2006:13

RESEARCH REPORT

GNSS Antenna Arrays

Hardware requirements for algorithm implementation

Staffan Backén
Dennis M. Akos

Luleå University of Technology

Research Report
-
Department of Computer Science and Electrical Engineering
Division of EISLAB

2006:13 - ISSN: 1402-1528 - ISRN: LTU-FR--06/13--SE

GNSS Antenna Arrays - Hardware requirements for algorithm implementation

Staffan Backén, Dennis M. Akos
Luleå University of Technology

April 4, 2006

Abstract

GNSS antenna arrays and beam forming/nulling have long been studied from an algorithm perspective. However, the hardware and implementation design issues are not as well understood.

A low cost data collection platform for continuous recording of eight simultaneous IF streams are presented with an analysis of its performance. Also, the requirements on front end components are examined, specifically their impact on array processing versus single antenna processing.

Focus is exclusively on digital beam forming as that is the only feasible approach for advanced algorithm implementation, and differences between traditional receiver architectures and a software defined radio approach will be emphasized.

It is found that simultaneous recording of multiple IF streams allows for great flexibility in the field of antenna array processing for GNSS. Also, commercial off the shelf components shows sufficient performance, although care must be taken regarding issues such as phase consistency and cross talk between front ends.

1 Introduction

With the prospect of addressing several of the drawbacks of single antenna GNSS systems, antenna arrays have long drawn interest from the navigation community. However, the hardware implementation details have not been thoroughly investigated. In this paper the advantages and disadvantages of ASIC (Application Specific Integrated Circuit) based front ends are evaluated, and the performance of a low cost, low complexity data collection system for GNSS antenna arrays is investigated. With the purpose of examining important performance measures such as SNR (Signal to Noise Ratio) and phase estimation, extensions to the traditional GPS tracking loops are developed and evaluated.

1.1 GNSS ASIC front ends

The purpose of a GNSS front end is to filter, amplify, downmix and sample the GNSS signal so that it is possible to process it using digital signal processing. As the GNSS signal power is very low, a large amount of gain is required. Also, with the high carrier frequency (1575.42MHz for GPS L1), the signal needs to be moved down in the frequency domain so that sampling and subsequent signal processing is feasible. This process is called down mixing, and is performed by multiplying the GNSS signal with a sinusoid whose frequency is close to the carrier frequency of the signal of interest. A typical frequency plan for a GPS L1 front end is a common clock of 16.3676MHz driving a frequency synthesizer with a multiplication factor of 96, resulting in a LO (Local Oscillator) frequency of 1571.2896MHz. The resulting IF (Intermediate frequency) will then be:

$$f_{IF} = f_{L1} - 96 \cdot f_s = 4.1304MHz \tag{1}$$

This particular frequency plan is only one example, and typically ASIC front ends supports different frequency plans. After filtering in a sufficiently narrow bandpass filter, the IF signal may be sampled with the same common clock used to generate the LO, while observing the Nyquist criterion.

When the signal have been properly conditioned in the analog domain, it will undergo analog to digital conversion before it finally is ready for processing. Most commercial ASIC front ends uses between one and four bits to represent the signal digitally. Unless only one bit quantization is used, an AGC (Automatic Gain Control) is required to control the amplitude of the signal before quantization. For antenna array processing in the absence of strong interference, a small dynamic range is sufficient, as shown in [DDG$^+$04].

However, many ASIC front ends have an option (often proprietary and undocumented) to output the analog IF signal directly as it is typically required during the verification of the chip after production. This output, together with an external ADC, may be used to provide greater dynamic range if required. Besides the obvious antenna RF input and digital output lines, ASIC based front ends typically have additional inputs providing a method to measure and/or control the AGC voltage, among other.

Figure 1 shows a schematic view of a typical ASIC front end. This particular front end uses the same frequency plan as in equation 1. Further, it has 2 bits of dynamic resolution and a digital

feedback loop to the AGC, where a simple regulator controls the gain to keep the LSB (Least Significant Bit) active during the optimal 32.70% of the time, thus minimizing quantization loss [BAMR03]. Some front ends uses analog feedback instead, but the principle is similar. Another

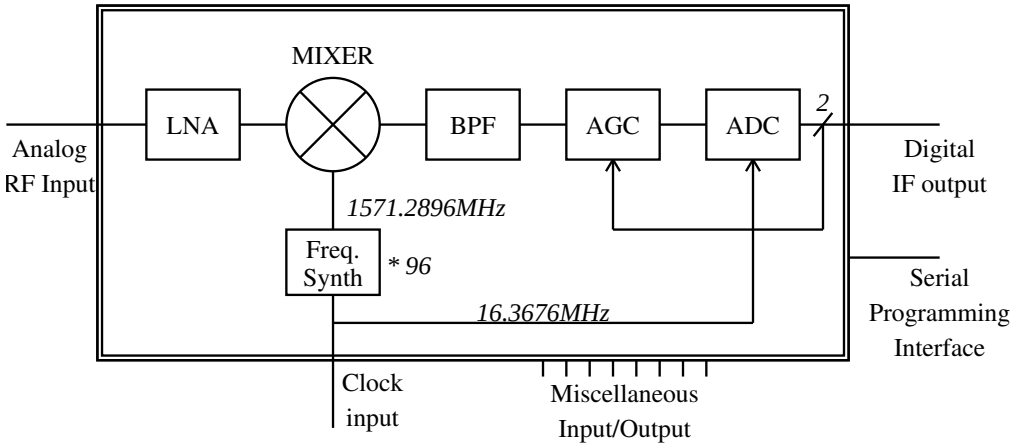


Figure 1: Simplified ASIC front end block diagram

common feature of ASIC front ends is a serial programming interface. Using this, different parts of the front end can be switched on/off, and additional features can be enabled when desired. Also, several additional I/O pins may be accessible.

1.2 GNSS ASICs for array processing

Antenna array processing of GNSS signals provide, compared to its single antenna counterparts, the means to mitigate interference and multipath more effectively as beams can be pointed towards signals of interest. This is achieved by phase shifting the signals from the different channels and finally summing them all together, providing gain towards a signal of interest and possibly nulls towards sources of interference. The algorithms used to control the phase shifts, known as steering vectors, is thoroughly explored in the area of signal processing. For GNSS specific antenna array processing, [Gra00] gives it a thorough treatment. Further, as shown in [DDG$^+$04], beam forming can be done pre correlation (i.e. raw IF data, sampled in the MHz range), as well as post correlation (outputs in the kHz range). However, both methods require several front end/antenna pairs to be connected in parallel and synchronized.

Regarding the processing of the data, two different methods are available. Either specialized array hardware, capable of handling the data in real time, or the software approach, were the simplest approach is to design an interface capable of storing enough data for post processing. As the software approach provides great flexibility and rapid development cycles it have been the method of choice for the authors.

As beam forming/nulling algorithms relies on the phase shift between channels (the arrays are generally very small compared to the wavelength of one chip), the phase behavior of the different

channels must be verified to perform sufficient. Also, the gain of the different channels should be tested although the differences here are expected to be small and depend mostly on the noise figure of the first LNA (Low Noise Amplifier). When referring to gain and phase, it is with regard to the front end components only, not the full system with the antenna unless explicitly indicated.

A discrete component front end for antenna array processing could, besides providing great flexibility, offer the possibility of using the same LO for the mixer stages. With an ASIC on the other hand, we typically do not have any input for this very high frequency signal. Instead, this signal will be generated in a frequency synthesizer based on a common clock, with the unfortunate result of independent phase noise between front ends. This is avoided in a discrete component multiple front end design, where a common LO signal is shared among the mixers responsible for down converting. The penalty of independent phase noise have not been evaluated in this paper.

Another effect of using individual frequency synthesizers is that the phase measurements must be verified to be constant between power cycles, proving that the PLL (Phase Locked Loop) controlling the frequency synthesizer locks to the same part of the duty cycle of the common clock. Also, the amount of crosstalk, defined as the power leakage between channels, needs to be measured.

2 Measurements on GNSS ASICs for antenna arrays

To examine the behavior of ASIC based front ends, a data collection system was designed using eight evaluation boards based on 2-bit ASIC front ends. With a sampling rate of 16.3676MHz and assuming continuous transfer, this means 31MB/s is required, with every sample being 16 bits wide. USB2 have a maximum transfer rate of 480Mb/s (60 MB/s), and with its low cost and relative ease of implementation, the choice of protocol was easy.

Compared to traditional GNSS signal processing hardware, software defined radio provided several advantages for this type of research. Quick development time and the ability to replay identical data with different settings was important, but most important was the possible extensions to the standard tracking algorithms that will provide accurate gain and phase measurements. A schematic view of the system is provided in figure 2.

2.1 Extension of tracking algorithm

The GPS tracking algorithms of a specific satellite signal, be it in hardware or software, is normally comprised of three pairs of local replicas of the signal. They are denoted Early, Prompt and Late, and are typically spaced half a chip apart.

Each replica is in turn divided into one Inphase and one Quadrature component, phase shifted 90° from each other. From now on, the replica vectors are denoted $\boldsymbol{v}_{s,p}^{n}$, where

$n \in \{1..N\}$	N number of streams, n is the channel being tracked
$s \in \{E, P, L\}$	Early, Prompt or Late
$p \in \{I, Q\}$	Inphase or Quadrature

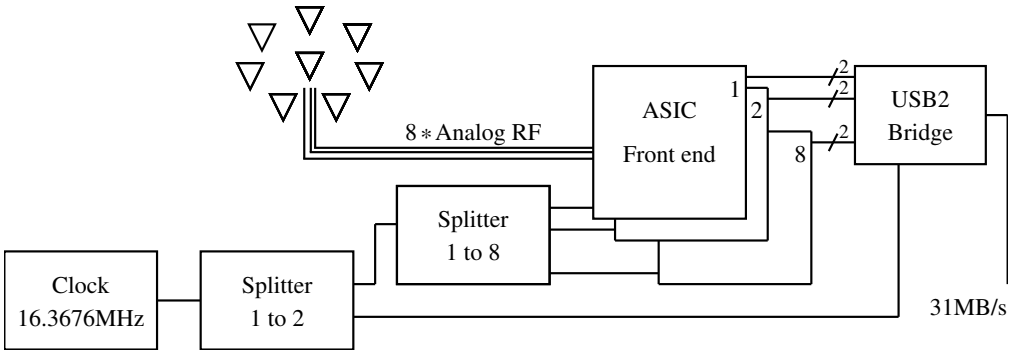


Figure 2: Data collection system

Also, the accumulator value a is defined as

$$a_{s,p}^{n,m} = \mathbf{v}_{s,p}^{n\ T} \cdot \mathbf{d}^m \tag{2}$$

where $\boldsymbol{v}_{s,p}^n$ is computed for stream n, the channel being tracked, and $\mathbf{d}^m$ is the data stream m. The power measure w is defined as

$$w_s^{n,m} = \sqrt{a_{s,I}^{n,m} + a_{s,Q}^{n,m}} \tag{3}$$

During tracking $(m = n)$ of stream n, a DLL (Delay Locked Loop) will steer $w_E^{n,n}$ and $w_L^{n,n}$ to be equal, thus maximizing $w_P^{n,n}$ [ME01]. It is an implementation of the early power minus late power DLL. Figure 3 is an example of the performance of a typical DLL. Here, lock is achieved after approximately 1.5 seconds.

Simultaneously, a PLL will maximize the signal energy of $a_{P,I}^{n,n}$ while $a_{P,Q}^{n,n}$ will consist of noise

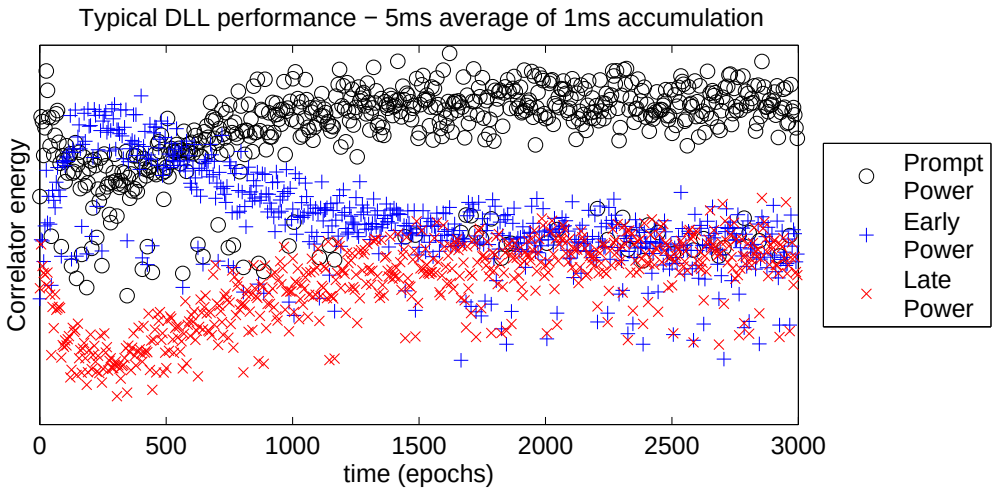


Figure 3: Typical DLL performance

with zero mean. See figure 4 for an example of typical PLL performance. Phase lock is achieved after approximately 100ms.

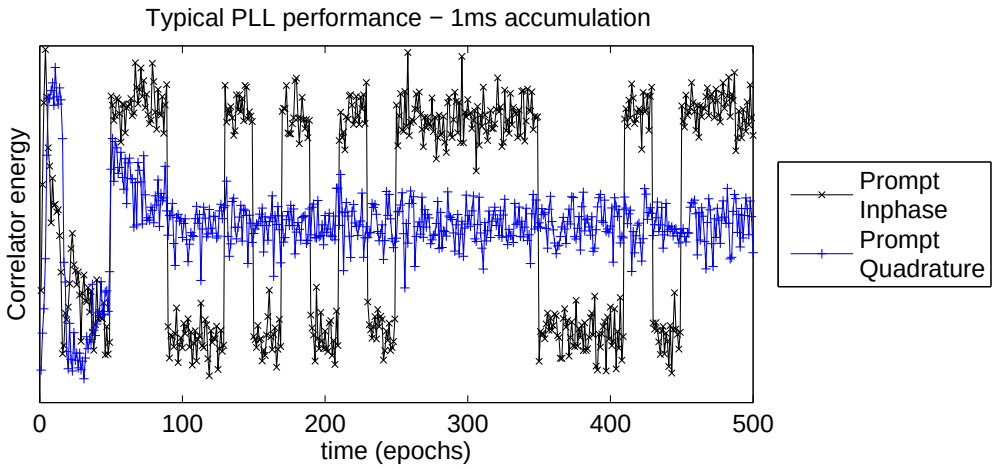


Figure 4: Typical PLL performance

The extension of the tracking algorithm consists of storing $\mathbf{v}_{P,I}^n$ and $\mathbf{v}_{P,Q}^n$ for every epoch during tracking. This allows for the computation of all $w_P^{n,m}$, that is used to estimate the SNR of stream m, and also $a_{P,I}^{n,m}$ and $a_{P,Q}^{n,m}$ that will be used to estimate the phase shift between stream m and stream n. These estimations will be good as long as the size of the array is very small compared to the period of one chip, generally referred to as the narrowband assumption in the array processing literature [KV96]. Assuming a perfect triangular correlation peak, we can express the correlation power offset from the center as

$$w_P^{n,m} = w_P^{n,n} \cdot \left(1 - \frac{|\tau^m|}{T_C}\right) \tag{4}$$

where T_C is the chip length and τ^m is the offset of antenna m from the center antenna, n, in the same units as T_C. With a chip length of 300m and a distance of one wavelength (0.1904m) between antenna n and antenna m, the power measure $w_P^{n,m}$ will, on average, be $0.9994 \cdot w_P^{n,n}$.

This extension is only valid for multiple front ends if they all share the same reference clock, otherwise the frequency drift between front end oscillators will cause the replicas to be useless for the other channels.

2.2 Gain estimation

The SNR of a GPS signal is traditionally expressed as a bandwidth free estimate, C/N_0, with unit dBHz, and several different methods have been proposed to compute it. The vsm (variance summing method) have been chosen here [PAT03]. The vsm algorithm requires a series of $w_P^{n,m}$, but it is no different from computing the SNR using a series of $w_P^{n,n}$ and its output will be denoted $\hat{p}^{n,m}$.

To verify the C/N_0 estimation on the slaved streams, four separate data sets was recorded with a single channel simulator source. The simulator output was connected via a 1-to-4 splitter to four of the front ends, and the simulator power was adjusted to match four different power levels between 40dBHz and 55dBHz. However, the C/N_0 of the signal before the front ends were likely 0.5 to 1.5dB higher, as we do not have infinite sampling frequency nor unlimited dynamic

range. [BAMR03] Each data set was seven seconds long, and the C/N_0 estimation was performed on the last 5 seconds to allow the DLL to lock. During tracking of channel 1, $\mathbf{v}_{P,I}^{1}$ was stored to allow for the computation of $w_P^{1,m}$ and finally $\hat{p}^{1,m}$ where $m \in \{1, 2, 3, 4\}$. The results, as shown in table 1, shows that C/N_0 will be accurately estimated across channels.

C/N_0 (dBHz)		Channel			
		1	2	3	4
SNR	55	55.38	55.80	55.82	55.73
	50	49.90	50.32	50.43	50.23
	45	44.88	45.37	45.22	44.81
	40	39.73	40.26	40.47	39.94

Table 1: C/N_0 estimation across channels

2.3 Phase Estimation

When tracking, the PLL is maximizing $a_{P,I}^{n,n}$ while minimizing $a_{P,Q}^{n,n}$, steering the phase shift between them to be $90°$. This is shown for channel 1 in figure 5.

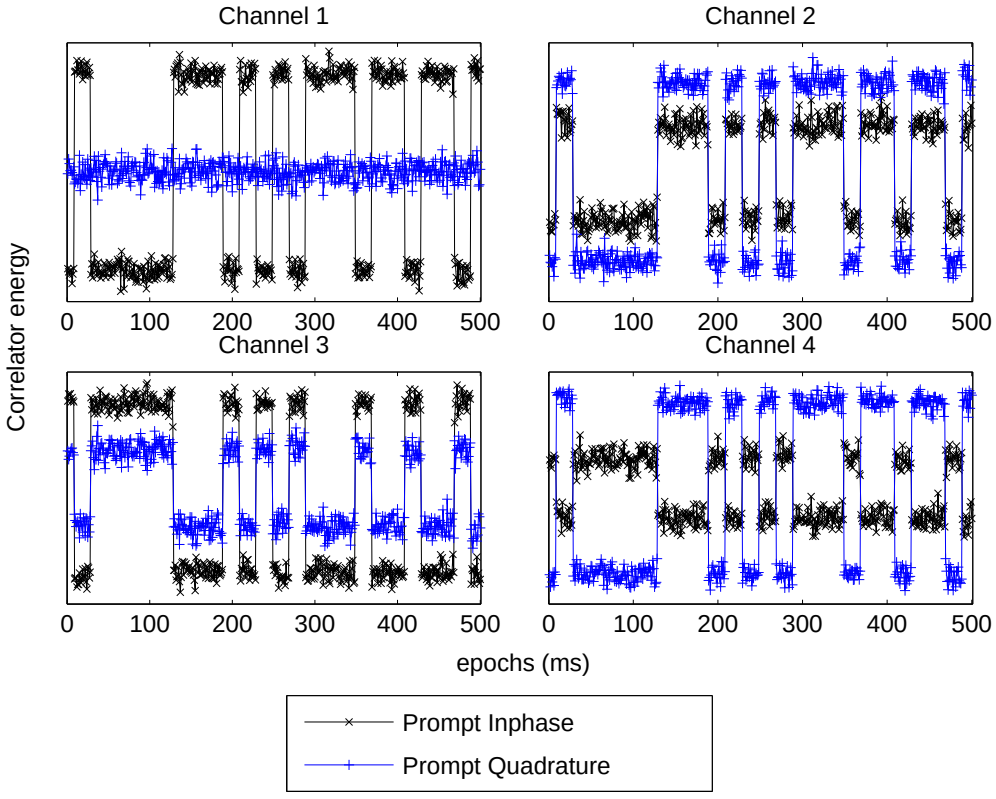


Figure 5: Accumulator inphase and quadrature

For the other channels, $a_{P,I}^{n,m}$ and $a_{P,Q}^{n,m}$ will not necessarily be orthogonal, rather the angle between them will be an estimate of the delay in channel m with respect to channel n, plus the effect of clock line biases.

We can express the estimate of the delay in radians as

$$\phi^{n,m} = \textit{arctan2}\left(a_{P,Q}^{n,m}, a_{P,I}^{n,m}\right) \tag{5}$$

where *arctan2* is the four quadrant arctangent function. In figure 5, channel 2-4 are clearly not orthogonal. Also, for the channel being tracked, the data bits show up clearly in the inphase component. Thus, we can define the data bit $b[e]$ in epoch e as

$$b[e] = \frac{\left|a_{P,I}^{n,n}[e]\right|}{a_{P,I}^{n,n}[e]} \tag{6}$$

Now, assuming constant phase offset during a short data set, we can compute the sum of the prompt inphase and prompt quadrature data bit free accumulators using the results from equation 2 and equation 6 as

$$\hat{a}_{P,I}^{n,m} = \sum_{e} a_{P,I}^{n,m}[e]b[e] \tag{7}$$

$$\hat{a}_{P,Q}^{n,m} = \sum_{e} a_{P,Q}^{n,m}[e]b[e] \tag{8}$$

Combining equation 7 and 8 with equation 5, gives

$$\hat{\phi}^{n,m} = \textit{arctan2}\left(\hat{a}_{P,Q}^{n,m}, \hat{a}_{P,I}^{n,m}\right) \tag{9}$$

where $\hat{\phi}^{n,m}$ is the ambiguous least squares estimate of of the delay between channel n and m.

2.4 Phase Shift Consistency

To measure the phase consistency of the setup, eight data sets, each three seconds long was recorded with different C/N_0 levels. A 1-to-4 splitter was used, distributing the GPS L1 signal from the simulator. The cable length to each front end was different to ensure a phase shift, and only the last second of the data sets was used to allow the DLL and PLL to lock properly. Between every data collection, the clock and USB2 connection (i.e. the front end power supply) was disconnected and reconnected. In table 2, the results of the phase shift consistency test are shown. (Note that the phase shift is in degrees and not in wavelengths.)

2.5 Crosstalk

With eight front ends in close proximity, it is important that the amount of crosstalk, i.e. co-channel influence, is measured and verified to be within acceptable bounds. It is reasonable to assume that the most devastating effect is when the signals are offset 90° from each other. We will model the signal as pure sinusoids, where the reference channel is

$$d^n = \mu^n sin(wt) \tag{10}$$

$\hat{\phi}^{1,m}$		Channel (m)			
		1	2	3	4
SNR (dBHz)	54	$0.06°$	$61.68°$	$-155.84°$	$109.66°$
	52	$-0.00°$	$61.62°$	$-157.51°$	$108.89°$
	50	$-0.03°$	$61.82°$	$-155.94°$	$108.98°$
	48	$-0.09°$	$60.29°$	$-157.44°$	$108.79°$
	46	$0.03°$	$60.48°$	$-156.69°$	$108.28°$
	44	$-0.03°$	$62.02°$	$-155.60°$	$109.76°$
	42	$0.01°$	$62.00°$	$-155.73°$	$109.94°$
	40	$-0.02°$	$61.06°$	$-156.02°$	$109.78°$

Table 2: Verification of phase shift consistency, 1 ms integration

and is interfered by the signal

$$d^m = \mu^m cos(wt) \tag{11}$$

The sum of 10 and 11, where the crosstalk is σ, will then be

$$d^{n+m} = \sqrt{(\mu^n)^2 + (\mu^m\sigma)^2} sin\left(wt + \arctan\left(\frac{\mu^m\sigma}{\mu^n}\right)\right) \tag{12}$$

Thus, for this example the crosstalk from channel m, will cause the signal in channel n to be both out of phase and slightly larger in gain. Solving for a maximum phase error of $1°$ (approximately the phase accuracy for 1Hz updates) and assuming equally strong signals ($\mu^n = \mu^m$), the maximum allowed crosstalk is

$$\delta_{dB} = 20\log_{10}\left(\arctan\left(1°\right)\right) \approx -35.2dB \tag{13}$$

For this low level of crosstalk, the increase in gain will be negligible. However, although it is a very crude model, it nevertheless give some guidance into isolation requirements between channels.

To measure the amount of crosstalk, an experiment was designed using a 1-to-2 splitter and a 0-80dB variable attenuator. The setup is shown in figure 6, and clock lines are omitted for clarity. A source signal from a single channel simulator was fed through an amplifier to the splitter. One of the outputs was routed to front end 1, and the other to front end 2 via the variable attenuator. For simplicity, the simulator was set to an output mode without data bits. This enables long incoherent summation intervals without taking the $180°$ phase shifts due to data bits into consideration. The signal was adjusted to match 50dBHz on channel 1, where tracking was performed with 1ms coherent averaging time. $\mathbf{v}^1_{P,I}$ and $\mathbf{v}^1_{P,Q}$ was stored for every epoch to allow for the computation of $a^{1,2}_{P,I}$ and $a^{1,2}_{P,Q}$ according to (2).

Ignoring the initial 1s of the data set to allow the DLL and PLL to lock, incoherent summation of $a^{1,2}_{P,I}$ and $a^{1,2}_{P,Q}$ and subsequent computation of $\tilde{w}^{1,2}_A$ where n is the attenuation according to

$$\tilde{w}^{1,2}_A = \sqrt{\left(\sum_{e=1}^{500} a^{1,2}_{P,I}[e]\right)^2 + \left(\sum_{e=1}^{500} a^{1,2}_{P,Q}[e]\right)^2} \tag{14}$$

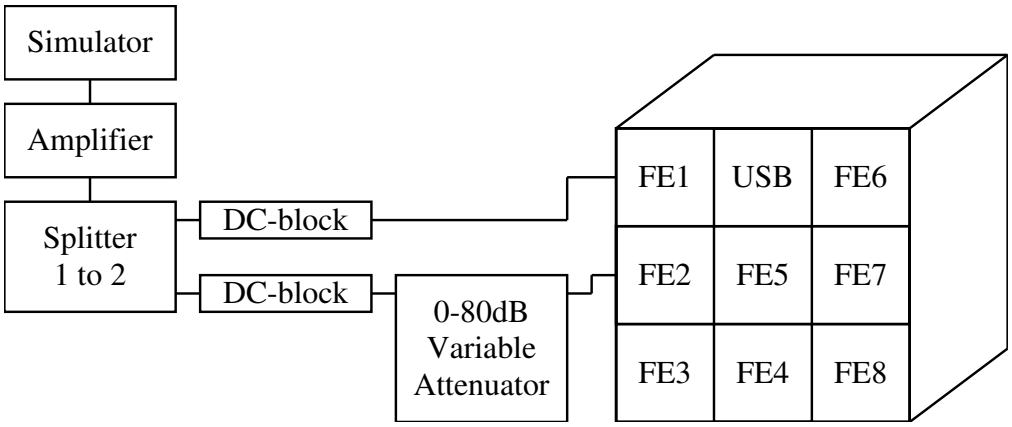


Figure 6: Setup for crosstalk measurements

was performed during each 500ms of the remaining 10 seconds.

A sequence of 43 data sets was recorded, where the attenuator was varied in 2 dB steps from 0dB to 80dB, and finally two data sets with front end 2 terminated and open. Figure 7 shows both the C/N_0 estimation of channel 2, and the correlator power ratio, defined as the ratio $\tilde{w}_A^{1,2}/\tilde{w}_{0dB}^{1,2}$. The correlator power ratio was computed for each of the 20 estimates, and the mean and $\pm$ 1 standard deviation of those samples are plotted.

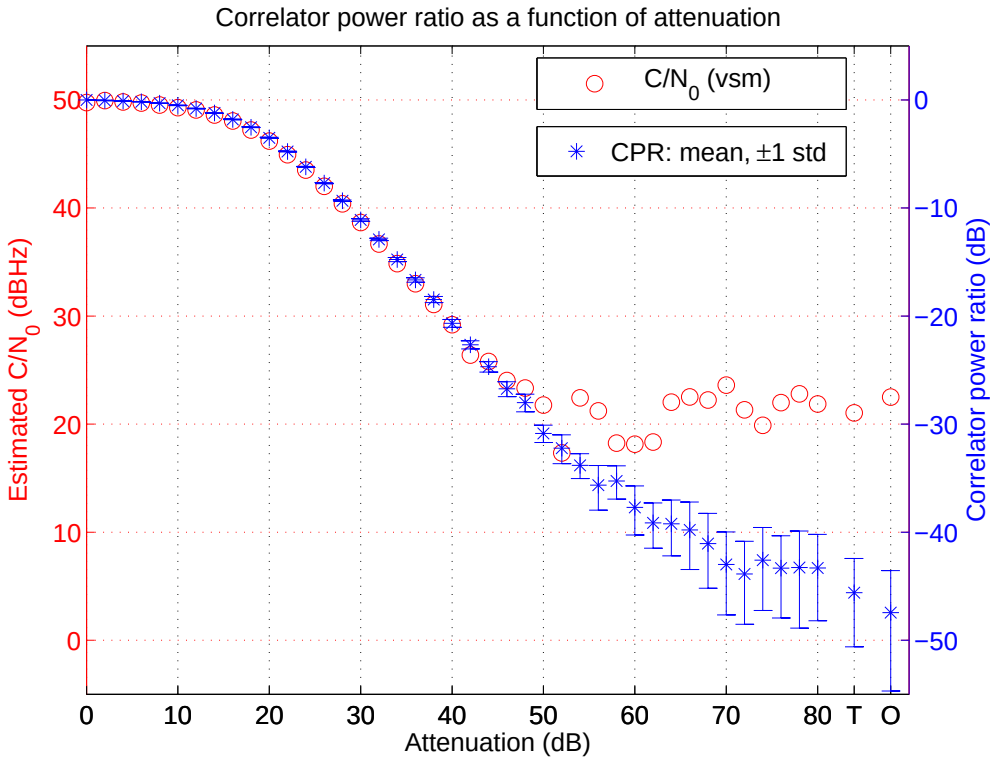


Figure 7: C/N_0 and correlator power ratio

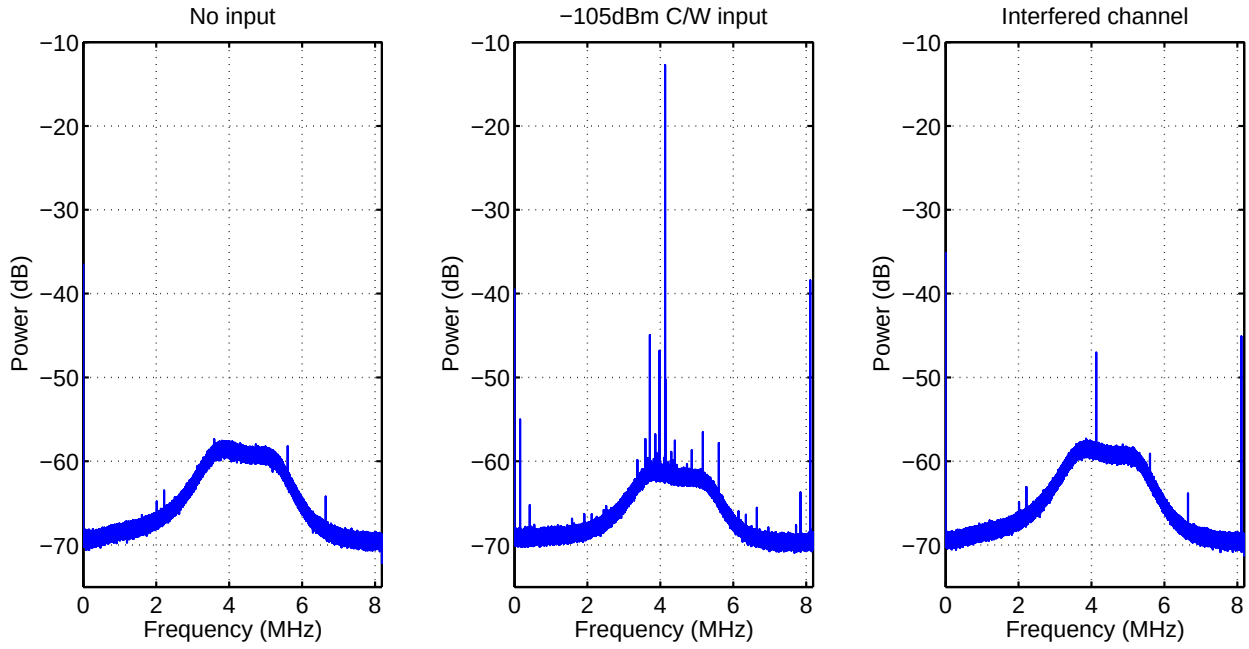
A few things are worth mentioning regarding figure 7. Firstly, the curved nature of the initial estimates are expected, as the attenuator dampens both signal and noise for reasonably high power signals. We are basically changing the noise level of the signal, not the SNR. Further attenuation will eventually give a straight line behavior, for this setup around 25dB. Secondly, the estimate of C/N_0 using the vsm algorithm is only valid to approximately 30dBHz as the coherent integration time is only 1ms. Finally, for the two last data sets front end 2 was terminated with 50Ω (T on the horizontal axis), and left open (O). This last value is the estimate of isolation between front end 1 and front end 2. As shown, the correlator power provides a relatively accurate estimate of low power signals when the incoherent summation time of the inphase and quadrature components are long. Although the standard deviation of the estimates for very low power signals (below -40dB) is rather high, the accuracy is acceptable for estimates where crosstalk is likely to be an issue.

Expanding on the correlator power ratio method, a second data collection was performed where the input signal was shifted between front ends for every data set. The layout is similar to figure 6, although the attenuator is no longer present, and the input signal is connected to front end 1 for data set 1; front end 2 for data set 2 and so forth. The mean of the correlator power ratio was computed for all eight front ends for every data set, giving the results shown in table 3 (the reference correlator power for front end n is from row n, column n). As in the previous test, the average of the correlator power ratio is computed for 20 sets of 500ms second long incoherent summation.

Crosstalk (dB)		Front end							
		1	2	3	4	5	6	7	8
Data set	1	0.0	-45.7	-35.8	-47.7	-49.7	-49.8	-51.5	-50.0
	2	-33.8	0.0	-33.9	-48.7	-49.9	-49.6	-55.3	-49.0
	3	-34.3	-43.3	0.0	-38.4	-50.7	-48.3	-51.2	-51.1
	4	-43.5	-51.4	-36.5	0.0	-50.6	-51.8	-50.9	-48.8
	5	-48.1	-53.4	-48.8	-49.4	0.0	-37.1	-51.1	-38.9
	6	-46.9	-52.1	-48.3	-50.2	-32.9	0.0	-44.5	-43.3
	7	-51.0	-52.0	-49.6	-52.1	-39.4	-37.2	0.0	-34.7
	8	-51.1	-53.1	-53.6	-53.9	-33.0	-41.1	-41.8	0.0

Table 3: Average correlator power ratio between front ends

However, to further validate that the isolation is sufficient, a spectral test was devised. Instead of a simulator source, as used in the previous approach, a C/W (continuous wave) signal was connected to each of the different front ends, and for every setting one 8 seconds long data set was recorded. To estimate very low levels of crosstalk, the signal level have to be set to a maximum of what the front end can support, and also a very long data set is required. For this test, Welch's averaged, modified periodogram method was used, with a data length of 128 MSamples divided into eight sections with a 50% overlap between sections.
The spectrum of a clean channel is shown in figure 8(a), a channel with -105dBm C/W input in figure 8(b) and a relatively interfered channel in figure 8(c). The estimate of crosstalk is the difference between the peak level and the nominal level at the intermediate frequency. With these settings the maximum level of interference that can be detected is $\approx$ -47dB.



(a) No input (b) C/W input (c) Example of crosstalk

Figure 8: Spectral estimation of crosstalk

Regarding figure 8(b), the abundance of peaks is a result of the mixer being a non linear device, thus very strong signals generate multiple harmonics. This level of C/W interference could probably not be mitigated using array processing. Using the spectral C/W method, an estimate of crosstalk was computed for all eight front ends. The result is shown in table 4.

Crosstalk (dB)		Front end							
		1	2	3	4	5	6	7	8
Data set	1	0.0	-47.2	-35.4	-48.3	-47.7	-48.0	-48.0	-48.0
	2	-37.0	0.0	-38.4	-48.2	-47.5	-48.1	-48.1	-47.9
	3	-33.9	-47.3	0.0	-40.0	-47.6	-47.1	-48.1	-47.9
	4	-43.8	-47.4	-40.1	0.0	-47.6	-47.3	-48.3	-48.0
	5	-47.3	-47.5	-47.5	-48.3	0.0	-39.8	-48.1	-40.7
	6	-46.4	-47.9	-47.2	-48.2	-33.9	0.0	-45.1	-43.1
	7	-46.5	-47.8	-47.8	-48.2	-40.5	-39.2	0.0	-34.0
	8	-47.0	-47.8	-47.9	-48.1	-34.7	-46.8	-41.6	0.0

Table 4: Spectral peak to noise difference between front ends

The estimates agrees well with the results obtained using the correlator power ratio method, within a few dBs for the higher levels.

Apparently, we almost meet our previously defined limit of 35dB. If needed, the shielding on

the front ends might need to be improved to provide higher isolation.

2.6 Example of live data

Using eight low cost patch antennas and an aluminium ground plane, a simple antenna array was constructed. The elements was placed in a 3+2+3 configuration, and each of them was powered from its respective front end. The array was placed on the roof of the university building, and several short data sets was recorded with the setup depicted in figure 2. A skyplot of the visible satellites during one of the data collections is shown in figure 9.

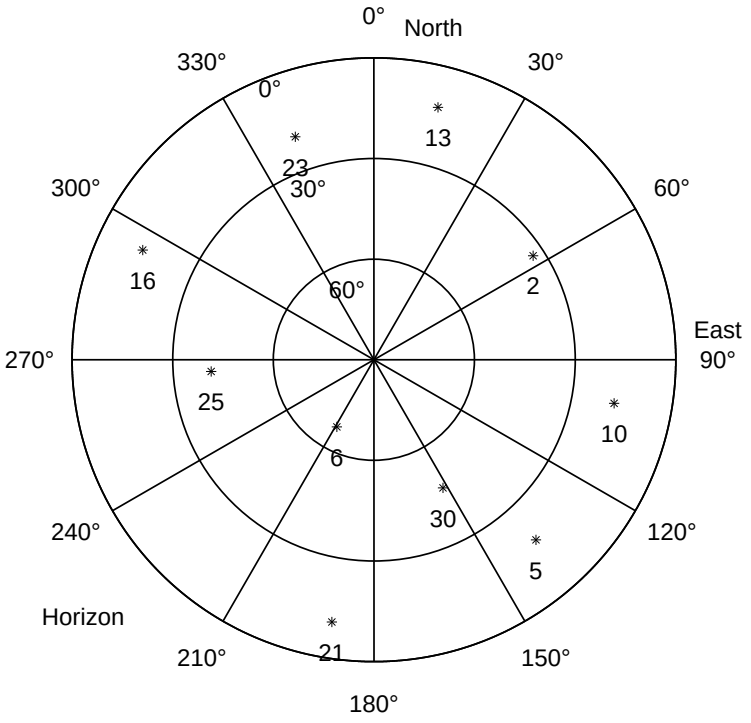


Figure 9: Skyplot of satellite positions

After acquisition and tracking, the estimate of C/N_0 on all channels ($\hat{p}^{1,m}$ where $m \in [1..8]$) was computed for each satellite found during tracking. The results are shown in figure 10.

The high variation of C/N_0 between elements is due to mutual coupling between antenna elements. When several antennas are placed close to each other, the gain and phase pattern as a function of elevation and azimuth will change. This is a well understood phenomenon in general [Bal82], and also in the navigation community [LET+02] [Kim05] [DDG+04].

With this large difference of C/N_0 between elements, (for example 8dB for prn 10), the performance of beam forming with this antenna would be severely degraded compared to an antenna with a significantly more even C/N_0 distribution.

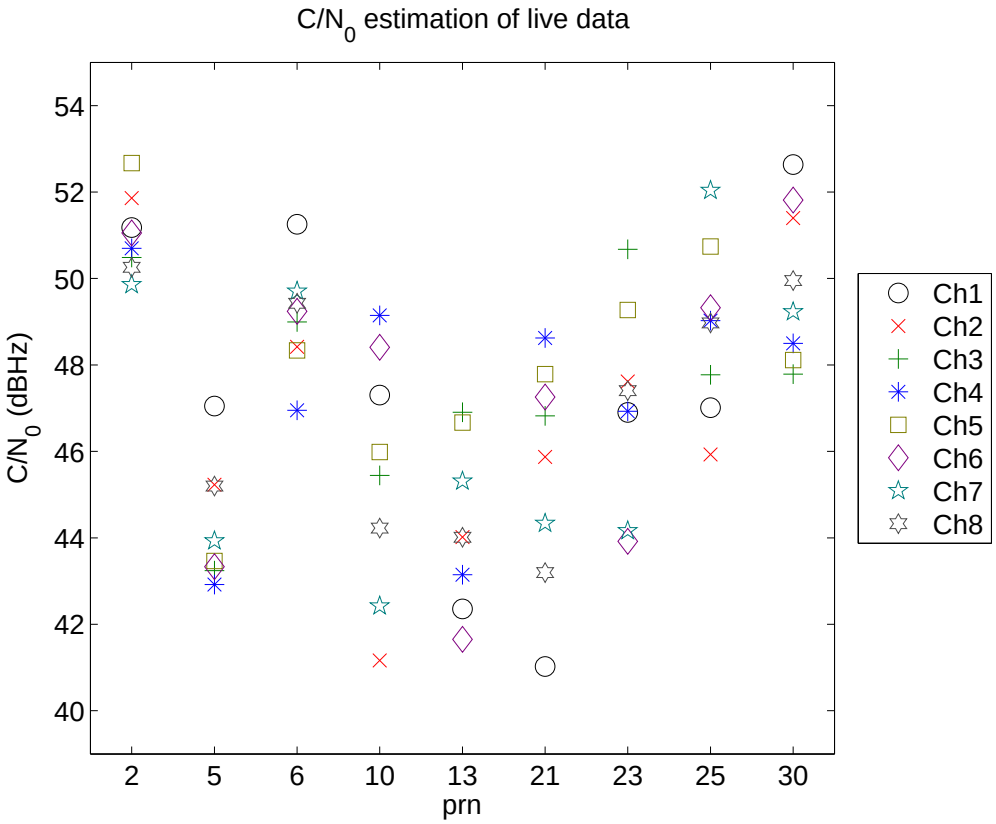


Figure 10: C/N_0 estimation of live data

3 Summary and conclusion

A low cost, ASIC front end based, eight channel data collection system have been evaluated regarding several key performance measures for antenna array processing. Also, simple extensions to tracking loops have been proposed for the purpose of verifying that the performance is acceptable. Gain variation, phase estimation accuracy and consistency as well as crosstalk have been investigated. Also, an example of live data demonstrated the need of a carefully designed antenna to minimize the effects of mutual coupling.

The tests presented in this paper are all interpretations of and/or extensions to standard GPS tracking loops, with the exception of the spectral method used to estimate crosstalk. Within the limits outlined (for example phase accuracy), post correlation beam forming is likely to produce predictable results. However, in the case of pre correlation beam forming, the absence of a common LO (as mentioned in section 1.2) might degrade the performance slightly. This is a subject that eventually requires evaluation. The difference between pre and post correlation beam forming is explained in [DDG+04].

Also, with the low dynamic range, the performance of interference mitigation algorithms would be severely degraded. The option of an analog IF output together with an external A/D converter would resolve this issue. The requirement of good isolation between front ends can not be over-

stated. It could be a major source of error in the phase, and eventually, gain estimates of the underlying signal of interest.

Finally, antenna array processing of GNSS signals is a powerful concept, and the low cost research platform approach investigated in this paper will hopefully be of aid in advancing the current state of science.

References

[Bal82] A.C. Balanis. *Antenna Theory, analysis and design*. John Wiley and sons, 1982.

[BAMR03] F Bastide, D.M. Akos, C Macabiau, and B Roturier. Automatic gain control (AGC) as an interference assessment tool. *Proc. ION GNSS 2003*, September 2003.

[DDG+04] D.S. DeLorenzo, S. David, J. Gautier, P. Enge, and D.M. Akos. GPS receiver architecture effects on controlled reception pattern antennas for JPALS. *Proc. ION GNSS 2004*, pages 2010–2020, 2004.

[Gra00] G.A. Granados. *Antenna arrays for multipath and interference mitigation in GNSS receivers*. PhD thesis, Universitat Politcnica de Catalunya, 2000.

[Kim05] Ung Suok Kim. Analysis of carrier phase and group delay biases introduced by crpa hardware. *ION2005, update this entry!*, pages 2010–2020, 2005.

[KV96] Hamid Krim and Mats Viberg. Two decades of array signal processing research. *IEEE Signal Processing Magazine*, 1996.

[LET+02] H. Ly, P. Eyring, E Traum, H. Tseng, K Stolk, R. Kurtz, A. Brown, D. Nathans, and E. Wong. Design, simulation, and testing of a miniaturized gps dual-frequency (l1/l2) antenna array. *Proc. ION GPS 2002*, 2002.

[ME01] P. Misra and P. Enge. *Global Positioning System - Signals, Measurements and Performance*. John Wiley and sons, 2001.

[PAT03] M.L. Psiaki, D.M. Akos, and J. Thor. A comparison of ”direct RF sampling” and ”down-convert & sampling” GNSS receiver architectures. *Proc. ION GNSS*, 2003.