delta_n r_{L1}}$
- $C_{\Delta_n r_{L2}}(t)$ is the correction to $\tilde{\Delta}_n r_{L2}(t)$, $\Delta_n r_{L2} = \tilde{\Delta}_n r_{L2} + C_{\Delta_n r_{L2}}$
- $C_{\Delta'_n r_{L2}}(t)$ is the correction to $\tilde{\Delta}'_n r_{L2}(t)$, $\Delta_n r_{L2} = \tilde{\Delta}'_n r_{L2} + C_{\Delta'_n r_{L2}}$

### 4.4.2 Estimation Order

The relations in equations (3), (8) and (9) can be used to estimate one quantity given some of the other quantities. In these relations, one difference is estimated by multiplying the other difference with some scalar. In the following, *a* is one integer quantity, *b* some other integer quantity less than or equal to *a* and *k* is the scalar that relates the two. The question is which of the quantities to use as a basis when estimating the other. The goal is to minimize the needed integer correction.

The estimations can be expressed analogous to the following equations.

(12) $$a = \tilde{a} + c$$

(13) $$b = \tilde{b} + d$$

*a* and *b* are the true values, $\tilde{a}$ and $\tilde{b}$ are the estimates and *c* and *d* are the corrections. We assume that *k* is the best scalar that estimates *a* given *b*, according to equation (14).

(14) $$\tilde{a} = b \cdot k$$

The best estimate of *b* given *a* will then be $1/k$, as stated in equation (15).

(15) $$\tilde{b} = a/k$$

Starting with equation (13), the following reasoning can be made:

(16) $$\begin{aligned} b &= \tilde{b} + d \\ &= a/k + d \\ &= (\tilde{a} + c)/k + d \\ &= (b \cdot k + c)/k + d \\ &= b + c/k + d \end{aligned}$$

From equation (16) it follows that $d = -c/k$, or $|d| = |c/k|$. As stated above, $a \geq b$. Therefore, $k \geq 1$. It follows that $|d| \geq |c|$.

I.e. we get smaller integer corrections if larger values are used to estimate smaller values. Therefore, $\Delta_n f_{L1}$ is used as a base for all estimates since this is the largest quantity of the four.

There are several orders in which to estimate/reconstruct the differences. Given that $\Delta_n f_{L1}$ is used as a base the following options exist:

$$
\begin{array}{ccc|ccc}
\Delta_n f_{L1} & \rightarrow & \Delta_n r_{L1} & \Delta_n f_{L1} & \rightarrow & \Delta_n r_{L1} \\
 & & \downarrow & \downarrow & & \downarrow \\
\Delta_n f_{L2} & \leftarrow & \Delta_n r_{L2} & \Delta_n f_{L2} & & \Delta_n r_{L2} \\
\hline
\Delta_n f_{L1} & \rightarrow & \Delta_n r_{L1} & \Delta_n f_{L1} & & \Delta_n r_{L1} \\
\downarrow & & & \downarrow & & \uparrow \\
\Delta_n f_{L2} & \rightarrow & \Delta_n r_{L2} & \Delta_n f_{L2} & \rightarrow & \Delta_n r_{L2}
\end{array}
$$

*Figure 7. The different orders to estimate and reconstruct the four differences.*

Appendix B shows that the corrections are virtually independent. Other aspects must determine the estimation order.

Analysis of measurements shows that $\Delta_n f_{L1}$ is good to use as a base for estimates of $\Delta_n f_{L2}$ – the needed correction is small (compare the histograms in appendix A, sections A.3, A.5, A.7 and A.9). This rules out the top left order in Figure 7.

The estimate of $\Delta_n r_{L1}$ given $\Delta_n r_{L2}$ is equivalent to the estimate of $\Delta_n r_{L2}$ given $\Delta_n r_{L1}$. Therefore, the only difference between the two orders to the right in Figure 7 is that the top right order uses $\Delta_n f_{L1}$ as a base for the pseudorange difference and the bottom right order uses $\Delta_n f_{L2}$. $\Delta_n r_{L1}$ and $\Delta_n r_{L2}$ are roughly equal and, as stated previously, $\Delta_n f_{L1}$ is larger in magnitude than $\Delta_n f_{L2}$. The bottom right order can be ruled out – the integer correction will be smaller when $\Delta_n f_{L1}$ is used compared to if $\Delta_n f_{L2}$ is used.

Two orders in Figure 7 remains, the top right and the bottom left order. They only differ in how they estimate $\Delta_n r_{L2}$. Perhaps these two ways to predict $\Delta_n r_{L2}$ can be combined to form a better prediction. However, this has not been closely investigated since the expected improvement in the estimation is limited. It is also likely that any detected correlation is receiver dependent.

The upper right order is selected since this estimate of $\Delta_n r_{L2}$ has a little better statistics, which is shown by the histograms in appendix A, sections A.2, A.5, A.6 and A.9.

### 4.4.3 Calculation of Estimates and Corrections

The encoder needs to calculate the corrections to the estimates. The decoder only needs these corrections and $\Delta_n f_{L1}$ in order to reconstruct $\Delta_n f_{L2}$, $\Delta_n r_{L1}$ and $\Delta r_{L2}$. Since $\Delta_n f_{L2}$, $\Delta_n r_{L1}$ and $\Delta_n r_{L2}$ are integer quantities, the corrections are integers if the estimates are integers. Having the corrections as integers simplifies the transmission encoding and is highly desirable. All estimates are therefore rounded to integers.

The estimates of $\Delta_n f_{L2}$ and $\Delta_n r_{L1}$ are given by equation (3) and (8) but are adjusted to integers. $\Delta_n r_{L1}$ is used to estimate $\Delta_n r_{L2}$. The following equations are the result:

(17) $$\tilde{\Delta}_n f_{L2} = \left\lfloor \Delta_n f_{L1} \times \frac{f_{L2}}{f_{L1}} + 0.5 \right\rfloor$$

(18) $$\tilde{\Delta}_n r_{L1} = \left\lfloor -\Delta_n f_{L1} \times \frac{c}{f_{L1}} \times \frac{1/0.02}{256} + 0.5 \right\rfloor$$

(19) $$\tilde{\Delta}_n r_{L2} = \Delta_n r_{L1}$$

Since it is vital to have the same values after decoding, the estimates have to be calculated in the exact same way in the encoder and in the decoder. The decoder will be implemented in a DARC receiver, with the limitation that only 32 bit integer arithmetic can be used (see section 2.5). Equations (17) and (18) can be simplified as shown by equations (20) and (21). The value used for $c$ is $2.99792458 \cdot 10^8 \ m/s$ (as defined in section 20.3.4.3 of [5]).

(20) $$\tilde{\Delta}_n f_{L2} = \left\lfloor \Delta_n f_{L1} \times \frac{60}{77} + 0.5 \right\rfloor$$

(21) $$\tilde{\Delta}_n r_{L1} = \left\lfloor -\Delta_n f_{L1} \times \frac{21413747}{576153600} + 0.5 \right\rfloor$$

We will now check if these equations can be evaluated using 32 bit arithmetic. The large numerator in equation (21) limits $|\Delta_n f_{L1}|$ to 100, which is good for deltas of only 0.1 milliseconds. According to [9] (section 8.1), the satellites can appear to move at radial speeds up to $800 m/s$ as seen from a point on the earth. This translates to $|\Delta_n f_{L1}| = 800 \cdot 256 / l_{L1} \cong 1.1 \cdot 10^6$. However, $800 m/s$ is an estimate. To get some margin, we assume that $|\Delta_n f_{L1}|$ can grow approximately $1.2 \cdot 10^6$ each second. To enable 15 second deltas, $|\Delta_n f_{L1}|$ must be allowed to grow up to $18 \cdot 10^6$, which is clearly larger than the allowed 100. However, the quotient in (21) can be approximated by another quotient with a smaller numerator. Using standard multiplication followed by division to calculate $\tilde{\Delta}_n r_{L1}$, this limits the numerator to $2^{31} / (18 \cdot 10^6) \cong 119$. The best quotient satisfying these criteria is $85/2287$, with an error of $1.4 \cdot 10^{-7}$. When $|\Delta_n f_{L1}|$ is large ($18 \cdot 10^6$), this error will translate into a bias in the estimate as large as $1.4 \cdot 10^{-7} \times 18 \cdot 10^6 \cong 2.5$ (the unit is 0.02 meters).

If the technique described in appendix D is used the largest possible numerator and denominator is 32767. The best quotient satisfying these new criteria is $467/12565$, with an error of only $2.0 \cdot 10^{-11}$. With this quotient, the bias is kept less than $2.0 \cdot 10^{-11} \times 18 \cdot 10^{6} \cong 3.6 \cdot 10^{-4}$. The new equation for $\tilde{\Delta}_n r_{L1}$ is given in equation (22).

$$\tilde{\Delta}_n r_{L1} = \left\lfloor -\Delta_n f_{L1} \times \frac{467}{12565} + 0.5 \right\rfloor \tag{22}$$

Equation (20) can be evaluated using the technique described in appendix D or using standard multiplication followed by division since the numerator is small.

Using equations (20), (22) and (19) the encoder performs the following calculations to generate the corrections:

$$C_{\Delta_n f_{L2}} = \Delta_n f_{L2} - \tilde{\Delta}_n f_{L2} \tag{23}$$

$$C_{\Delta_n r_{L1}} = \Delta_n r_{L1} - \tilde{\Delta}_n r_{L1} \tag{24}$$

$$C_{\Delta_n r_{L2}} = \Delta_n r_{L2} - \tilde{\Delta}_n r_{L2} \tag{25}$$

On the other side of the transmission link, the decoder recreates $\tilde{\Delta}_n f_{L2}$, $\Delta_n f_{L2}$, $\tilde{\Delta}_n r_{L1}$, $\Delta_n r_{L1}$, $\tilde{\Delta}_n r_{L2}$ and $\Delta_n r_{L2}$, in that order. It is vital that the decoder is using the same equations as the encoder for $\tilde{\Delta}_n f_{L2}$, $\tilde{\Delta}_n r_{L1}$ and $\tilde{\Delta}_n r_{L2}$.

$$\Delta_n f_{L2} = \tilde{\Delta}_n f_{L2} + C_{\Delta_n f_{L2}} \tag{26}$$

$$\Delta_n r_{L1} = \tilde{\Delta}_n r_{L1} + C_{\Delta_n r_{L1}} \tag{27}$$

$$\Delta_n r_{L2} = \tilde{\Delta}_n r_{L2} + C_{\Delta_n r_{L2}} \tag{28}$$

### 4.4.4 Summary

Section 4.4 has shown that it is possible to recreate $\Delta_n f_{L1}$, $\Delta_n f_{L2}$, $\Delta_n r_{L1}$ and $\Delta_n r_{L2}$ given $\Delta_n f_{L1}$, $C_{\Delta_n f_{L2}}$, $C_{\Delta_n r_{L1}}$ and $C_{\Delta_n r_{L2}}$. The UDS is more effective if it contains the corrections instead of the raw differences.

## 4.5 Temporal Compression Revisited

This section will address the temporal compression of $\Delta_n f_{L1}$ for the different methods mentioned in section 4.3 (IR, AR and IAR).

### 4.5.1 Incremental Relativity

For IR, the Initialization Data Set (IDS) would contain $f_{L1}$, $f_{L2}$, $r_{L1}$ and $r_{L2}$. The Update Data Set (UDS) would contain $\Delta_n f_{L1}$, $C_{\Delta_n f_{L2}}$, $C_{\Delta_n r_{L1}}$ and $C_{\Delta_n r_{L2}}$ where *n* would always be the epoch spacing in seconds. However, $\Delta_n f_{L1}$ is changing slowly while $C_{\Delta_n f_{L2}}$, $C_{\Delta_n r_{L1}}$ and $C_{\Delta_n r_{L2}}$ have small temporal redundancies (see appendix C, sections C.3, C.4 and C.5). If we extend the IDS with the latest available $\Delta_n f_{L1}$, the UDS would not need $\Delta_n f_{L1}$. The UDS would only need the difference between the current $\Delta_n f_{L1}$ and the previous. Let us call this difference $\Delta_{n,n} f_{L1}$ which we define according to expression (29).

(29) $$\Delta_{n,n} f_{L1}(t) \equiv \Delta_n f_{L1}(t) - \Delta_n f_{L1}(t-n)$$

As can be seen in appendix C, section C.1, $\Delta_{n,n} f_{L1}$ also contains a fair share of temporal redundancies. The same procedure is repeated and the IDS is extended with $\Delta_{n,n} f_{L1}$ so that the UDS only needs the difference between the current $\Delta_{n,n} f_{L1}$ and the previous. Let us call this difference $\Delta_{n,n,n} f_{L1}$ defined according to expression (30).

(30) $$\Delta_{n,n,n} f_{L1}(t) \equiv \Delta_{n,n} f_{L1}(t) - \Delta_{n,n} f_{L1}(t-n)$$

The triple difference is the method presented in [13] that is used to compress RINEX files. The temporal redundancies of $\Delta_{n,n,n} f_{L1}$ are shown in appendix C, section C.2. However, it is not certain that $\Delta_{n,n,n} f_{L1}$ is the best way to encode the dynamic behavior of $f_{L1}$. Inspection of real sequences indicates that $\Delta_{n,n} f_{L1}$ is somewhat dependent on $\Delta_n f_{L1}$. The relationship was estimated to the expression in equation (31).

(31) $$\Delta_{1,1} f_{L1}(t) \cong 1.4 \cdot 10^{-6} \times \left(\Delta_1 f_{L1}(t-n)\right)^2 - 140$$

However, the relationship is dependent on the path of the satellite as seen from the base station. Inspection of real sequences also indicates that the acquired estimation errors are on par with those obtained with the $\Delta_{n,n,n} f_{L1}$ method. The expectation of $\Delta_{n,n,n} f_{L1}$ is zero, while the expectation of equation (31) is zero when averaged over time. Using $\Delta_{n,n,n} f_{L1}$ is the safer option of the two. In addition, equation (31) is only valid for epoch intervals of one second. Another problem with equation (31) is that the estimate may vary with the geographic location of the base station due to local patterns of satellite paths at that location. Other temporal redundancies have not been detected in the $f_{L1}$ variable. Therefore, the $\Delta_{n,n,n} f_{L1}$ method is selected.

If the decoder knows $\Delta_n f_{L1}(t-n)$ and $\Delta_{n,n} f_{L1}(t-n)$ it can reconstruct $\Delta_n f_{L1}(t)$ and $\Delta_{n,n} f_{L1}(t)$ given $\Delta_{n,n,n} f_{L1}(t)$ using equations (32) and (33).

(32) $$\Delta_{n,n} f_{L1}(t) = \Delta_{n,n} f_{L1}(t-n) + \Delta_{n,n,n} f_{L1}(t)$$

(33) $$\Delta_n f_{L1}(t) = \Delta_n f_{L1}(t-n) + \Delta_{n,n} f_{L1}(t)$$

The decoder is now ready for the next epoch. This way the decoder is incrementally updating the different deltas. The IDS of the IR method contain $f_{L1}(t)$, $\Delta_n f_{L1}(t)$, $\Delta_{n,n} f_{L1}(t)$, $f_{L2}(t)$, $r_{L1}(t)$ and $r_{L2}(t)$. The UDS of the IR method $m$ epochs after the IDS contain $\Delta_{n,n,n} f_{L1}(t + m \cdot n)$, $C_{\Delta_n f_{L2}}(t + m \cdot n)$, $C_{\Delta_n r_{L1}}(t + m \cdot n)$ and $C_{\Delta_n r_{L2}}(t + m \cdot n)$.

### 4.5.2 Absolute Relativity and Interleaved Absolute Relativity

AR and IAR demand a different strategy to reduce the temporal redundancies of $\Delta_n f_{L1}$. With IR, all UDSes between the current UDS and the current IDS are required. With AR and IAR, all UDSes can only depend on the current IDS. If the curve in Figure 2 is linear over short periods, this suggests that an estimate of $\Delta_1 f_{L1}$ in the IDS can be used to estimate the future of $\Delta_n f_{L1}$ according to equation (34).

$$\tilde{\Delta}_n f_{L1}(t+n) = n \cdot \tilde{\Delta}_1 f_{L1}(t) \tag{34}$$

However, this estimate is quite biased as can be seen in Figure 8.

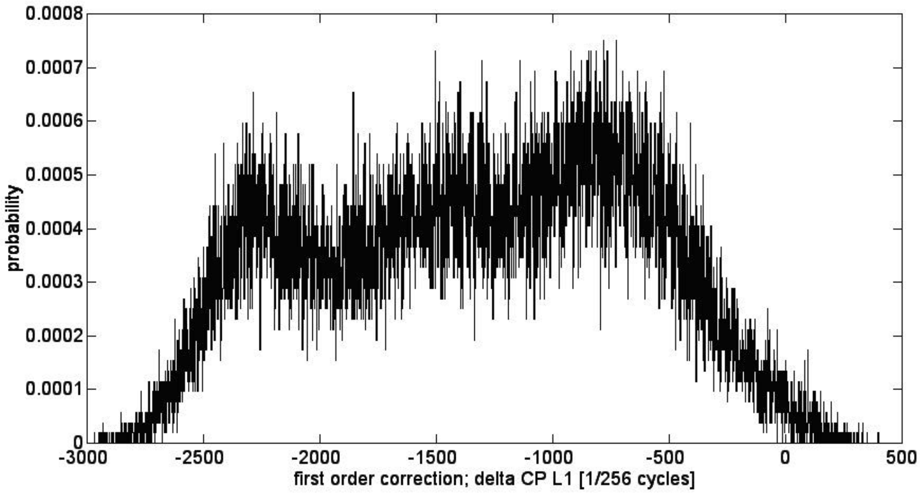



*Figure 8: Histogram of* $\tilde{\Delta}_5 f_{L1}(t+5) - 5 \cdot \tilde{\Delta}_1 f_{L1}(t)$.

If the second derivative is included, this suggests that $\Delta_n f_{L1}$ is changing linearly. The IDS would have to be extended with an estimate of $\Delta_{1,1} f_{L1}$ as well. The new estimate of $\Delta_n f_{L1}$ is expressed in equation (35).

$$\begin{aligned} \tilde{\Delta}_n f_{L1}(t+n) &= \sum_{i=1}^{n} \left( \tilde{\Delta}_1 f_{L1}(t) + i \cdot \tilde{\Delta}_{1,1} f_{L1}(t) \right) \\ &= n \cdot \tilde{\Delta}_1 f_{L1}(t) + \frac{n^2+n}{2} \cdot \tilde{\Delta}_{1,1} f_{L1}(t) \end{aligned} \qquad n \in I \tag{35}$$

The improvement can be seen in the histogram in Figure 10. The histogram in Figure 10 has a peak at zero, in contrast to the histogram in Figure 8.

One problem with equation (35) is that it only supports calculation of $\tilde{\Delta}_n f_{L1}$ when $n$ is an integer. However, the final expression in equation (35) is in fact valid for all $n$. This result is shown in appendix E.

The encoder needs to calculate $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$. If the epoch intervals of the incoming messages are whole seconds, a line can be fitted to the most recent deltas using the least mean square method [8], as is shown in Figure 9.

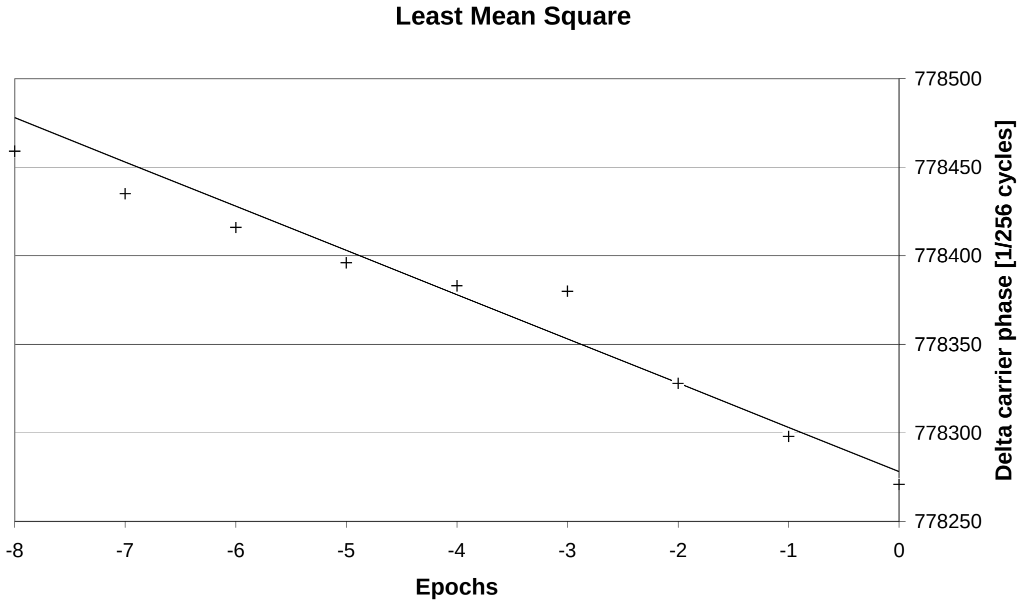



*Figure 9: $\tilde{\Delta}_1 f_{L1}(t)$ and $\tilde{\Delta}_{1,1} f_{L1}(t)$ are calculated with the least mean square method using $\Delta_1 f_{L1}(t-8)$ through $\Delta_1 f_{L1}(t)$. The epoch interval is one second.*

In Figure 9, $\tilde{\Delta}_1 f_{L1}(t)$ is found as the value of the estimated line at epoch zero and $\tilde{\Delta}_{1,1} f_{L1}(t)$ is found as the slope of the line. In this example $\tilde{\Delta}_1 f_{L1}(t) \cong 778278$ and $\tilde{\Delta}_{1,1} f_{L1}(t) \cong -25$. If the incoming epochs are not arriving with one second intervals, but with half second intervals, the encoder needs to perform the following calculations. First, the encoder calculates $\tilde{\Delta}_{1/2} f_{L1}(t)$ and $\tilde{\Delta}_{1/2,1/2} f_{L1}(t)$ analogous to how $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ were calculated in Figure 9. It then calculates $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ using equations (36) and (37). These equations are derived in appendix E, but are restated here.

(36) $$\tilde{\Delta}_{1,1} f_{L1}(t) = 4 \cdot \tilde{\Delta}_{1/2,1/2} f_{L1}(t)$$

(37) $$\tilde{\Delta}_1 f_{L1}(t) = 2 \cdot \tilde{\Delta}_{1/2} f_{L1}(t) - \tilde{\Delta}_{1/2,1/2} f_{L1}(t)$$

If epoch intervals of two seconds were used when doing the least mean square estimation, $\tilde{\Delta}_2 f_{L1}(t)$ and $\tilde{\Delta}_{2,2} f_{L1}(t)$ would have been the result. In that case, those two values have to be related to $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$. These relations are shown in equations (38) and (39). These relations are also derived in appendix E.

(38) $$\tilde{\Delta}_{1,1} f_{L1}(t) = \frac{1}{4} \cdot \tilde{\Delta}_{2,2} f_{L1}(t)$$

(39) $$\tilde{\Delta}_1 f_{L1}(t) = \frac{1}{2} \cdot \tilde{\Delta}_2 f_{L1}(t) + \frac{1}{8} \cdot \tilde{\Delta}_{2,2} f_{L1}(t)$$

When the encoder has calculated $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ for the IDS, it needs to calculate $\tilde{\Delta}_n f_{L1}$ and the correction for this variable for every UDS until the next IDS. As was the case with the estimates calculated in section 4.4.3, the encoder and the decoder must calculate the estimate in the exact same way. The expression for $\tilde{\Delta}_n f_{L1}$ is found in equation (40).

(40) $$\tilde{\Delta}_n f_{L1} = n \cdot \tilde{\Delta}_1 f_{L1} + \frac{n^2 + n}{2} \cdot \tilde{\Delta}_{1,1} f_{L1}$$

This equation is straightforward to evaluate if $n \in I$, and both $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ are integers. If $n \notin I$, care must be taken. Specifically, if $2n \in I$ equation (40) should be evaluated according to equation (41).

(41) $$\begin{aligned} \tilde{\Delta}_n f_{L1} &= \left\lfloor \frac{8n \cdot \tilde{\Delta}_1 f_{L1} + \left(4n^2 + 4n\right) \cdot \tilde{\Delta}_{1,1} f_{L1}}{8} + 0.5 \right\rfloor \\ &\langle m = 2n \rangle \\ &= \left\lfloor \frac{4m \cdot \tilde{\Delta}_1 f_{L1} + \left(m^2 + 2m\right) \cdot \tilde{\Delta}_{1,1} f_{L1}}{8} + 0.5 \right\rfloor \end{aligned}$$

This equation works for all $n \in I$ as well. The advantage of this equation is that $n$ is the only non-integer quantity the decoder needs to handle.

The correction is labeled $C_{\Delta_n f_{L1}}$ and the relationship between the quantities is stated in equation (42).

(42) $$C_{\Delta_n f_{L1}} = \Delta_n f_{L1} - \tilde{\Delta}_n f_{L1}$$

In equation (42), $\tilde{\Delta}_n f_{L1}$ is given by equation (41).

The decoder needs to receive the IDS (containing $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$) and then the UDS (containing $C_{\Delta_n f_{L1}}$). The decoder also needs to know $n$, i.e. the time in seconds between the IDS and the UDS. With these values, it can calculate the true $\Delta_n f_{L1}$ as shown in equation (43).

(43) $$\Delta_n f_{L1} = \tilde{\Delta}_n f_{L1} + C_{\Delta_n f_{L1}}$$

Again, $\tilde{\Delta}_n f_{L1}$ is given by equation (41).

### 4.5.3 Determining the needed resolution for $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$

In this section $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ are determined using the least mean square method described in the previous section. The nine previous $\Delta_1 f_{L1}$ were used in the estimation process. $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ were then used to calculate $C_{\Delta_n f_{L1}}$ for $n$ =1, 2 … 10 according to equation (42). When $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ were determined to eight binary decimals, the histogram in Figure 10 was obtained.

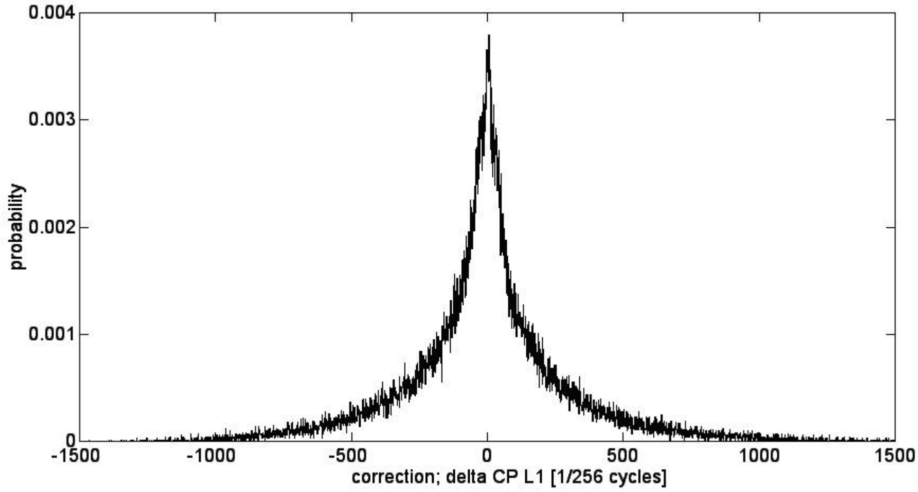



*Figure 10: Histogram of $C_{\Delta_n f_{L1}}$ with $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ determined to 8 binary decimals.*

The standard deviation $s$ of the data in Figure 10 is 301. If $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ are rounded to integers, $s$ becomes 303. Having $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ as integers reduces the decoder complexity. However, it is interesting to see how the data is dispersed when more bits are removed from $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$. But before this is investigated, it is also interesting to know how much bandwidth is saved per removed bit from $\tilde{\Delta}_1 f_{L1}$ or $\tilde{\Delta}_{1,1} f_{L1}$. Since it is the least significant bits that are interesting to remove, we assume that the contents of the removed bits are random and that the removed bits therefore represent the same number of bits of removed information. Further, we assume that the epoch interval is one second and that the interval between IDSes is ten seconds. One last assumption is made and that is that there is an average of eight satellites tracked by the base station. This translates into a bit rate reduction of $1 \cdot 8/10 = 0.8$ bits per second per removed bit.

*Table 2: The standard deviation of $C_{\Delta_n f_{L1}}$ as a function of the number of least significant bits removed from $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$.*

| s | removed bits | $\tilde{\Delta}_{1,1} f_{L1}$ 0 | 1 | 2 | 3 |
|---|---|---|---|---|---|
| $\tilde{\Delta}_1 f_{L1}$ | 0 | 303 | 308 | 325 | 413 |
| | 1 | 303 | 308 | 325 | 413 |
| | 2 | 304 | 308 | 325 | 413 |
| | 3 | 305 | 309 | 327 | 414 |
| | 4 | 309 | 314 | 330 | 417 |
| | 5 | 325 | 328 | 345 | 430 |
| | 6 | 379 | 382 | 396 | 470 |

Table 2 shows that it is more severe to remove precision from $\tilde{\Delta}_{1,1} f_{L1}$ than from $\tilde{\Delta}_1 f_{L1}$. This is expected because of the quadratic term in equation (41). Table 2 suggests that somewhere around 3-5 bits can be removed from $\tilde{\Delta}_1 f_{L1}$ and that around 0-2 bits can be removed from $\tilde{\Delta}_{1,1} f_{L1}$ without affecting the standard deviation very much. Removing more than these seven bits affect $C_{\Delta_n f_{L1}}$ so much that the attempt to gain bandwidth in the IDS surely results in even more bandwidth lost in the UDS. Losing one bit in the UDS is ten times as bad as losing it in the IDS. If all these seven bits are removed from $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$, only five bits per second is gained. This insignificant gain is therefore ignored and the idea to remove additional bits is discarded.

Conclusion: the encoder rounds $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ to the nearest integer.

## 4.6 Integer Encoding

The different methods (IR, AR and IAR) include a number of integers that are to be transmitted from the base to the rover. Some integers appear in the IDS and some in the UDS. In the previous section, it was stated that the IDS is more sensitive to extra bits than the UDS. This is the case since the UDS will be transmitted more frequently. Focus should therefore be on reducing the needed bits in the UDS.

### 4.6.1 Integer Encoding in the IDS of the IR Method

The integers in the IDS of the IR method are $f_{L1}$, $\Delta_n f_{L1}$, $\Delta_{n,n} f_{L1}$, $f_{L2}$, $r_{L1}$ and $C_{r_{L2}}$. The range of $f_{L1}$, $f_{L2}$ and $r_{L1}$ are 32 bits since these values are taken directly from the RTCM messages. $\Delta_n f_{L1}$ and $\Delta_{n,n} f_{L1}$ are dependant on $n$. However, a typical RTK service must have an update interval of no more than 2 seconds to be useful. In section 4.4.3 it was deduced that $|\Delta_n f_{L1}|$ could grow no more than $1.2 \cdot 10^6$ per second. With $n = 2$, this translates into one sign bit and 22 value bits. I.e. 23 bits are enough to encode $\Delta_n f_{L1}$.

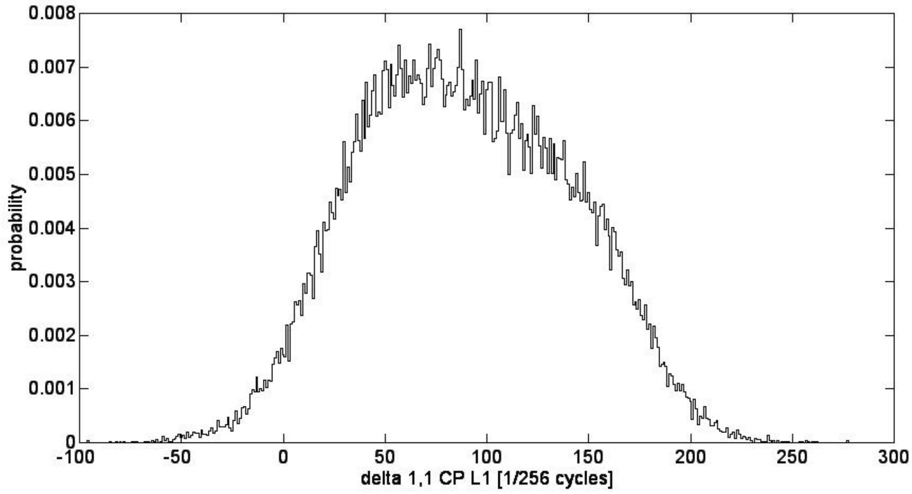



*Figure 11: Histogram of* $\Delta_{1,1} f_{L1}$.

Figure 11 shows an histogram of $\Delta_{1,1} f_{L1}$. Assuming that $\Delta_{2,2} f_{L1}$ has the double magnitude, $\Delta_{n,n} f_{L1}$ will need a range of –200 to 600. To simplify encoding, we use one sign bit and 10 value bits. I.e. 11 bits are used to encode $\Delta_{n,n} f_{L1}$.

The last integer in the IDS of the IR method is $C_{r_{L2}}$. It is quite hazardous to make any statements about this integer since its distribution is a function of the ionosphere. Therefore, it is changing over time. The exact number is needed for $C_{r_{L2}}$, otherwise the compression algorithm gets lossy.

During a day with high ionospheric activity (total electron content (TEC) of $10^{18}$ electrons/$m^2$), the difference in pseudorange between the L1 and the L2 frequency due to the ionosphere is about 10 meters. This is true for a satellite that is directly overhead of the base station. If the satellite has a low inclination angle, the signal path through the ionosphere is approximately 3.2 times as long. I.e. a total of 32 meters difference. Section 9.3 of [9] is recommended reading for a GPS oriented introduction to the ionosphere.

32 meters translates into values of $32/0.02 = 1600$ for $C_{r_{L2}}$. The ionosphere is always making $C_{r_{L2}}$ larger, but other effects make it smaller. Therefore, it can be negative as well. The range for $C_{r_{L2}}$ is selected as –255 to 1791 (768±1023). If any other value is needed for $C_{r_{L2}}$, –256 is used as an escape code that is followed by the value for $r_{L2}$. This encoding of $C_{r_{L2}}$ requires 11 bits in the normal case and 43 bits when extreme values appear.

### 4.6.2 Integer Encoding in the IDS of the AR and IAR Methods

The IDS of the IR method is similar to that in the AR and IAR methods. The difference is that the IR method includes $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ instead of $\Delta_n f_{L1}$ and $\Delta_{n,n} f_{L1}$. However, since $n$ could be to 2 in the IDS of the IR method $\tilde{\Delta}_1 f_{L1}$ and $\tilde{\Delta}_{1,1} f_{L1}$ require one bit less. I.e. 22 bits are used to encode $\tilde{\Delta}_1 f_{L1}$ and 10 bits are used to encode $\tilde{\Delta}_{1,1} f_{L1}$.

### 4.6.3 Integer Encoding in the UDS of the IR Method

The UDS of the IR method contain $\Delta_{n,n,n} f_{L1}$, $C_{\Delta_n f_{L2}}$, $C_{\Delta_n r_{L1}}$ and $C_{\Delta_n r_{L2}}$. A histogram of $\Delta_{n,n,n} f_{L1}$ for $n=1$ is shown in Figure 12 and for $n=2$ in Figure 13. The case $n=1/2$ have not been covered since I do not have access to sequences with half second intervals. These two curves show definite Gaussian shapes. The estimated standard deviation $s$ for the two histograms is 38. Their mean values are almost zero.

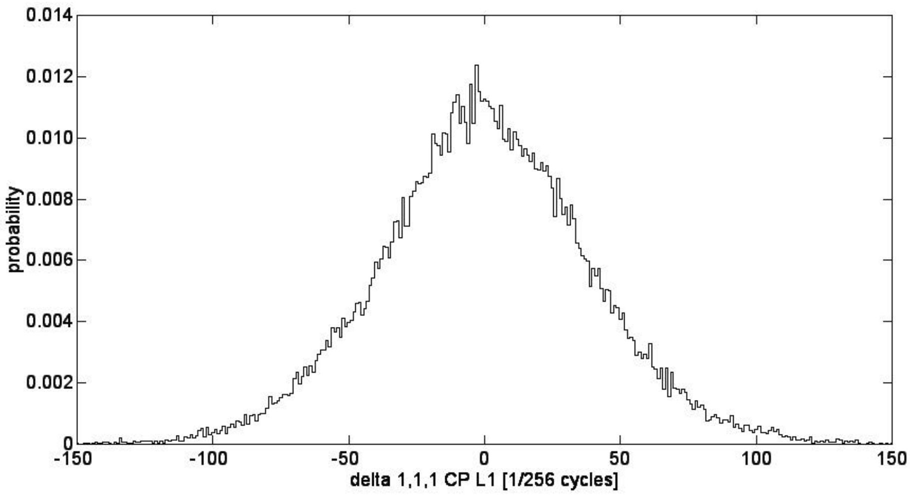



*Figure 12: Histogram of* $\Delta_{1,1,1} f_{L1}$.

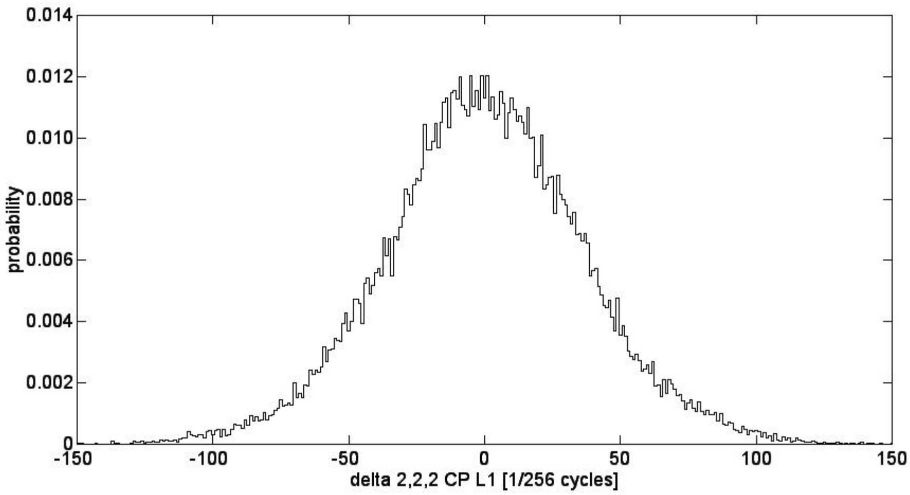



*Figure 13: Histogram of* $\Delta_{2,2,2} f_{L1}$.

Applying the Source Coding Theorem [10] on the two test sequences used to generate the histograms result in entropies of 7.3 bits per symbol. I.e. there is no hope to encode $\Delta_{n,n,n} f_{L1}$ using fewer bits than this.

Assuming a Gaussian shape of the histograms, they can be expressed according to equation (44) taken from [11].

(44) $$f(x,s) = \frac{1}{s\sqrt{2p}} e^{-x^2/2s^2}$$

Further, we assume that the standard deviation of a sequence with half second intervals is also 38.

The value for $\Delta_{n,n,n} f_{L1}$ will be encoded in a field using an integer with $s$ bits. If $\Delta_{n,n,n} f_{L1}$ does not fit in the $s$ bits, this field is set to $-2^{s-1}$. In this case, the first field is followed by a second field. The second field contains $t$ bits. If it is not possible to fit $\Delta_{n,n,n} f_{L1}$ in the second field either, the encoder generates a new IDS instead of a UDS. It is a matter of minimizing the cost in bits by adjusting $s$ and $t$.

However, in order to minimize the cost, there must exist some cost for transmitting the IDS instead of the UDS. We assume this cost as large as 10000 bits. The total cost can now be expressed according to equation (45).

(45)

$$\begin{aligned} c(s,t,s) = & \sum_{i=-\infty}^{-2^{t-1}-1} 10000 \cdot f(i,s) + \sum_{i=-2^{t-1}}^{-2^{s-1}} (s+t) \cdot f(i,s) \\ & + \sum_{i=-2^{(s-1)}+1}^{2^{s-1}-1} s \cdot f(i,s) \\ & + \sum_{i=2^{s-1}}^{2^{t-1}-1} (s+t) \cdot f(i,s) + \sum_{i=2^{t-1}}^{\infty} 10000 \cdot f(i,s) \end{aligned}$$

Evaluating equation (45) over $s = [0,9]$ and $t = [s,15]$ for $s = 38$ yields the results recorded in Table 3.

*Table 3: Cost in bits for different selections of s and t.*

| | ***s* = 0** | **2** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| ***t* = 0** | $10^4$ | | | | | | | | | |
| **1** | 9790 | 9790 | | | | | | | | |
| **2** | 9580 | 9580 | 9580 | | | | | | | |
| **3** | 9162 | 9162 | 9162 | 9162 | | | | | | |
| **4** | 8333 | 8333 | 8334 | 8334 | 8333 | | | | | |
| **5** | 6739 | 6739 | 6739 | 6740 | 6740 | 6739 | | | | |
| **6** | 4001 | 4002 | 4002 | 4003 | 4003 | 4002 | 4001 | | | |
| **7** | 928 | 929 | 930 | 930 | 930 | 930 | 929 | 928 | | |
| **8** | 15.6 | 16.5 | 17.3 | 18.0 | 18.3 | 18.0 | 16.8 | 15.3 | 15.6 | |
| **9** | 9.0 | 9.9 | 10.7 | 11.3 | 11.6 | 11.2 | 9.7 | 7.9 | 8.0 | 9.0 |
| **10** | 10.0 | 10.9 | 11.7 | 12.3 | 12.4 | 11.8 | 10.1 | 8.0 | 8.0 | 9.0 |
| **11** | 11.0 | 11.9 | 12.7 | 13.2 | 13.3 | 12.5 | 10.5 | 8.0 | 8.0 | 9.0 |
| **12** | 12.0 | 12.9 | 13.6 | 14.1 | 14.1 | 13.2 | 10.9 | 8.1 | 8.0 | 9.0 |
| **13** | 13.0 | 13.9 | 14.6 | 15.1 | 15.0 | 13.9 | 11.3 | 8.2 | 8.0 | 9.0 |
| **14** | 14.0 | 14.9 | 15.6 | 16.0 | 15.8 | 14.6 | 11.7 | 8.3 | 8.0 | 9.0 |
| **15** | 15.0 | 15.8 | 16.5 | 16.9 | 16.7 | 15.3 | 12.1 | 8.4 | 8.0 | 9.0 |

As can be seen in Table 3, the cost is only marginally higher (7.9 bit per symbol) than the theoretic limits calculated with the Source Coding Theorem (7.3 bits per symbol).

7.9 bits per symbol is acquired if $s = 7$ and $t = 9$. In order to protect ourselves from cases where the standard deviation is higher that 38 – just in case this is a bad estimate – we add an extra bit to $s$ and two extra bits to $t$. The additional cost for the extra bits is insignificant.

Since the histograms of $C_{\Delta_n f_{L2}}$, $C_{\Delta_n r_{L1}}$ and $C_{\Delta_n r_{L2}}$ also have (approximate) Gaussian shapes, the above procedure can be iterated for these fields as well. The iteration yields the results recorded in Table 4.

*Table 4: s and t for $\Delta_{n,n,n} f_{L1}$, $C_{\Delta_n f_{L2}}$, $C_{\Delta_n r_{L1}}$ and $C_{\Delta_n r_{L2}}$.*

| Field | $s$ | $t$ | Cost $n = 1/2$, 1, 2 |
|---|---|---|---|
| $\Delta_{n,n,n} f_{L1}$ | 8 | 11 | 8.0, 8.0, 8.0 |
| $C_{\Delta_n f_{L2}}$ | 3 | 6 | 3.0, 3.0, 3.2 |
| $C_{\Delta_n r_{L1}}$ | 6 | 9 | 6.0, 6.0, 6.1 |
| $C_{\Delta_n r_{L2}}$ | 6 | 9 | 6.0, 6.0, 6.5 |

When cost is different for different *n*, the cause is that the standard deviation is different for different *n*. The standard deviation for $n = 1$ and $n = 2$ were measured to the following quantities:

- 1.1 and 1.7 for $C_{\Delta_n f_{L2}}$
- 12.5 and 13.6 for $C_{\Delta_n r_{L1}}$
- 12.6 and 17.1 for $C_{\Delta_n r_{L2}}$

It is assumed that the standard deviation is lower for $n = 1/2$, but not significantly lower than the standard deviation for $n = 1$.

### 4.6.4 Integer Encoding in the UDS of the AR and IAR Methods

In the UDS of the IR method, equation (45) was minimized over $s$ and $t$ given the estimated standard deviation. This scheme has to be adjusted somewhat to suit AR and IAR. In IR, only $n = 1/2$, 1 and 2 had to be considered. In AR and IAR, *n* can be as large as 10. This has a major impact on the distribution. The shape of the histogram for $C_{\Delta_n f_{L1}}$ is Gaussian. Measuring the standard deviations for different *n* yields the following results for $n = [1,10]$: 30, 60, 100, 140, 190, 240, 300, 370, 440 and 520. Instead of minimizing the cost *c* in equation (45), the average of the cost for the different standard deviations is minimized according to equation (46).

$$c(s,t) = \frac{1}{10}\sum_{n=1}^{10} c(s,t,s_n) \tag{46}$$

$s_n$ in equation (46) is the 10 different measured standard deviations.

Minimizing equation (46) over *s* and *t* yields $s = 10$ and $t = 12$. The cost becomes 11.0 bits per symbol. Again, *s* is increased with one and *t* with two. The cost is increased to 11.1 bits per symbol by this. This should be compared with the theoretic minimum, which occurs when a different encoding is used depending on the time interval between the IDS and the UDS. Applying the Source Coding Theorem on the Gaussian distributions with the measured standard deviations and then averaging the results yields 9.5 bits per symbol.

Iterating the above for $C_{\Delta_n f_{L2}}$, $C_{\Delta_n r_{L1}}$ and $C_{\Delta_n r_{L2}}$ generates the costs presented below in Table 5.

*Table 5: s and t for* $C_{\Delta_n f_{L1}}$, $C_{\Delta_n f_{L2}}$, $C_{\Delta_n r_{L1}}$ *and* $C_{\Delta_n r_{L2}}$.

| Field | *s* | *t* | Average cost |
|---|---|---|---|
| $C_{\Delta_n f_{L1}}$ | 11 | 14 | 11.1 |
| $C_{\Delta_n f_{L2}}$ | 5 | 8 | 5.0 |
| $C_{\Delta_n r_{L1}}$ | 7 | 10 | 7.0 |
| $C_{\Delta_n r_{L2}}$ | 7 | 10 | 7.2 |

The estimated standard deviations for the different time differences $n = [1,10]$ behind the numbers in Table 5 are given below:

- $s$ = 1.1, 1.7, 2.4, 3.0, 3.7, 4.3, 4.9, 5.5, 6.1 and 6.7 for $C_{\Delta_n f_{L2}}$.
- $s$ = 12.5, 13.6, 14.0, 14.4, 14.8, 15.2, 15.3, 15.5, 15.6 and 15.7 for $C_{\Delta_n r_{L1}}$.
- $s$ = 12.6, 17.1, 20.1, 22.5, 24.5, 26.3, 27.8, 29.1, 30.3 and 31.7 for $C_{\Delta_n r_{L2}}$.

The histograms of $C_{\Delta_n f_{L2}}$, $C_{\Delta_n r_{L1}}$ and $C_{\Delta_n r_{L2}}$ are not as 'Gaussian' as those of $C_{\Delta_n f_{L1}}$ (and $\Delta_{n,n,n} f_{L1}$ from the IR section above). This difference is compensated by extra bits. Extra bits protect from situations with abnormal standard deviations and from divergence from the Gaussian distribution.

# 5 Transmission Protocol

This chapter presents transmission protocols for the three methods IR, AR and IAR. In addition, a fourth protocol is presented. This protocol is called ‘Rearranged Message’.

It should be possible to configure the desired IDS interval in the encoder. The encoder should try to maintain this interval. However, the encoder may fail this for a number of reasons. The main reasons for the encoder to transmit a new IDS instead of a UDS are:

- The epoch content is changing for a satellite.
- The needed correction does not fit in the assigned bits in the UDS.
- The L1 or the L2 smoothing interval has changed.
- Two adjacent epochs are separated by more that two seconds.

The last two reasons are not applicable to the IAR method and the last reason is only applicable to the IR method.

It can be noted that transmitting IDSes more often than desired have a larger impact on the IR and the AR methods since the encoder needs to transmit a new IDS for all satellites. With IAR, the encoder can transmit a new IDS for one satellite only. Figure 14 and Figure 15 displays this difference. The encoder has been configured to transmit new IDSes every 4 seconds. However, in epoch 4 something happens that forces an IDS for satellite 3. With the IR and the AR methods, this forces IDSes prematurely for all satellites. The IAR method handles this much more smoothly.

| Satellite | Epoch 1 | Epoch 2 | Epoch 3 | Epoch 4 | Epoch 5 |
|---|---|---|---|---|---|
| **1** | IDS | UDS | UDS | IDS | UDS |
| **2** | IDS | UDS | UDS | IDS | UDS |
| **3** | IDS | UDS | UDS | IDS | UDS |
| **4** | IDS | UDS | UDS | IDS | UDS |

*Figure 14: IDS transmission using the IR or the AR method.*

| Satellite | Epoch 1 | Epoch 2 | Epoch 3 | Epoch 4 | Epoch 5 |
|---|---|---|---|---|---|
| **1** | IDS | UDS | UDS | UDS | IDS |
| **2** | UDS | IDS | UDS | UDS | UDS |
| **3** | UDS | UDS | IDS | IDS | UDS |
| **4** | UDS | UDS | UDS | IDS | UDS |

*Figure 15: IDS transmission using the IAR method.*

If the encoder has no data to fill into the $\Delta_n f_{L1}$ and the $\Delta_{n,n} f_{L1}$ fields in the IDS of the IR method, these fields should be set to zero. The same can be said about the $\tilde{\Delta}_1 f_{L1}$ and the $\tilde{\Delta}_{1,1} f_{L1}$ fields in the IDS of the AR and IAR methods. In addition, if the encoder wants to set any of these fields to some value that does not fit into the assigned bits, it should set it to the closest value possible. The encoder should of course use the replacement value instead of the desired value when UDSes are generated.

## 5.1 Incremental Relativity

This section describes the IR method.

### 5.1.1 IDS

*Table 6: Initialization Data Set (IR)*

| Bits | Condition | Field |
|---|---|---|
| 8 | | Header, 0x02 |
| 10 | | Reference station id |
| 4 | | Epoch id |
| 1 | | Split message, SM |
| 1 | SM | First part |
| 3 | | Reference station health |
| 13 | | Half second within hour |
| 1 | | Microsecond correction present, MSCP |
| 12 | MSCP | Microsecond correction |
| 1 | | Carrier phase L1 present, CPL1P |
| 1 | CPL1P | Carrier phase L1 P-Code |
| 1 | | Carrier phase L2 present, CPL2P |
| 1 | CPL2P | Carrier phase L2 P-Code |
| 1 | | Pseudorange L1 present, PRL1P |
| 1 | PRL1P | Pseudorange L1 P-Code |
| 1 | | Pseudorange L2 present, PRL2P |
| 1 | PRL2P | Pseudorange L2 P-Code |
| 2 | PRL1P | L1 smoothing interval |
| 2 | PRL2P | L2 smoothing interval |
| 4 | | Number of satellites |
| ***The rest of the table is repeated for every satellite*** | | |
| 5 | | Satellite id |
| 1 | | Local content, LC |
| 1 | LC | Local carrier phase L1 present, CPL1P |
| 1 | LC | Local carrier phase L2 present, CPL2P |
| 1 | LC | Local pseudorange L1 present, PRL1P |
| 1 | LC | Local pseudorange L2 present, PRL2P |
| | ***CPL1P*** | ***Carrier phase L1*** |
| 3 | | Data quality carrier phase L1 |
| 5 | | Cumulative loss of continuity L1 |
| 32 | | $f_{L1}$ |
| 23 | | $\Delta_n f_{L1}$ |
| 11 | | $\Delta_{n,n} f_{L1}$ |
| | ***CPL2P*** | ***Carrier phase L2*** |
| 3 | | Data quality carrier phase L2 |
| 5 | | Cumulative loss of continuity L2 |
| 32 | | $f_{L2}$ |
| | ***PRL1P*** | ***Pseudorange L1*** |
| 4 | | Data quality pseudorange L1 |
| 4 | | Multipath error L1 |
| 32 | | $r_{L1}$ |
| | ***PRL2P*** | ***Pseudorange L2*** |
| 4 | | Data quality pseudorange L2 |
| 4 | | Multipath error L2 |
| 11 | PRL1P | $C_{r_{L2}}$ |
| 32 | ¬PRL1P | $r_{L2}$ |

### 5.1.2 Explanation of the Fields in the IDS

- *Header*
  8 bits indicating that this is an IDS message of the IR method.

- *Reference station id*
  The Reference station id taken from the same field in the RTCM message. This field can be used to send data from several reference stations on the same channel. If this is done the Epoch id described below is local to the reference station.

- *Epoch id*
  The epoch id is incremented (modulo 16) for each message (IDS and UDS) that is sent.

- *Split message, SM*
  If a message is too large to fit in one DARC layer 4 message, it is split in two parts. The fields Header, Reference station id, Epoch id, Split message and First part are added to the front of the second part. This field is set to one if the message was split.

- *First part*
  Only appears if SM is set. The First part field is set to 1 in the first part and 0 in the second part.

- *Reference station health*
  The reference station health field from the RTCM message.

- *Half second within hour, HSIH*
  This field is calculated according to equation (47).

  (47) $$HSIH = \left\lfloor \frac{ZCount \cdot 600000 + GNSSTime + 250000}{500000} \right\rfloor$$

  ZCount and GNSSTime come from the RTCM message. The reverse calculations are described under Microsecond correction.

- *Microsecond correction present, MSCP*
  When the epoch is not exactly aligned to a second, this field is set to one.

- *Microsecond correction, MSC*
  Only appears if MSCP is set. The field is calculated according to equation (48).

  (48) $$MSC = ZCount \cdot 600000 + GNSSTime - HSIH \cdot 500000$$

  ZCount and GNSSTime come from the RTCM message. The reverse calculations are stated below in equations (49) and (50).

  (49) $$GNSSTime = (HSIH \cdot 500000 + MSC) \% 600000$$

  (50) $$ZCount = \left\lfloor \frac{HSIH \cdot 500000 + MSC - GNSSTime}{600000} \right\rfloor$$

- *Carrier phase L1 present, CPL1P*
  This field is set to one if any of the satellites have the carrier phase L1 observable.

- *Carrier phase L1 P-Code*
  Only appears if CPL1P is set. The value is taken from the same field in the RTCM message type 18. If both variants are available to the encoder, it must select one of them.
- *Carrier phase L2 present, CPL2P*
  This field is set to one if any of the satellites have the carrier phase L2 observable.
- *Carrier phase L2 P-Code*
  Only appears if CPL2P is set. The value is taken from the same field in the RTCM message type 18. If both variants are available to the encoder, it must select one of them.
- *Pseudorange L1 present, PRL1P*
  This field is set to one if any of the satellites have the pseudorange L1 observable.
- *Pseudorange L1 P-Code*
  Only appears if PRL1P is set. The value is taken from the same field in the RTCM message type 19. If both variants are available to the encoder, it must select one of them.
- *Pseudorange L2 present, PRL2P*
  This field is set to one if any of the satellites have the pseudorange L2 observable.
- *Pseudorange L2 P-Code*
  Only appears if PRL2P is set. The value is taken from the same field in the RTCM message type 19. If both variants are available to the encoder, it must select one of them.
- *L1 smoothing interval*
  Only appears if PRL1P is set. The value is taken from the same field in the RTCM message type 19.
- *L2 smoothing interval*
  Only appears if PRL2P is set. The value is taken from the same field in the RTCM message type 19.
- *Number of satellites*
  The number of satellites. Satellite ids are sorted from lower to higher ids in the IDS. Satellite id 0 is first and satellite id 31 is last.
- *Satellite id*
  The satellite id taken from the RTCM message.
- *Local content, LC*
  If some observable is not present for this satellite that is present for some other satellite, this bit is set to 1.
- *Local carrier phase L1 present, CPL1P*
  Only appears if LC is set. This field is set to one if this satellite have the carrier phase L1 observable.
- *Local carrier phase L2 present, CPL2P*
  Only appears if LC is set. This field is set to one if this satellite have the carrier phase L2 observable.

- *Local pseudorange L1 present, PRL1P*
  Only appears if LC is set. This field is set to one if this satellite have the pseudorange L1 observable.
- *Local pseudorange L2 present, PRL2P*
  Only appears if LC is set. This field is set to one if this satellite have the pseudorange L2 observable.
- *Data quality carrier phase L1*
  Only appears if CPL1P is set. This field is taken from the same field in the RTCM message type 18.
- *Cumulative loss of continuity L1*
  Only appears if CPL1P is set. This field is taken from the same field in the RTCM message type 18.
- $f_{L1}$
  Only appears if CPL1P is set. This field is taken from the same field in the RTCM message type 18.
- $\Delta_n f_{L1}$
  Only appears if CPL1P is set. Encoded according to section 4.6.1.
- $\Delta_{n,n} f_{L1}$
  Only appears if CPL1P is set. Encoded according to section 4.6.1.
- *Data quality carrier phase L2*
  Only appears if CPL2P is set. This field is taken from the same field in the RTCM message type 18.
- *Cumulative loss of continuity L2*
  Only appears if CPL2P is set. This field is taken from the same field in the RTCM message type 18.
- $f_{L2}$
  Only appears if CPL2P is set. This field is taken from the same field in the RTCM message type 18.
- *Data quality pseudorange L1*
  Only appears if PRL1P is set. This field is taken from the same field in the RTCM message type 19.
- *Multipath error L1*
  Only appears if PRL1P is set. This field is taken from the same field in the RTCM message type 19.
- $r_{L1}$
  Only appears if PRL1P is set. This field is taken from the same field in the RTCM message type 19.
- *Data quality pseudorange L2*
  Only appears if PRL2P is set. This field is taken from the same field in the RTCM message type 19.
- *Multipath error L2*
  Only appears if PRL2P is set. This field is taken from the same field in the RTCM message type 19.

- $C_{r_{L2}}$
  Only appears if PRL2P and PRL1P are set. Encoded according to section 4.6.1.

- $r_{L2}$
  Appears if PRL2P is set and PRL1P is unset. Also appears if $C_{r_{L2}}$ is set to -256. This field is taken from the same field in the RTCM message type 19.

### 5.1.3 UDS

*Table 7: Update Data Set (IR)*

| Bits | Condition | Field |
|---|---|---|
| 8 | | Header, 0x03 |
| 10 | | Reference station id |
| 4 | | Epoch id |
| 1 | | Reference station health changed, RSHC |
| 3 | RSHC | Reference station health |
| 2 | | Half seconds since last epoch |
| 2 | MSCP | Microsecond correction update, MSCU |
| 12 | MSCU = -2 | Microsecond correction |
| 1 | | Carrier phase sign bit present, CPSP |
| ***The rest of the table is repeated for every satellite*** | | |
| | ***CPL1P*** | ***Carrier phase L1*** |
| 1 | | Status changed carrier phase L1, SCPL1 |
| 3 | SCPL1 | Data quality carrier phase L1 |
| 5 | SCPL1 | Cumulative loss of continuity L1 |
| 8(11) | | $\Delta_{n,n,n} f_{L1}$ |
| 1 | CPSP | Sign of $f_{L1}$ |
| | ***CPL2P*** | ***Carrier phase L2*** |
| 1 | | Status changed carrier phase L2, SCPL2 |
| 3 | SCPL2 | Data quality carrier phase L2 |
| 5 | SCPL2 | Cumulative loss of continuity L2 |
| 3(6) | CPL1P | $C_{\Delta_n f_{L2}}$ |
| 1 | CPSPn CPL1P | Sign of $f_{L2}$ |
| 32 | ¬CPL1P | $f_{L2}$ |
| | ***PRL1P*** | ***Pseudorange L1*** |
| 1 | | Status changed pseudorange L1, SPRL1 |
| 4 | SPRL1 | Data quality pseudorange L1 |
| 4 | SPRL1 | Multipath error L1 |
| 6(9) | CPL1P | $C_{\Delta_n r_{L1}}$ |
| 32 | ¬CPL1P | $r_{L1}$ |
| | ***PRL2P*** | ***Pseudorange L2*** |
| 1 | | Status changed pseudorange L2, SPRL2 |
| 4 | SPRL2 | Data quality pseudorange L2 |
| 4 | SPRL2 | Multipath error L2 |
| 6(9) | PRL1Pn CPL1P | $C_{\Delta_n r_{L2}}$ |
| 11 | PRL1Pn ¬CPL1P | $C_{r_{L2}}$ |
| 32 | ¬PRL1P | $r_{L2}$ |

### 5.1.4 Explanation of the Fields in the UDS

The satellites in the UDS appear in the same order as in the IDS. The same observables are present for each of the satellites.

- Header
  8 bits indicating that this is an UDS message of the IR method.
- Reference station id
  The reference station id taken from the same field in the RTCM message. See the IDS for further details.
- *Epoch id*
  The Epoch id is incremented modulo 16 for each message (IDS and UDS) that is sent. When a UDS arrives at the decoder, it can only be decoded if the decoder has received the message with epoch id one less the current epoch id. The previous message must have been received no more than 5 seconds ago.
- *Reference station health changed, RSHC*
  If the Reference station health field from the previous message needs to be updated, this field is set to one.
- *Reference station health*
  Only appears if RSHC is set. See the IDS for encoding details.
- *Half seconds since last epoch*
  Number of half seconds since the last epoch. 4 is encoded as 0.
- *Microsecond correction update, MSCU*
  Only appears if MSCP was set in the IDS. This field represents the change in the MSC field since the last message. If MSC has changed by more than one microsecond, MSCU contains -2.
- *Microsecond correction, MSC*
  Only appears if MSCP was set in the IDS and if MSCU is -2. See the IDS for encoding details.
- *Carrier phase sign bit present, CPSP*
  Indicates if the sign bit of the carrier phase values are present. The encoder should set this bit if the sign is ambiguous for *any* of the final carrier phase values in the UDS. The sign of $f_{L1}(t)$ is ambiguous if $f_{L1}(t-n)$ is small and $f_{L1}(t-n)+\Delta_n f_{L1}(t)$ has opposite sign of $f_{L1}(t-n)$. Zero has positive sign in this reasoning.
- *Status changed carrier phase L1, SCPL1*
  Only appears if CPL1P is set. This field indicates that either or both of Data quality and Cumulative loss of continuity has changed for carrier phase L1.
- *Data quality carrier phase L1*
  Only appears if CPL1P and SCPL1 are set. See the IDS for encoding details.
- *Cumulative loss of continuity L1*
  Only appears if CPL1P and SCPL1 are set. See the IDS for encoding details.

- $\Delta_{n,n,n} f_{L1}$
  Only appears if CPL1P is set. Encoding according to section 4.6.3. The decoder performs the following calculations:

$$\begin{aligned} \Delta_{n,n} f_{L1}(t) &= \Delta_{n,n} f_{L1}(t-n) + \Delta_{n,n,n} f_{L1}(t) \\ \Delta_{n} f_{L1}(t) &= \Delta_{n} f_{L1}(t-n) + \Delta_{n,n} f_{L1}(t) \\ f_{L1}(t) &= f_{L1}(t-n) + \Delta_{n} f_{L1}(t) \end{aligned} \tag{51}$$

  Care should be taken when performing the last calculation. If $f_{L1}(t-n)$ is large and positive and $\Delta_n f_{L1}(t)$ is positive so that $f_{L1}(t)$ becomes larger than $2^{31}-1$, $2^{31}$ should be subtracted from $f_{L1}(t)$. If $f_{L1}(t-n)$ is large and negative and $\Delta_n f_{L1}(t)$ is negative so that $f_{L1}(t)$ becomes smaller than $-2^{31}$, $2^{31}$ should be added to $f_{L1}(t)$.

- *Sign of* $f_{L1}$
  Only appears if CPSP and CPL1P are set. The sign bit of $f_{L1}(t)$ should be set to the content of this field.

- *Status changed carrier phase L2, SCPL2*
  Only appears if CPL2P is set. This field indicates that either or both of Data quality and Cumulative loss of continuity has changed for carrier phase L2.

- *Data quality carrier phase L2*
  Only appears if CPL2P and SCPL2 are set. See the IDS for encoding details.

- *Cumulative loss of continuity L2*
  Only appears if CPL2P and SCPL2 are set. See the IDS for encoding details.

- $C_{\Delta_n f_{L2}}$
  Only appears if CPL2P and CPL1P are set. Encoding according to section 4.6.3. The decoder performs the following calculations:

$$\begin{aligned} \tilde{\Delta}_n f_{L2}(t) &= \left\lfloor \Delta_n f_{L1}(t) \cdot \frac{60}{77} + 0.5 \right\rfloor \\ \Delta_n f_{L2}(t) &= \tilde{\Delta}_n f_{L2}(t) + C_{\Delta_n f_{L2}}(t) \\ f_{L2}(t) &= f_{L2}(t-n) + \Delta_n f_{L2}(t) \end{aligned} \tag{52}$$

  The same care should be taken with the last calculation as in the $\Delta_{n,n,n} f_{L1}$ section.

- *Sign of* $f_{L2}$
  Only appears if CPSP, CPL2P and CPL1P are set. The sign bit of $f_{L2}(t)$ should be set to the content of this field.

- $f_{L2}$
  Only appears if CPL2P is set and CPL1P is unset. This field is taken from the same field in the RTCM message type 18.

- *Status changed pseudorange L1, SPRL1*
  Only appears if PRL1P is set. This field indicates that either or both of Data quality and Multipath error has changed for pseudorange L1.

- *Data quality pseudorange L1*
  Only appears if PRL1P and SPRL1 are set. See the IDS for encoding details.
- *Multipath error L1*
  Only appears if PRL1P and SPRL1 are set. See the IDS for encoding details.
- $C_{\Delta_n r_{L1}}$
  Only appears if PRL1P and CPL1P are set. Encoding according to section 4.6.3. The decoder performs the following calculations:

$$\begin{aligned} \tilde{\Delta}_n r_{L1}(t) &= \left\lfloor -\Delta_n f_{L1}(t) \cdot \frac{467}{12565} + 0.5 \right\rfloor \\ \Delta_n r_{L1}(t) &= \tilde{\Delta}_n r_{L1}(t) + C_{\Delta_n r_{L1}}(t) \\ r_{L1}(t) &= r_{L1}(t-n) + \Delta_n r_{L1}(t) \end{aligned} \tag{53}$$

- $r_{L1}$
  Only appears if PRL1P is set and CPL1P is unset. This field is taken from the same field in the RTCM message type 19.
- *Status changed pseudorange L1, SPRL1*
  Only appears if PRL1P is set. This field indicates that either or both of Data quality and Multipath error has changed for pseudorange L1.
- *Data quality pseudorange L1*
  Only appears if PRL1P and SPRL1 are set. See the IDS for encoding details.
- *Multipath error L1*
  Only appears if PRL1P and SPRL1 are set. See the IDS for encoding details.
- $C_{\Delta_n r_{L2}}$
  Only appears if PRL2P, PRL1P and CPL1P are set. Encoding according to section 4.6.3. The decoder performs the following calculations:

$$\begin{aligned} \tilde{\Delta}_n r_{L2}(t) &= \Delta_n r_{L1}(t) \\ \Delta_n r_{L2}(t) &= \tilde{\Delta}_n r_{L2}(t) + C_{\Delta_n r_{L2}}(t) \\ r_{L2}(t) &= r_{L2}(t-n) + \Delta_n r_{L2}(t) \end{aligned} \tag{54}$$

- $C_{r_{L2}}$
  Only appears if PRL2P and PRL1P are set and CPL1P is unset. See the same field in the IDS for details.
- $r_{L2}$
  Appears if PRL2P is set and PRL1P is unset. Also appears if $C_{r_{L2}}$ is set to -256. This field is taken from the same field in the RTCM message type 19.

## 5.2 Absolute Relativity

This section describes the AR method. The UDS is always relative to the last transmitted IDS. The IDS and the UDS have been split in two since the IAR method shares the satellite parts.

### 5.2.1 IDS

*Table 8: Initialization Data Set (AR)*

| Bits | Condition | Field |
|---|---|---|
| 8 | | Header, 0x04 |
| 10 | | Reference station id |
| 6 | | IDS id |
| 1 | | Split message, SM |
| 1 | SM | First part, FP |
| 6 | ¬FP | Byte align |
| 3 | | Reference station health |
| 13 | | Half second within hour |
| 1 | | Microsecond correction present, MSCP |
| 12 | MSCP | Microsecond correction |
| 1 | | Carrier phase L1 present, CPL1P |
| 1 | CPL1P | Carrier phase L1 P-Code |
| 1 | | Carrier phase L2 present, CPL2P |
| 1 | CPL2P | Carrier phase L2 P-Code |
| 1 | | Pseudorange L1 present, PRL1P |
| 1 | PRL1P | Pseudorange L1 P-Code |
| 1 | | Pseudorange L2 present, PRL2P |
| 1 | PRL2P | Pseudorange L2 P-Code |
| 2 | PRL1P | L1 smoothing interval |
| 2 | PRL2P | L2 smoothing interval |
| 4 | | Number of satellites |
| ***The rest of the table is repeated for every satellite*** | | |
| 5 | | Satellite id |
| 1 | | Local content, LC |
| 1 | LC | Local carrier phase L1 present, CPL1P |
| 1 | LC | Local carrier phase L2 present, CPL2P |
| 1 | LC | Local pseudorange L1 present, PRL1P |
| 1 | LC | Local pseudorange L2 present, PRL2P |
| – | | According to Table 9 |

*Table 9: Satellite Initialization Data Set (AR and IAR)*

| | | |
|---|---|---|
| ***CPL1P*** | | ***Carrier phase L1*** |
| 3 | | Data quality carrier phase L1 |
| 5 | | Cumulative loss of continuity L1 |
| 32 | | $f_{L1}$ |
| 22 | | $\tilde{\Delta}_1 f_{L1}$ |
| 10 | | $\tilde{\Delta}_{1,1} f_{L1}$ |
| ***CPL2P*** | | ***Carrier phase L2*** |
| 3 | | Data quality carrier phase L2 |
| 5 | | Cumulative loss of continuity L2 |
| 32 | | $f_{L2}$ |
| ***PRL1P*** | | ***Pseudorange L1*** |
| 4 | | Data quality pseudorange L1 |
| 4 | | Multipath error L1 |
| 32 | | $r_{L1}$ |
| ***PRL2P*** | | ***Pseudorange L2*** |
| 4 | | Data quality pseudorange L2 |
| 4 | | Multipath error L2 |
| 11 | PRL1P | $C_{r_{L2}}$ |
| 32 | ¬PRL1P | $r_{L2}$ |

### 5.2.2 Explanation of the Fields in the IDS

Only the fields that are different from the IDS of the IR method are described.

- *Header*
  8 bits indicating that this is an IDS message of the AR method.
- *Reference station id*
  Similar to the Reference station id of the IR method. The IDS id is local to the reference station if several reference stations are transmitted on the same channel.
- *IDS id*
  Similar to the Epoch id of the IR method. This field is incremented modulo 64 with each new IDS that is transmitted.
- *Split message*
  The same functionality as in the IR method, but these fields are added to the front of the second message: Header, Reference station id, IDS id, Split message, First part and Byte align.
- *Byte align*
  Added to the second part of a split message in order to simplify the rejoining of the two parts. This field is filled with zeroes.
- $\tilde{\Delta}_1 f_{L1}$
  Only appears if CPL1P is set. Encoded according to section 4.6.2.
- $\tilde{\Delta}_{1,1} f_{L1}$
  Only appears if CPL1P is set. Encoded according to section 4.6.2.

### 5.2.3 UDS

*Table 10: Update Data Set (AR)*

| Bits | Condition | Field |
|---|---|---|
| 8 | | Header, 0x05 |
| 10 | | Reference station id |
| 6 | | IDS id |
| 1 | | Reference station health changed, RSHC |
| 3 | RSHC | Reference station health |
| 6 | | Half seconds since the IDS |
| 2 | MSCP | Microsecond correction update, MSCU |
| 12 | MSCU = -2 | Microsecond correction |
| 1 | | Carrier phase sign bit present, CPSP |
| ***The rest of the table is repeated for every satellite*** | | |
| – | | According to Table 11 |

*Table 11: Satellite Update Data Set (AR and IAR)*

| | | |
|---|---|---|
| ***CPL1P*** | ***Carrier phase L1*** | |
| 1 | | Status changed carrier phase L1, SCPL1 |
| 3 | SCPL1 | Data quality carrier phase L1 |
| 5 | SCPL1 | Cumulative loss of continuity L1 |
| 11(14) | | $C_{\Delta_n f_{L1}}$ |
| 1 | CPSP | Sign of $f_{L1}$ |
| ***CPL2P*** | ***Carrier phase L2*** | |
| 1 | | Status changed carrier phase L2, SCPL2 |
| 3 | SCPL2 | Data quality carrier phase L2 |
| 5 | SCPL2 | Cumulative loss of continuity L2 |
| 5(8) | CPL1P | $C_{\Delta_n f_{L2}}$ |
| 1 | CPSPn CPL1P | Sign of $f_{L2}$ |
| 32 | ¬CPL1P | $f_{L2}$ |
| ***PRL1P*** | ***Pseudorange L1*** | |
| 1 | | Status changed pseudorange L1, SPRL1 |
| 4 | SPRL1 | Data quality pseudorange L1 |
| 4 | SPRL1 | Multipath error L1 |
| 7(10) | CPL1P | $C_{\Delta_n r_{L1}}$ |
| 32 | ¬CPL1P | $r_{L1}$ |
| ***PRL2P*** | ***Pseudorange L2*** | |
| 1 | | Status changed pseudorange L2, SPRL2 |
| 4 | SPRL2 | Data quality pseudorange L2 |
| 4 | SPRL2 | Multipath error L2 |
| 7(10) | PRL1Pn CPL1P | $C_{\Delta_n r_{L2}}$ |
| 11 | PRL1Pn ¬CPL1P | $C_{r_{L2}}$ |
| 32 | ¬PRL1P | $r_{L2}$ |

### 5.2.4 Explanation of the Fields in the UDS

Only the fields that are different from the UDS of the IR method are described.

- *Header*
  8 bits indicating that this is an UDS message of the AR method.

- *IDS id*
  This field identifies the IDS that this UDS is an update to. The decoder must not apply a UDS to an IDS that is older than 25 seconds.

- *Half seconds since IDS*
  The number of half seconds since the IDS message. 64 is encoded as 0.

- $C_{\Delta_n f_{L1}}$
  Only appears if CPL1P is set. Encoding according to section 4.6.4. The decoder performs the following calculations:

$$\tilde{\Delta}_n f_{L1}(t) = \left\lfloor \frac{4m \cdot \tilde{\Delta}_1 f_{L1} + \left(m^2 + 2m\right) \cdot \tilde{\Delta}_{1,1} f_{L1}}{8} + 0.5 \right\rfloor$$

(55) $$\Delta_n f_{L1}(t) = \tilde{\Delta}_n f_{L1}(t) + C_{\Delta_n f_{L1}}(t)$$

$$f_{L1}(t) = f_{L1}(t-n) + \Delta_n f_{L1}(t)$$

  In equation (55), *m* is the value found in Half seconds since IDS. The same care should be taken with the last calculation as in the $\Delta_{n,n,n} f_{L1}$ section of the IR method.

- $C_{\Delta_n f_{L2}}$, $C_{\Delta_n r_{L1}}$ and $C_{\Delta_n r_{L2}}$
  Encoding according to section 4.6.4, otherwise the same as in the IR method.

## 5.3 Interleaved Absolute Relativity

This section describes the IAR method. The UDS is satellite local and is always relative to the last transmitted IDS for that satellite.

### 5.3.1 IDS/UDS

*Table 12: Common Data Set (IAR)*

| Bits | Condition | Field |
|---|---|---|
| 8 | | Header, 0x06 |
| 10 | | Reference station id |
| 1 | | Split message, SM |
| 4 | SM | Message id |
| 1 | SM | First part |
| 3 | | Reference station health |
| 13 | | Half second within hour |
| 1 | | Microsecond correction present, MSCP |
| 12 | MSCP | Microsecond correction |
| 1 | | Carrier phase L1 present, CPL1P |
| 1 | CPL1P | Carrier phase L1 P-Code |
| 1 | | Carrier phase L2 present, CPL2P |
| 1 | CPL2P | Carrier phase L2 P-Code |
| 1 | | Pseudorange L1 present, PRL1P |
| 1 | PRL1P | Pseudorange L1 P-Code |
| 1 | | Pseudorange L2 present, PRL2P |
| 1 | PRL2P | Pseudorange L2 P-Code |
| 2 | PRL1P | L1 smoothing interval |
| 2 | PRL2P | L2 smoothing interval |
| 1 | | Carrier phase sign bit present, CPSP |
| 4 | | Number of satellites |
| ***The rest of the table is repeated for every satellite*** | | |
| 5 | | Satellite id |
| 1 | | Local content, LC |
| 1 | LC | Local carrier phase L1 present, CPL1P |
| 1 | LC | Local carrier phase L2 present, CPL2P |
| 1 | LC | Local pseudorange L1 present, PRL1P |
| 1 | LC | Local pseudorange L2 present, PRL2P |
| 6 | | Satellite IDS id |
| 1 | | Initialization Data Set, IDS |
| – | IDS | According to Table 9 |
| – | ¬IDS | According to Table 11 |

### 5.3.2 Explanation of the Fields in the IDS/UDS

Only the fields that are different from the IDS/UDS of the AR method are described.

- *Header*
  8 bits indicating that this is an IAR message.
- *Reference station id*
  Similar to the Reference station id of the AR method. The Message id is local to the reference station if several reference stations are transmitted on the same channel.

- *Split message, SM*
  The same functionality as in the AR method, but these fields are added to the front of the second message: Header, Reference station id, Split message, Message id and First part.
- *Message id*
  Only appears if SM is set. Incremented modulo 16 for each message that is split. This field is only used to recognize the two parts of a split message. The first and second part must be received by the decoder within 5 seconds of each other.
- *Satellite IDS id*
  Similar to the IDS id of the AR method but this field is local to the satellite in the IAR method. Incremented modulo 64 for each IDS that is sent out for this satellite. The decoder must not apply a UDS to an IDS that is older than 25 seconds.
- *Initialization Data Set, IDS*
  Indicates if this satellite is initialized or updated by this message.

## 5.4 Rearranged Message

As a reference, this message has been included. It contains all information that is present in standard RTCM 18/19 messages. Any duplicate information has been removed. This separates the two sources of bandwidth gain that has been introduced – removing duplicate information and 'real' compression.

*Table 13: Complete Data Set (RM)*

| Bits | Condition | Field |
|---|---|---|
| 8 | | Header, 0x07 |
| 10 | | Reference station id |
| 1 | | Split message, SM |
| 4 | SM | Message id |
| 1 | SM | First part, FP |
| 3 | | Reference station health |
| 13 | | Modified Z-Count |
| 20 | | GNSS time of measurement |
| 1 | | Carrier phase L1 present, CPL1P |
| 1 | CPL1P | Carrier phase L1 P-Code |
| 1 | | Carrier phase L2 present, CPL2P |
| 1 | CPL2P | Carrier phase L2 P-Code |
| 1 | | Pseudorange L1 present, PRL1P |
| 1 | PRL1P | Pseudorange L1 P-Code |
| 1 | | Pseudorange L2 present, PRL2P |
| 1 | PRL2P | Pseudorange L2 P-Code |
| 2 | PRL1P | L1 smoothing interval |
| 2 | PRL2P | L2 smoothing interval |
| 4 | | Number of satellites |
| ***The rest of the table is repeated for every satellite*** | | |
| 5 | | Satellite id |
| 1 | | Local content, LC |
| 1 | LC | Local carrier phase L1 present, CPL1P |
| 1 | LC | Local carrier phase L2 present, CPL2P |
| 1 | LC | Local pseudorange L1 present, PRL1P |
| 1 | LC | Local pseudorange L2 present, PRL2P |
| | ***CPL1P*** | ***Carrier phase L1*** |
| 3 | | Data quality carrier phase L1 |
| 5 | | Cumulative loss of continuity L1 |
| 32 | | $f_{L1}$ |
| | ***CPL2P*** | ***Carrier phase L2*** |
| 3 | | Data quality carrier phase L2 |
| 5 | | Cumulative loss of continuity L2 |
| 32 | | $f_{L2}$ |
| | ***PRL1P*** | ***Pseudorange L1*** |
| 4 | | Data quality pseudorange L1 |
| 4 | | Multipath error L1 |
| 32 | | $r_{L1}$ |
| | ***PRL2P*** | ***Pseudorange L2*** |
| 4 | | Data quality pseudorange L2 |
| 4 | | Multipath error L2 |
| 32 | | $r_{L2}$ |

# 6 Bandwidth Estimates

This chapter presents estimations of the bandwidth requirements of the protocols in chapter 5. The following assumptions have been made:

- All four RTCM messages (Type 18 and 19 for both L1 and L2) are available in every epoch.
- The epochs are separated by one second.
- The header of the DARC packets (layer 4) is four bytes.
- "RTCM 18/19" and "Stripped 18/19" are not transmitted as four DARC packets. I.e. they are appended and transmitted in as few DARC packets as possible.
- "IAR", "AR" and "IR" generate Initialization Data Sets every ninth epoch.
- "IAR" is assumed to generate Initialization Data Sets for all satellites at the same time. This is a disadvantage to availability and bandwidth. However, the penalty in bandwidth is slight and comparisons should be valid.

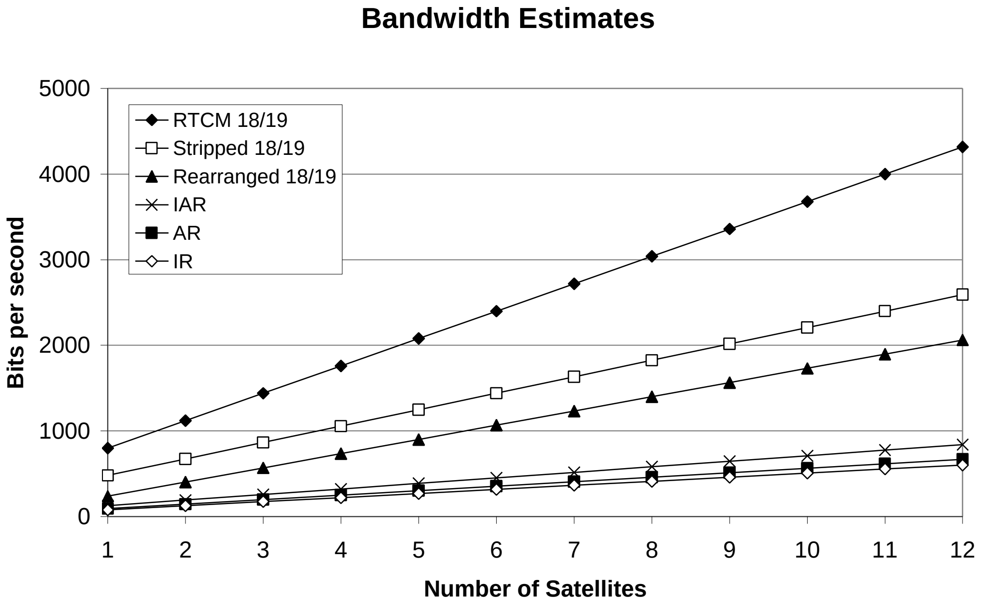



*Figure 16: Estimates of bandwidth requirements for different transmission protocols when there are no lower transmission layers.*

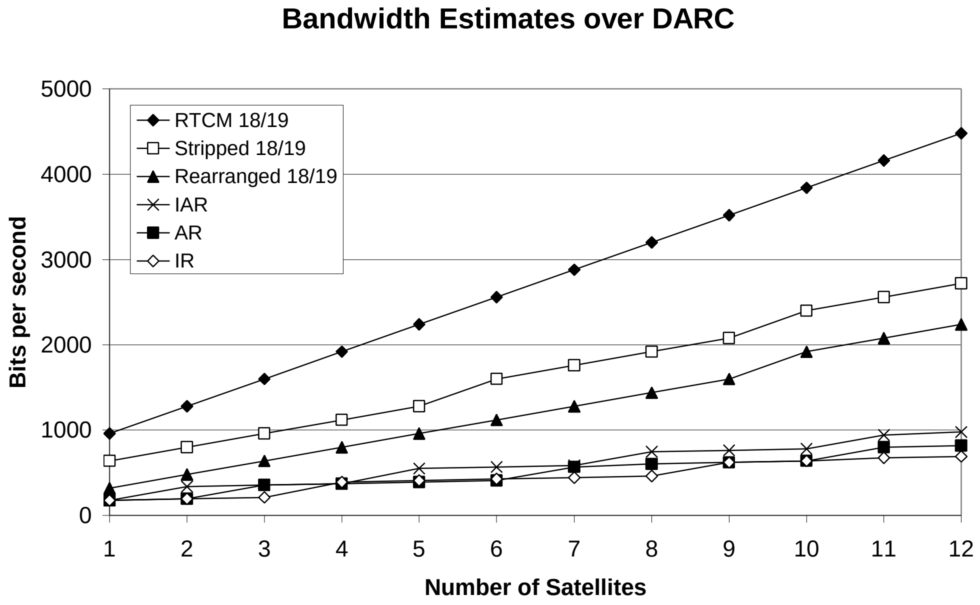



*Figure 17: Estimates of bandwidth requirements for different transmission protocols when they are transmitted over the DARC LMCh.*

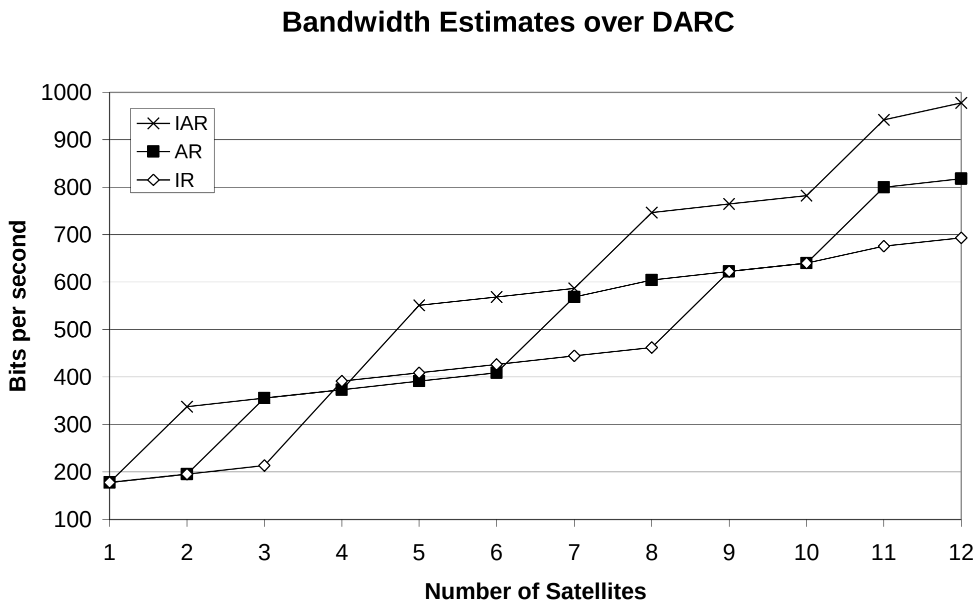



*Figure 18: The same as Figure 17 but with special interest in the transmission protocols with temporal compression.*

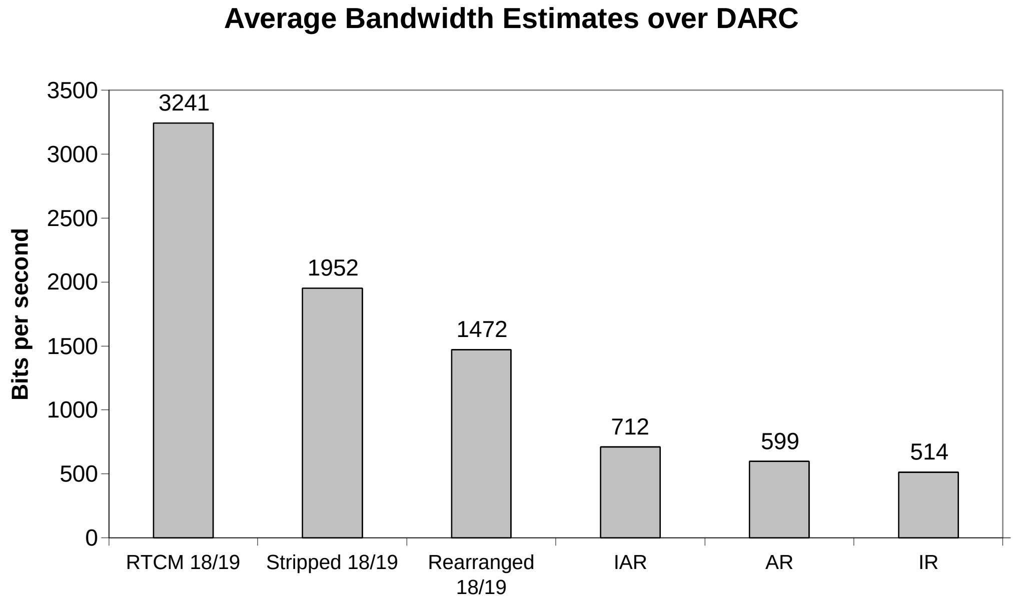



*Figure 19: Bandwidth requirement averaged over time.*

In Figure 19, the probability of a specific number of satellites in view is multiplied with the bandwidth requirement. The products for the different transmission protocols were then added to form the graph. The probabilities of the number of satellites in view were taken from [6]. The elevation angle of the satellites is limited to only 5 degrees, which perhaps is a little low for RTK. The probabilities also assume a clear view of the sky, but this is typically the case for a professional installation of a base station.

The assumption that there should be nine epochs between each epoch was chosen since this seems to be a likely selection. However, this might not always be the preferred selection. A small IDS interval increases the availability. Figure 20 has been included in order to aid in the selection of a suitable IDS interval.

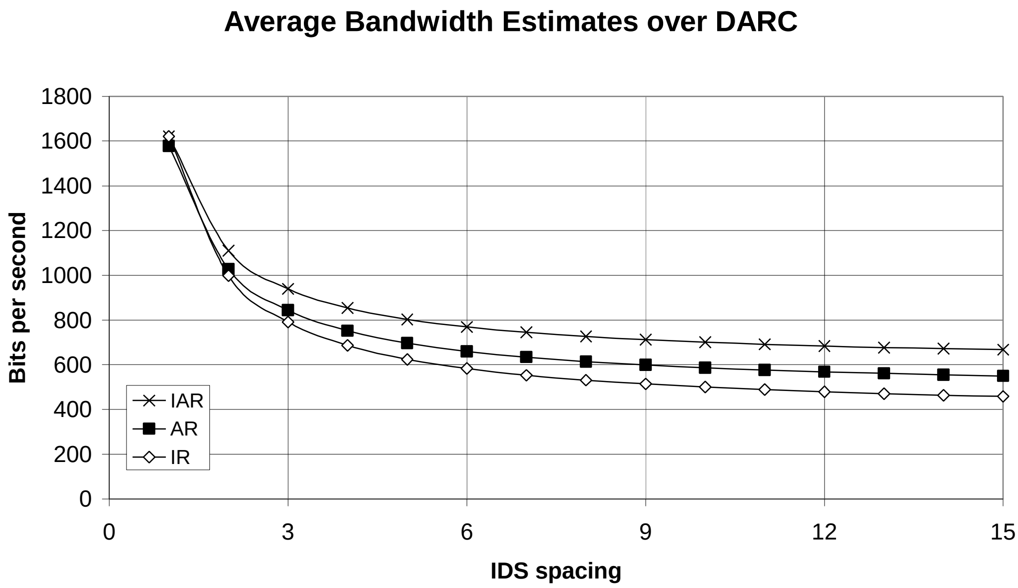



*Figure 20: Bandwidth requirements averaged over time as a function of the IDS interval in seconds.*

# 7 Conclusions and Further Studies

## 7.1 Conclusions

Five methods for compressing RTK data have been presented. Two of these methods have been included only to emphasize the reduction in bandwidth introduced by non-trivial compression strategies. The three real methods are Incremental Relativity (IR), Absolute Relativity (AR) and Interleaved Absolute Relativity (IAR). All these methods remove temporal redundancies, which impose a threat to the availability of the service – the user is more vulnerable to bad reception conditions.

If the RTK service is interrupted for a couple of seconds, the roving GPS receiver may have to reinitialize when the service reappears. Reinitializing can take as short as 10 seconds with optimal conditions, but with less than optimal conditions it can take several minutes.

The following discussion assumes 1 second between epochs and 10 seconds between initializations of the compression algorithm.

If no compression is utilized, one lost message will interrupt the service for 1 second. GPS receivers can typically handle such interruptions.

IR is very vulnerable to bad reception. If the decoder loses one message, delivery is interrupted for anything from 1 to 10 seconds. The longer the interruption, the higher probability that the GPS receiver is forced to reinitialize. This method is only interesting for scenarios where the DARC receiver is known to operate in good reception conditions.

AR is not as vulnerable as IR, but when the wrong message is lost, the service is interrupted for 10 seconds. When this happens, the GPS receiver is likely to reinitialize. The scenarios for AR are typically similar to the scenarios where IR is interesting.

IAR is improving availability even more compared to IR and AR. When one message is lost, the service is interrupted for 1 second for all satellites and possibly interrupted for 10 seconds for individual satellites. As long as there are RTK messages for more than 4 satellites available to the GPS receiver, it should be able to continue without reinitializing. One requirement is that the GPS receiver is tracking the same 4 satellites (at least) that it is receiving measurements and corrections for. IAR supports operation where occasional messages are lost, without forcing the GPS receiver to reinitialize. The IAR method imposes no major vulnerability and is interesting wherever RTK over DARC is interesting.

Since the difference in required bandwidth for IR, AR and IAR are comparably minor – a fact that was not anticipated – IAR should be the method of choice most of the time.

## 7.2 Further Studies

- **Error control**

  The RTCM error control was thrown out without a thorough investigation of the effects. Although it was argued that the DARC error control was *much* stronger, this has not been proven.

This has not been investigated in detail for two reasons. The outcome seems obvious and focus has been on other areas.

- **Scheduling of the transmission of IDSes for IAR**

  Some algorithm for scheduling the transmission of the IDS for different satellites when the IAR method is used is needed. The algorithm should typically distribute the IDS of the different satellites evenly over time, so that as few IDSes as possible are transmitted in the same message. This should be done while keeping the distance between two IDSes for the same satellite less than some configurable number of epochs/seconds.

  However, bandwidth could be gained if two or more IDSes are transmitted in the same message since DARC is block oriented.

  If the estimates of the carrier phase in the IDS are bad, it might be interesting to make new estimates and transmit a new IDS in order to reduce the corrections in the UDSes.

- **Spatial correlation between satellites**

  It has not been investigated if there exist inter-satellite redundancies. E.g. if the receiver has a clock error, this might generate a bias that is common to all observations. Perhaps there exist other sources of inter-satellite redundancies as well.

  This has not been investigated since it is likely that the redundancies are receiver dependent, which would require several different receivers from several different manufacturers in the investigation.

- **GLONASS base stations**

  The Russian equivalence of GPS is called GLONASS (Global Navigation Satellite System). The RTCM standard supports transmission of measurements and corrections for this system in parallel to GPS. It would be nice if these measurements and corrections were also compressed. Since no inter-satellite redundancies have been exploited, the addition of GLONASS is a matter of defining a new message. However, different satellites use different carrier frequencies in GLONASS, which makes it a bit more complicated.

  This has not been investigated since it requires a GLONASS receiver or some recorded GLONASS RTCM data.

- **Comparison with RTCM type 20/21**

  Instead of using RTCM Type 18/19, it is possible to use RTCM Type 20/21. As the messages are defined in the RTCM standard the required bandwidth of 18/19 and 20/21 are equal, but who knows what the results are after compression?

  This has not been investigated since it requires a GPS receiver that generates the 20/21 pair or some recorded data.

- **Evaluation in a real installation**

  The encoder and the decoder have not been implemented as a part of this thesis. It is of course interesting to see the results of an implementation in a real RTK installation.

  This has not been pursued since it is out of scope.

- **P(Y)-Code and C/A-Code messages in conjunction**

  Some base stations can generate messages for both P-Code and C/A-Code. This is not supported by the protocols. It is interesting to investigate the relation between the two and possibly exploit any redundancies.

  This has not been investigated further since it requires access to a receiver that delivers P-Code and C/A-Code RTCM messages in parallel.

# 8 References

[1] European Telecommunications Standards Institute, *Radio Broadcasting systems; System for Wireless Infotainment Forwarding and Teledistribution (SWIFT)*, ETS 300 751, 1997

[2] RTCM Special Committee No. 104, *RTCM Recommended Standards for Differential GNSS Service*, RTCM 11-98/SC104-STD Version 2.2, Radio Technical Commission for Maritime Services, 1998

[3] Shu Lin, Daniel J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, ISBN 0-13-283796-X, 1983

[4] Richard B. Langley, "RTK GPS", *GPS World Magazine*, Advanstar Communications, Inc., September 1998

[5] ARINC Research Corporation, *NAVSTAR GPS Space Segment/Navigation User Interfaces, Interface Control Document (ICD)*, ICD-GPS-200 Rev. C-PR (Public Release Version), ARINC Research Corporation, 1997

[6] NAVSTAR GPS Joint Program Office, *Global Positioning System Standard Positioning Service Signal Specification*, 2$^{nd}$ Edition, US Coast Guard Navigation Center, 1995

[7] Elliot D. Kaplan et al., *Understanding GPS: Principles and Application*, Artech House, Inc., ISBN 0-89006-793-7, 1996

[8] Lars Eldén, Linde Wittmeyer-Koch, *Numerisk analys – en introduktion*, Andra upplagan, Studentlitteratur, ISBN 91-44-25652-3, 1992

[9] Alfred Leick, *GPS Satellite Surveying*, 2$^{nd}$ Edition, John Wiley & Sons, Inc., ISBN 0-471-30626-6, 1994

[10] Robert Forchheimer, *Image Coding and Data Compression*, Dept. of Electrical Engineering, Linköping University, 1994

[11] Gunnar Blom, *Sannolikhetsteori och statistikteori med tillämpningar*, Fjärde upplagan, Studentlitteratur, ISBN 91-44-03594-2, 1989

[12] Werner Gurtner, Gerald M. Mader, "Receiver Independent Exchange Format Version 2", *CSTG GPS Bulletin*, National Geodetic Survey, vol.3, no.3, Sept/Oct, 1990

[13] Yuki Hatanaka, "A RINEX Compression Format and Tools", *Proc. ION-GPS-96*, September 17-20, 1996, pp. 177-183

# A Histograms of Corrections

The data sequences were generated using two different receivers. One sequence was generated using a Trimble 4000 SSi an afternoon in Austria in March 1998. The other sequence was generated with an Ashtech Z-Surveyor an evening in Sweden in December 1998. The Ashtech receiver did not perform as well as the Trimble. The reasons can be many, including the fact that only half the sky was visible to the Ashtech while the Trimble had a clear view. The Ashtech receiver was also subjected to multipath due to an unfavorable antenna installation.

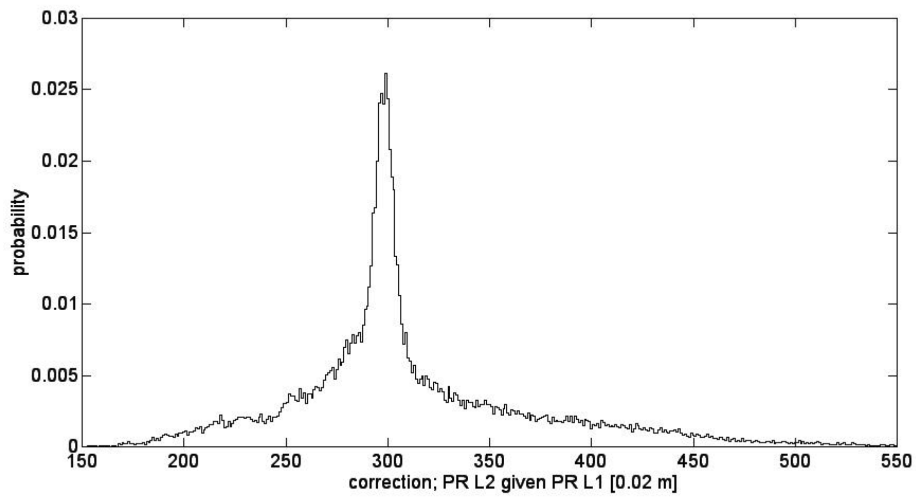
The axes in the plots are labeled in the following way:

- $C_{r_{L2}}$ – correction; PR L2 given PR L1
- $C_{\Delta_n f_{L2}}$ – correction; delta CP L2 given delta CP L1
- $C_{\Delta_n r_{L1}}$ – correction; delta PR L1 given delta CP L1
- $C_{\Delta_n r_{L2}}$ – correction; delta PR L2 given delta PR L1
- $C_{\Delta'_n r_{L2}}$ – correction; delta PR L2 given delta CP L2
- $\Delta_{n,n} f_{L1}$ – delta *n*,*n* CP L1
- $\Delta_{n,n,n} f_{L1}$ – delta *n*,*n*,*n* CP L1

This appendix uses the Trimble sequence only. The Ashtech sequence is used in later appendices.

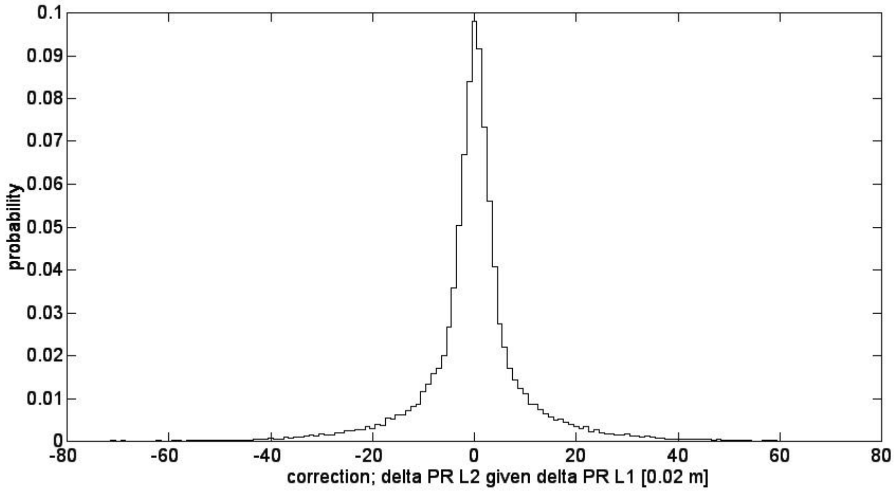
## A.1 Histogram of $C_{r_{L2}}$

This difference is mostly due to the ionosphere and the troposphere. The other errors affect the two frequencies in similar ways. It should be noted that the peak at 300 (6 meters) is depending on the status of the ionosphere and the troposphere. A much longer test is needed to get a more accurate view of the distribution. It can be noted that the ionosphere among other things is a function of the number of sunspots, which have a cycle of 11 years…

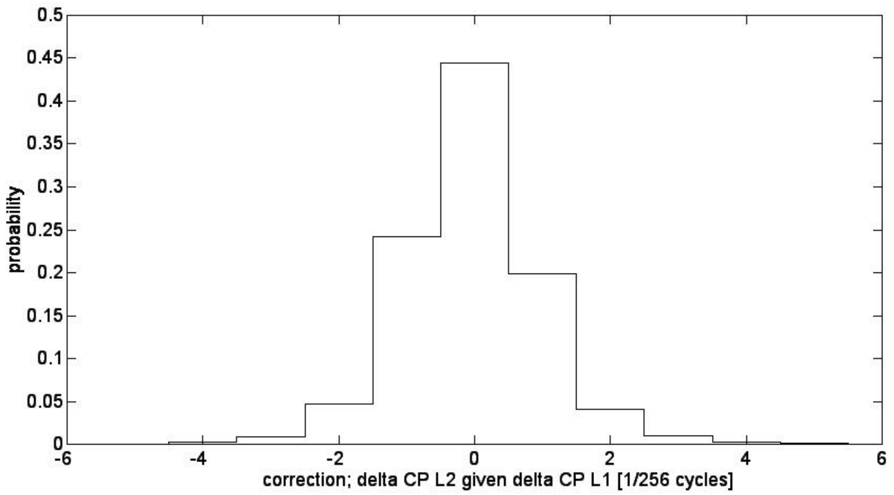


## A.2 Histogram of $C_{\Delta_1 r_{L2}}$

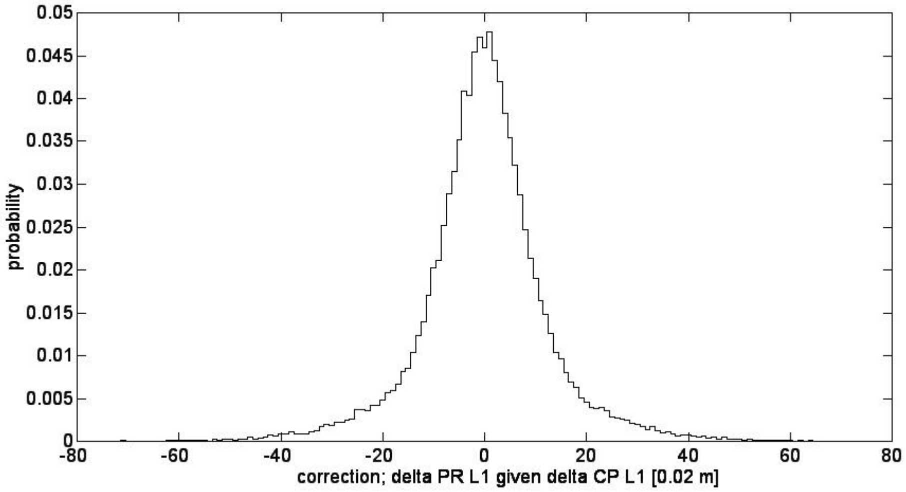
Compared to the correction in section A.1, this correction is iono-free. I.e. the effect of the ionosphere has canceled out.



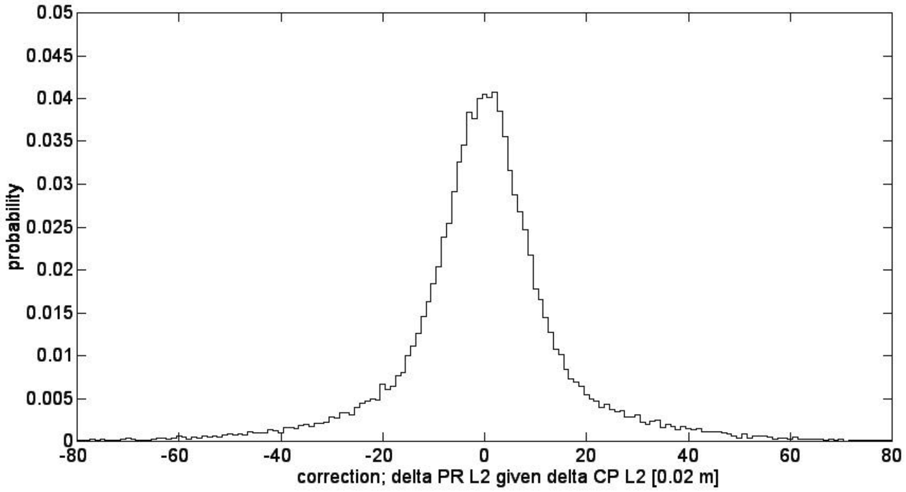

## A.3 Histogram of $C_{\Delta_1 f_{L2}}$



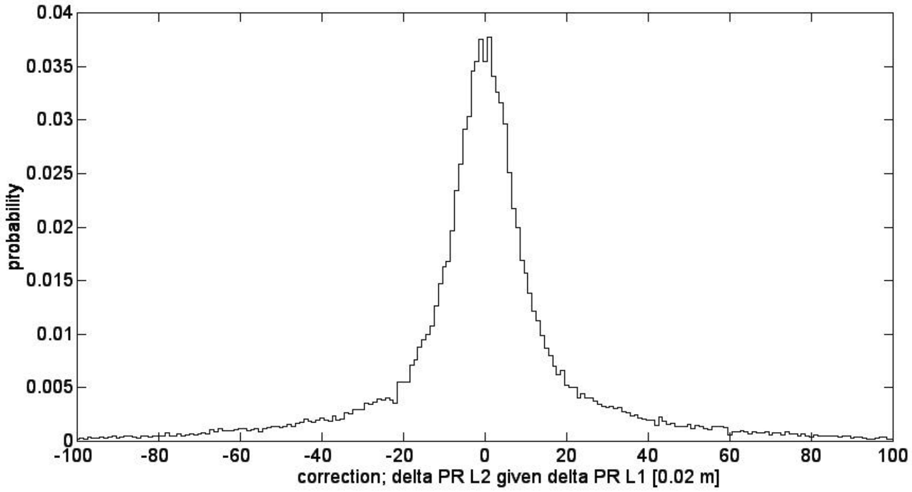
## A.4 Histogram of $C_{\Delta_1 r_{L1}}$



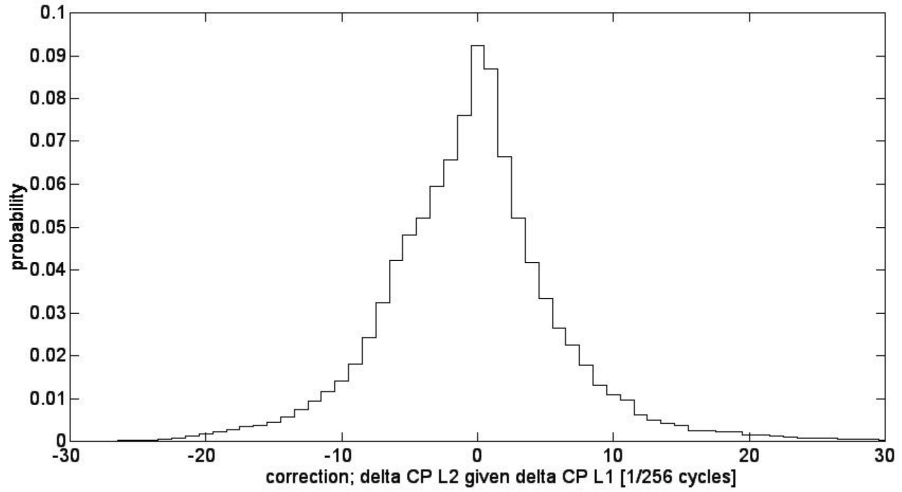

## A.5 Histogram of $C_{\Delta_1' r_{L2}}$



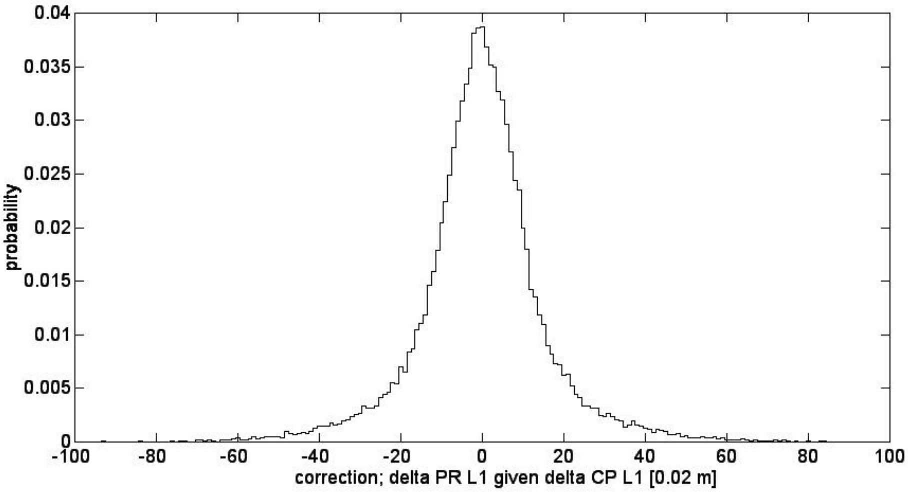

## A.6 Histogram of $C_{\Delta_{10} r_{L2}}$



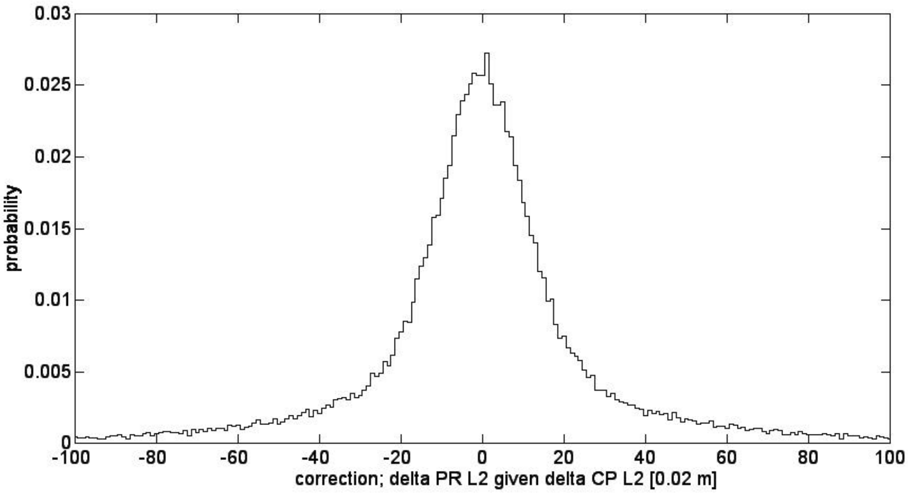
## A.7 Histogram of $C_{\Delta_{10} f_{L2}}$




## A.8 Histogram of $C_{\Delta_{10} r_{L1}}$




## A.9 Histogram of $C_{\Delta'_{10} r_{L2}}$

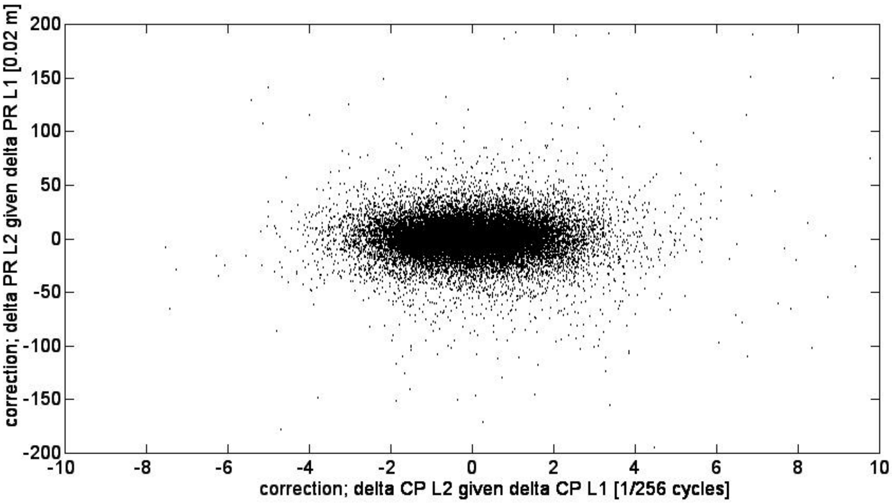


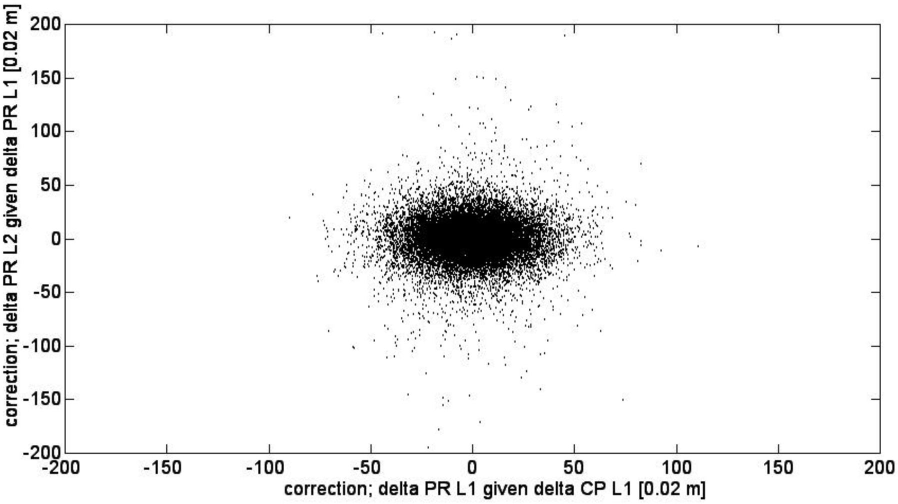
# B Spatial Correlation Between Corrections

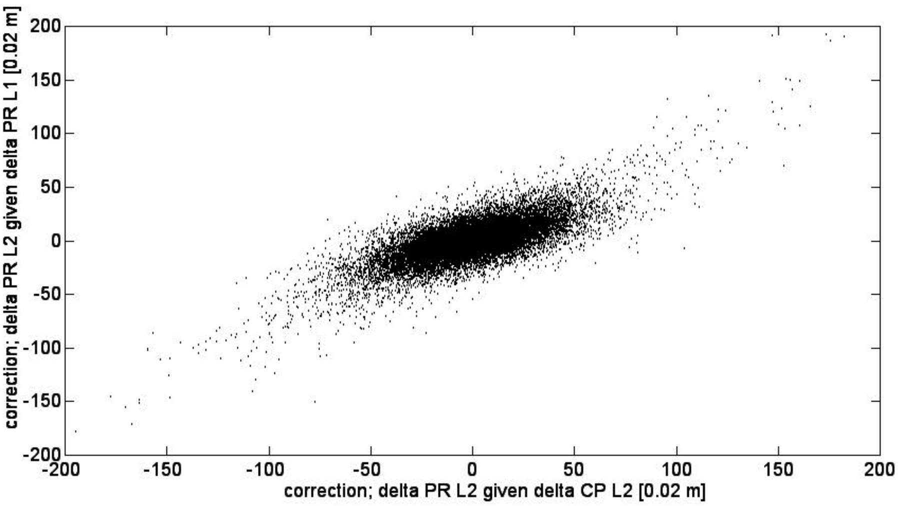
This appendix illustrates the correlation between the estimation errors derived in section 4.4. Eighteen plots are included. See appendix A for a description of the sequences used and for the notation in the figures.

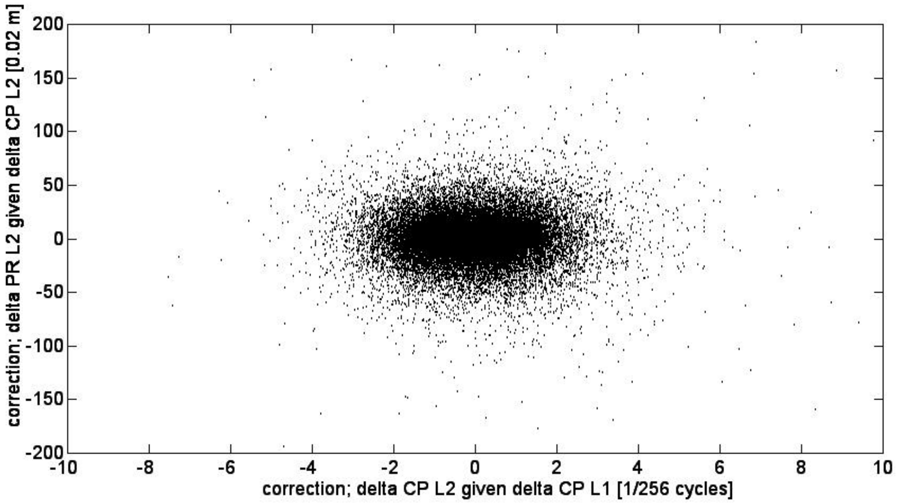
The first six plots represent the correlation between $C_{\Delta_1 f_{L2}}$, $C_{\Delta_1 r_{L1}}$, $C_{\Delta_1 r_{L2}}$ and $C_{\Delta'_1 r_{L2}}$ using the Trimble data sequence. The following six plots represent the correlation between $C_{\Delta_{10} f_{L2}}$, $C_{\Delta_{10} r_{L1}}$, $C_{\Delta_{10} r_{L2}}$ and $C_{\Delta'_{10} r_{L2}}$, again using the Trimble data sequence. The last six plots again represent $C_{\Delta_{10} f_{L2}}$, $C_{\Delta_{10} r_{L1}}$, $C_{\Delta_{10} r_{L2}}$ and $C_{\Delta'_{10} r_{L2}}$, but using the Ashtech data sequence.

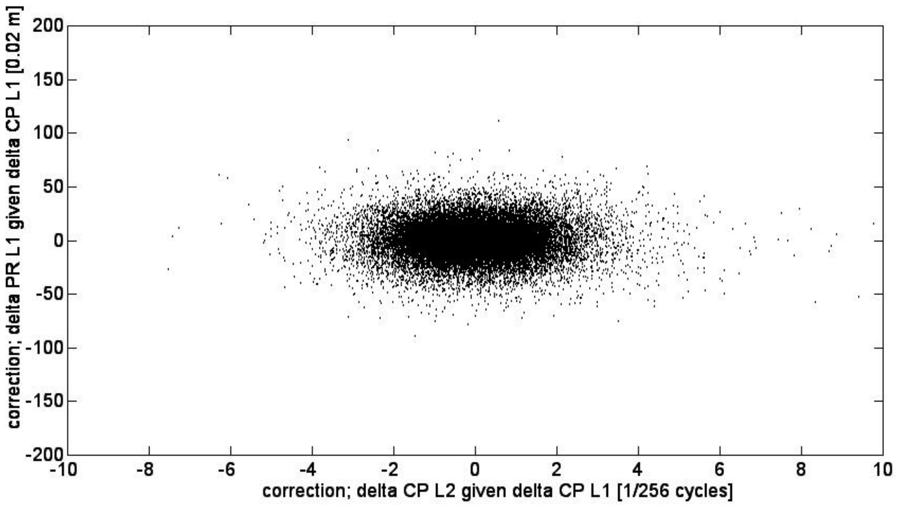
If two variables are correlated, there will be a clear slope of the noise blob – a narrow blob represents high correlation. The plots show that $C_{\Delta_n r_{L2}}$ and $C_{\Delta'_n r_{L2}}$ are somewhat dependent, but the correlation is only slight. Some other dependencies also appear, but these are also minor.

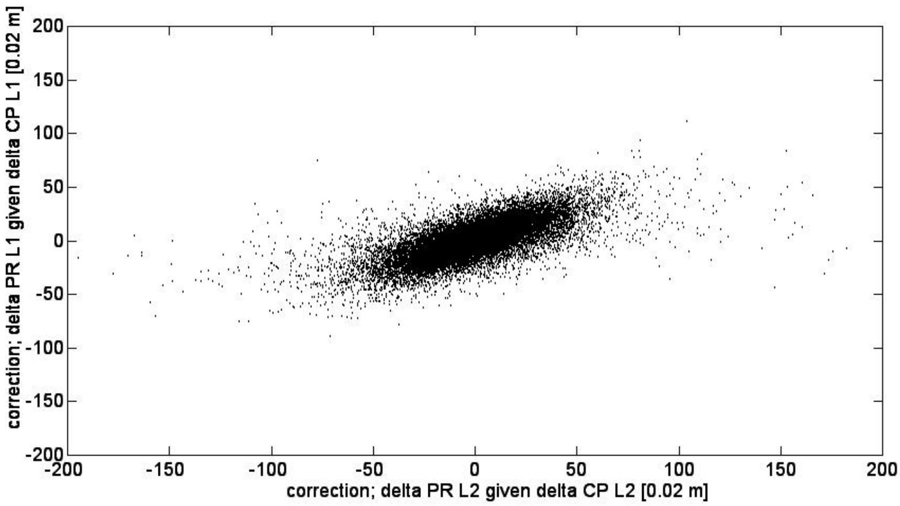

correction; delta PR L2 given delta PR L1 [0.02 m]
correction; delta CP L2 given delta CP L1 [1/256 cycles]

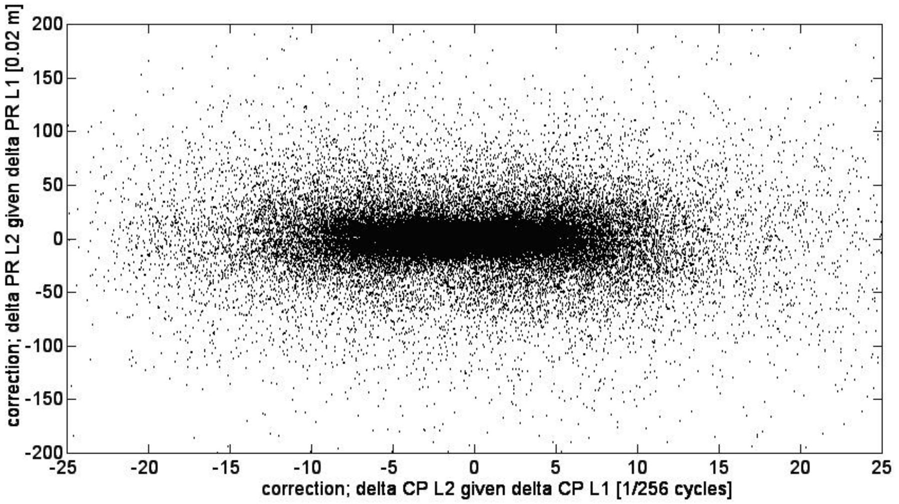

correction; delta PR L2 given delta PR L1 [0.02 m]
correction; delta PR L1 given delta CP L1 [0.02 m]

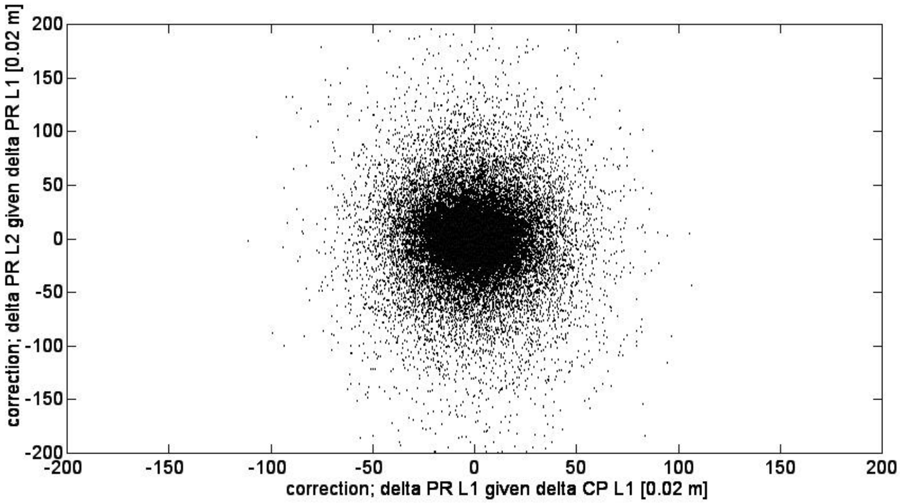

correction; delta PR L2 given delta PR L1 [0.02 m]
correction; delta PR L2 given delta CP L2 [0.02 m]

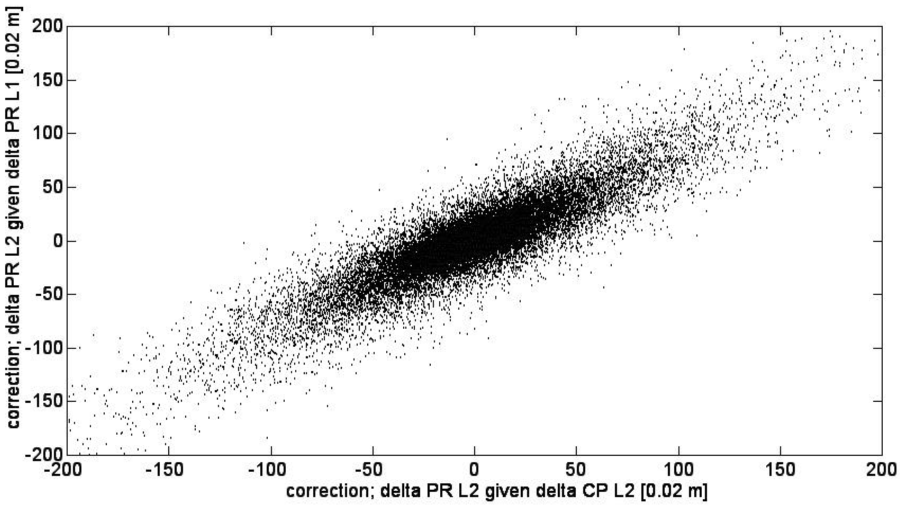

correction; delta PR L2 given delta CP L2 [0.02 m]
correction; delta CP L2 given delta CP L1 [1/256 cycles]

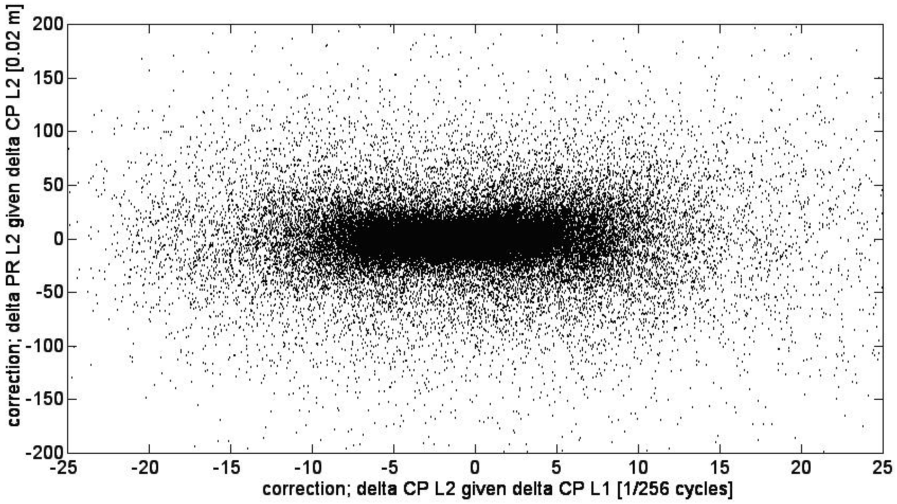

correction; delta PR L1 given delta CP L1 [0.02 m]
correction; delta CP L2 given delta CP L1 [1/256 cycles]

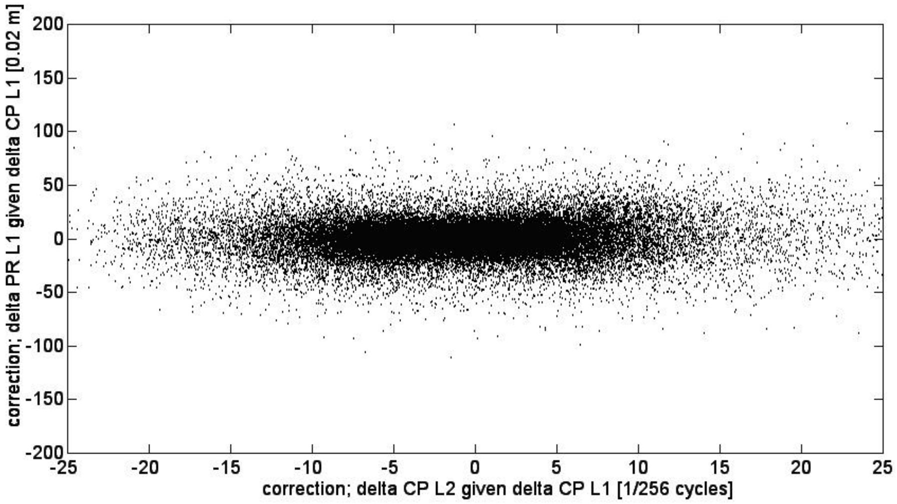

correction; delta PR L1 given delta CP L1 [0.02 m]
correction; delta PR L2 given delta CP L2 [0.02 m]

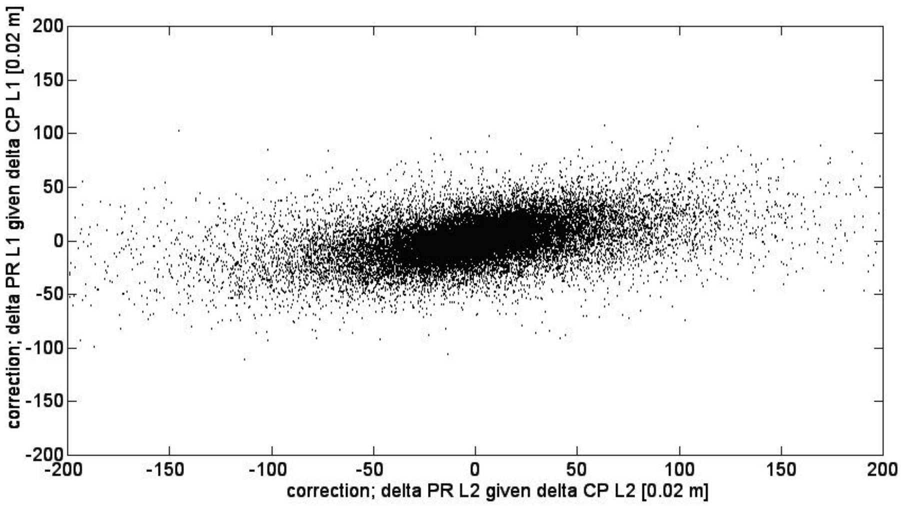

correction; delta PR L2 given delta PR L1 [0.02 m]
correction; delta CP L2 given delta CP L1 [1/256 cycles]

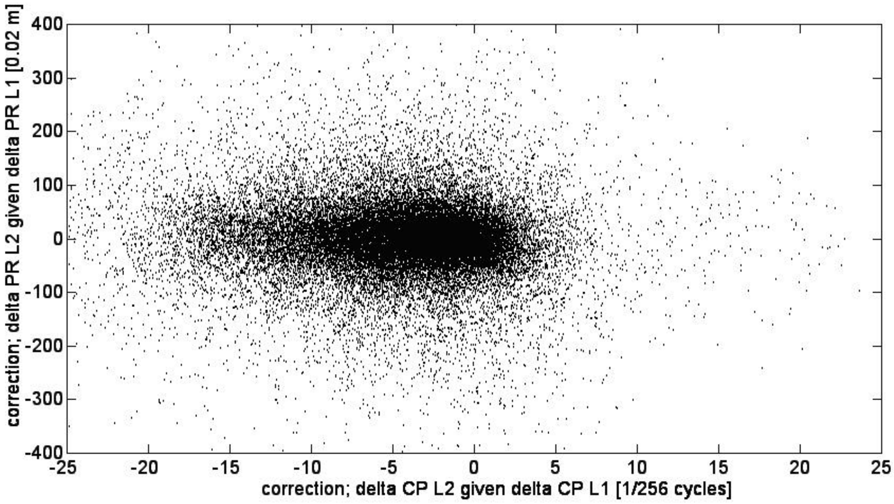

correction; delta PR L2 given delta PR L1 [0.02 m]
correction; delta PR L1 given delta CP L1 [0.02 m]

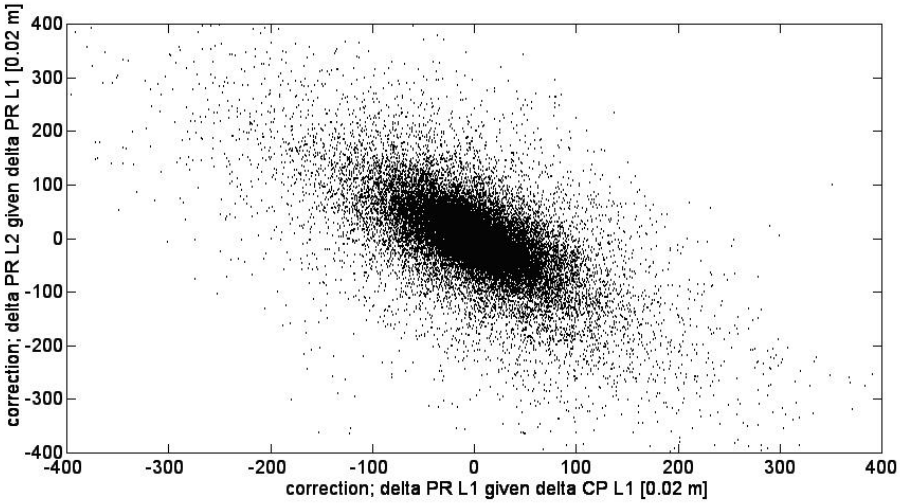

correction; delta PR L2 given delta PR L1 [0.02 m]
correction; delta PR L2 given delta CP L2 [0.02 m]

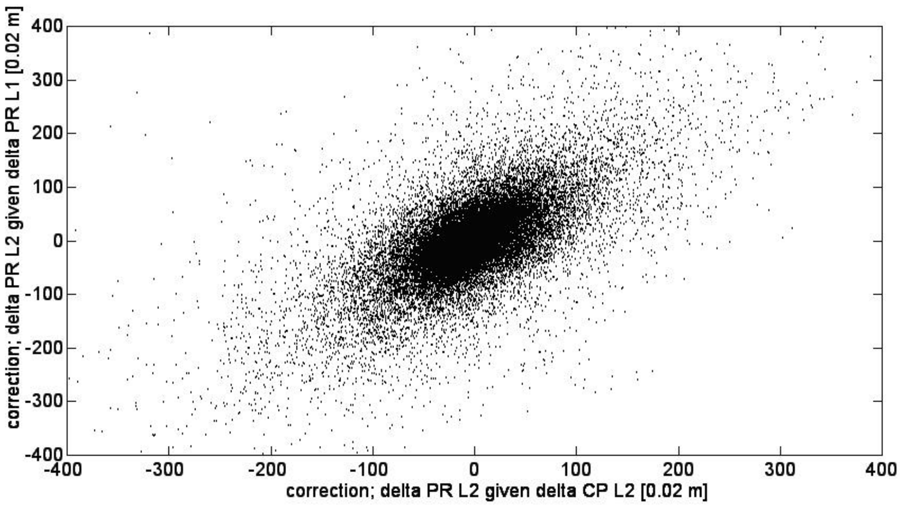

correction; delta PR L2 given delta CP L2 [0.02 m]
correction; delta CP L2 given delta CP L1 [1/256 cycles]

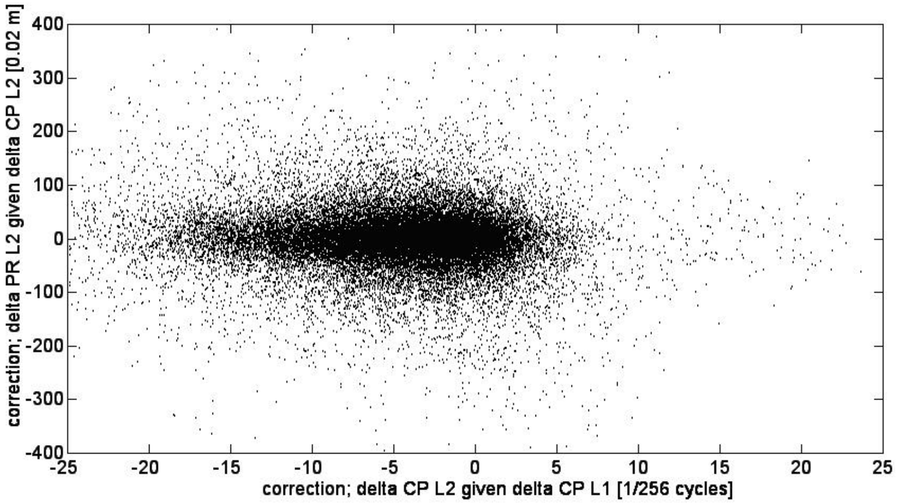

correction; delta PR L1 given delta CP L1 [0.02 m]
correction; delta CP L2 given delta CP L1 [1/256 cycles]

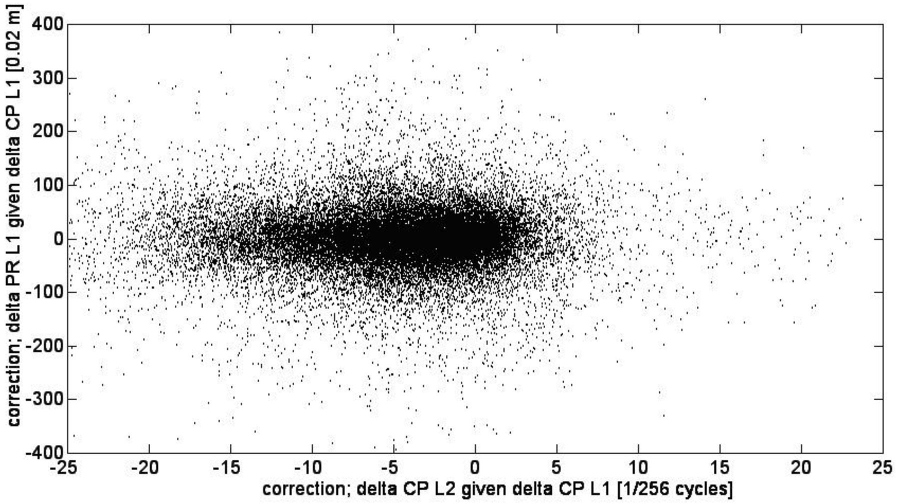

correction; delta PR L1 given delta CP L1 [0.02 m]
correction; delta PR L2 given delta CP L2 [0.02 m]

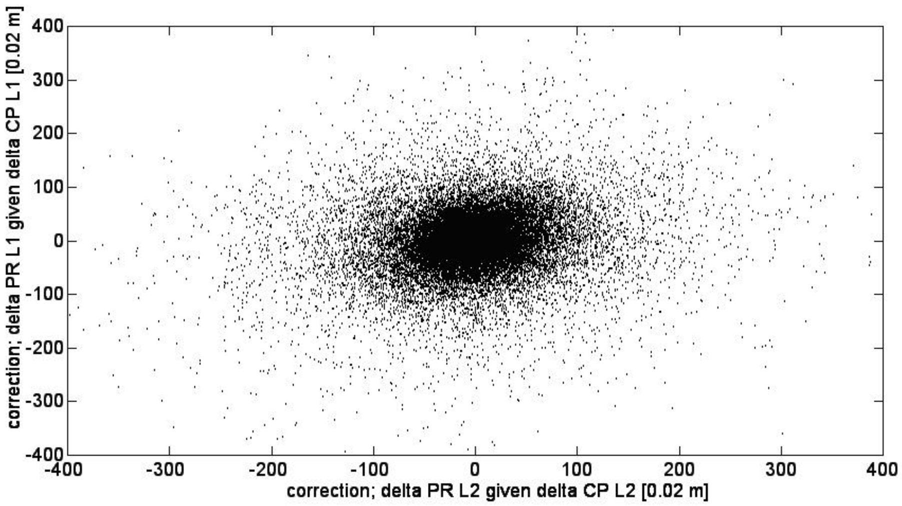

correction; delta PR L2 given delta PR L1 [0.02 m]
correction; delta CP L2 given delta CP L1 [1/256 cycles]


correction; delta PR L2 given delta PR L1 [0.02 m]
correction; delta PR L1 given delta CP L1 [0.02 m]


correction; delta PR L2 given delta PR L1 [0.02 m]
correction; delta PR L2 given delta CP L2 [0.02 m]


correction; delta PR L2 given delta CP L2 [0.02 m]
correction; delta CP L2 given delta CP L1 [1/256 cycles]


correction; delta PR L1 given delta CP L1 [0.02 m]
correction; delta CP L2 given delta CP L1 [1/256 cycles]

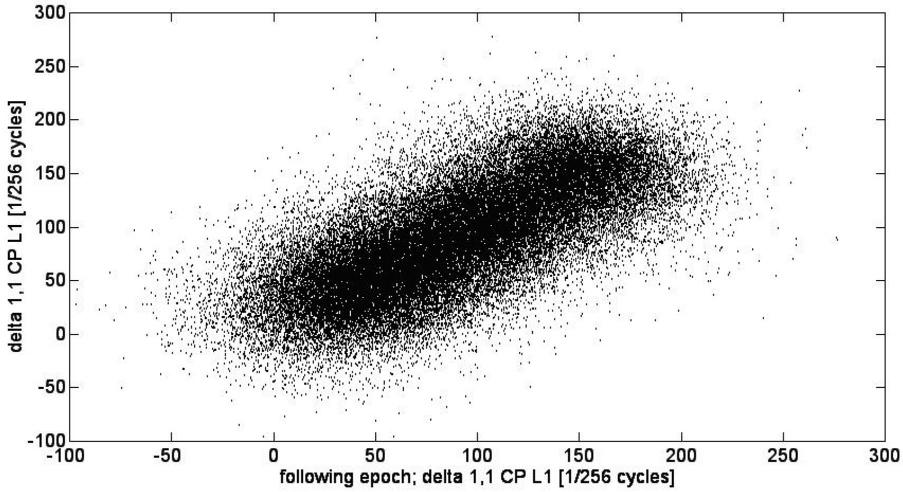

correction; delta PR L1 given delta CP L1 [0.02 m]
correction; delta PR L2 given delta CP L2 [0.02 m]

# C Temporal Statistics

This appendix presents data from the Trimble sequence. See appendix A for a description of the sequences used and for the notation in the figures.

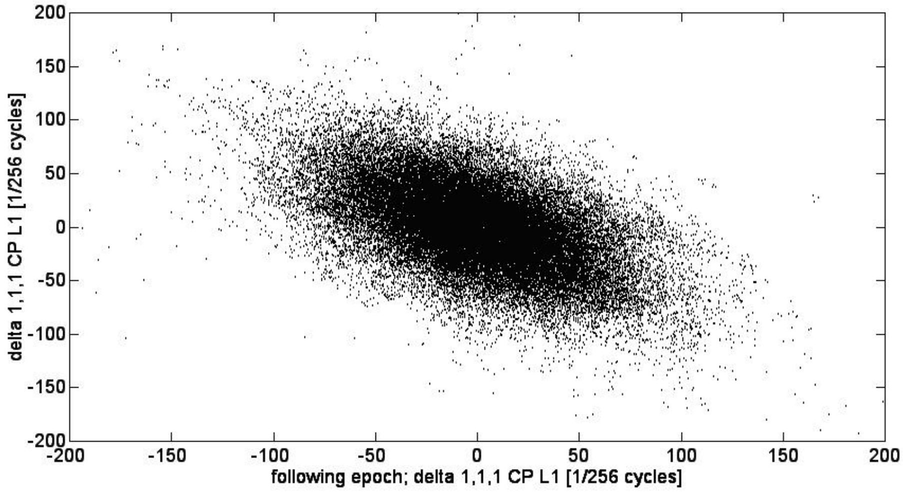
## C.1 Temporal Correlation of $C_{\Delta_{1,1} f_{L1}}$

The data is not centered on zero and large values in one epoch are followed by large values in the following epoch.

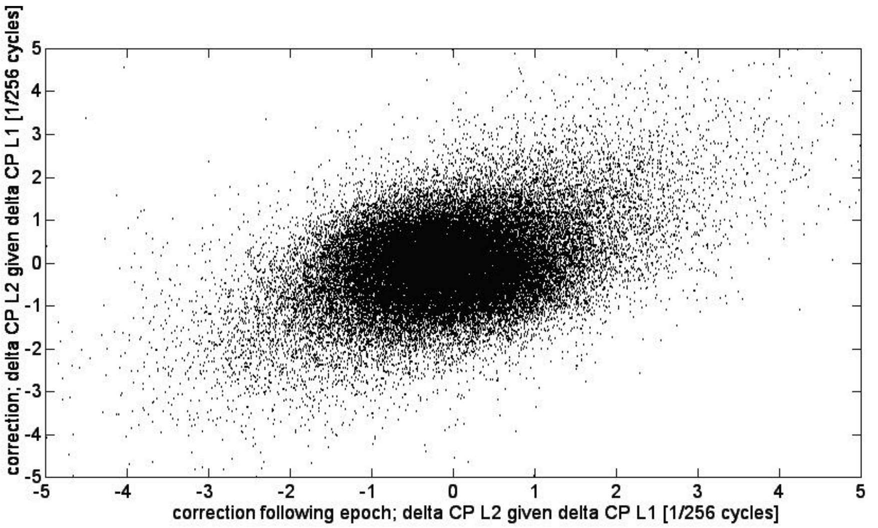



## C.2 Temporal Correlation of $C_{\Delta_{1,1,1} f_{L1}}$

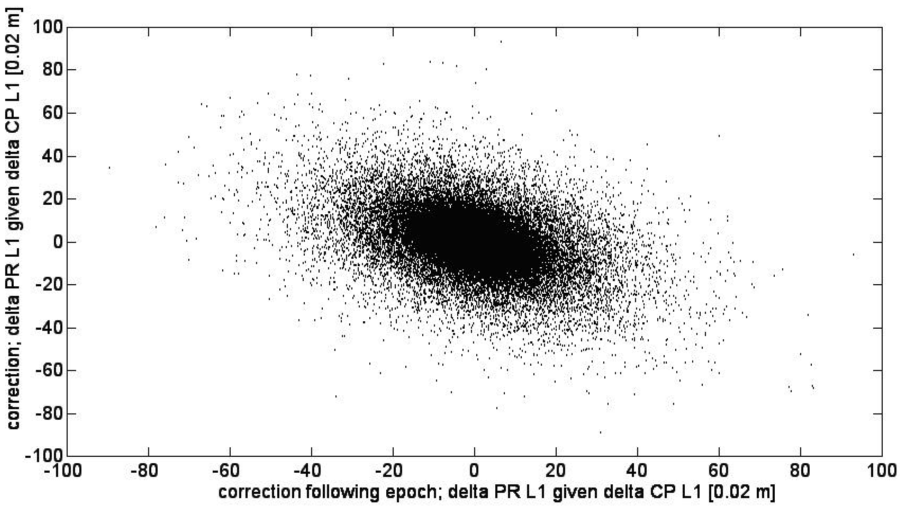
The data is now centered on zero and the blob is more circular.



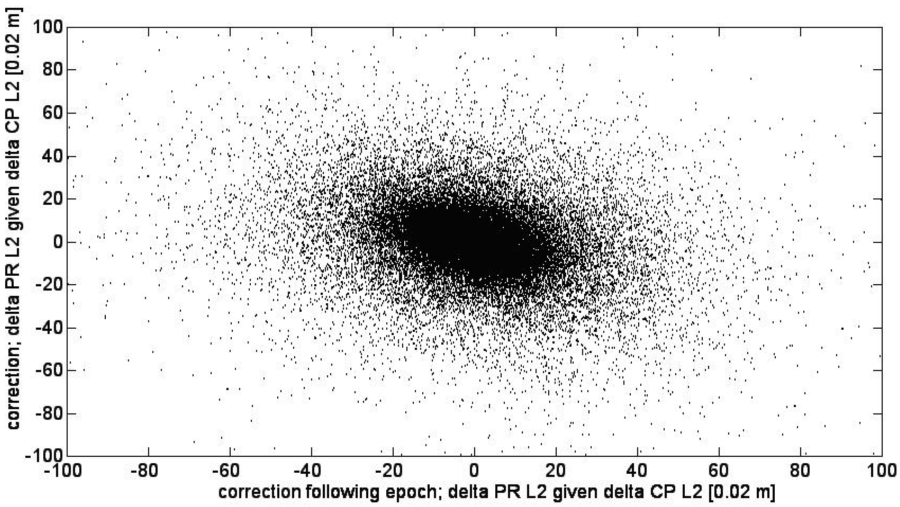
## C.3 Temporal Correlation of $C_{\Delta_1 f_{L2}}$




## C.4 Temporal Correlation of $C_{\Delta_1 r_{L1}}$




## C.5 Temporal Correlation of $C_{\Delta_1 r_{L2}}$



# D Integer Multiplication by a Rational Number

This appendix describes one way to multiply an integer by a rational number given that the integer is a 32 bit signed number and that the rational number is represented as a quotient of two 15 bit unsigned integers.

The expression we are trying to calculate is shown in equation (56), where x is the 32 bit signed number and $y/z$ is the rational number.

$$r = \left\lfloor x \cdot \frac{y}{z} + 1/2 \right\rfloor \tag{56}$$

Since x and y are integers and since z is positive, this can be rewritten as:

$$r = \left\lfloor \frac{x \cdot y + \lfloor z/2 \rfloor}{z} \right\rfloor \tag{57}$$

If % represent the modulo operator, equations (58) and (59) are true. Beware however, the modulo operator have different implementations for negative operands in different environments. Therefore, x should be non-negative.

$$x/z = \lfloor x/z \rfloor + \frac{x\%z}{z} \tag{58}$$

$$x \cdot y = \lfloor x/2^k \rfloor \cdot y \cdot 2^k + \left(x\%2^k\right) \cdot y \tag{59}$$

Equations (58) and (59) are used to rewrite equation (57).

$$\begin{aligned}
r &= \left\lfloor \frac{\lfloor x/2^k \rfloor \cdot y \cdot 2^k + \left(x\%2^k\right) \cdot y + \lfloor z/2 \rfloor}{z} \right\rfloor \\
&= \left\lfloor \frac{\lfloor x/2^k \rfloor \cdot y}{z} \cdot 2^k + \frac{\left(x\%2^k\right) \cdot y + \lfloor z/2 \rfloor}{z} \right\rfloor \\
&= \left\lfloor \left\lfloor \frac{\lfloor x/2^k \rfloor \cdot y}{z} \right\rfloor \cdot 2^k + \frac{\left(\lfloor x/2^k \rfloor \cdot y\right)\%z}{z} \cdot 2^k + \frac{\left(x\%2^k\right) \cdot y + \lfloor z/2 \rfloor}{z} \right\rfloor \\
&= \left\lfloor \frac{\lfloor x/2^k \rfloor \cdot y}{z} \right\rfloor \cdot 2^k + \left\lfloor \frac{\left(\left(\lfloor x/2^k \rfloor \cdot y\right)\%z\right) \cdot 2^k + \left(x\%2^k\right) \cdot y + \lfloor z/2 \rfloor}{z} \right\rfloor
\end{aligned} \tag{60}$$

For this final expression to work, all numerators have to fit in 32 bits. If k is set to 16, this is the case. The numerator in the first term is easily verified to fit in 32 bits, but the second term is more delicate.

$$\begin{aligned}
numerator &= \left(\left(\lfloor x/2^k \rfloor \cdot y\right)\%z\right) \cdot 2^k + \left(x\%2^k\right) \cdot y + \lfloor z/2 \rfloor \\
&\le \left(2^{k-1} - 2\right) \cdot 2^k + \left(2^k - 1\right) \cdot \left(2^{k-1} - 1\right) + \left(2^{k-2} - 1\right) \\
&= 2^{2k-1} - 2^{k+1} + 2^{2k-1} - 2^k - 2^{k-1} + 1 + 2^{k-2} - 1 \\
&= 2^{2k} - 2^{k+1} - 2^k - 2^{k-1} + 2^{k-2} \\
&= 2^{2k} - 13 \cdot 2^{k-2} < 2^{2k}
\end{aligned} \tag{61}$$

Since z is a 15 bit number, $z \le 2^{k-1} - 1$. It follows that $w\%z \le 2^{k-1} - 2$.

## D.1 Implementation in C

This section presents a reference implementation of the above calculation. Remember the following things:

- y and z have to be positive and less than $2^{15}$.
- Although x was defined as any signed 32 bit integer, $x = -2^{31}$ will not work as input since this number is too large to represent when positive[1].
- The result will have to fit in a signed 32 bit integer as well, or the result is undefined[2].

```
int
mult_rational(int x, int y, int z)
{
  int result, high_xy, high_res;
  unsigned int high_rest, low_rest;

  int negative = 0;
  if (x < 0) {
    negative = 1;
    x = -x;
  }

  high_xy   = (x >> 16) * y;
  high_res  = (high_xy / z) << 16;
  high_rest = (high_xy % z) << 16;
  low_rest  = (x & 0xffff) * y + (z >> 1);

  result = high_res + (high_rest + low_rest) / z;

  return negative ? –result : result;
}
```

[1] Assuming that integers are implemented as two-complement binary numbers.

[2] The definition of the result is left as an exercise to the reader.

# E Aspects of the Carrier Phase Estimate

All expressions in this appendix apply to the estimates of the carrier phase used in section 4.5.2. The index of $f$ has been dropped to improve readability.

$\Delta_{1,1} f(t)$ was defined in equation (29), but a variant of the definition is restated in equation (62).

$$\tilde{\Delta}_{1,1} f(t) = \tilde{\Delta}_1 f(t) - \tilde{\Delta}_1 f(t-1) \tag{62}$$

This definition can be used to state equation (63). In chapter 4, $\tilde{\Delta}_1 f(t)$ was defined to change linearly over time, and this is the relation.

$$\tilde{\Delta}_1 f(t) = \tilde{\Delta}_1 f(t_0) + (t - t_0) \cdot \tilde{\Delta}_{1,1} f(t_0) \tag{63}$$

We are now interested in how the equation for $\tilde{f}(t)$ – that has this property of $\tilde{\Delta}_1 f(t)$ – looks. A variant of equation (2) is restated here as equation (64).

$$\tilde{\Delta}_1 f(t) = \tilde{f}(t) - \tilde{f}(t-1) \tag{64}$$

Equation (64) can be used to realize that equation (65) is true.

$$\begin{aligned} \tilde{f}(t) &= \langle t = t_0 + s \rangle = \tilde{f}(t_0 + s) \\ &= \tilde{f}(t_0) + \sum_{i=1}^{s} \tilde{\Delta}_1 f(t_0 + i) \end{aligned} \tag{65}$$

Substituting equation (63) into equation (65) gives equation (66).

$$\begin{aligned} \tilde{f}(t) &= \tilde{f}(t_0) + \sum_{i=1}^{s} \tilde{\Delta}_1 f(t_0) + (t_0 + i - t_0)\tilde{\Delta}_{1,1} f(t_0) \\ &= \tilde{f}(t_0) + s\tilde{\Delta}_1 f(t_0) + \sum_{i=1}^{s} i\tilde{\Delta}_{1,1} f(t_0) \\ &= \tilde{f}(t_0) + s\tilde{\Delta}_1 f(t_0) + \frac{s^2 + s}{2}\tilde{\Delta}_{1,1} f(t_0) \\ &= \frac{\tilde{\Delta}_{1,1} f(t_0)}{2} \cdot s^2 + \left( \tilde{\Delta}_1 f(t_0) + \frac{\tilde{\Delta}_{1,1} f(t_0)}{2} \right) \cdot s + \tilde{f}(t_0) \\ &\langle s = t - t_0 \rangle \\ &= \frac{\tilde{\Delta}_{1,1} f(t_0)}{2} \cdot (t - t_0)^2 + \left( \tilde{\Delta}_1 f(t_0) + \frac{\tilde{\Delta}_{1,1} f(t_0)}{2} \right) \cdot (t - t_0) + \tilde{f}(t_0) \end{aligned} \tag{66}$$

The same exercise is now going to be performed for half second intervals in order to find out how $\tilde{\Delta}_1 f(t_0)$ and $\tilde{\Delta}_{1,1} f(t_0)$ are related to $\tilde{\Delta}_{1/2} f(t_0)$ and $\tilde{\Delta}_{1/2,1/2} f(t_0)$.

The half second variant of equation (62) is found in equation (67).

$$\tilde{\Delta}_{1/2,1/2} f(t) = \tilde{\Delta}_{1/2} f(t) - \tilde{\Delta}_{1/2} f(t - 1/2) \tag{67}$$

Equation (68) can be derived from equation (67) in the same way equation (63) was derived from equation (62).

$$\tilde{\Delta}_{1/2} f(t) = \tilde{\Delta}_{1/2} f(t_0) + 2(t - t_0) \cdot \tilde{\Delta}_{1/2,1/2} f(t_0) \tag{68}$$

The equivalence of equation (64) for half second intervals is stated in equation (69).

(69) $$\tilde{\Delta}_{1/2} f(t) = \tilde{f}(t) - \tilde{f}(t - 1/2)$$

Equation (69) can be used to realize that equation (70) is true, in the same way equation (64) explained equation (65).

(70) $$\begin{aligned} \tilde{f}(t) &= \langle t = t_0 + s \rangle = \tilde{f}(t_0 + s) \\ &= \tilde{f}(t_0) + \sum_{i=1}^{2s} \tilde{\Delta}_{1/2} f(t_0 + i/2) \end{aligned}$$

Substituting equation (68) into equation (70) gives equation (71).

(71) $$\begin{aligned} \tilde{f}(t) &= \tilde{f}(t_0) + \sum_{i=1}^{2s} \tilde{\Delta}_{1/2} f(t_0) + 2(t_0 + i/2 - t_0)\tilde{\Delta}_{1/2,1/2} f(t_0) \\ &= \tilde{f}(t_0) + 2s\tilde{\Delta}_{1/2} f(t_0) + \sum_{i=1}^{2s} i\tilde{\Delta}_{1/2,1/2} f(t_0) \\ &= \tilde{f}(t_0) + 2s\tilde{\Delta}_{1/2} f(t_0) + \left(2s^2 + s\right)\tilde{\Delta}_{1/2,1/2} f(t_0) \\ &= 2\tilde{\Delta}_{1/2,1/2} f(t_0) \cdot s^2 + \left(2\tilde{\Delta}_{1/2} f(t_0) + \tilde{\Delta}_{1/2,1/2} f(t_0)\right) \cdot s + \tilde{f}(t_0) \\ &\langle s = t - t_0 \rangle \\ &= 2\tilde{\Delta}_{1/2,1/2} f(t_0) \cdot (t - t_0)^2 + \left(2\tilde{\Delta}_{1/2} f(t_0) + \tilde{\Delta}_{1/2,1/2} f(t_0)\right) \cdot (t - t_0) + \tilde{f}(t_0) \end{aligned}$$

Equations (66) and (71) represent the same thing. Identifying the coefficients of $(t - t_0)^2$ gives the relation shown in equation (72).

(72) $$\begin{aligned} \frac{\tilde{\Delta}_{1,1} f(t_0)}{2} &= 2\tilde{\Delta}_{1/2,1/2} f(t_0) \\ \tilde{\Delta}_{1,1} f(t_0) &= 4\tilde{\Delta}_{1/2,1/2} f(t_0) \end{aligned}$$

Similarly, identifying the coefficients of $(t - t_0)$ gives the relation stated in expression (73).

(73) $$\begin{aligned} \tilde{\Delta}_1 f(t_0) + \frac{\tilde{\Delta}_{1,1} f(t_0)}{2} &= 2\tilde{\Delta}_{1/2} f(t_0) + \tilde{\Delta}_{1/2,1/2} f(t_0) \\ \tilde{\Delta}_1 f(t_0) &= 2\tilde{\Delta}_{1/2} f(t_0) + \tilde{\Delta}_{1/2,1/2} f(t_0) - \frac{\tilde{\Delta}_{1,1} f(t_0)}{2} \\ \tilde{\Delta}_1 f(t_0) &= 2\tilde{\Delta}_{1/2} f(t_0) + \tilde{\Delta}_{1/2,1/2} f(t_0) - 2\tilde{\Delta}_{1/2,1/2} f(t_0) \\ \tilde{\Delta}_1 f(t_0) &= 2\tilde{\Delta}_{1/2} f(t_0) - \tilde{\Delta}_{1/2,1/2} f(t_0) \end{aligned}$$

The same exercise can be performed for intervals of two seconds. This is not done here, but the results are presented in equation (74).

(74) $$\begin{aligned} \tilde{\Delta}_2 f(t_0) &= 2\tilde{\Delta}_1 f(t_0) - \tilde{\Delta}_{1,1} f(t_0) \\ \tilde{\Delta}_{2,2} f(t_0) &= 4\tilde{\Delta}_{1,1} f(t_0) \end{aligned}$$

Solving for $\tilde{\Delta}_1 f(t_0)$ and $\tilde{\Delta}_{1,1} f(t_0)$ results in equation (75).

$$\begin{aligned} \tilde{\Delta}_1 f(t_0) &= \frac{1}{2}\tilde{\Delta}_2 f(t_0) + \frac{1}{8}\tilde{\Delta}_{2,2} f(t_0) \\ \tilde{\Delta}_{1,1} f(t_0) &= \frac{1}{4}\tilde{\Delta}_{2,2} f(t_0) \end{aligned} \tag{75}$$

Expressions for $\tilde{\Delta}_1 f(t_0)$ and $\tilde{\Delta}_{1,1} f(t_0)$ given $\tilde{\Delta}_{1/2} f(t_0)$ and $\tilde{\Delta}_{1/2,1/2} f(t_0)$ or $\tilde{\Delta}_2 f(t_0)$ and $\tilde{\Delta}_{2,2} f(t_0)$ have been derived. The expressions are collected in (76) for clarity.

$$\begin{aligned} \tilde{\Delta}_1 f(t_0) &= 2\tilde{\Delta}_{1/2} f(t_0) - \tilde{\Delta}_{1/2,1/2} f(t_0) \\ \tilde{\Delta}_{1,1} f(t_0) &= 4\tilde{\Delta}_{1/2,1/2} f(t_0) \\ \tilde{\Delta}_1 f(t_0) &= \frac{1}{2}\tilde{\Delta}_2 f(t_0) + \frac{1}{8}\tilde{\Delta}_{2,2} f(t_0) \\ \tilde{\Delta}_{1,1} f(t_0) &= \frac{1}{4}\tilde{\Delta}_{2,2} f(t_0) \end{aligned} \tag{76}$$

A correction to $\tilde{\Delta}_n f(t_0 + n)$ is transmitted in the UDS of the AR and the IAR methods. It will now be shown how $\tilde{\Delta}_n f(t_0 + n)$ should be calculated using $\tilde{\Delta}_1 f(t_0)$ and $\tilde{\Delta}_{1,1} f(t_0)$.

$$\begin{aligned} \tilde{\Delta}_n f(t_0 + n) &= \tilde{f}(t_0 + n) - \tilde{f}(t_0) \\ &= \frac{\tilde{\Delta}_{1,1} f(t_0)}{2} \cdot (t_0 + n - t_0)^2 \\ &\quad + \left(\tilde{\Delta}_1 f(t_0) + \frac{\tilde{\Delta}_{1,1} f(t_0)}{2}\right) \cdot (t_0 + n - t_0) + \tilde{f}(t_0) \\ &\quad - \frac{\tilde{\Delta}_{1,1} f(t_0)}{2} \cdot (t_0 - t_0)^2 \\ &\quad - \left(\tilde{\Delta}_1 f(t_0) + \frac{\tilde{\Delta}_{1,1} f(t_0)}{2}\right) \cdot (t_0 - t_0) - \tilde{f}(t_0) \\ &= \frac{\tilde{\Delta}_{1,1} f(t_0)}{2} \cdot n^2 + \left(\tilde{\Delta}_1 f(t_0) + \frac{\tilde{\Delta}_{1,1} f(t_0)}{2}\right) \cdot n \\ &= n \cdot \tilde{\Delta}_1 f(t_0) + \frac{n^2 + n}{2} \cdot \tilde{\Delta}_{1,1} f(t_0) \end{aligned} \tag{77}$$