J Markkanen

EISCAT Digital Receiver Signal Processing

EISCAT Sodankylä 2014

Jussi I Markkanen[1]

EISCAT Digital Receiver Signal Processing

EISCAT Sodankylä

October 2014

This document was produced with LaTeX
in TeXLive-2014 distribution,
using TeXShop for the editing.

[1] http://www.sgo.fi/~jussi

Contents

List of Figures

1 Introduction

This article describes the operation of the EISCAT digital receiver in a self-contained manner. Apart from leaving out finite word length effects, the description is also fairly complete. That was not the original aim when I started the writing. My original purpose was just to explain the origin of a certain odd disturbance in the EISCAT UHF experiment "quip" that I carried out together with my Finnish colleagues in May 2014. That there is something to explain is not in doubt: if one expects data like in Fig. 1, but finds data like in Fig. 2–3, an explanation is called for.

In Chapter 2, after this Introduction, I give an account, from a quite personal point of view, of the puzzle I encountered when I started inspecting the quip data; and of the resulting period of confusion; and about how my EISCAT colleagues and I finally came to an understanding about what was happening. I have told the story with the student[2] in mind. Those believing that technical writing and stories mix badly, might want to skip to Chapter 3.

It turned out that in order to quantitatively explain what was happing in the digital receiver, I had to review the basics in the theory of digital signal processing, such as sampling, filtering, frequency translation and decimation, both in the time domain and in the frequency domain. I'm not entirely happy with the way the basics of digital signal processing are sometimes formulated, with dimensionless frequencies (unphysical), z-transformations (unnecessary), filter design (even more so), many pages devoted to FFT (unwarranted), and so on. Here I have aimed for a concise, elementary presentation that is readily and quantitatively applicable to the EISCAT digital receiver, but does not contain much extra.

I present what I need from the signal processing theory in Chapters 3 (energy signals) and 6 (noise-like signals), starting from the sampling theorem of energy signals, and finishing with the decimation theorem of noise and the concept of noise equivalent bandwidth. In these two Chapters I either prove[3], or at least justify, the results that relate to the digital signal processing. The few results about analog signal processing that are needed to discuss the effect of the anti-aliasing analog filter are given without proof.

My notation, especially in frequency domain, is slightly non-standard. Most often in theory presentation, people use the dimensionless angular frequency $\omega = 2\pi f/f_s$ as their variable of choice. This does not matter much in practice as long as one has only a single relevant sampling frequency f_s to account for, but can become confusing in multi-rate context, and when one needs to handle systems containing both digital and analog parts. I prefer to work explicitly with the physical frequency f throughout—unphysical as that might strictly speaking be in the digital domain—and have written all the formulas accordingly. In the radar context, digital signal processing is, as Clausewitz would have quipped, "merely the continuation of analog signal processing by other means". In that spirit, I do not do z-transforms either, but work exclusively with Fourier transforms.

For fulfilling my original aim, explaining the disturbance in the quip experiment, it would have been sufficient to handle only narrow-band deterministic signals; basically, sinusoids; the stuff of Chapter 3. But during writing, I revisited a 1997 paper by

[2] Such as a student in the EISCAT radar school.

[3] Of course, only in the normal carefree way of elementary physics, without paying attention to anything related to existence, elegance, convergence, and so on. However, I try to make it clear which of the equations are definitions and which are results.

Wannberg et al. in Radio Science on the EISCAT Svalbard radar, a paper where I'm one of the many co-authors. I realised that I was unable to make full sense of the very piece of the paper—a drawing of the ESR receiver—which contained my sole contribution.

I now think it is clear that the part in the drawing concerning the digital receiver is not entirely correct. There are two problems. One is related to a line in the drawing said to represent "filter shape"; the other is related to the functional block diagram of the receiver. Instead of just stating what and why is problematic, I use the issue as a pretext to give a rather lengthly discussion about the consequences of having decimators in the system. With the ESR system, multi-rate digital signal processing was introduced into EISCAT for the first time. When re-reading the paper, I got the feeling that at that time we perhaps were not fully appreciating the conceptual implications.

I present in Chapter 4 a detailed analysis of the function-level operation of the HSP43220 decimating digital filter that forms the core of the digital receiver hardware. This is partly in order to provide a proper functional block diagram of the digital receiver, to replace the one in Wannberg at al., and partly to provide a quantitative basis for solving the quip disturbance problem. Part of the HSP43220 operation is based on the so-called CIC, or Hogenauer, filter structure. My exposition of that interesting structure is more detailed than that in, say, the Wikipedia.

One reason of handling also noise-signals in this article is to acquire a quantitative understanding on how the "background spectra", casually shown in the EISCAT real-time displays during incoherent scatter experiments, relate to the actual shape of the digital filter (once that shape has been properly defined). For decimating filters in general, and the filters in current operational use in EISCAT in particular, the noise spectrum can differ significantly from the shape of the filter power response, for often there is as a significant amount of noise aliasing. The predicted noise spectra of several filters currently in operational use in EISCAT are shown. One day, I may test those predictions against actual data.

The decimating digital filters that are used in EISCAT experiments have all been designed by a Matlab script called ddfplan.m, which I wrote a long time ago to quickly get something usable for the first ESR experiments. I have never documented precisely how the filter that is produced by the script relates to the filter hardware. The discussion in Chapter 4 underlines that the relation is not self-evident, because the decimating filter hardware contains six finite impulse response filters, five infinite impulse response filters, and an untold (yep) number of decimators. How those can be replaced by a single equivalent filter plus a single equivalent decimator, as the design script does, is explained in this article. The equivalent filter is used in GUISDAP incoherent scatter analysis suite via my script get_impresp.m. I have something to complain about that practice, too.

The reason for the "disturbance" in the quip experiment is explained qualitatively in Chapter 2, and quantitatively in Chapter 5. The reason is not any kind of malfunction, or external disturbance. Instead, the observed behaviour is an orderly manifestation of the properties of the digital receiver, when the system is pushed near its bandwidth limits. The disturbance became visible in quip because that experiment was the first one where anyone tried to increase the receiver's final sampling rate beyond the 2.5 Msamples per second that had been successfully used so far. After all the hassle that resulted from the effort to produce a really wide-band filter, it would be nice to finish the article in the Summary with a definitive statement about the maximum usable sampling rate. Unfortunately, the answer must be that "it depends". A rate of 5 Msamples/s can be

achieved. The usefulness probably depends on the intended use of the data, and the cleverness of the data analysis.

The article contains 32 illustrations. I have placed them all together, after the main text. This is only partly due to my editorial laziness. All the figures are referred to from the main text. But I have also tried to make them rather independent of the main text, and it should be possible to make sense of them without referring to the main text. On several occasion, the figures and their captions provide details that are not covered in the main text. It should be possible to get a reasonably good idea about the contents of this article just by leafing through the Figures.

2 An oscillating disturbance in the experiment quip

In May 7–9, 2014, the Finnish-Norwegian special experiment quip was used to test 4-phase phase coding at EISCAT UHF radar for the first time. In the test, several different code sets are transmitted, including both 4-phase codes and binary phase codes. Apart from the different phase codes and different sampling arrangements needed to record voltage-level data, the experiment was based on Ingemar Häggström's beata experiment. Both ion-line and plasma-line data were taken, using multiple receiver channels. In particular, channel 1 was for ion-line data, recording samples at 1 MHz rate. The filter b500d15 was used to configure channel 1 for the required factor of 15 sampling rate reduction, or decimation, from the 15 MHz A/D primary sampling rate to the final 1 MHz rate. The filter was expected to function without trouble, for I had used it during several previous experiments.

It is nowadays routine in EISCAT to record "transmission samples". The transmitted waveform is probed just before it leaves the antenna and the captured voltage is fed to the analog receiver, so that it traverses essentially the same signal path as the actual radar echo signal. That signal is then handled by the digital receiver as any other signal. The resulting voltage-level data are referred to as the transmission samples. Transmission samples of a 4-phase code on channel 1 were as shown in Fig. 1. These data indeed were as expected.

When inspecting the beata source files during preparation of the quip experiment, I had learned that IH uses three receiver channels to cover the plasma line. These channels use the decimation 6 filter b800d6 to achieve a 2.5 MHz sampling rate. In UHF beata, the channels 3, 4 and 5 are tuned 2.4 MHz apart from each other. I did not quite understand this configuration. The nominal 3 dB width of the b800d6 filter, ±800 kHz= 1600 kHz, appeared in the low side for covering the 2.4 MHz distance from channel to channel.[4] But more puzzlingly, why would this filter be functioning at all?

Since the introduction of the digital receiver channels to EISCAT, via the "channel boards", at ESR in the 1990s, it has been common knowledge that the fastest sampling rate we can achieve is 1.6 MHz (decimation 9), because otherwise the decimating filter's internal speed specifications are violated. I thought that my filter-design program ddf-plan.m (Listing 1) actually checks that the specifications given in the Intersil HSP43220 decimating digital filter's data sheet[5] are respected. How could IH squeeze a 2.5 MHz

[4] When preparing the experiment, I did not quite realise how slowly the EISCAT digital filters actually die off in the stop band. The b800d6 filter is exceptionally narrow for its sampling rate, and it still has 20% of its maximum gain left at a frequency 1.2 MHz away from the centre of the filter, in the middle point between two plasma line channels.

[5] https://www.intersil.com/content/dam/Intersil/documents/hsp4/hsp43220.pdf

sampling speed out of those devices?

When I executed the latest version of ddfplan, with the parameters required to create the b800d6 file (the parameters are documented in the filter file), the program immediately complained about too small decimation value, as it should. Inspecting into the mess, I found that at the time when the b800d6 file that IH was using had been generated, the official ddfplan program in EISCAT CVS had had a bug and had not been able to catch the speed violation. But the beata experiment had been performed many times, and the "illegal" decimation=6 filter had therefore been in standard use, without any harmful effects being noticed. Assar Westman concluded that the speed specifications given in the filter's data sheet probably simply were too conservative.

It appeared that we were allowed to breach the filter speed limit without consequences, and encouraged, I decided to push the luck further still. After changing the speed traps in ddfplan.m from errors to warnings, a little experimentation in Tromsø before the quip campaign showed that decimation down to three appeared to work; at least one got a sensible looking output stream when using a sinusoidal analog input signal. Decimation 3 corresponds to 5 MHz sampling rate, so this looked like a significant improvement over the 2.5 MHz achieved in beata. I also wanted to make the filter's 3 dB width to be near the "ideal" value half of the sampling rate, that is, about 2500 kHz. My solution for plasma-line sampling for the quip campaign was the filter b2300d3. That would cover an almost 5 MHz wide frequency band, so that a single plasma line channel would be enough. In quip, digital receiver channel 4 is used for the plasma-line sampling. That channel takes its input data from a plasma-line dedicated analogue receiver path, which has a built-in frequency offset compared to transmission, and therefore, does not see the transmission sample data. To record the transmission samples with the 0.2 μs resolution, one of the available standard (non-offseted) channels, channel 3, was reserved for that purpose.

During the May 7–9 campaign runs, everything appeared to go smoothly by the quick-look monitoring which just displayed the raw data, though I wondered a little why the transmission samples on channel 3 (Fig. 2) looked "noisy" compared to channel 1 data (Fig. 1). But the main concern in the campaign was to verify that the new 4-phase phase shifter hardware was working, and by the look of the transmission samples on the real-time display, that seemed to be the case. So I only realised that there were serious problems when I started to look at the data more closely immediately after the run.

Channels 1 and 3 (as well as the channel 2 which is used for gain calibration in beata and quip) process data from a same A/D converter. Fig. 2 shows how the same transmitted pulse that is presented in Fig. 1 with 1 MHz sampling on channel 1, appears on channel 3 with 5 MHz sampling. When zooming-in to the high sampling rate data as in the top two panels of Fig. 3, it became evident that the power and phase variation was mostly not noise but actually a rather regular periodic disturbance.

The spectrum computed from the zoomed-in segment, the bottom panel of Fig. 3, shows that in addition of the expected DC component, the transmission sample data had a component also at about 200 kHz. This component was down by about 28 dB from the DC component.

Unfortunately I did not produce spectrum plots like that in the bottom panel of Fig. 3 in the initial analysis. Such plots clearly show the presence of two distinct frequencies in the data. Rather, for several days after the experiment, I was only working with the time-domain data; and I became quite confused. I noticed that the data in the top two panels of Fig. 3 had very nearly the frequency of 200 kHz, but I had no clue about what

that would mean. Some other characteristics of the oscillating disturbance are given in the caption of Fig. 3.

Remember, there were new hardware in the signal path in the experiment, the new phase shifter. Perhaps the new hardware somehow was responsible for the disturbance. But the disturbance appeared to have so low frequency that if it was some external signal, present already in the A/D's analogue input, it should have been able to pass through the channel 1 filter, too, and should have been visible also in the channel 1 data. Those data, after all, were computed literary from the same primary 15 MHz A/D samples as the channel 3 data. On the other hand, I could not understand how the sinusoid-looking disturbance could be an external signal, for a sinusoidal analog signal, after quadrature detection, should result in the complex signal $A\exp(i\omega t)$ which has constant power (such as the clearly DC-level signal on channel 1). Very confusing. I totally failed to appreciate that I was dealing with two different frequencies in the data. Mike Rietveld, to whom I described the oscillating disturbance during the weekend May 10–11 in Tromsø, immediately remarked that to him this felt like some kind of beat phenomenon. That remark in the latest should have kicked me into the correct track of reasoning, but didn't.

During the weekend, other oddities in the data became apparent. In any given recorded data dump, there were multiple radar cycles with numerically identical data. I mentioned the problem to AW during a phone call in the weekend. The next week, he was able to identify, and then solve, the problem as being due to my attempt to transfer too much data. In the worst-case version of the quip experiment, three channels were sampling at the 5 MHz sampling rate (almost) continuously, and in addition, one channel sampled continuously with 1 MHz. AW found out that the program that is responsible for transferring data from the receiver output buffers to the process computer, could no more handle the resulting data rate. The program occasionally did not actually fetch new data from the channel output buffer. Instead, it delivered old data from its internal buffers, but failed to give any explicit warning. This problem turned out to be unrelated to the oscillating disturbance, and I mention it here only because it was an additional source of confusion during an already bad time.

When discussing these problems with the Tromsø site staff on Monday 12th May, it was agreed that more, and more varied, data was needed. We decided to vary the transmission frequencies; to switch on and off the control signals of the new phase shifter; to change channel filters; and to switch the channel hardware (the channel boards) between the channels. We also decided to test the digital receiver just by feeding a pulse train from a signal generator directly to the A/D input. At that time, the reason for the repeating data was not yet understood, but by luck, I had anyway reduced the number of the wide-band channels from three to one. This data rate can be safely transferred, so quip version v2.1, which was used in the tests on 13th May, was free from the data-flow problem.

I did the additional tests on the 13th May together with Stian Grande, with the rest of Tromsø site staff peeking into the experiment control room occasionally for discussion. Our main observations during the tests were the following.

- Removing the phase-flip control signals going from the radar controller to the 4-phase phase shifter removed the phase coding as expected, but did not change the oscillations in any obvious way.
- Switching hardware between channels did not have any effect.

- Of the filters tried, only the filter b2300d3 seemed to have the oscillation problem.
- When using the signal generator to emulate non-phase-coded transmission, the oscillation was still there more or less similarly as what we got when removing the phase shifter control signals.
- When the transmission frequency, and the corresponding channel tuning frequency (the setting of the channel's numerically controlled oscillator, NCO), was F13 (corresponding to NCO frequency 10.1 MHz), the oscillation had a frequency of 200 kHz. With F12 (10.4 MHz), the oscillation frequency increased to 800 kHz, and also the amplitude of the disturbance, both in terms of power, and in terms of phase variation, increased, as is shown in Fig. 4. With F11 (10.7 MHz), the oscillation frequency again chanced, now to 1.4 MHz, and again the amplitude of the disturbance increased.

The first four items convinced us that the disturbance must be related to the new wide-band filter, but what could be wrong with it? The last point in the observation list provided the crucial clue. It shows that the disturbance is closely tied to the transmission rather than being some independent external signal, or a signal generated by some malfunction within the filter itself. But what could the connection between transmission and the disturbance be?

As I have already mentioned, we did not have spectral plots like the bottom panel of Fig. 3 or Fig. 4 available at that time. We had only the time series data, and we tried to estimate the disturbance periods from the real-time display raw data plots by the eye. I did the initial period-to-frequency conversion with Matlab, and managed to get that calculation wrong, adding to the confusion. Fortunately SG did not believe my numbers, but used a calculator in his smartphone to do the conversion. This gave the frequencies listed in the last bulleted item above. So we had an increase from 0.2 MHz to 0.8 MHz to 1.4 MHz when the transmission frequency was changed from F13 to F12 to F11. Then Arild Steinberg also came to chat about the results. He noticed that the change of the transmission frequencies F_n was in steps of 0.3 MHz, and observed that the change of the disturbance frequency therefore was two times the change of transmission frequency.

Only then did the number two finally register in my mind as something important. Up to that point, I had tacitly assumed that what ever the source of the problem was, it was a single source only. But now I became convinced that a single source simply could not possess this many properties. There must be more than one source of the disturbance in order to make this kind of frequency multiplication by two possible. In a small *aha* moment, I realised that we do, indeed, have two signals in our data after all, namely, the two frequency components of the real-valued analogue input signal. A quick sketch of the frequency mapping and aliasing taking place in the digital receiver showed how the receiver could produce the measured disturbance frequencies: the unwanted spectral component of the analog input, which one normally assumes is filtered out, would end-up in producing precisely the observed numbers.

For instance, the transmission frequency F12, which shows up at 10.4 MHz in the A/D input, has spectral components at 10.4 MHz and −10.4 MHz. In the channel board, as I very well knew, the negative frequency component is shifted to zero frequency, and the positive frequency component, which thereby is shifted to 20.8 MHz, is filtered out. But if that filtering is less than perfect, with 15 MHz sampling, the 20.8 MHz has a live alias at 20.8 MHz-15 MHz=5.8 MHz, and when this is then decimated to the final 5 MHz

sampling frequency, the unwanted spectral component will appear in the final 5 MHz baseband as a 0.8 MHz complex sinusoid[6]. The desired frequency component of the analogue signal goes to zero-frequency, and when these two frequencies, DC and the unwanted component at 0.8 MHz, interfere, a 0.8 MHz beat frequency can result.

It was immediately clear to me that the failure to suppress the unwanted frequency component in detection must be the reason for the observed disturbance. But it still remained to be checked whether this failure was due to some malfunction of the filter, or whether it simply was a consequence of how the filter had been designed.

In Chapter 5 of this document, I will show that the filter behaves correctly, that is, as designed. The b2300d3 filter has such a design that it—as an unintended but unavoidable consequence—does not suppress the unwanted frequency very well. I will also demonstrate that the decimation 3 is almost the only case one needs to be worried about; with the possible exception of the b800d6 filter used for plasma-line reception in beta (see Fig. 27), all other filters currently in use suppress the unwanted frequency well enough to allow it be entirely ignored. In the next two Chapters, we build the machinery needed to model the disturbance.

3 Operation of the EISCAT digital receiver

The Tromsø UHF radar's receiver has two analogue receiver chains, one for the ion line and the other for plasma line. Both analog chains start from the antenna and end in an A/D converter which samples continuously at 15 Msamples per second. Both A/D converters feed three identical digital channels each, shown schematically in Fig. 6. The real-valued A/D input signal $x^a(t)$ is band-limited by a bandpass filter centred at 11.2 MHz and having 3 dB width of slightly over 7 MHz.

In sampling, according to the sampling theorem, the spectrum of the stuff in the analog passband becomes periodic by the primary sampling frequency f_s (corresponding to the sampling interval $\tau_s = 1/f_s$). In this chapter, we assume the "stuff to be deterministic signals $x(t)$ for which the Fourier transform exists; these are called "energy signals". In Chapter 6, we handle noise-like, or random, signals.

The sampling theorem relates the discrete-time spectrum and the continuous-time spectrum via

$$X^s(f) = \sum_{m=-\infty}^{\infty} X^a(f - m f_s)\,. \tag{1}$$

The $X^a(f)$ is the spectrum (Fourier transform) of the continuous-time signal $x^a(t)$,

$$X^a(f) \triangleq \int_{-\infty}^{\infty} x^a(t) \mathrm{e}^{-i2\pi f t} dt\,. \tag{2}$$

The $X^s(f)$ is the spectrum of the sampled signal, which is defined as the discrete time Fourier transform of the sample sequence $x_n^s \triangleq x^a(n\tau_s)$,

$$X^s(f) \triangleq \frac{1}{f_s} \sum_{n=-\infty}^{\infty} x_n^s \mathrm{e}^{-i2\pi(f/f_s)n}\,. \tag{3}$$

[6] By complex sinusoid in this article I mean the complex exponential function $A\mathrm{e}^{i\omega t}$, not the function sin with a complex argument.

The right-hand-side of Eq. (1) is called the periodic summation of the analog spectrum $X^a(f)$.

Note the factor $1/f_s = \tau_s$ in the definition of the discrete spectrum, Eq. (3). Many authors do not use it. But I want to have the discrete-time spectrum to be as similar to the continuous-time spectrum as possible, starting from them having the same physical unit (V/Hz). As typically in signal processing, I take the samples to have the unit of a Volt, and will also assume unit impedance, to that the power has unit of V^2.

Also note that it can occasionally be slightly confusing not to have the sampling frequency explicit in the notation of the discrete spectrum, e.g. $X(f_s'; f)$ or $X(f/f_s)$. But the former notation would be too cumbersome for the simple needs of this article, and the latter I do not like because I want to consider the discrete-time spectrum to be function of the physical frequency f, not a dimensionless frequency. A problem that can occur is that if two spectra refer to different sampling frequencies, one can have $X(f) = Y(f)$ for all f, even though the underlying sequences are different.

Equation (3) is sometimes explained by stating that the sample sequence x_n is obtained by multiplying $x(t)$ by the Dirac comb $\mathrm{Ш}_T(t)$ (defined in Eq. (9), below). If one then computes the continuous time spectrum Eq. (2) of the product $x(t)\mathrm{Ш}_T(t)$, one gets the r.h.s. of Eq. (3). (That product will show up in the proof of the sampling theorem, below.) The problem with this argument is that the sample sequence x_n clearly is not actually obtained in that way; the samples are most naturally modelled as numbers, not as delta-functions. Nevertheless, the argument provides an a priori reason to expect that the properties of the discrete time spectra are to some degree similar to the properties of the continuous time spectra.

In my view, Eq. (3) is logically strictly a definition, not a result. That it is a good definition, can ultimately be justified only a posteriori, by showing that so defined, the discrete time spectrum keeps as many of the properties of the physical spectrum as one can hope for. Most especially, one expects some ambiguity by the sampling frequency, because two sinusoids that differ my a multiple of the sampling frequency will produce precisely the same set of samples. On the other hand, for a sufficiently slowly wiggling signal, one expects that the samples should have enough information to allow recovering the original spectrum. It should be possible to compute from the samples a quantity that can give back the original spectrum, modulo the ambiguity. Then it makes sense to define that quantity as the discrete time spectrum. The sampling theorem and its consequences show that this indeed is possible.

The inverse relation to Eq. (3) is

$$x_n^s = \int_0^{f_s} X^s(f) \mathrm{e}^{i2\pi(f/f_s)n} df \,. \tag{4}$$

The sampling theorem can be neatly proved using properties of the Dirac comb $\mathrm{Ш}(x)$[7], which is the periodic summation of Dirac delta-functions,

$$\mathrm{Ш}(x) \triangleq \sum_{n=-\infty}^{\infty} \delta(x-n) \,. \tag{5}$$

This function is periodic, by unity, so it can be expanded into Fourier series

$$\mathrm{Ш}(x) = \sum_n c_n \mathrm{e}^{i2\pi x n} \,. \tag{6}$$

[7] The spiky glyph is the cyrillic letter *sha*.

The coefficients c_n are found as usual with the inverse transformation, and are all unity:

$$c_n = \int_{-1/2}^{1/2} \text{Ш}(x) \mathrm{e}^{-i2\pi xn} dx = \int_{-1/2}^{1/2} \delta(x-0) \mathrm{e}^{-i2\pi xn} dx = 1\,. \tag{7}$$

Thus, we can represent the Dirac comb also in the form

$$\text{Ш}(x) = \sum_{n=-\infty}^{\infty} \mathrm{e}^{i2\pi xn}\,. \tag{8}$$

Using the delta-function property

$$\delta(ax) = \frac{1}{|a|}\delta(x)\,,$$

one gets from Eq. (8) another representation of the time-dimensional Dirac comb Ш_T with period T,

$$\text{Ш}_T(t) \triangleq \sum_{n=-\infty}^{\infty} \delta(f - mT) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \mathrm{e}^{i2\pi tn/T}\,. \tag{9}$$

The sampling theorem follows from Eq. (9) almost immediately. Starting from the right-hand-side of Eq. (1), and freely changing the order of integrations and summations, we get

$$\begin{aligned}
\sum_{m=-\infty}^{\infty} X^a(f - mf_s) &= \sum_{m=-\infty}^{\infty} \int_{-\infty}^{\infty} x^a(t) \mathrm{e}^{-i2\pi(f-mf_s)t} dt \\
&= \frac{1}{f_s} \int x^a(t) \left(\frac{1}{\tau_s} \sum_m \mathrm{e}^{i2\pi tm/\tau_s} \right) \mathrm{e}^{-i2\pi ft} dt \\
&= \frac{1}{f_s} \int x(t) \text{Ш}_{\tau_s}(t) \mathrm{e}^{-i2\pi ft} dt \\
&= \frac{1}{f_s} \int x(t) \sum_{m=-\infty}^{\infty} \delta(f - m\tau_s) \mathrm{e}^{-i2\pi ft} dt \\
&= \frac{1}{f_s} \sum_{m=-\infty}^{\infty} x(m\tau_s) \mathrm{e}^{-i2\pi f\tau_s m} \\
&= X^s(f)\,.
\end{aligned}$$

In the EISCAT receiver, the location of the centre point and width of the analog passband is carefully matched to the sampling frequency, as they in general must be in band-pass sampling. The analog passband is so selected that the periodic replicas in the spectral domain do not overlap (to prevent possibly distorting spectral information), but on the other hand cover the frequency axis more or less uniformly (so that receiver noise will look like white noise; see the discussion immediately after Eq. (91)). And in any case, the passband must be sufficiently wide to cover the analogue frequencies of interest. The top panel of Fig. 8 shows schematically the analog passband and how it is periodically extended in sampling. The 15 MHz intervals on the frequency axis with endpoints 7.5+m×15 MHz are called the Nyquist zones of the 15 MHz sampling. In most

cases, spectral data are presented using the Nyquist zone around the zero-frequency. In this article, I call that zone, $[-7.5, 7.5]$ MHz, the primary baseband.

The bottom part of Fig. 8 illustrates schematically how the spectrum of a narrow band (narrow compared to the passband) real-valued analog signal typically is transformed in a digital channel. Referring to the Figure, it can be appreciated that that there is a need for two "basebands". The primary A/D output samples x_n^s, flowing at the rate $f_s = 15$ MHz, are never directly used in EISCAT. Instead, a decimating filter is used to reduce—decimate—the sampling rate by some integer factor M, so that the final sample stream z_n that is stored to the channel buffer memory and is available for further processing or permanent storage, represents a sampling rate $f_s' = f_s/M$. This rate is termed the final sampling rate in this article.

The operation of a factor-M decimator, such as the block marked by "M" in Fig. 6, on a sequence y_n is formally defined via the relation

$$z_n \triangleq y_{Mn}\,, \tag{10}$$

which says that the decimated sequence z_n is obtained by picking every M'th element of the original sequence y_n, and disregarding others.

Before proceeding to inspect the spectrum of the decimated sequence, an early warning must be given. The decimation operation, like the digital filtering operation in Eq. (14), is a linear operation on the sample sequence, and so can, e.g., represented by a matrix. But there is a significant qualitative difference between these two basic linear signal processing operations and operators. The digital filtering operator is "shift invariant", while the decimation operator is not. In practical terms, the shift invariance means that if we have a long input sequence to the filter, and then we shift the sequence in time, the output sequence is essentially the same, it is just shifted by the same amount as the input. But if the input sequence to the decimator is shifted by anything else than a multiple of the decimation factor, an entirely different output sequence can result.

We will also see that filtering operators are "commutative", so that if one first filters by filter H_1, and then by H_2, the same result is be achieved if the operations are done in the opposite order. And clearly, two decimation operators are commutative. But instead, a decimation operator and a filtering operator do not commute with each other, as we will see later in this article. It also follows that while two filters can be combined to a single filter, a filter and decimator cannot be combined into a filter (see footnote 9 on p. 18.).

Recalling the sampling theorem, one expects that the spectrum $Z(f)$ of the z_n sequence is periodic by the associated sampling frequency f_s'. By the same token, the spectrum of the input sequence y_n to the decimator in Fig. 6 still has the periodicity f_s. The change of periodicity is achieved by kind of "finite periodic summation", by adding together M shifted copies of the spectrum $Y(f)$, with the shifts taken in steps of $f_s' = f_s/M$,

$$Z(f) = \sum_{m=0}^{M-1} Y(f - m f_s')\,. \tag{11}$$

We refer to Eq. (11) as the decimation theorem. We will occasionally use the notation of a "decimation operator" $\mathbf{M}^{\downarrow}$ as a shorthand for the finite periodic summation like in Eq. (11), and write Eq. (11) in the form

$$Z(f) = \mathbf{M}^{\downarrow}.Y(f)\,.$$

The decimation theorem could be proved from the sampling theorem, by comparing the results of two different ways of producing samples with the final sampling rate: either directly with the final rate f_s'; or sampling initially with rate f_s, followed by decimation (without any filtering involved) by M.

But I give here an independent proof that is analogous to the proof of the sampling theorem, but instead of the Dirac delta function, involves the Kronecker delta. Starting from the r.h.s. of Eq. (11), we get

$$\begin{aligned}\sum_{m=0}^{M-1} Y(f - m f_s') &= \frac{1}{f_s} \sum_{m=0}^{M-1} \sum_{n=-\infty}^{\infty} y_n \mathrm{e}^{-i2\pi[(f-mf_s')/f_s]n} \\ &= \frac{1}{f_s} \sum_n y_n \left(\sum_{m=0}^{M-1} \mathrm{e}^{i2\pi \frac{m}{M} n} \right) \mathrm{e}^{-i2\pi(f/f_s)n} \,.\end{aligned}$$

The sum of type $1 + x + \cdots + x^{M-1}$ in parenthesis equals M when n is an integer multiple of M, and is zero otherwise, so is equal to $M \sum_k \delta_{n,kM}$. We have

$$\begin{aligned}\sum_{m=0}^{M-1} Y(f - m f_s') &= \frac{M}{f_s} \sum_k y_{kM} \mathrm{e}^{-i2\pi(f/f_s)kM} \\ &= \frac{1}{f_s'} \sum_k z_k \mathrm{e}^{-i2\pi(f/f_s')k} \\ &= Z(f) \,,\end{aligned}$$

which completes the proof.

The effect of decimation is sketched in Fig. 8 as the step from $Y(f)$ to $Z(f)$. I call the frequency interval $[-f_s'/2, f_s'/2]$ the final baseband in this article.

Referring to Fig. 8 and Fig. 6, the first step of processing after sampling is a frequency shift towards the right, to move a negative frequency component of the spectrum $X^s(f)$ to, or near to, the zero frequency. This step is called quadrature detection. In hardware, the frequency shift is achieved by multiplying the still real-valued samples x_n^s by a complex exponential sequence in the NCOM module, to produce the complex-valued sequence x_n

$$x_n = \mathrm{e}^{i2\pi\nu_{\mathrm{nco}} n} \, x_n^s \,. \tag{12}$$

The dimensionless NCO frequency parameter ν_{nco} is

$$\nu_{\mathrm{nco}} = f_{\mathrm{nco}} / f_{\mathrm{s}} \,. \tag{13}$$

That multiplication by a complex sinusoid in the time domain corresponds to a frequency shift in the frequency domain, follows directly from the definition of the spectrum,

$$\begin{aligned}X(f) &= \frac{1}{f_s} \sum_n x_n \mathrm{e}^{-i2\pi(f/f_s)n} \\ &= \frac{1}{f_s} \sum_n \left(x_n^s \mathrm{e}^{i2\pi\nu_{\mathrm{nco}} n} \right) \mathrm{e}^{-i2\pi(f/f_s)n} \\ &= \frac{1}{f_s} \sum_n x_n^s \mathrm{e}^{-i2\pi[(f-f_{\mathrm{nco}})/f_s]n} \\ &= X^s(f - f_{\mathrm{nco}}) \,.\end{aligned}$$

The frequency f_{nco} is a user-configurable channel-specific tuning value, just called the NCO value of the channel in EISCAT. In Fig. 8, f_{nco} is equal to f_2^a, the location of the positive frequency component of the analogue spectrum. Thus it is the negative frequency component at $f_1^a = -f_2^a$ that moves to zero frequency in the detection.

The general strategy is to ensure that the transmission frequency f_{tx}, somewhere around 11 MHz as it appears at the A/D input, is shifted to zero frequency, so one normally takes $f_{\mathrm{nco}} = f_{\mathrm{tx}}$. After the primary sampling and the detection, the negative frequency component f_1^a of the analogue signal will appear at position f_1 near zero in the primary baseband. An alias of the positive frequency component of the signal at f_2^a will show up in the primary baseband at position f_2. The frequencies f_1 and f_2 are computed by first translating f_1^a and f_2^2 to the interval $[0\,,\,f_s]$ using the modulus operator, and then, if the result is larger than $f_s/2$, subtracting f_s, as in Algorithm 1.

---

for k=1,2 **do**
  $f_k = \mathrm{mod}\,(f_k^a + f_{\mathrm{nco}}\,,\,f_s)$
  if $f_k > f_s/2$ **then**
    $f_k = f_k - f_s$
  end
end

Algorithm 1: Pseudocode for mapping the analogue spectrum with delta-peaks at f_1^a and f_2^a to the primary baseband, $f_k^a \to f_k$. To map directly to the final baseband, $f_k^a \to f_k'$, replace f_k by f_k' and f_s by f_s'.

Because the decimation is by an integer factor, the recipe of Algorithm 1 can be used to compute also the positions f_1' and f_2' of the frequencies in the final baseband, just by replacing f_s by f_s'. Nevertheless, even if the frequencies f_1' and f_2' in the final baseband were indeed computed directly by the above recipe, one still needs to compute also the intermediate frequencies f_1 and f_2 on the primary baseband, for it is these frequencies that are relevant in determining the attenuation of the signal in the filtering.

Filtering of the sequence x_n by a by a filter with impulse response sequence h_k (the elements h_k are called the filter taps) is by definition via the convolution sum [8],

$$y_n \triangleq \sum_{k=-\infty}^{\infty} h_k x_{n-k} = \sum_{k=-\infty}^{\infty} h_{n-k} x_k\,. \tag{14}$$

The EISCAT channel filters are of the FIR (finite impulse response) type, which have only a finite number of non-zero filter taps h_k. Moreover, they are real-valued, meaning that the taps h_k are all real-valued, even though the sequences x_n and y_n are complex-valued.

In the frequency domain, the filtering transforms the spectrum $X(f)$ to $Y(f)$ by simple multiplication,

$$Y(f) = H(f)\,X(f)\,, \tag{15}$$

where the transfer function $H(f)$ of a digital filter is defined as the Fourier transform of

[8] To be consistent with my quest of making the digital quantities to correspond as closely as possible with the analogue quantities, one should replace the dimensionless taps h_k with $h_k\tau_s$ on the right-hand-side of both Eq. (14) and Eq. (16), so as to force h_k to have dimension 1/s. But we do not compare digital and analog impulse responses in this article, so including the τ_s seems excessively pedantic.

the filter's inpulse response sequence h_n,

$$H(f) \triangleq \sum_k h_k \mathrm{e}^{-i2\pi(f/f_s)k} \,. \tag{16}$$

To prove Eq. (15) from Eq. (14), we use the definitions of spectrum and transfer function, and make a change of summation index ($n \to m + k$) in a suitable place. Thus,

$$\begin{aligned}
Y(f) &= \frac{1}{f_s} \sum_n y_n \mathrm{e}^{-i2\pi(f/f_s)n} \\
&= \frac{1}{f_s} \sum_n \left(\sum_k h_k x_{n-k} \right) \mathrm{e}^{-i2\pi(f/f_s)n} \\
&= \sum_k h_k \left(\frac{1}{f_s} \sum_n x_{n-k} \mathrm{e}^{-i2\pi(f/f_s)n} \right) \\
&= \sum_k h_k \left(\frac{1}{f_s} \sum_m x_m \mathrm{e}^{-i2\pi(f/f_s)(k+m)} \right) \\
&= \sum_k h_k \left(\frac{1}{f_s} \sum_m x_m \mathrm{e}^{-i2\pi(f/f_s)m} \right) \mathrm{e}^{-i2\pi(f/f_s)k} \\
&= H(f)\, X(f) \,.
\end{aligned}$$

In this article, as with the spectra, we consider also the transfer function to be a function of the physical frequency f rather than the dimensionless normalised frequency f/f_s. The transfer function is a dimensionless quantity, both in the digital and in the analog domain. In the digital domain, also the impulse response is a dimensionless quantity. In continuous-time domain, the impulse response $h(t)$ defines filtering via the convolution integral

$$y(t) = \int_{-\infty}^{\infty} h(t')x(t - t')dt' \,,$$

which implies for $h(t)$ the unit $1/\mathrm{s}$. But because the continuous-time transfer function is defined via the normal Fourier integral,

$$H(f) \triangleq \int_{-\infty}^{\infty} h(t)\mathrm{e}^{-i2\pi f t} dt \,,$$

the transfer function stays dimensionless, and Eq. (15) holds in continuous-time also.

In both digital and analog domain, the squared magnitude of the transfer function, $|H(f)|^2$, can be viewed as kind of "relative gain" of the filter, which describes how much attenuation or amplification takes place at a given frequency. Typically, for theoretical considerations, one normalises the transfer function so that $H(0) = 1$, but we do not require this here.

The transfer function is periodic by the sampling frequency f_s. The squared magnitude of the transfer function is also called the power (-domain) response of the filter. In discrete-time, it has the property

$$\int_0^{f_s} |H(f)|^2 df = f_s \sum_n |h_n|^2. \tag{17}$$

The "reference frequency" f_s in Eq. (16) is the sampling frequency of the samples coming into the filter, the primary 15 MHz sampling frequency f_s in our case. Thus, the filter $H(f)$ lives naturally in the primary 15 MHz wide baseband. On the other hand, the spectrum $Z(f)$ of the final decimated data z_n that are visible to user has period equal to the the final sampling frequency f_s'. So one needs to address the question about how $H(f)$ relates to the relevant things, spectra especially, in the final baseband.

It would be nice to have the channel's band-shaping filter also being periodic by f_s', so that it could be fully specified by showing it on the final baseband only. Motivated by the decimation theorem of signal spectra, Eq. (11), one can form a quantity, call it $H'(f)$, by

$$H'(f) \triangleq \mathbf{M}^{\downarrow}.H(f) = \sum_{m=0}^{M-1} H(f - m f_s') \,, \tag{18}$$

which has the required periodicity. Could the squared magnitude of $H'(f)$ perhaps be the desired band-shaping function? As far as I can recall, this was vaguely the concept I had in mind when I made a drawing of EISCAT Svalbard radar digital receiver for a paper by Wannberg et al. that was published in 1997 in Radio Science.

So it is lamentable that the above strategy cannot be made to work. One cannot form a proper filter whose transfer function would have the period of the final sampling frequency and which would map the input spectrum $X(f)$ to the final spectrum $Z(f)$ by a simple multiplication. A proper filter, one respecting the defining equation Eq. (14), will not change the sampling frequency.[9] A decimator is needed somewhere between the input and the output, and the non-shift-invariant linear operation of filtering together with decimation cannot, in general, be replaced by the shift-invariant linear operation of filtering alone.

4 Revisiting a 1997 paper on the EISCAT Svalbard Radar

As far as I know, the only peer-reviewed account that covers the modern-day EISCAT digital receiver is a 1997 paper by Wannberg et al. in Radio Science, which I will here refer to as [W97][10]. The digital receiver is explained extremely briefly in the paper, and for that reason alone, expanding on the paper a little might be in order. But it now seems to me that Figure 6 of the paper, depicting the ESR receiver, is not quite correct. For convenience, I have reproduced the relevant part of that figure as Fig. 9 of this article. The right-hand-side column shows the frequency mapping of the EISCAT Svalbard radar receiver, somewhat analogously to Fig. 8 of this article. Both columns have something that could usefully be changed. I handle the functional block diagram first.

[9] It is bad that the expression "decimating filter" is in common use in the signal processing literature; I consider it basically to be an oxymoron. In the DSP context, I would prefer the word "filter" to mean solely something that respects the convolution Eq. (14), and therefore does not change the sampling rate. In this narrow view, a decimating filter is not a filter; not a good way to use the language. But of course, the game in this respect was lost long time ago, and a "filter" can be just about any processing unit that somehow strips something from its input.

[10] Radio.Sci., 36, 2283–2307, 1997, *The EISCAT Svalbard radar, A case study of modern incoherent scatter radar system design* by Wannberg, Wolf, Vanhainen, Röttger, Postila, Markkanen, Jacobsen, Stenberg, Larsen, Eliassen, Heck and Huuskonen.

Functional block representation of the HSP43220 decimating filter

The part of the [W97] drawing that represents the digital receiver, the left-hand-side column of the diagram of Fig. 9, shows two low-pass filters and a single decimator located between them. By the look of it, the drawing compares badly both with the Fig. 6,which has one filter and one decimator, and the more detailed Fig. 7, which has two filters and two decimators. All these three drawings purport to represent precisely the same channel hardware, implemented via the EISCAT channel board. One should not become confused with the particular values of the frequencies in the ESR diagram; both the primary sampling frequency and the location and size of analog passband were different in the early ESR receiver compared to the present EISCAT receivers.

I think the way the digital receiver is drawn in [W97] is wrong. How we arrived at the drawing I cannot recall. It is not impossible that we simply interpreted the decimating filter's data sheet slightly wrongly. In the HSP43220 data sheet, the top level functional diagram of the decimating digital filter (DDF) is drawn as two cascaded blocks, labeled "high order decimation filter" (HDF) and "fir decimation filter" (FDF)[11]. This is clear enough, but trouble begins when one tries to find out, using the more detailed diagrams, where the filter(s) and decimator(s) are within these two main blocks.

It is especially difficult to find decimator blocks—they simply are not all there. There is one, but only one, block that has a promising name: a "decimation register". That block is embedded in the middle of the HDF, between two blocks called the integrator (H_I) and the comb (H_C). But there is no decimator block marked for the FDF anywhere in the data sheet. On the other hand, the name "fir decimation filter" suggests that the FDF would be doing decimation also, in addition of filtering. In the [W97] drawing, we show only one decimator, so presumably we thought that the FDF would only be doing filtering. (Why would we have shown a decimator for the HDF, but not for the FDF, if we had thought that both of them actually do decimation?) I will denote by H_{FDF} the functional block that does the actual filtering in the FDF.

For the [W97] paper, so it seems, we had concluded that the hardware configuration really is

$$\mathrm{DDF} = \langle H_I \to M_1 \to H_C\, H_{FDF} \rangle \quad ?! \ ,$$

where I have denoted the decimator inside the HDF by M_1. This can be shortened by noting that the last two filters can be combined to a single filter $H' = H_C\, H_{FDF}$, whose transfer function is the product of the transfer functions of the two filters, as follows from Eq. (15). Then the configuration would become

$$\mathrm{DDF} \hookrightarrow \langle H_I \to M_1 \to H' \rangle \quad ?! \ ,$$

which could be what our drawing in [W97] shows. In this shorthand diagrammatic notation, I use the sign "$\to$" as an abbreviation for "followed by", and the sign "$\hookrightarrow$" for "can be replaced by", or "is equivalent to". I do not draw an arrow between two filters, because the actual order does not matter there.

But then we should have been left wondering, in addition of the attribute "decimating" in the name FDF, also why the programming instructions in the data sheet refer to two decimation parameters, one for the HDF, one for the FDF. I will show later in this section that the FDF's decimation functionality cannot simply be absorbed into the HDF decimator. So why is there not any FDF decimator block shown in the data sheet?

[11] The actual label used in the data sheet for the fir decimation filter is just "FIR". That obviously is a completely unusable name for our purposes here.

A moment's thought suggest that decimation in the FDF block is such a trivial operation that it does not, *per se*, merit its own block in the hardware. Decimation can be done by not computing at all, to at least not sending forward, those samples that are not actually needed. So from the hardware point of view, the decimation in the FDF is more of a control operation, rather than an actual computation.

But that does not make it any less real. For a functional model of the system, we need a block diagram where we can clearly point out where the filtering is done, and where the decimation. I will show below that the actual hardware is equivalent both to my detailed block diagram, Fig. 7, and the simplified diagram of Fig. 6. I will also show that the drawing in [W97] is not equivalent with either of these and is, therefore, basically wrong.

If I were to draw the receiver diagram of Fig. 9 now, I would use Fig. 7. It differs from [W97] in that the final decimator M_2 is missing from [W97]. Even if dropping out samples after the final filter is trivial from the hardware point of view, the effect to the final spectra can be entirely non-trivial, as will be discussed in the next section. The final decimator is an essential feature of the system, and must not be left out of the functional block diagram.

To analyse the filter's data sheet in order to verify the actual filter structure, we need the result that in a certain case, it is possible to change the order of a decimator and filter in a functional block diagram. In the same exhausted late-night spirit that prompted Intersil (then Harris Semiconductors) engineers to choose the name "FIR" as the proper name for one of their FIR filters, I will call the result the *order-change rule*, and write it down diagrammatically as

$$\langle \mathrm{M} \to \mathrm{H} \rangle \hookrightarrow \langle \mathrm{H}^{(\mathrm{M})} \to \mathrm{M} \rangle \,. \tag{19}$$

The order change rule (19) says that

- In any system of cascaded digital filters and decimators, it is permissible to change the configuration "decimator followed by filter",

  $$\langle \mathrm{M} \to \mathrm{H} \rangle \,,$$

  to the configuration "a filter followed by the same decimator",

  $$\langle \mathrm{H}^{(\mathrm{M})} \to \mathrm{M} \rangle \,,$$

  where the new filter $\mathrm{H}^{(\mathrm{M})}$ is formed from the original filter H by inserting $M-1$ zeros between each pair of consecutive filter taps, as in Eq. (20).

We call those two decimating filter configurations functionally equivalent, to mean that for any input, both systems will produce the same output.

In the end of the section, I will show that the other direction of the order change is not in general possible: a given configuration [H → M] cannot in general be replaced by an [M → H'], no matter what H'.

The impulse responses $h_n^{(M)}$ and h_n of the two filters in Eq. (19) are related by

$$h_n^{(M)} = \begin{cases} h_k & \text{when } n = Mk \text{ for some } k \\ 0 & \text{otherwise} \end{cases} \,. \tag{20}$$

Eq. (20) implies that

$$h_k = h_{Mk}^{(M)} \quad \text{for any } k \,. \tag{21}$$

The order-change rule is most directly proven in the time domain. We denote the transforming signal in the processing chain [M → H] by x_n, y_n and z_n,

$$x_n \rightsquigarrow \mathrm{M} \rightsquigarrow y_n \rightsquigarrow \mathrm{H} \rightsquigarrow z_n$$

and the signal in the processing chain [$\mathrm{H}^{(\mathrm{M})}$ → M] by x_n, y'_n and z'_n,

$$x_n \rightsquigarrow \mathrm{H}^{(\mathrm{M})} \rightsquigarrow \mathrm{y}'_\mathrm{n} \rightsquigarrow \mathrm{M} \rightsquigarrow \mathrm{z}'_\mathrm{n}\,,$$

and verify that for any input x_n, the outputs z_n and z'_n are the same:

$$\begin{aligned}
z_n &= \sum_k h_k y_{n-k} && \Leftarrow \text{definition of filter, Eq. (14)} \\
&= \sum_k h_k x_{M(n-k)} && \Leftarrow \text{definition of decimator, Eq. (10)} \\
&= \sum_k h^{(M)}_{Mk} x_{Mn-Mk} && \Leftarrow \text{Eq. (21)} \\
&= \sum_k h^{(M)}_{k} x_{Mn-k} && \Leftarrow \text{Eq. (20)} \\
&= y'_{Mn} \\
&= z'_n\,.
\end{aligned}$$

We will prove the order-change rule also in the frequency domain, using the decimation theorem. We will first show that it follows from the definition Eq. (20) of the filter $\mathrm{H}^{(\mathrm{M})}$, that as function of the physical frequency f, the transfer functions $H^{(M)}$ and H are equal,

$$H^{(M)}(f) = H(f) \quad \text{for all } f\,. \tag{22}$$

The transfer functions can be the same, even though the impulse responses are different, because the two filters refer to two different sampling frequencies; as a function of physical time, they would cover the same time interval[12]. Formally, keeping careful track of the relevant sampling frequency in the definition Eq. (16) of the transfer function, and making use of Eq. (21) and Eq. (20), we have

$$\begin{aligned}
H(f) &= \sum_k h_k \mathrm{e}^{-i2\pi(f/f'_s)k} && \Leftarrow H \text{ refers to } f'_s \\
&= \sum_k h_k \mathrm{e}^{-i2\pi(f/f_s)Mk} \\
&= \sum_k h^{(M)}_{Mk} \mathrm{e}^{-i2\pi(f/f_s)Mk} \\
&= \sum_k h^{(M)}_{k} \mathrm{e}^{-i2\pi(f/f_s)k} \\
&= H^{(M)}(f)\,.
\end{aligned}$$

[12] This is one of the cases where it would be good to the have the relevant sampling frequency explicit in the notation of the transfer function. As functions of two variables, f_s and f, the transfer functions $H^{(M)}$ and H are, of course, different. The notational problem is avoided in the "standard" DSP notation which uses the normalised frequency as the principal frequency variable. But the argument does not become less confusing by that. Instead, you will find those authors, too, constantly reminding you that their formulas refer now to the higher sampling frequency, now to the lower.

I mention in the passing that the operation of inserting $M-1$ zeros after each element of the sequence x_n so that the new sequence represents M times faster sampling rate, is called "interpolation", and is kind of inverse operation to the decimation, so that we could denote it by $\mathbf{M}^{\uparrow}$. Under this operation, the spectrum of the sequence, as function of the physical parameter f, does not change, but only scales by $1/M$,

$$\mathbf{M}^{\uparrow}.X(f) = \frac{1}{M} X(f)\,.$$

The scaling becomes from our normalisation factor $1/f_s$ in the definition Eq. (3) of the spectrum. We do not have such normalisation in the definition of the transfer function, and so the 1/M is missing in Eq. (22). Nevertheless, we will call the filter $\mathrm{H}^{(\mathrm{M})}$ the "interpolated form" of the filter H, and may use the alternative notation $\mathbf{M}^{\uparrow}.\mathrm{H}$ for it.

We now use the basic property Eq. (15) of the transfer function, and the decimation theorem Eq. (11), to show that for any input spectrum $X(f)$ to the two systems in Eq. (19), the output spectra will be the same. We denote the spectra of signals x_n ,y_n, z_n, y'_n and z'_n by the corresponding capital letters, and verify that for any given input X, the output spectra Z and Z' are equal:

$$\begin{aligned}
Z(f) &= H(f)Y(f) && \Leftarrow \text{Eq. (15)} \\
&= H(f)\sum_{m=0}^{M-1} X(f - m f'_s) && \Leftarrow \text{Eq. (11)} \\
&= \sum_m H(f)X(f - m f'_s) \\
&= \sum_m H(f - m f'_s)X(f - m f'_s) && \Leftarrow H \text{ is periodic by } f'_s \\
&= \sum_m H^{(M)}(f - m f'_s)X(f - m f'_s) && \Leftarrow \text{Eq. (11)} \\
&= \sum_m Y'(f - m f'_s) && \Leftarrow \text{Eq. (15)} \\
&= Z'(f)\,.
\end{aligned}$$

This proof of the the order change rule in the spectral domain is illustrated by the drawings in Fig. 13.

With the order-change rule, we conclude from the data sheet that the actual hardware structure of the HDF as "integrator-decimator-comb", or

$$\mathrm{HDF} = \langle \mathrm{H_I} \to \mathrm{M}_1 \to \mathrm{H_C} \rangle\,, \tag{23}$$

can be replaced by the equivalent system

$$\mathrm{HDF} \hookrightarrow \langle \mathrm{H_I}\, \mathrm{H_C^{(M_1)}} \to \mathrm{M}_1 \rangle\,. \tag{24}$$

It follows from the basic property of the transfer functions that the two cascaded filters in the chain (24) can be combined into a single filter with transfer function

$$H_{\mathrm{CIC}}^{(M_1)}(f) = H_I(f)\, H_{\mathrm{C}}^{(M_1)}(f)\,. \tag{25}$$

The abbreviation "CIC" stands for "cascaded integrator-comb"[13]. Thanks to Eq. (22), the superscripts M_1 are not needed in the transfer functions, so we conclude that the high-order decimating filter HDF is functionally equivalent to a single filter with transfer function

$$H_{\mathrm{CIC}}(f) = H_I(f)H_C(f)\,, \tag{26}$$

followed by the decimator M1. We note down this result diagrammatically as

$$\mathrm{HDF} \hookrightarrow \langle \mathrm{H_{CIC}} \to \mathrm{M_1} \rangle\,. \tag{27}$$

The HSP42330's other main block, the fir decimating filter FDF, essentially computes the convolution sum of an $N/2$ taps, even or odd symmetric, finite impulse response filter, with freely settable tabs, in a straightforward way, using a single multiplier-accumulator. The input data are the filter coefficients and the latest N samples from the HDF, which are available in a RAM buffer. Only those output samples are computed that are actually needed according to the FDF's decimation parameter M_2. The FDF generates a signal to tell external hardware when a new sample is ready to be read from the FDF's output register. This functionality clearly can, and should, be represented as a filter followed by a decimator,

$$\mathrm{FDF} \hookrightarrow \langle \mathrm{H_{FDF}} \to \mathrm{M_2} \rangle\,. \tag{28}$$

To my mind, the most natural functional block diagram for the whole HSP443220 decimating digital filter therefore is the equivalent representation

$$\mathrm{DDF} \hookrightarrow \langle \mathrm{H_{CIC}} \to \mathrm{M_1} \to \mathrm{H_{FDF}} \to \mathrm{M_2} \rangle\,, \tag{29}$$

which corresponds to the one in Fig. 7. Using the order-change rule to move the HDF decimator $\mathrm{M_1}$ to be after the FDF, one gets the representation of Fig. 6

$$\begin{aligned} \mathrm{DDF} &\hookrightarrow \langle \mathrm{H_{CIC}}\, \mathrm{H_{FDF}^{(M_1)}} \to \mathrm{M_1}\, \mathrm{M_2} \rangle & (30)\\ &= \langle \mathrm{H_{DDF}} \to \mathrm{M_{DDF}} \rangle\,. & (31) \end{aligned}$$

The total transfer function of the filter part $\mathrm{H_{DDF}}$ of the HSP43220 is

$$H_{\mathrm{DDF}}(f) = H_{\mathrm{CIC}}(f)H_{\mathrm{FDF}}(f)\,. \tag{32}$$

The total decimation of the HSP43220 is the product of the decimation M_1 of the HDF and M_2 of the FDF,

$$M_{\mathrm{DDF}} = M_1 M_2\,. \tag{33}$$

Corresponding to Eq. (32), the total equivalent impulse response h_{DDF} of the system in the one-filter one-decimator equivalent form of Eq. (31), is the convolution of the impulse responses of the CIC and the zero-stuffed FDF,

$$h_{\mathrm{DDF}} = h_{\mathrm{CIC}} * h_{\mathrm{FDF}}^{(M_1)}\,. \tag{34}$$

This filter operates on samples flowing with the primary sampling rate f_s. In the context of the EISCAT receiver, we may call the filter part of, $\mathrm{H_{DDF}}$, the DDF the channel's postdetection filter, PDF.

[13] E.B. Hogenauer, An Economic Class of Digital Filters for Decimation and Interpolation, *IEEE Trans. on Acoust., Speed, Signal Processing*, vol ASSP-29, pp. 155-162, Apr. 1981.

Nothing much needs to be said about the FDF; the ddfplan.m program provides a way for the filter designer to assign the filter taps, by imagining that the FDF is a cascade of N boxcar-in-time filters, each of length L. For the FDF, the decimation, M_2, can be set independently of the filter taps (within filter speed specs). The caption of Fig. 6 gives some of the details. The created FDF is a linear phase filter.

For the HDF, we still need dig one level deeper into the HSP43220 data sheet, to find out what more specifically goes into the integrator $\mathrm{H_I}$ and the comb $\mathrm{H_C}$. These consist of five identical basic integrators and combs, respectively, which we denote by $\mathrm{H_i}$ and $\mathrm{H_c}$, so that the actual hardware can be diagrammed as

$$\mathrm{HDF} = \langle \mathrm{H_I} \to \mathrm{M_1} \to \mathrm{H_C} \rangle = \langle \mathrm{H_i H_i H_i H_i H_i} \to \mathrm{M_1} \to \mathrm{H_c H_c H_c H_c H_c} \rangle \,. \tag{35}$$

Using the order-change rule five times to move $\mathrm{M_1}$ to the end and then rearranging the order of the cascaded basic integrators and the zero-stuffed equivalent basic combs, we get an equivalent representation

$$\mathrm{HDF} \hookrightarrow \langle \langle \mathrm{H_i\, H_c^{(M_1)}} \rangle^{\mathrm{K}} \to \mathrm{M_1} \rangle \,, \tag{36}$$

where we have marked the number of sections actually used to by K, which can be from 1 to 5. The number of sections is programmable configuration parameter, and is set in the ddfplan.m script. By comparing Eq. (36) to Eq. (27), we identify the filter part, the CIC, of the HDF, as the equivalent filter

$$\mathrm{CIC} \hookrightarrow \langle \mathrm{H_i\, H_c^{(M_1)}} \rangle^{\mathrm{K}} = \mathrm{H_{cic}}^{\mathrm{K}} \,, \tag{37}$$

where we defined the basic cascaded integrator-comb, $\mathrm{H_{cic}}$, as the filter consisting of the basic integrator $\mathrm{H_i}$ followed by the zero-stuffed equivalent basic comb $\mathrm{H_c^{M_1}}$,

$$\mathrm{H_{cic}} \triangleq \langle \mathrm{H_i\, H_c^{(M_1)}} \rangle \hookrightarrow \langle \mathrm{H_i\, M_1\, H_c} \rangle \,. \tag{38}$$

The transfer function H_{cic} of the basic CIC is (no substribt M_1 needed)

$$H_{\mathrm{cic}}(f) = H_i(f) H_c(f) \,, \tag{39}$$

and the transfer function of the whole CIC is

$$H_{\mathrm{CIC}}(f) = [H_i(f) H_c(f)]^K = [H_{\mathrm{cic}}(f)]^K \,. \tag{40}$$

We now need to find out the transfer functions of the basic integrator $\mathrm{H_i}$ and the basic comb $\mathrm{H_c}$, or directly the transfer function $H_{\mathrm{cic}}(f)$ of the basic CIC. We will compute $H_i(f)$ and $H_c(f)$ separately below using Fourier-technics, but for better insight into the basic CIC's operation, it is better to compute $H_{\mathrm{cic}}(f)$ directly in the time domain. We could use either of the two equivalent representations of the basic CIC in Eq. (38), but it is easier to work with the configuration which incorporates the zero-stuffed comb $\mathrm{H_i^{(M_1)}}$. We will get directly the sought-for transfer function of the equivalent filter, instead of needing to extract it from the input–output relation of a system having a decimator in the middle.

We inspect the data sheet to see what precisely the basic integrator and the basic comb do. First the integrator. From the data sheet one finds, curiously, that the integrator $\mathrm{H_i}$ actually is not a finite impulse response filter at all, but an infinite impulse response filter,

namely, a simple accumulator. Such a device corresponds to a recursive filter which has an input (x_n)-output (y_n) relation

$$y_n = y_{n-1} + x_n \,. \tag{41}$$

This means that the integrator just keeps endlessly accumulating the sum of all the samples it has received, and so can write the solution to Eq. (41) in the form

$$y_n = \sum_k^n x_k \,, \tag{42}$$

where we do not really care what the precise start index in the summation is (it might be a large negative number in this notation if the accumulator has been accumulating for a long time).

Then the basic comb. In the original, hardware, configuration, the comb just computes the difference between current and the last-but-one sample of its input stream, that is, its input output relation is

$$y'_n = x'_n - x'_{n-1} \,. \tag{43}$$

Thus, the basic comb is a two-tap FIR filter, with impulse response

$$h_c = (1, -1) \,. \tag{44}$$

The impulse response $H_c^{(M)}$ of the equivalent filter therefore is

$$h_c^{(M)} = (1, 0, \ldots, 0, -1) \quad (M_1 - 1 \text{ zeros between the 1 and } -1) \,. \tag{45}$$

The input-output, $y_n \to z_n$, relation of the zero-stuffed basic comb is

$$z_n = \sum_k h_{c,k}^{(M)} y_{n-k} = \sum_k h_{c,k} y_{n-M_1 k} = y_n - y_{n-M_1} \,. \tag{46}$$

That is, the zero-stuffed comb computes the difference between its current input sample and the by-M_1 earlier input sample. The comb input samples y_n are coming from the basic integrator. We can now state the input-output relation of the basic CIC. In terms of its input samples x_n, the output samples z_n of the basic CIC are

$$z_n = \sum_k^n x_k - \sum_k^{n-M_1} x_k = x_n + x_{n-1} \cdots + x_{n-M_1+1} \,. \tag{47}$$

That is, the basic CIC computes the sum of the latest M_1 samples coming to it. So the basic CIC is a finite impulse response boxcar filter, with M_1 taps, all equal to unity,

$$h_{\text{cic}} = (1, \ldots, 1) \quad M_1 \text{ times “1”} \,. \tag{48}$$

We have found the precise impulse response of the basic CIC, so have basically solved the operation of the HDF. But there is still one item that needs to be addressed before we can claim that we understand how the HDF works. This relates to the basic integrator, Eq. (42). Don’t we have the danger of an overflow in the ever-accumulating sum y_n at some point? The interesting answer is that yes, we for sure are going to have overflows,

but that does not matter. This is due to the wrapping-around property of the two's complement arithmetic that both the integrator and the comb use.

The result is that even though the ever-integrating sum can, and will, overflow, the basic CIC's output, the *difference* of two such sums in Eq. (47), which we actually use, will still be correct, provided that a certain condition holds. The condition is that that correct answer, the sum of the M_1 latest x_n, itself must not overflow.

I do not know of any simple proof for this property of 2's complement arithmetic, but consider the following example. Assume word length of 3 bits, so the legal numbers are $-4, -3, \cdots +3$. Denote the two's complement addition by $\oplus$, so that, for example, $2 \oplus 3 = -3\,(= 5 - 8)$. Then assume that $y_{n_1} = 3$, and assume that first a 2, and then a 1, is added to it, to give

$$y_{n_2} = (3 \oplus 2) \oplus 1 = -3 \oplus 1 = -2,$$

so that the difference, when also computed in the two's complement scheme, is

$$y_{n_2} \ominus y_{n_1} = -2 \ominus 3 = 3\,,$$

which is the correct difference between 3 and 3+2+1. As about the "certain condition", consider the following. Take $y_{n_1} = 1$, and add first 2 and then 3 to it to get y_{n_2}. This time, the difference is

$$y_{n_2} \ominus y_{n_1} = [(1 \oplus 2) \oplus 3] \ominus 1 = -2 \ominus 1 = -3\,,$$

which is *not* the actual difference between $1 + 2 + 3$ and 1, but rather, its complement. The difference between the two cases is that in the correctly working one, the correct difference 3 is a legal number in the 3-bit system, while in the problem case, the actual difference 5 no more is a legal 3-bit two's complement number. Both of these cases fit into the general rule,

$$a \oplus b \oplus c \ominus a = b \oplus c\,,$$

which shows that we will get trouble when the correct difference overflows so that $b \oplus c \neq b + c$.

From the impulse response Eq. (48) of the basic CIC, we get the transfer function of the basic CIC as

$$\begin{aligned} H_{\text{cic}}(f) &= \sum_{n=0}^{M_1-1} \mathrm{e}^{-i2\pi(f/f_s)n} \\ &= \frac{1 - \mathrm{e}^{-i2\pi(f/f_s)M_1}}{1 - \mathrm{e}^{-i2\pi(f/f_s)}} \qquad (49) \\ &= M_1 \times \mathrm{e}^{-i\pi(M_1-1)(f/f_s)} \times D_{M_1}(\pi \frac{f}{f_s})\,, \qquad (50) \end{aligned}$$

where the Dirichlet kernel $D_{M_1}(.)$ is defined by

$$D_M(x) \triangleq \frac{\sin(xM)}{M \sin(x)}\,. \qquad (51)$$

The complete, to unity normalized transfer function of the HDF's filter, with K sections in use, therefore is

$$H_{\text{CIC}}(f) = \mathrm{e}^{-i\pi K(M_1-1)(f/f_s)} \times \left[D_{M_1}(\pi \frac{f}{f_s}) \right]^K\,. \qquad (52)$$

The CIC is a linear phase filter, and because the FDF also is, the whole DDF is a linear phase filter. The CIC has zeros at frequencies that are multiples of the HDF output rate f_s/M_1. Its impulse response, the h_{CIC} in Eq. (34), is cascade of K boxcar filters, each having M_1 taps,

$$h_{\mathrm{CIC}} = \mathrm{box}_{M_1} * \cdots * \mathrm{box}_{M_1} \quad (K \text{ times}) \,. \tag{53}$$

In Eq. (50), we found the transfer function of the basic CIC element by first finding its impulse response, using time-domain arguments. We never computed the H_i and H_c of Eq. (39) separately. A more standard way is to compute the CIC transfer function by computing separately the $H_i(f)$ and $H_c(f)$, and then multiplying them. The transfer function of the comb $\mathrm{H_c}$ one can easily get from its impulse response $(1, -1)$, but to compute the transfer function of the infinite impulse response filter $\mathrm{H_i}$, requires more powerful tools.

The standard way in literature to compute the transfer functions $H_i(f)$ is by solving the recursive equation Eq. (41) with z-transforms. I promised in the Introduction to do without z-transforms[14], so we will solve the equation using Fourier transforms instead.

To find the transfer function $H_i(f)$, we write the three sequencies, y_n, y_{n-1} and x_n in Eq. (41) in terms of their spectra, using the relation Eq. (4). This gives for $X(f)$ and $Y(f)$ the relation

$$\int_0^{f_s} Y(f) \mathrm{e}^{i2\pi(f/f_s)n} df = \int_0^{f_s} Y(f) \mathrm{e}^{i2\pi(f/f_s)(n-1)} df - \int_0^{f_s} X(f) \mathrm{e}^{i2\pi(f/f_s)n} df$$

or

$$\int_0^{f_s} \left[Y(f) - Y(f) \mathrm{e}^{-i2\pi(f/f_s)} + X(f) \right] \mathrm{e}^{i2\pi(f/f_s)n} df = 0 \,.$$

For this to be true for all n requires

$$Y(f) = \frac{1}{1 - \mathrm{e}^{-i2\pi(f/f_s)}} X(f) \,.$$

This implies that the transfer function of the basic integrator is

$$H_i(f) = \frac{1}{1 - \mathrm{e}^{-i2\pi(f/f_s)}} \,. \tag{54}$$

The transfer function $H_c(f)$ of the basic comb we get directly from its impulse response $h_0 = 1$, $h_1 = -1$. With this basic impulse response (not the zero-filled one), the filter is to be considered to sit in its original place, *after* the decimator. Thus the transfer function $H_c(f)$ refers to the reduced sampling rate f_s/M_1. We find

$$H_c(f) = 1 - \mathrm{e}^{-i2\pi(f/f_s)M_1} \,. \tag{55}$$

Combining Eq. (54) and Eq. (55) gives Eq. (49), as before.

I have now shown how the HSP43220 decimating digital filter, according to its data sheet, works, and have shown that both Fig. 7 and Fig. 6 are correct representations of its operation. For the sake of the argument, would it be possible that also our drawing

[14] Which is probably as good as anything. When preparing this article, I realised an error in the ddfplan.m script, related to the z-transforms. I had confused the basic z-transform concepts of a root and a zero. All the several tens of EISCAT filter files therefore contain comments where I list the locations of filter "poles". In reality, a FIR filter never has any poles. Strangely, no one has ever complained.

in [W97], Fig. 9, were a correct equivalent representation, with some clever definition of the two filters and the one decimator in between them?

The answer is no. We have seen that the actual hardware is something that terminates with a proper filter followed by a decimator,

$$\rightarrow \langle H_{FDF} \rightarrow M_2 \rangle \, .$$

For the drawing in [W97] to be equivalent to this, it must be possible to transform this actual hardware configuration to the [W97] form. In the very least, it must be possible to move the decimator M_2 from the end of the processing chain to somewhere earlier. But that is not possible. I will show below that the canonical decimating filter

$$\langle H \rightarrow M \rangle$$

cannot in general be equivalently, that is, so that it is valid with any input, be replaced by the system

$$\langle M \rightarrow H' \rangle \, ,$$

no matter what filter the H′ would be. (It is clear that the decimator M must be the same in both cases, for a proper filter does not change the sampling rate.)

The problem is that although the order-change rule, Eq. (19), says that a decimator in front of any filter can be moved after the filter, by modifying the filter in a suitable way, the opposite is not true. Rather, the following holds.

- One cannot, in general, move a decimator M situated after a filter H, to the front of that filter, so that—with possibly some changes in the filter—one could get an equivalent system.

To show this, consider the counterexample in Fig. 14, presented in the frequency domain. The input consist of two differently shaped, nearby spectral bumps (red and blue in the figure). The filter H is such that it will kill one of the bumps (blue) but leave the other intact. The decimator M is such that if the two bumps are driven though it, by the periodic replication of Eq. (11), the bumps will overlap, distorting the shape of both, now, actually, make the output constant. Then, in the case [H → M], the output is series of replicates of the bump (red) that is in the filter's passband. But in the [M → H′] case, the output is series of distorted bumps (flat line in the figure). To remove the distortion, we would need a new H′ for each input, but that is not allowed here. A counterexample in the time domain is presented in Fig. 15. There, when decimation is done first, the operation removes odd-numbered samples, so that the output can not depend on them, no matter how the filtering is done. Instead, when the filtering is done first, the output is arranged to depend on both even-numbered and odd-numbered samples. Therefore, the two filters cannot be made equal.

The asymmetry with regards of the decimator plus filter order change stems from the fact that the purpose of the filter is, often, to reduce the "information bandwidth" of the incoming stream so that the decimator can handle it; that is, that the smaller sampling rate after decimator is still a correct one. If that protection is removed, by moving the decimator into the front of the filter, disaster looms. That is why with decimating digital filters, the filter always is first and the decimator is after it. This arrangement is analogous to having an anti-aliasing filter in front of the A/D, which in the information bandwidth context can be viewed as kind of decimator, too.

About the band shape of a multirate digital receiver channel

There is no uncertainty about what one should mean by the band shape of an analog receiver channel. The input-output relation of such a channel is similar to Eq. (15),

$$|Y(f)|^2 = |H(f)|^2|X(f)|^2 \text{ for all } f\,, \tag{56}$$

and the band shape means the squared modulus $|H(f)|^2$ of the channels's transfer function. Nor would there be any ambiguity in that respect in a purely digital channel if there would be no sampling rate changes between input and output, for then Eq. (56) is in force. But as soon as there is any sampling rate conversions, aliasing can occur, and one needs to be more careful about what one means by band shape, and how one displays it.

In the [W97] drawing about the ESR receiver, our Fig. 9, in the bottom-right panel a dashed line is shown, with the label "post-detection filtering matched to final sampling rate". That line is drawn with the same periodicity as the signal spectrum (50 KHz in the figure). The idea presumably was to show that this is the relevant band shape of the receiver, periodically extended due to the sampling. This idea does not work.

The basic problem is that as soon as there is sampling rate changes between input and output, the relation Eq. (15) no more holds as a general rule. Here already the primary sampling, by the A/D, counts as kind of sampling rate conversion. With the notations of Fig. 6, the analog input spectrum to the A/D is of the form

$$X^a(f) = H^A(f)X^A(f)\,,$$

where $H^A(f)$ is the transfer function of the analog filter in front of the A/D, and X^A is the input signal spectrum to that filter. Then according to the sampling theorem, the spectrum of the sampled signal x^s_n is

$$X^s(f) = \sum_{m=-\infty}^{\infty} H^A(f - mf_s)X^A(f - mf_s)\,. \tag{57}$$

For the multirate system consisting of the analog receiver and the A/D, no simple multiplicative filter-like relation, like Eq. (56), between the input spectrum $|X^A(f)|^2$ and the output spectrum $|X^s(f)|^2$ no more exists. This violates the central idea of the concept of a "band shape", which is that the input spectrum can be simply multiplied by the band shape to get the output spectrum.

There is a special case when one actually can salvage a filter-like relation even for a multirate system. This is the case of a strongly band limited system. By that I mean a system where the input signal and the filter both are so narrow (and, in case of real band-pass signals, suitably placed) that there is practically no spectral overlap between any two Nyquist components for different zone index m. In particular, we then have

$$H^A(f - m_1f_s)X^A(f - m_2f_s) = 0 \text{ for all } f \text{ if } m_1 \neq m_2\,. \tag{58}$$

In that case, the r.h.s. of Eq. (57) factorises to the form

$$X^s(f) = \left[\sum_m H^A(f - mf_s)\right] \cdot \left[\sum_m X^A(f - mf_s)\right]\,, \tag{59}$$

where the first term in the r.h.s. is the periodic extension of the analog filter and the second term is the periodic extension of the analog signal. In addition, because also the shifted replicas $H^A(f - mf_s)$ for different m and the replicas $X^A(f - mf_s)$ for different m do not overlap either, in each Nyquist zone only a single term in both square brackets in Eq. (59) contributes.

That was approximately the situation one had with the original EISCAT receiver of the 1980s, where downconversion, detection and post detection filtering were done in analog domain, and the sampling, directly on the baseband, was the last step in the processing. In that old system, under the assumption of strong bandlimitness, it would make good sense to draw a receiver diagram like in Fig. 9. The dashed line there would simply be the squared magnitude of the first term in the r.h.s. of Eq. (59). On the other hand, even the old EISCAT receiver was typically not that strongly band limited, and Eq. (59) was only approximatively true. Indeed, when one talked about "band shape" with that system, one just quoted the power response of the analog post detection filter, using as much of the frequency axis as was needed, irrespective of what the sampling rate was.

With the new digital receiver, and especially with the new wide-band decimating filters, the non-factorization of the input-output relation can become quantitatively quite bad, and the periodic-by-final-frequency band shape curves can become meaningless. What one should show as a band shape of a digital receiver channel is not some periodic-by-f_s' curve, but just the power response $|H_{\mathrm{DDS}}(f)|^2$ of the equivalent filter (which is periodic by the primary sampling rate). Irrespective of what the final sampling rate f_s' of the channel is, enough of the frequency axis inside the primary baseband should be used to show the whole support region of that function.

We will now calculate the final spectrum in terms of the analog input spectrum, and see under what condition we can still recover a filter-like input-ouput relation. We handle here only narrow-band signals like sinusoids, or the IS ion-line signal[15]; wide-band noise is handled in Chapter 6.

In the EISCAT receiver, the primary sampling rate f_s is carefully selected with respect to the width and location of the analog passband, to ensure that the Nyquist replicas $X^A(f - mf_s)$ do not overlap after the primary sampling. And (as seen in Fig. 8), neither do the components of the anti-aliasing passband filter H^A, which defines the passband, overlap. Therefore, for the primary samples, the factorisation of Eq. (59) holds, and on the primary baseband, one has the simple multiplicative relation

$$X^s(f) = H^A(f)X^A(f), \text{ for } |f| < f_s/2\,. \tag{60}$$

Here, both $H^A(f)$ and $X^A(f)$ are to be understood as being an appropriate replica of the actual analog bandpass spectrum, deduced from Fig. 8. For narrow-band signals, one might take $H^A(f)$ to be constant over the support of $X^A(f)$. Thus the spectrum $X^s(f)$ that is the starting point in the digital processing proper, essentially is just the periodic extension of the analog spectrum $X^A(f)$, with period f_s, as sketched in Fig. 8.

The spectrum is then frequency translated in the NCOM, to $X(f) = X^s(f - f_{\mathrm{nco}})$, and then processed in the channel's equivalent decimating filter DDF, consisting of the post detection FIR filter $\mathrm{H_{DDF}}$ and the decimator M. The result is the final spectrum $Z(f)$.

[15] Strictly speaking, the IS signal is modelled as a random process, for which the spectrum $|Z(f)|^2$ does not exist. One has to use the power spectral density $G^z(f)$ instead, but such a thing we will meet only later, in Section **??** of this article. Nothing in the present discussion depends on this technicality.

From Eq. (15) and Eq. (11), we get the relation

$$\begin{aligned} Z(f) &= Z_1(f) + Z_2(f) \\ &= \sum_{m=0}^{M-1} H_{\mathrm{DDF}}(f - m f_s') X_1(f - m f_s') + \sum_{m=0}^{M-1} H_{\mathrm{DDF}}(f - m f_s') X_2(f - m f_s') \end{aligned} \quad (61)$$

between the input and the output of the DDF. We have explicitly denoted by X_1 and X_2 the spectral components deriving by a suitable frequency shift—different for X_1 and X_2—from the negative-frequency component and the positive-frequency component of the analog spectrum X^A, as in Fig. 8. Equation (61) is the complete recipe for predicting the observable spectrum, given the channel's equivalent filter $H_{\mathrm{DDF}}(f)$, total decimation M, and the analog input spectrum $X^A(f)$. In most cases, the positive frequency part Z_2 will be insignificant compared to the negative frequency part, and then one has the relation

$$Z(f) = Z_1(f) = \sum_{m=0}^{M-1} H_{\mathrm{DDF}}(f - m f_s') X_1(f - m f_s') \,. \quad (62)$$

The case when Z_2 is *not* insignificant results in the quip disturbance, and is handled in Chapter 5 (Eq. (74), etc.).

In the case of a strongly band limited system, where the support of both the $\mathrm{H}_{\mathrm{DDF}}$ and the signal in the primary baseband is essentially restricted to the narrower final baseband, the r.h.s. of Eq. (62) factorizes as

$$Z(f) = \left[\sum_{m=0}^{M-1} H_{\mathrm{DDF}}(f - m f_s') \right] \cdot \left[\sum_{m=0}^{M-1} X_1(f - m f_s') \right] . \quad (63)$$

In this case, in the final baseband, one has the relation

$$Z(f) = H_{\mathrm{DDF}}(f) \cdot X_1^A(f - \Delta_1) \quad \text{,when } |f| < f_s'/2 \,, \quad (64)$$

where the shift Δ_1 from the analog passband to the final baseband is computed according to Algorithm 1 on p. 16.

The finite periodic summation of the filter transfer function in Eq. (63) corresponds to the filter $\mathrm{H}' = \mathbf{M}^{\downarrow}\mathrm{H}_{\mathrm{DDF}}$ of Eq. (18), the one I vaguely had in mind when preparing the drawing for the [W97] paper. We have now seen that in order to there be any sense in using such an H′ as a basis for the dashed band-shape line in [W97], the channel's input-output relation Eq. (62) must factorise. A sufficient condition is that $\mathrm{H}_{\mathrm{DDF}}$ and X are strongly band limited to the final baseband. This situation is illustrated in Fig. 10.

For the factorisation, it is not necessary that both H and X are strongly bandlimited. It is only necessary that the terms $H(\ldots m_1) \cdot X(\ldots m_2)$ resulting from expanding the product in Eq. (63) are identically zero when $m_1 \neq m_2$. Thus, one can also consider the situation where only X is strongly bandlimited. One could then formally restrict H_{DDF} to the support region of X, and use that restricted function in place of $\mathrm{H}_{\mathrm{DDF}}$ in Eq. (63). This situation is sketched to Fig. 11. One could even argue that this case should be a common one in practice, for in a well-defined incoherent scatter experiment, if not the $H_{\mathrm{DDF}}(f)$, then at least the incoherent scatter spectrum itself should be strongly band-limited, otherwise the IS signal is not correctly sampled. In fact, because of the possibility of Doppler-shift, the final baseband should be considerably wider than the IS spectrum.

Some of the filters in the EISCAT filter collection are rather strongly band limited, (Fig. 28, Fig. 31), some others less so (Fig. 29), and some are not band-limited at all (have a constant power response), the filter b2300d3 in Fig. 32 is one of those.[16]

It should be also noted that even if the input signal itself were strongly band-limited, the input noise never is (this case is handled in Chapter 6). With such wide-band input and a wide filter, the channel's input output relation no more factors as in Eq. (63). In that situation, illustrated in Fig. 12, the power response of the decimated filter H' does not provide a good measure of the band shape as far as noise is concerned. A good measure of "band shape" does exist for the noise case, but it is given by the quantity $\sum_m |H_{\mathrm{DDF}}(f - m f_s')|^2$ (which I call the "noise response") in Eq. (104) in Chapter 6, rather than the power response $|\sum_m H_{\mathrm{DDF}}(f - m f_s')|^2$.

Summarising our discussion about the band shape, in many cases it is possible to argue that the dashed line in the [W97] receiver diagram represents some kind of band shape. However, even if it possible to defend the dashed line, there is no point in actually doing so. It is better to ditch that "filter-shape" line from the figure altogether. There is too much pain for too little gain. In a multi-rate digital receiver, the channel's equivalent filter (the PDF block of Fig. 6) lives naturally in the primary baseband. When the filter shape is drawn, it should not be restricted to the final baseband only, but should be shown everywhere in its support interval in the primary baseband. That is what the ddfplan.m does. This is analogous to the practice of showing analog filter shapes in the whole region where they are essentially non-zero.

In fact, it can be outright dangerous to show $H(f)$ only within the final baseband; there may be significant parts of it outside the final baseband. One way to make a drawing which shows simultaneously the filter shape and the final signal spectrum, is to simply plot both $|Z(f)|^2$ and $|H(f)|^2$ in a frequency interval around zero-frequency in the primary baseband that is wide enough to cover all the frequencies where at least one of those functions is non-zero to a significant degree. In addition, it might be useful also to plot the filter's noise response (using Eq. (103)), which shows how much noise aliasing to the final baseband takes place.

For many purposes, we are interested in the spectra of the measured data in the final baseband, like the quantity $|Z(f)|^2$, rather than the filter *per se*. We have well-defined ways, such as Eq. (62) and Eq. (103), to compute those spectra. Those formulas incorporate the effects of both the filtering and the decimation, and should be built-in directly into the data analysis machinery, instead of the analysis trying to emulate pre-historic single-rate signal processing.

Maybe this has now done in the EISCAT analysis, I don't know. But I do know that it was not done initially after ESR started operating. At that time, the GUISDAP analysis got its information about the channel filtering via my Matlab routine get_impresp.m (Listing 3). That routine returns not the impulse response taps, but an "chip"-interpolated continuous-time approximation of the impulse response.[17] That continuous-time impulse response is used to compute weighting functions for the lagged products of samples. This cannot be the correct way to do things in principle. Instead, one should assign

[16] Most of the filters currently in the collection are of the type of the filter b25d300, the ion-line channel filter in beata, Fig. 29, where the 3 dB point is very near to half of the final sampling rate f_s'. But more recently, IH has started using filters that are more strongly band-limited, like the plasma-line channel filter b800d6 in beata, Fig. 31.

[17] I have more recently augmented the get_impresp.m script so that it returns also the impulse response taps, as an extra parameter.

the weights for the primary 15 MHz samples, using the impulse response of the whole equivalent filter, and then trace the ambiguity functions through the filtering and decimation steps, like we have traced the spectra here.

We have now handled enough theory of the EISCAT digital receiver to allow us in the next Chapter to model quantitatively the data processing in the channel configurations such as b2300d3 that gave the curiously oscillating output, shown in Figures 3 and 4.

5 Modelling the disturbance in the quip experiment

The decimation 3 boxcar-in-time filter b2300d3 was designed with the ddfplan program. The program's standard graphical output is shown in Figures 16 and 17. Fig. 16 shows the impulse response h_{CIC} and power response $|H_{\mathrm{CIC}}(f)|^2$ of the hight order decimation filter; the impulse response h_{FDF} and the power response $|H_{\mathrm{FDF}}(f)|^2$ of the fir decimation filter; and the impulse response h_{DDF} and power response $|H_{\mathrm{DDF}}(f)|^2$ of the combined equivalent filter. The impulse response h_{DDF} is computed from h_{CIC} and h_{FDF} using Eq. (34). The ddfplan program does not use the explicit formulas like Eq. (52) for the transfer functions. Instead, it computes them from the impulse responses, by evaluating the definition Eq. (16) with a sufficiently long FFT.[18]

As seen from Fig. 16, the FDF filter has only a single non-zero tap, and two zero taps on its side. This is a trick to, in essence, to put the FDF in all-pass mode. The HSP43220 data sheet says that it is possible to put both the HDF and the FDF to bypass mode, together or separately, but as far as I know, we have never been able to get this feature working in EISCAT. Adding the M_1 zeros (now two of them) between the taps of the FDF, required by Eq. (34), does not change anything essential in this case; there is only one non-zero tap in the FDF. But in general, the convolutions $h_{\mathrm{CIC}} * h_{\mathrm{FDF}}$ and $h_{\mathrm{CIC}} * h_{\mathrm{FDF}}^{(M_1)}$ can be very different. One needs to keep this in mind e.g. when figuring out with pen and paper what kind of impulse response to expect for the equivalent filter, given h_{CIC} and h_{FDF}.

With the FDF in all-pass state, the total impulse response is essentially equal to h_{CIC}, a boxcar filter with three taps. Thus the filtering operation to be applied to the 15 MHz frequency-shifted sample stream is just the uniformly weighted moving average of three samples. Decimation by three, to the final 5 MHz rate, means that the final output is computed from the 15 MHz stream as the sum of the samples in adjacent blocks of three samples. With hindsight, this arrangement cannot be expected to provide an awful lot of averaging, that is, high-frequency suppression. But at the time when I designed the filter, I did not not notice the writing on the wall. I was happy to have found a way to tweak ddfplan so that I could produce one of those boxcar filters that my colleague Markku Lehtinen has been favouring. Normally, the EISCAT filters are not of boxcar form, but are more or less "triangular" or "gaussian" in shape in the time domain. The boxcar filters have the nice property that their output noise is white if the input noise is white (see Section **??**). This makes error estimation more straightforward. On the other hand, they are very far from the strongly band limited type of filters in the sense of Chapter 4.

[18] The FFT is a numerically very fast way to evaluate Fourier transforms such as the r.h.s. of Eq. (16) at the set of uniformly spaced frequencies $f_n = n f_s/N$, $n = 0 \ldots N-1$. N is the "length" of the FFT, and must, traditionally, be a power of two. If N is larger than the number of coefficients h_k, one adds a sufficient number of zeros to the end of the sequence h_k to satisfy the needs of the algorithm.

The bottom right panel in Fig. 7 gives the total power response $|H_{\mathrm{DDF}}(f)|^2$ of the filter, though only on the positive half of the primary baseband, from 0 to 7.5 MHz. The negative part needs not be drawn. Because the filter impulse response h_{DDF} is real-valued—that is, the "real part" and "imaginary part" of the signal path in the digital receiver have identical filters as indicated in Fig. 7—the transfer function has hermitian symmetry, and so the power response is symmetric around origin. It is noted that there is a null in the response curve at the final sampling frequency 5 MHz, but that there is not huge attenuation on either side of the null. However, Fig. 7 is logarithmic plot, and it is difficult to get proper feeling of the attenuation from it. Therefore, the second standard output plot of ddfplan, Fig. 17, shows the filter output also in linear scale. Unfortunately the standard plot only shows the filter in the final baseband. This needs to be changed in future versions of ddfplan; a somewhat wider region must be shown.

Another somewhat dubious feature in the standard plots—which might or might not need to be changed—is the way the total equivalent impulse response h_{DDF} in the top panel of Fig. 17, and similar, is drawn. The digital impulse response in reality is a sequence of taps h_n. What is plotted in the panel "DDF Impulse Response", is a linearly interpolated line through the points $(n\tau_s, h_n)$, where τ_s is the primary sampling interval $1/f_s$. This is probably alright when there are lot of the coefficients h_n, but the presentation becomes increasingly meaningless when there are only a few of them. It is probably better to just show the points $(n\tau_s, h_n)$, without connecting them, then.[19]

Having found the impulse response of the filter b2300d3—boxcar with three taps followed by decimation by three—probably the simplest way to check what to expect in the quip experiment, is by direct numerical simulation of the expected output in the time domain; and that is what I initially did after the experiment.

Referring to Fig. 6 for notation, one generates samples x_n^s, to represent the primary 15 MHz transmission samples, fresh out of the A/D. Phase coding was observed *not* to relate to the oscillations, so we need not simulate the phase coding, but can just generate real-valued sinusoidal samples

$$x_n^s = 2\cos(2\pi f_{\mathrm{tx}}\tau_s n) = 1\times \mathrm{e}^{i2\pi(-f_{\mathrm{tx}})\tau_s n} + 1\times \mathrm{e}^{i2\pi(+f_{\mathrm{tx}})\tau_s n} \tag{65}$$

at the quip IF2 frequency $f_{\mathrm{tx}} = 10.1$ MHz. The factor two in front of the cosine ensures that we will have unit amplitude both for the positive and the negative frequency part of the spectrum $X^s(f)$ in Eq. (68).

Next, one shifts the samples x_n^s in frequency by multiplying them with a complex sinusoid using f_{nco} equal to f_{tx} in Eq. (12). This results in samples x_n. Then one filters the x_n by taking convolution with the filter coefficients h_k,

$$y_n = \sum_k h_k x_{n-k}\,. \tag{66}$$

Finally, one decimates by M (=3, now) to get samples at the final sampling frequency,

$$z_n = y_{Mn}. \tag{67}$$

For comparison to Fig. 3 of measured data, one also wants the power spectrum in the final baseband, $|Z(f)|^2$. One way to get $Z(f)$ is to start from the spectrum $X^s(f)$ of

[19] The Matlab script get_impresp.m that is meant to extract the equivalent impulse response from EISCAT filter files, returns an "pchip" interpolated smooth curve to represent the impulse response.

the A/D output samples x_n^s. What the simulated spectrum $X^s(f)$ precisely looks like, depends on how many terms one uses in the definition Eq. (3). If we take that sum from $-\infty$ to ∞, the spectrum becomes a line-spectrum, that is, two periodic summations of delta functions, one summation containing $\delta(f - f_1^a)$, the other containing $\delta(f - f_2^a)$, with $f_1^a = -f_{\text{tx}}$ and $f_2^a = f_{\text{tx}}$. The periodic summation of frequency-dimensional delta functions is the Dirac comb Ш_{f_s},

$$\text{Ш}_{f_s}(f) \triangleq \sum_{m=-\infty}^{\infty} \delta(f - m f_s)\,. \tag{68}$$

We have

$$X^s(f) = \text{Ш}_{f_s}(f - f_1^a) + \text{Ш}_{f_s}(f - f_2^a)\,. \tag{69}$$

When there are only a finite number, N, of samples used for the spectrum, the delta peaks broaden to have finite width. In math terms, the two Dirac combs in Eq. (69) are replaced by two Dirichlet kernels $D_N()$, Eq. (51), multiplied by phase factors. These functions have main peaks at the positions $f_k^a + m f_s$. The more samples are used, the narrower the peaks and the lower the envelope of the side lobes associated with the peaks. The be exact, with N samples x_n^s in Eq. (65), $n = 0 \ldots N-1$, direct evaluation of the spectrum from its definition, gives

$$X^s(f) = \mathrm{e}^{i\pi\nu_1(N-1)} D_N(\pi \frac{f - f_1^a}{f_s}) + \mathrm{e}^{i\pi\nu_2(N-1)} D_N(\pi \frac{f - f_2^a}{f_s})\,. \tag{70}$$

For Eq. (70), we also normalised the spectrum by the number of samples, in order to have the main peak magnitudes equal to unity rather than N.

Given the explicit expression Eq. (70) for the spectrum of the A/D samples, and the filter transfer function $H(f)$ calculated from Eq. (16), the spectrum $Y(f)$ on the primary baseband could readily be calculated numerically. And then the spectrum at final baseband could be calculated from the decimation recipe Eq. (11).

However, when one anyway was going the calculate the samples z_n, the shorter way to get $Z(f)$—the one I actually used—was to compute the power spectrum $|Z(f)^2|$ directly from the definition Eq. (3),

$$Z(f) = \frac{1}{f_s} \sum_{n=0}^{N-1} z_n \mathrm{e}^{-i2\pi(f/f_s)n}\,. \tag{71}$$

This discrete time Fourier transform is efficiently computed using an FFT of sufficiently large length to yield a smooth curve for plotting.

Figure 18 shows the result of the simulation done for the EICAT UHF transmission frequency F13 which was used in the quip experiment. In the analog passband, at the A/D input, the frequency appears at 10.1 MHz. In addition of time series plots of $|z_n|^2$ (the top panel), the figure also shows the phase of the z_n, as well as the power spectrum $|Z(f)|^2$. Considering that the simulation is done in full double floating accuracy and assumes zero background noise, the agreement between data and simulation looks qualitatively fairly good. I think that this comparison alone is enough to make one believe that the b2300d3 filter works correctly (even though the hardware is driven beyond the official specs).

To make a quantitative comparison between the measured and observed beat behaviours, it is of interest to derive expressions for calculating the beat characteristics

(instead of just estimating them from the simulated data). This we will do in the next section. The section also serves as background for the program ddfsimu.m (Listing 2) that I have written to help to inspect especially the behaviour of wide-band filters, better than can be done with ddfplan.m. As a side-product, we will gain some insight into the aliasing behaviour of the sampled beat, which, like any other aliasing, could be quite confusing otherwise.

Modelling the beat

We assume a sinusoidal analogue input signal to the A/D, as in Eq. (65), but now not necessarily on the transmission frequency f_{tx}. Instead, we let it have any frequency f_{rx} in the analogue passband. The signal could be, for instance, a Doppler-shifted echo from a hard target, or could be an ionospheric plasma line echo. We assume that signal is recorded for a long time, so that the analog spectrum will essentially be two delta peaks. Thus, the analogue spectrum has narrow spectral components $f_1^a = -f_{\text{rx}}$ and $f_2^2 = f_{\text{rx}}$, of unit amplitude. After sampling, frequency-shifting in the NCO, and filtering in the digital filter, the spectrum $Y(f)$ restricted to the baseband is

$$Y(f) = H_1\delta(f - f_1) + H_2\delta(f - f_2)\,, \tag{72}$$

where

$$H_k \triangleq H_{\text{DDF}}(f_k)\,. \tag{73}$$

The frequencies f_k in the primary baseband are computed with Algorithm 1, which involves the NCO frequency f_{nco}. $H_{\text{DDF}}(f)$ is the filter's transfer function computed from Eq. (16). In decimation, $Y(f)$ is transformed to $Z(f)$ according to the decimation theorem Eq. (11), but now there is no spectral overlap in the $M-1$ time replication of Y, so the spectrum still has just the two delta-peaks in the final baseband,

$$Z(f) = H_1\delta(f - f_1') + H_2\delta(f - f_2') \quad \text{when } |f| < f_s'/2\,. \tag{74}$$

This is the portion of the Dirac comb, of the form of Eq. (69), which has support in the final baseband. The frequencies f_1' and f_2' in the final baseband can again be calculated with Algorithm 1.

The corresponding sample sequence z_n is inverted from $Z(f)$ using Eq. (4), and is, unsurprisingly, just two complex exponentials,

$$z_n = H_1 e^{i2\pi f_1'\tau_s' n} + H_2 e^{i2\pi f_2'\tau_s' n}\,, \tag{75}$$

where $\tau_s' = 1/f_s'$ is the final sampling interval. An aim in channel configuration is to make the magnitude of H_2 insignificant compared to H_1, and normally that is achieved. But with the filter b2300d3, the second term in Eq. (75) cannot be ignored, and that causes the beat.

In principle, the numbers z_n are all that one really needs deal with, but nevertheless, to inspect beat aliasing (see below), it is instructive to imagine that z_n are just a (correctly) sampled version of a continuous-time signal $z(t)$,

$$z(t) = H_1 e^{i2\pi f_1' t} + H_2 e^{i2\pi f_2' t}\,. \tag{76}$$

The signal $z(t)$ is what we would presumably get if the receiver would be doing all of the processing in the analogue domain, as in the EISCAT receiver of the old, where sampling

was done in the very end, just before the buffer memory (which was a remarkably big box).

From Eq. (75), one can readily derive expressions for the signal power $|z_n|^2$ and the phase ϕ of z_n. When $f_{\rm rx}$ is not equal to $f_{\rm nco}$, there is a linear phase variation in z, corresponding to the "Doppler-shift $f_{\rm D} = f_{\rm rx} - f_{\rm tx}$, which one wants to remove in plotting to make the variation due to the beat better visible. Note that because the receiver is turned on the transmission frequency, and the frequency shift by the NCO is done so as to move the negative frequency component of transmission to zero frequency, $f_{\rm D}$ is actually equal to the frequency f_1' (less than zero when $f_{\rm rx}$ is less than $f_{\rm tx}$). In the simulation, we consider $f_{\rm rx}$ as a known parameter (in actual experiments, it is of course a parameter to be determined from data), so to remove the Doppler, one just multiplies $z(t)$ and z_n by $\mathrm{e}^{-i2\pi f_{\rm D} t}$ and $\mathrm{e}^{-i2\pi f_{\rm D}\tau_s' n}$, respectively.

From Eq. (75), the signal power is found to be

$$|z_n|^2 = |H_1|^2 + |H_2|^2 + 2a\cos(2\pi\,\Delta\nu\, n + \alpha)\,. \tag{77}$$

where

$$a = |H_1 H_2|\,, \tag{78}$$

$$\alpha = \arg\left(\overline{H_1} H2\right), \tag{79}$$

and

$$\Delta\nu = \frac{f_2' - f_1'}{f_s'}\,. \tag{80}$$

Eq. (77) implies that the mean value of the power is $P = |H_1|^2 + |H_2|^2$ and that the power beat is strictly sinusoidal. Thus the relative peak to peak power variation $dP/P \triangleq 4a/P$ is given by

$$dP/P = 4 \times \frac{|H_1||H_2|}{|H_1|^2 + |H_2|^2}\,. \tag{81}$$

The beat frequency $f_{\rm beat}$ is equal to the frequency difference of the two spectral lines in the final baseband, and the beat period $T_{\rm beat}$ is its inverse,

$$T_{\rm beat} = \frac{1}{|f_2' - f_1'|}\,. \tag{82}$$

After removing the Doppler variation from $z(t)$ to inspect the residual phase variation, we are left with time development

$$w(t) = H_1 + H_2 \mathrm{e}^{i2\pi(f_2' - f_1')t} \tag{83}$$

and the corresponding sampled version $w_n = w(n/f_s')$. The complex numbers $w(t)$ trace a circle which has radius $|H_2|$ and centre at the complex number H_1, as sketched in Fig. 19. The minimum and maximum phase angle of $w(t)$ is found from Fig. 19 by drawing tangents from the coordinate origin to the circle. This gives the peak to peak phase variation as

$$\mathrm{P2P}_{\Delta\phi} = 2\arcsin\frac{|H_2|}{|H_1|}\,. \tag{84}$$

From Eq. (83) and Eq. (82), the phase variation is periodic with the same period $T_{\rm beat}$ as the signal power, but is no more strictly sinusoidal in time.

When applied to the situation of Fig. 3, where the frequencies in the final 5 MHz wide baseband are $f_1' = 0$ and $f_2' = 0.200$ MHz, $|H_1| = H(0) = 1$ and

$$|H_2| = |H(10.1 + 10.1 - 15)| = |H(5.2)| = 0.04714\,.$$

these formula predict beat period 5.000 μs, relative P2P power variation $dP/P = 0.1881$, peak to peak phase variation in the beat $\mathrm{P2P}_{\Delta\phi}$=5.404°, and the magnitude difference of the two peaks in the power spectrum $20 \times \log 10(|H_2|)$ = -26.52 dB. Considering how noisy the data is, these predictions appear consistent with the data. The observed spectral peak is 2.3 dB higher than predicted, but in other, less-noisy, cases, like the one shown in Fig. 3 (data) and Fig. 23 (simulation), the agreement is better.

These formulas are incorporated into the ddfsimu.m script (Listing 2) that provides graphics about the location and strength of the two spectral peaks in the various stages of processing in the digital channel, and also produces simulated time series data and spectral plot (the simulated samples z_n are computed directly from Eq. (77)). The filter $|H(f)|$ is plotted for the whole primary baseband, and shading is used to indicate the final baseband. Ddfsimu plots for comparison to data in Fig. 3 are shown as Fig. 20 and Fig. 21. The corresponding plots for comparison to Fig. 4 are shown as Fig. 22 and Fig. 23.

Beat aliasing

With f_1' and f_2' in Eq. (83) in the interval $[\frac{-f_s'}{2}, \frac{f_s'}{2}]$, their difference is in the interval $[-f_s', f_s']$, twice as large. The implication is that if we consider the samples w_n to be samples of a continuous-time signal $w(t)$ (similarly, with the samples z_n), the signal itself can be "correctly sampled" in the sense that its frequency contents is in the Nyquist interval, but the beat part in Eq. (83), might not be. The resulting aliasing can lead to surprising changes in the apparent periodicity of the plotted sequences $|z_n|^2$ and $\arg z_n$ when the difference $f_2' - f_1'$ is varied. For instance, with transmission and NCO at 10.1 MHz, both Doppler-shifted reception at 10.4 MHz (Fig. 24) and at 12.1 MHz (Fig. 25) produce the same apparent beat period of 1.25 μs of z_n. Nevertheless, the signal spectrum correctly identifies the contributing frequencies in both cases.

If there is any lesson here, it could be that just looking at the apparent periodicities in the sampled signal "by the eye" is not a good way to infer the actual spectral contents. We were lucky initially when we first encountered the mysterious oscillations, in that the actual situation was more like in Fig. 24 than in Fig. 25. Aliasing is always a danger when using sampled data, but here it can apparently pop up in a particularly nasty way: the signal is correctly sampled, but one can still become confused by aliasing.

6 Noise in the digital receiver

In the previous sections, we have seen how a sinusoidal signal, and in general, a narrow-band signal, is processed in the EISCAT digital receiver. Even though not very relevant from the point of view of understanding our original problem, the beat oscillations, we inspect here how wide-band noise that always is present in the analog input, gets transformed in the digital receiver. This helps us make sense of the commonly displayed EISCAT real-time plots which show spectra over the final baseband, computed from the whole incoming stuff, consisting both the narrow-band signal and the wide-band noise.

The main purpose here is to show how the "background spectrum" $G^\zeta(f)$—spectrum of the final baseband noise ζ_n in the absence of any signal—is related to the transfer function $H(f)$ of the digital filter. We will see that for many filters used nowadays in standard EISCAT experiments, the spectrum is not very different from the squared magnitude of the transfer function, restricted to the final baseband. However, in general, this can be a bad approximation, due to noise aliasing into the final baseband from the primary baseband. For instance, for the very wide filter b2300d3, $G^\zeta(f)$ is constant as function of f, provided that the filter's input noise, χ_n in Fig. 6, is white noise.

We begin by reviewing some aspects of the autocorrelation function based machinery that is needed to handle discrete-time noise.

For notation, refer to Fig. 6. We denote by χ_n the noise samples in front of the FDF. We denote by γ_n the samples out of the FDF but before the decimator. We denote by ζ_n the noise samples after the decimator. The A/D sampling rate is f_s, 15 MHz now. The digital filter has impulse response h_n and transfer function $H(f)$, the latter defined over the 15 MHz primary baseband $[-\frac{f_s}{2}, \frac{f_s}{2}]$. The decimator reduces the sampling rate by a factor M, to $f_s' = f_s/M$. We call the interval $[-\frac{f_s'}{2}, \frac{f_s'}{2}]$ the final baseband.

The noise is modelled as a random sequence, which is the discrete-time analogue of the random process. Thus, the noise samples are considered to be random variables. The basic tool of the noise analysis is the autocorrelation sequence g_l, formed from the expectation values of lagged products of the samples. For example, for the final baseband noise sequence ζ_n, the autocorrelation sequence g_l^ζ is

$$g_l^\zeta \triangleq \mathbb{E}\,\overline{\zeta_n}\zeta_{n+l}\,, \qquad \text{independent of n.} \tag{85}$$

We assume here zero-mean sequences; otherwise, one should subtract from ζ its mean value $\mathbb{E}\,\zeta$. Also, we do not divide by variance, so our "autocorrelation" should perhaps more precisely be called the autocovariance. But the definitions vary anyway. An equivalent way to write the definition (one used e.g. in Priestley[20]), is

$$g_l^\zeta = \mathbb{E}\,\zeta_n\overline{\zeta}_{n-l}\,.$$

As indicated, we assumed that the sequences are stationary, so that the expectation value of the lagged product only depends on the lag l, but not on the time n. The noise is said to be white if the autocorrelation is non-zero only at lag $l = 0$; that is, if distinct samples are uncorrelated. Autocorrelation zero-lag is the expected noise power P,

$$g_0^\zeta = \mathbb{E}\,|\zeta_n|^2 \triangleq P^\zeta \quad \text{, for all n}\,. \tag{86}$$

We define the noise spectrum $G(f)$ is the discrete-time Fourier transform of the autocorrelation sequence g_l, divided by the corresponding sampling rate. For example, for the final noise sequence ζ_n, the spectrum is

$$G^\zeta(f) \triangleq \frac{1}{f_s'}\sum_l g_l^\zeta \mathrm{e}^{-i2\pi(f/f_s')l}\,. \tag{87}$$

The spectrum is periodic by the appropriate sampling frequency, f_s' in the case of G^ζ. Analogously to Eq. (3), we have included normalisation by the sampling rate into the

[20] M. B. Priestley, Spectral Analysis and Time Series, Academic Press, 1981.

definition (but not into the notation $G^\zeta(f)$). With this normalisation, the integral of $G(f)$ over a Nyquist interval gives the power,

$$\int_0^{f_s'} G^\zeta(f)df = P^\zeta \,. \tag{88}$$

Therefore, the "noise spectrum" $G(f)$ more precisely is called the noise power spectral density. It has the unit V^2/Hz, just as is the case with continuous-time noise.[21] The inverse relation to Eq. (87) is

$$g_l^\zeta = \int_0^{f_s'} G^\zeta(f)\mathrm{e}^{i2\pi(f/f_s')l}df \,. \tag{89}$$

For white noise, it follows from Eq. (87) and Eq. (86) that the power spectral density is constant,

$$G(f) = \frac{P}{f_s} \,, \quad \text{for white noise.} \tag{90}$$

The converse is also true. When $G(f) = \text{const}$, the integral in Eq. (89) becomes equal to $f_s\delta_{l,0}$. Therefore, if the spectral density is constant, the autocorrelation has only its zero lag nonzero, and thus the noise is white.

In what follows, we will for the most part assume that the complex-valued noise χ_n into the channel's decimating filter is white, so that its autocorrelation sequence has only the zero lag non-zero, and the power spectral density is constant,

$$G^\chi(f) = \frac{P^\chi}{f_s} \,. \tag{91}$$

This should be a fairly good approximation in the EISCAT receiver. Even though the anti-aliasing analog bandpass filter is only slightly over 7 MHz wide, the analogue signal is a real-valued signal, and therefore has total spectral coverage of twice that much. Due to the carefully selected IF passband locations in the bandpass sampling, after the sampling, the frequency axis is covered quite uniformly by copies of the passband, as indicated schematically in the top panel of Fig. 8. This implies that provided that the noise into the anti-aliasing filter is of fairly uniform power spectral density over the passband, and the anti-aliasing filter itself is reasonably flat, then also the sequence of the primary noise samples χ_n^s, just after the A/D, has a fairly uniform spectral density and therefore, is reasonably white.

I will subsequently simply assume that the noise after the A/D is white, but for completeness, I will briefly outline, without proofs, the steps that are needed to be able to actually inspect the whiteness question quantitatively.

A sensible starting point is be to assume that the noise, call it $\chi^A(t)$, in front of the receiver's anti-aliasing filter is stationary white noise. In the analog domain, noise is described using a continuous-time version $g(t)$ of the autocorrelation Eq. (85), so that e.g. for the noise $\chi^A(t)$,

$$g^{\chi^A}(t) \triangleq \mathbb{E}\,\overline{\chi^A(t')}\chi^A(t'+t) \,. \tag{92}$$

For stationary noise, the time instant t' in Eq. (92) does not matter. In the EISCAT system, we would not need the complex conjugation here, either, for the noise $\chi^A(t)$ is real-valued; but it doesn't hurt.

[21] In the EISCAT real-time plots, the unit is taken to be K/Hz.

The continuous-time white noise is characterised as having infinite power and zero correlation length. This state of affairs can be expressed using the Dirac delta-function as

$$g^{\chi^A}(t) = \sigma^A \delta(t)\,. \tag{93}$$

Because the delta function has dimension of $1/\mathrm{s} = \mathrm{Hz}$, the constant of proportionality σ^A has dimension of V^2/Hz. σ^A is the power spectral density of the white noise (see Eq. (95)). Typically, it is quantified via the system noise temperature T_{sys}, as

$$\sigma^A = \mathrm{gain} \times \mathrm{k_B T_{sys}}\,. \tag{94}$$

The constant of proportionality is an overall receiver "gain factor", which one normally needs not worry about. Its actual value will cancel out from the final formulas where it enters, typically, because those formulas involve only power measured relative to a calibration power.

The continuous-time noise spectrum $G^{\chi}(f)$ is defined as the Fourier transform of the autocorrelation function $g^{\chi}(t)$, so that for the white noise $\chi^A(t)$ of Eq. (93), one has

$$G^{\chi^A}(f) \triangleq \int_{-\infty}^{\infty} g^{\chi^A}(t) \mathrm{e}^{-i2\pi f t}\, dt = \sigma^A. \tag{95}$$

The analog anti-aliasing filter is described by a transfer function $H^A(f)$. In general, an analog filter with transfer function $H(f)$ changes the power spectrum via multiplication by $|H(f)|^2$, a recipe formally identical to Eq. (102), below. Thus the spectrum of the noise $\chi^a(t)$ after the filter, just in front of the A/D converter, is

$$G^{\chi^a}(f) = |H^A(f)|^2 G^{\chi^A}(f) = \sigma^A\, |H^A(f)|^2\,. \tag{96}$$

In sampling, with sampling frequency f_s, the analog noise spectrum $G^{\chi^a}(f)$ becomes periodic with period f_s, via a periodic summation similar to the summation in the sampling theorem, Eq. (1).

$$G^{\chi^s}(f) = \sum_{m=-\infty}^{\infty} G^{\chi^a}(f - m f_s)\,. \tag{97}$$

If the input is white noise, Eq. (97) becomes

$$G^{\chi^s}(f) = \sigma^A \sum_{m=-\infty}^{\infty} |H^A(f - m f_s)|^2\,. \tag{98}$$

Eq. (97) gives quantitatively the sought-for noise spectrum at the start point of the digital processing. One might call Eq. (97) the sampling theorem for noise.

It may be noted that for Eq. (97) to be true, $\chi^a(t)$ does not really need to be "noise", but can be any stationary signal for which an autocorrelation function can be defined as in Eq. (92). Thus $\chi^a(t)$ could also be the incoherent scatter signal in the case when no coding is used in transmission. (With coding, things become more complicated, a case I will not attempt to cover here.) If the IS signal is band-limited to the primary baseband—and it surely should be, for it should be band-limited even to the final baseband—then the replicas $G^{\chi^a}(f - m f_s)$ when $m \neq 0$ to not contribute anything to the baseband $m = 0$, and the discrete-time IS spectrum is simply the same as the analog spectrum.

The power response $|H^A(f)|^2$ of the anti-aliasing filter is a known—or at least knowable—quantity in a well-defined system such as EISCAT. Thus Eq. (97) can be used to inspect

how white the spectrum $G^{\chi^s}(f)$ actually is. Here we will take it to be is precisely constant everywhere,

$$G^{\chi^s}(f) = \frac{P^{\chi^s}}{f_s} \quad (\propto \sigma^A)\,. \tag{99}$$

We denoted by P^{χ^s} is the noise power $\mathbb{E}\,(\chi_n^s)^2$ after the A/D.

With the noise χ_n^s white, we do not need in our analysis pay much attention to the complex multiplication in the NCOM module of Fig. 6. White noise does not change in any essential way in the frequency shifting. To verify this in the time domain, we note that with

$$\mathbb{E}\,\chi_n^s \chi_{n+l}^s = \delta_{l,0} P^{\chi^s}\,,$$

after the frequency-shifting multiplication of Eq. (12) we have

$$\mathbb{E}\,\overline{\chi_n}\chi_{n+l} = \mathrm{e}^{i2\pi\nu_{\mathrm{nco}}l} \times \mathbb{E}\,\chi_n^s \chi_{n+l}^s = \delta_{l,0} P^{\chi^s}\,. \tag{100}$$

Thus also the frequency-shifted noise χ_n, which is the input for the decimating filter, is white and has power equal to P^{χ^s}.

We will now inspect how the noise spectrum transforms in the decimating digital filter, first in the filter h_n which converts χ_n to γ_n, and then in the decimator M which converts γ_n to ζ_n. In the filtering by $H(f)$,

$$G^{\gamma}(f) = |H(f)|^2 G^{\chi}(f)\,, \tag{101}$$

and in decimation by M,

$$G^{\zeta}(f) = \sum_{m=0}^{M-1} G^{\gamma}(f + m f_s')\,. \tag{102}$$

The spectra G^{χ} and G^{γ} are periodic by the primary sampling frequency f_s, while the final spectrum G^{ζ} is periodic by the final sampling frequency f_s'. We will prove Eq. (101) and Eq. (102) at the end of this chapter. Combining the two equations gives the main result of this chapter

$$G^{\zeta}(f) = \sum_{m=0}^{M-1} |H(f + m f_s')|^2 G^{\chi}(f + m f_s')\,. \tag{103}$$

Equation (103) shows how an input power spectral density $G^{\chi}(f)$ transforms in a decimating digital filter with transfer function $H(f)$, decimation M and final sampling rate f_s'. For white noise input, $G^{\chi}(f)$ is constant, and Eq. (103) simplifies to

$$G^{\zeta}(f) = \sigma^{\chi} \sum_{m=0}^{M-1} |H(f + m f_s')|^2\,, \tag{104}$$

where

$$\sigma^{\chi} \triangleq G^{\chi}(f) = P^{\chi}/f_s \quad \text{, for all } f\,,$$

is the power spectral density at the filter's input.

Eq. (104) allows one to predict the shape of the "background spectrum". The shape of the background spectrum can differ drastically from the shape of the power spectrum $|H(f)|^2$ that is relevant for narrow-band signals. For example, $G^{\zeta}(f)$ is constant for the

decimating-by-M filters like the b2300d3 which have constant impulse response of M taps. That this particular type of decimating filter maps white noise to white noise can be argued without any computation. In these devices, an output sample ζ_n is just the sum of M input samples χ_n in such a way that each consecutive group of M input samples produces a new output samples. Therefore the blocks of input samples used for distinct output samples have no common samples. For white noise samples, that implies that also the output samples are uncorrelated, and hence have a constant power spectral density.

The filters with boxcar impulse response have $|H(f)|^2$ of the general form of Dirichlet kernel $D_M(\pi f/f_s')$, definitely not a constant. Eq. (104) shows how the shape change happens in decimation: spectral power is aliased to position f in the final baseband from all the positions $f + m f_s'$ in the primary baseband where the filter $H(f)$ is nonzero.

The above argument for the constant final spectral density shows how decimating a sequence reduces inter-sample correlations; in the above boxcar case, all the correlations were removed. As a consequence, the decimated sequence tends to be more "white" than the original sequence, and hence, its spectrum tends to be more flat. Note, though, that it is only the amount of correlations and the spectral shape that change in decimation. The total power does not change, for

$$P^\zeta = \mathbb{E}\,|\zeta_n|^2 = \mathbb{E}\,|\gamma_{Mn}|^2 = P^\gamma\,. \tag{105}$$

Noise equivalent bandwidth

The last topic I'm going to mention in this article is the concept of noise equivalent bandwidth, B_{eq}, though I'm not sure of how much use, or interest, that is in the EISCAT context. I first encounter the concept in my space debris work when ESA measurement modellers wanted to know what is the value of B_{eq} in EISCAT measurements. So I added B_{eq} to the set of output parameters of the ddfplan.m script (though, until recently, it was sometimes computed wrongly).

A receiver's noise equivalent bandwidth is a figure of merit that can be to some degree useful if one wants to know what kind of signal-to-noise ratio,

$$\text{SNR} = P_{\text{sig}}/P_{\text{noise}} \tag{106}$$

one can expect in a measurement, and wants to estimate P_{sig} from the radar equation, and the P_{noise} using the formula based on system temperature and some kind of "receiver bandwidth" B,

$$P_{\text{noise}} = k_{\text{B}} T_{\text{sys}} \times B\,. \tag{107}$$

Then the quantity that should be plugged into Eq. (107) in the place of B is the receiver's noise equivalent bandwidth.

The quantity $k_{\text{B}} T_{\text{sys}}$ has dimension W/Hz and the idea is that it represents the power spectral density of the incoming white noise into the receiver. Some filtering takes place in the receiver, and the effect of the filter is summarised by the noise equivalent bandwidth. The B_{eq} is the width of a hypothetical boxcar-in-frequency filter that let's through the same amount of noise to the final baseband as the actual filter. With white noise at input, according to Eq. (88), Eq. (105) and Eq. (101), the final noise power P^ζ is computable as

$$P^\zeta = P^\gamma = G^\chi \times \int_0^{f_s} |H(f)|^2 df \tag{108}$$

where G^{χ} is the power spectral density of the input noise.

By comparison of Eq. (107) and Eq. (108), one might hastily conclude that the noise equivalent bandwidth is just the integral $\int |H|^2 df$ in Eq. (108). However, that conclusion does not take into account that the apparent "power" $P^{\zeta} = \mathbb{E}\,|\zeta_n|^2$ also depends on the overall gain of the receiver, not just the shape of the gain curve $|H|^2$. This might not matter in the SNR computation, for whatever the absolute scale of the gain, both signal and noise would presumably be subject to the same scale.[22] But to get a measure for receiver bandwidth that does not depend upon the gain scale, is is customary to normalise the gain to unity at zero frequency in the power integral. Thus, the noise equivalent bandwidth of a digital filter in general is to be computed from the (possibly non-normalised) filter transfer function $H(f)$ as

$$B_{\text{eq}} = \frac{1}{|H(0)|^2} \int_0^{f_s} |H(f)|^2 df \,. \tag{109}$$

From the Parseval's theorem for Fourier series, this can also be expressed directly in terms of the filter coefficients, as

$$B_{\text{eq}} = f_s \frac{\sum |h_n|^2}{|\sum h_n|^2} \,. \tag{110}$$

For an M-tap boxcar-in-time, decimation-by-M filter such as the b3200d3, Eq. (110) gives $B_{\text{eq}} = f_s/M = f_s'$. This is consistent with our earlier result that the noise spectrum is constant in the final baseband of width f_s'.

Proof of the noise filtering theorem, Eq. (101)

Abbriviating $f/f_s = \nu$; using the definitions of the noise spectrum; the definition of the noise autocorrelation; and the definition of the filtering, we get

$$\begin{aligned} f_s\, G^{\gamma}(f) &= \sum_l g_l^{\gamma} \mathrm{e}^{-i2\pi\nu l} \\ &= \sum_l \mathbb{E}\,\overline{\gamma}_n \gamma_{n+l}\, \mathrm{e}^{-i2\pi\nu l} \\ &= \sum_{lkk'} \overline{h}_k h_{k'}\, \mathbb{E}\,\overline{\gamma}_{n-k} \gamma_{n+l-k'}\, \mathrm{e}^{-i2\pi\nu l} \\ &= \sum_{lkk'} \overline{h}_k h_{k'} g^{\chi}_{l+k-k'}\, \mathrm{e}^{-i2\pi\nu l} \end{aligned}$$

Then changing order of summation to make the l sum first, and changing in the l-sum the summation index to l', $l' = l + k - k'$, the triple sum separates to the product of three sums, giving the asserted result.

$$\begin{aligned} f_s\, G^{\gamma}(f) &= \sum_{kk'} \overline{h}_k h_{k'} \Big(\sum_{l'} g^{\chi}_{l'} \mathrm{e}^{-i2\pi\nu l'}\Big) \mathrm{e}^{+i2\pi\nu k} \mathrm{e}^{-i2\pi\nu k'} \\ &= \overline{H}(f) H(f)\, (f_s\, G^{\chi}(f)) \,. \end{aligned}$$

[22] This assumption is actually not safe in EISCAT where the filters are not at all flat in frequency. The assumption does not take into account that the narrow-band signal can experience quite different gain depending upon it's Doppler-shift. Then one has anyway to be aware of the gain shape $|H(f)|^2$ for all f, and the whole concept of noise-equivalent bandwidth becomes rather unnecessary.

Proof of the noise decimation theorem, Eq. (102)

The result is analogous to the decimation theorem for the spectra of the deterministic signal, Eq. (11), and follows from the definition of the spectrum as the Fourier transform of a sequence, and the definition of decimation. First we note that from the definition of decimation as picking every M'th element of the sequence,

$$\zeta_n = \gamma_{Mn} \,, \tag{111}$$

together with the assumed stationarity of the noise, a result similar to Eq. (111) holds also for the noise autocorrelation,

$$g_l^\zeta = \mathbb{E}\,\overline{\zeta}_n \zeta_{n+l} = \mathbb{E}\,\overline{\gamma}_{Mn}\gamma_{Mn+Ml} = g_{Ml}^\gamma \,. \tag{112}$$

Then, starting from the right-hand side of Eq. (102), we have

$$\begin{aligned} \sum_{m=0}^{M-1} G^\gamma(f + m f_s') &= \frac{1}{f_s} \sum_m \sum_l g_l^\gamma \mathrm{e}^{-i2\pi \frac{f+m f_s'}{f_s} l} \\ &= \frac{1}{f_s} \sum_l \left(g_l^\gamma \mathrm{e}^{-i2\pi \frac{f}{f_s} l} \sum_{m=0}^{M-1} \mathrm{e}^{-i2\pi \frac{m}{M} l} \right) . \end{aligned} \tag{113}$$

On the right-hand side of Eq. (113), the sum over m equals M when l is a multiple of M, and is zero otherwise, so in the l-sum most terms drop out. Then using $f_s' = f_s/M$ and Eq. (112), we get from Eq. (113)

$$\begin{aligned} \sum_{m=0}^{M-1} G^\gamma(f + m f_s') &= \frac{M}{f_s} \sum_l g_{Ml}^\gamma \mathrm{e}^{-i2\pi \frac{f}{f_s} Ml} \\ &= \frac{1}{f_s'} \sum_l g_{Ml}^\gamma \mathrm{e}^{-i2\pi \frac{f}{f_s'} l} \\ &= \frac{1}{f_s'} \sum_l g_l^\zeta \mathrm{e}^{-i2\pi \frac{f}{f_s'} l} \\ &= G^\zeta(f) \,, \end{aligned}$$

as we claimed in Eq. (102).

7 Summary

I have provided a fairly detailed functional description of the EISCAT digital receiver, augmenting the very short account given in the 1997 paper of Wannberg et al., about the ESR receiver. The model of the system built here has allowed me to compute quantitatively the expected output of the receiver both for sinusoidal signals and for noise. I have especially inspected what kind of output to expect in the case of the quip experiment, which used the filter b2300d3 that provides 5 MHz final sampling rate.

The model is able to reproduce the observed "disturbance" very well, so I consider the disturbance mostly understood. It is simply caused by beat-type interference between two complex-valued sinusoidal signals in the final baseband. Both these complex sinusoids derive from the single original real-valued analog sinusoid, which has two spectral

component, symmetric around zero in the analog domain. In the detection, both of the those two components map to the baseband. Normally, the channel's post detection filtering should, and does, kill the higher-frequency component. But the new filter is so wide that not enough suppression of unwanted frequencies can take place.

As about the consequences, I think the jury is still out. For one thing, it is possible to make other filters with 5 MHz sampling rate, that are narrower than the boxcar-in-time b2300d3. But it seems difficult to get rid of the beat altogether even then. A related issue is that if one wants to make more use of boxcar-in-time filters in general, as I would like for the space debris work at least, the beat problem can creep-in already with a 1 MHz sampling rate, depending on the precise frequencies of transmission and the signal; and will necessarily become a problem with 2.5 MHz rate. (The 1 MHz filter that I have been using so far, b500d15, is not affected by the beat).

So for the fastest sampling rates, the real question is, does the presence of the beat make data unusable. In high-accuracy space debris work, the beat probably would distort the predictability of the pulse response shape, which I use for high range accuracy. But it might be possible to filter the unwanted spectral component out in the analysis phase.

I state above that the disturbance is "mostly" understood. That means that I can predict the spectrum correctly at a level that is about 40–50 dB down from the main spectral peak. But as Fig. 5 shows, in the data taken when the transmitter was operating, there are regular-looking features in the spectrum that do not fit in the model.

On the −50 dB level below the wanted signal, the unwanted signals should not be of any consequence in any practical EISCAT measurement. It would nevertheless be of interest so find out where those features come from. This story might get a part two, some day.

8 FIGURES

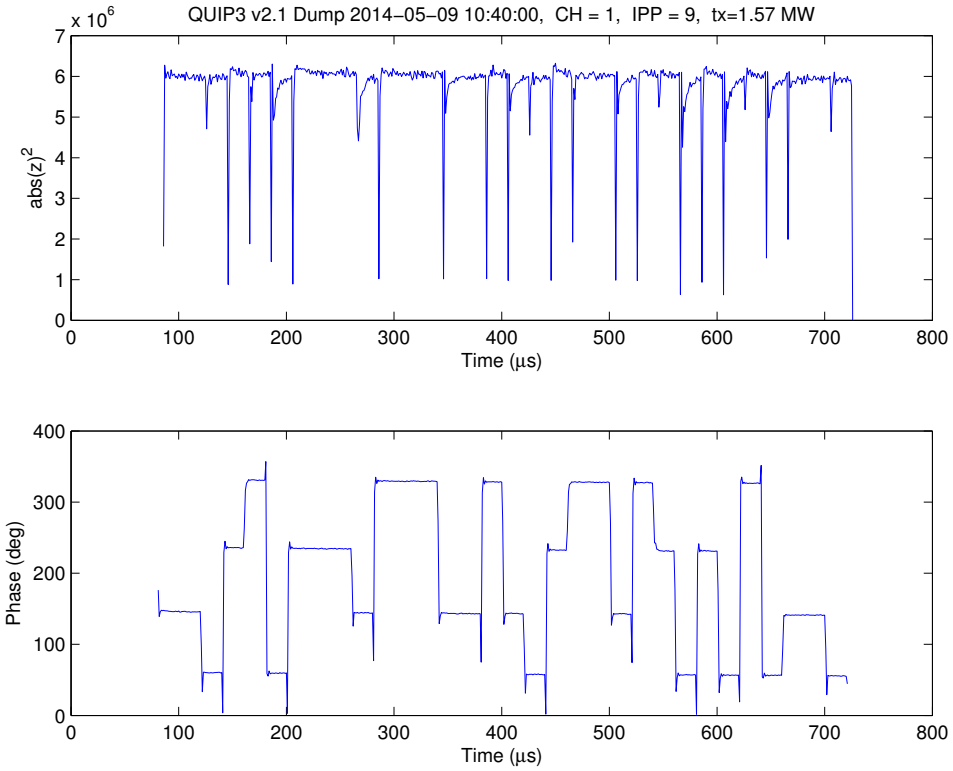

Figure 1: Experiment quip in May 2014, channel 1 transmission-sample signal. These transmission-monitoring voltage level data on the EISCAT Tromsø UHF radar are from a 640 μs 4-phase code, taken with the final sampling rate of 1 MHz. The channel's decimating filter was the decimation-by-15 filter b500d15 (Fig. 30), which had been tested earlier in several satellite measurements. The top panel shows the squared magnitude of the samples, the bottom panel shows the phase of the samples. The time counts from the beginning of the radar cycle. By and large, these data were as expected.

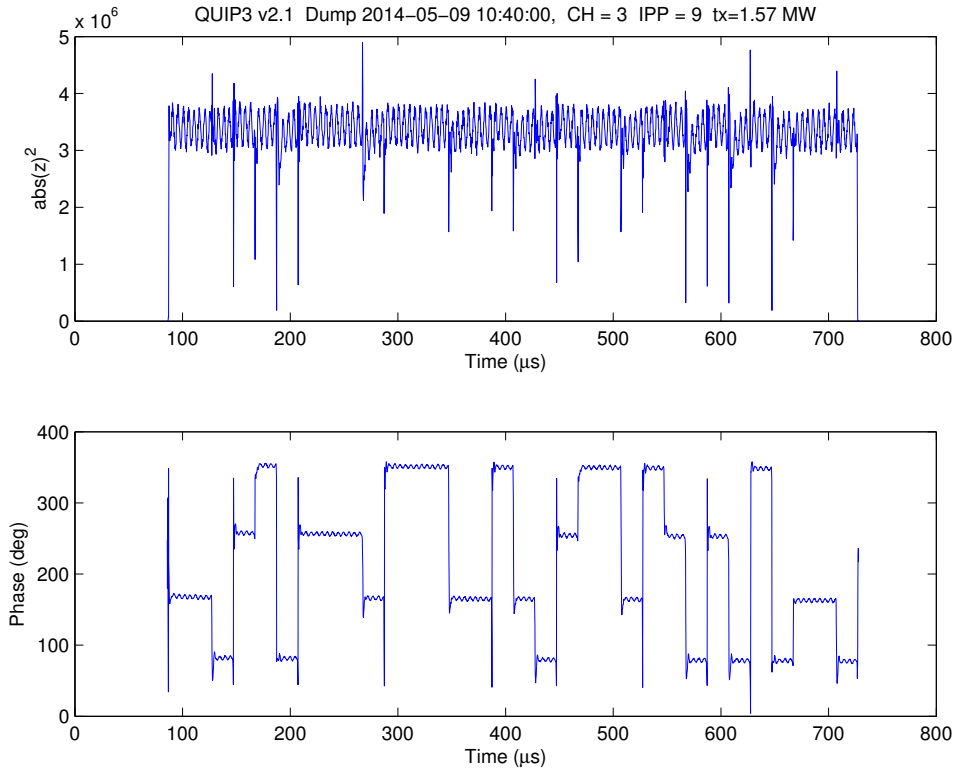

Figure 2: Quip, channel 3 transmission data. These data are from the same code, and actually literarily the same primary 15 MHz A/D samples, as in Fig. 1, but decimated to 5 MHz final sampling rate, using the new, poorly tested wide-band decimating filter b2300d3 (Fig. 32). This filter has a large bandwidth, 4.7 MHz. Therefore, even though the data immediately appeared "noisier" than the 1 MHz data in Fig. 1, during the experiment is was thought that this probably was OK. But after the experiment, a zoom-in, as in Fig. 3, revealed that the "noise" was not so much noise at all, but a rather regular oscillation.

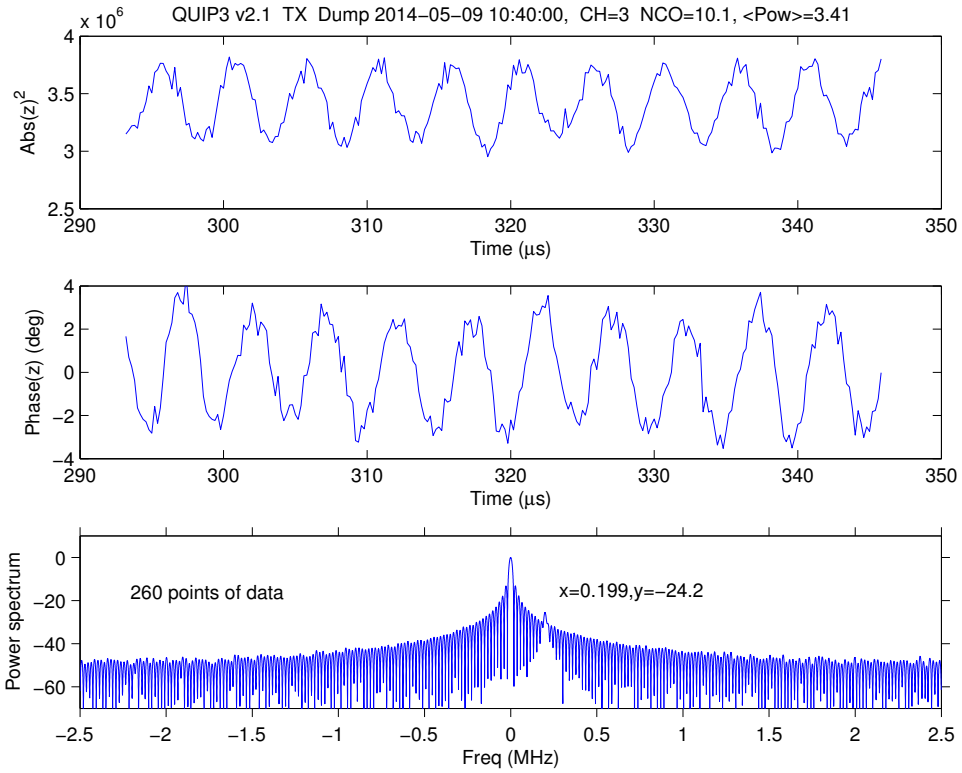

Figure 3: Quip, the wide band transmission samples at 10.1 MHz zoomed-in. The top two panels are a zoomed-in version of Fig. 2, zoomed into the longest constant-phase segment of the code. The plots reveal an unexpected oscillatory behaviour both in the squared magnitude and in the phase of the samples. In the second panel, the peak-to-peak variation of the phase angle is about 5 degrees (the mean phase is set to zero in the plot). The period of the oscillations in the top two panels is about 5 μs. The maximum peak-to-peak variation of the power in the top panel is about 22 % around the mean power. The bottom panel is power spectrum computed from the samples in the longest constant-phase segment of the code and normalised to maximum value of 0 dB. Unfortunately, I understood to start plotting spectra only after we had already come to grips with the oscillation problem, after several days of stressful confusion. In the spectrum, there is the expected DC peak, but in addition, an unexpected peak at about +200 kHz. This could have been a crucial clue.

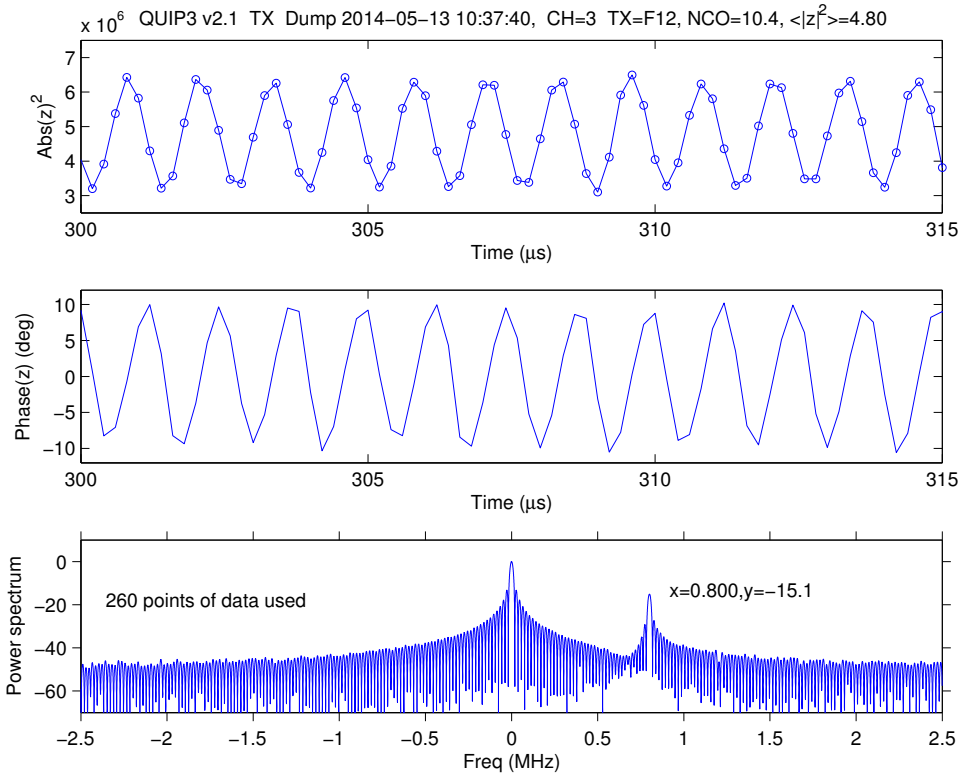

Figure 4: Wide-band transmission samples at 10.4 MHz. To generate ideas about what the oscillating disturbance on the 5 MHz channel in quip might depend on, the transmission frequency was changed from F13 as in Fig. 3 to F12 and the receiver's tuning was correspondingly changed from 10.1 MHz to 10.4 MHz. The period of the disturbance decreased to about 1.25 μs from 5.00 μs. The relative peak-to-peak power variation increased to about 65% from 22%, and the peak-to-peak phase variation increased to about 20° from about five degrees. The actual transmission peak power was 1.66 MW for these data. Between Fig. 3 and Fig. 4, nothing changes in the filter itself, so this test strongly suggested that the disturbance was not generated by some malfunction of the filter, but had somehow to be present in the incoming analog signal. But then, on the other hand, why were we not seeing it in the 1 MHz data on channel 1?

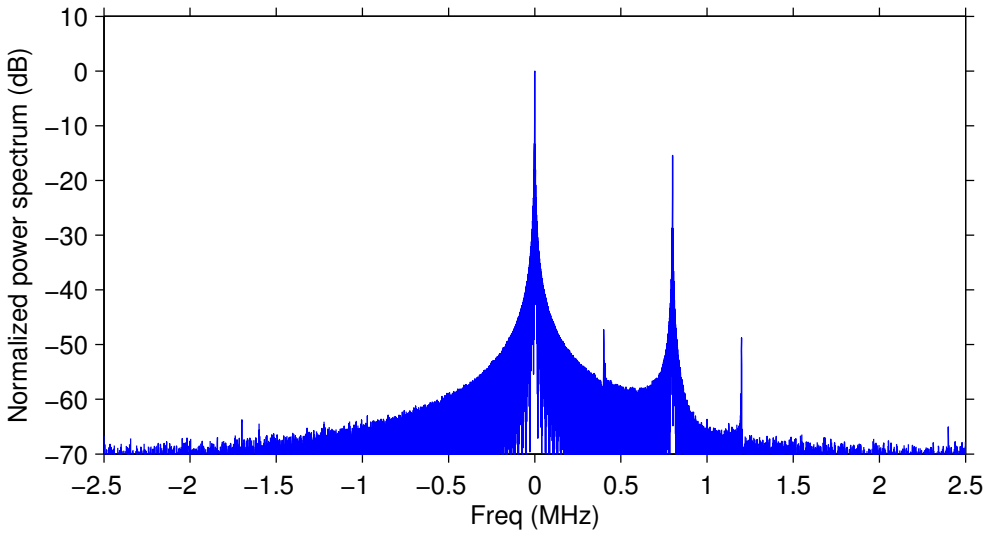

Figure 5: Quip, a higher-sensitivity spectrum of the wide band data. When inspecting the oscillating disturbance, at some point also the 4-phase phase shifter was disabled during transmission by disconnecting its control signals. This made the transmission effectively a 640 μs longpulse, which allows the whole pulse easily to be used for spectrum computation. This figure is basically a higher-sensitivity version of the spectrum shown in Fig. 4. In addition of using a longer FFT, the data are also integrated over a few pulses. The maxima of the four peaks, as read from the FFT curve, are at (0.0000, 0.0000), (0.4004, −47.23), (0.7999, −14.44) and (1.200, −48.79). Today, we have a good understanding of the two main peaks. Where the two minor peaks come from, requires further study.

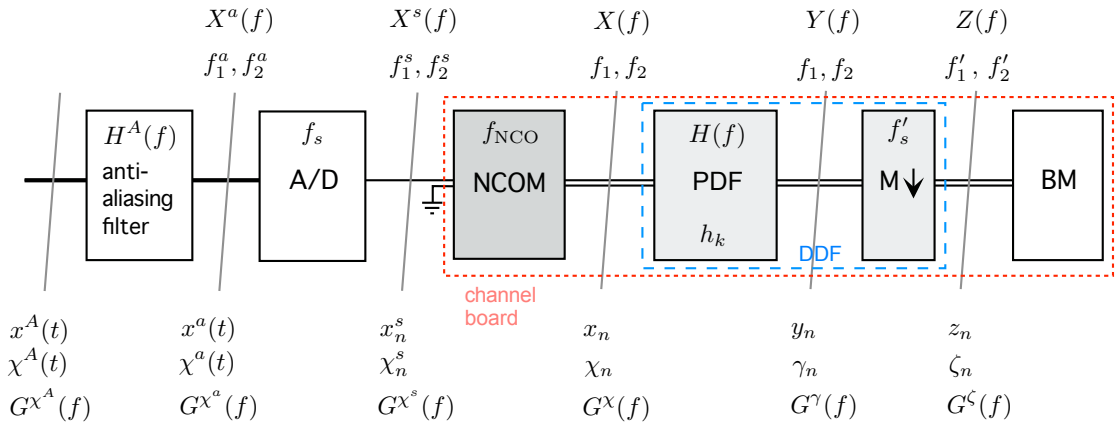

Figure 6: The EISCAT digital channel, overview and notations. Functionally, each EISCAT digital receiver channel consists of a numerically controlled oscillator (NCO) and complex multiplier module (NCOM), the post detection filter (PDF) which is a finite impulse response digital filter, a decimator (M), and a memory buffer (BM) for temporal storage of the channel output. The dotted red line encircles the modules that are implemented in hardware as the EISCAT "channel board", designed by Markku Postila in the early 1990s at the EISCAT Sodankylä site. The PDF and the decimator summarise the functionality of the HSP43200 decimating digital filter (DDF) that, together with the HSP45116 NCOM, forms the core of the channel board hardware. There are separate, but identical, DDFs for real and imaginary part to precess complex-valued digital data after the NCOM. The figure also shows the anti-aliasing filter and the analogue to digital converter in front of the digital channel. In the EISCAT receivers, the output of the A/D is copied to several channels, but only one channel is shown in this figure. The A/D samples continuously at f_s = 15 Msamples/s, using the second intermediate frequency (IF2) of the EISCAT analogue receiver as its input. In front of the A/D, there is the anti-aliasing bandpass filter, which in the UHF system is about 7 MHz wide and centred at 11.2 MHz. Between the decimator M, which reduces the sampling rate from the 15 MHz primary rate f_s to the final sampling rate $f_s' = f_s/M$, and the output buffer BM, there is gating logic (not shown) that allows the data stream to the buffer be blocked as required, with 100 ns time resolution. The various f-labels at the top of the diagram give the notion used in the main text to refer to the frequencies occurring in the processing of a sinusoidal analog input signal $x^a(t)$. The spectral processing begins from the spectrum $X^a(f)$ of the real-valued analog signal. That spectrum has two delta-function like components, at f_1^a and f_2^a ($f_1^a = -f_2^a$). The processing finishes with the spectrum $Z(f)$ of the final decimated complex-valued sampling stream z_n. Under the receiver blocks in the diagram, the symbols $x^A(t) \ldots z_n$ give the notation for the time domain signal in various stages of processing. The symbols $\chi^A(t) \ldots \zeta_n$ refer to the time domain noise. The symbols $G^{\cdot\cdot}(f)$ refer to the power spectral density of the noise.

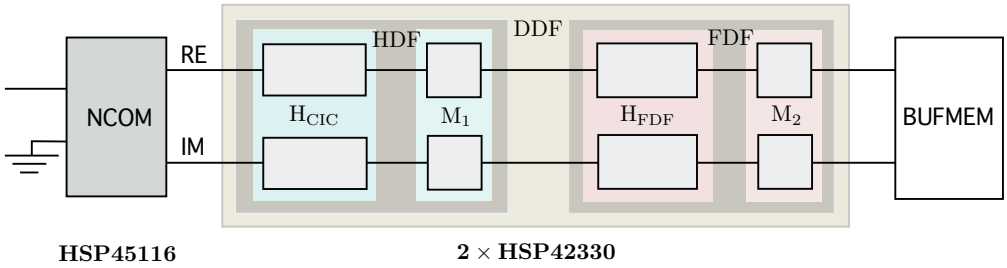

Figure 7: EISCAT digital channel, a programmer's block diagram. The channel consists of a numerically controlled oscillator and complex multiplier module (NCOM), a decimating digital filter (DDF), and an output buffer. The NCOM is a single chip, Intersil's HSP45116. The DDF is implemented by two HSP43220 decimating digital filters, one for the real part and one for the imaginary part of the NCOM output.The HSP43220 consists of two decimating filters, termed the HDF and the FDF in this document (in the HSP43220 data sheet, the latter is referred to as "FIR", which is an unusably generic name). Both decimating filters are shown here in the canonical representation: a filter, which is labelled by H_{CIC} and H_{FDF} here, followed by a decimator, M_1 and M_2, respectively. These notations are also used in the ddfplan.m Matlab script that is used in EISCAT to prepare filter configuration files, which are known as the "fir files" in EISCAT. The "high-order decimating filter" HDF allows decimation M_1 up to 1024 and down (so it now seems) to 3. The filter designer's control over the impulse response of the H_{CIC} is very limited. The impulse response is constrained to be equivalent to the impulse response of a cascade of N_1, from 1 to 5, boxcar filters, each with M_1 taps. The designer can only specify the number of the filters in the cascade and the decimation factor. The "fir decimating filter" FDF provides decimation M_2 from 1 to 16, and up to 512 independently definable coefficients, to build a symmetric filter H_{FDF} with twice that number of taps. However, the ddfplan.m script does not allow even that much flexibility, for it models also the FDF as a cascade of N_2 boxcar filters, each with L_2 equal taps (the decimation factor M_2 can be different from L_2, though). The script provides graphical output about the filter shape $|H_{DDF}(f)|^2$, etc (see, e.g. Fig. 16 and 17), and the normal way of using the program is to vary the five input parameters until the result looks acceptable for the experiment in question. The program then generates the required filter configuration file, which by EISCAT convention is named as bxxxdmmm.fir, where xxx is approximately the 3 dB point in kilohertz and mmm is the total decimation factor M_1M_2.

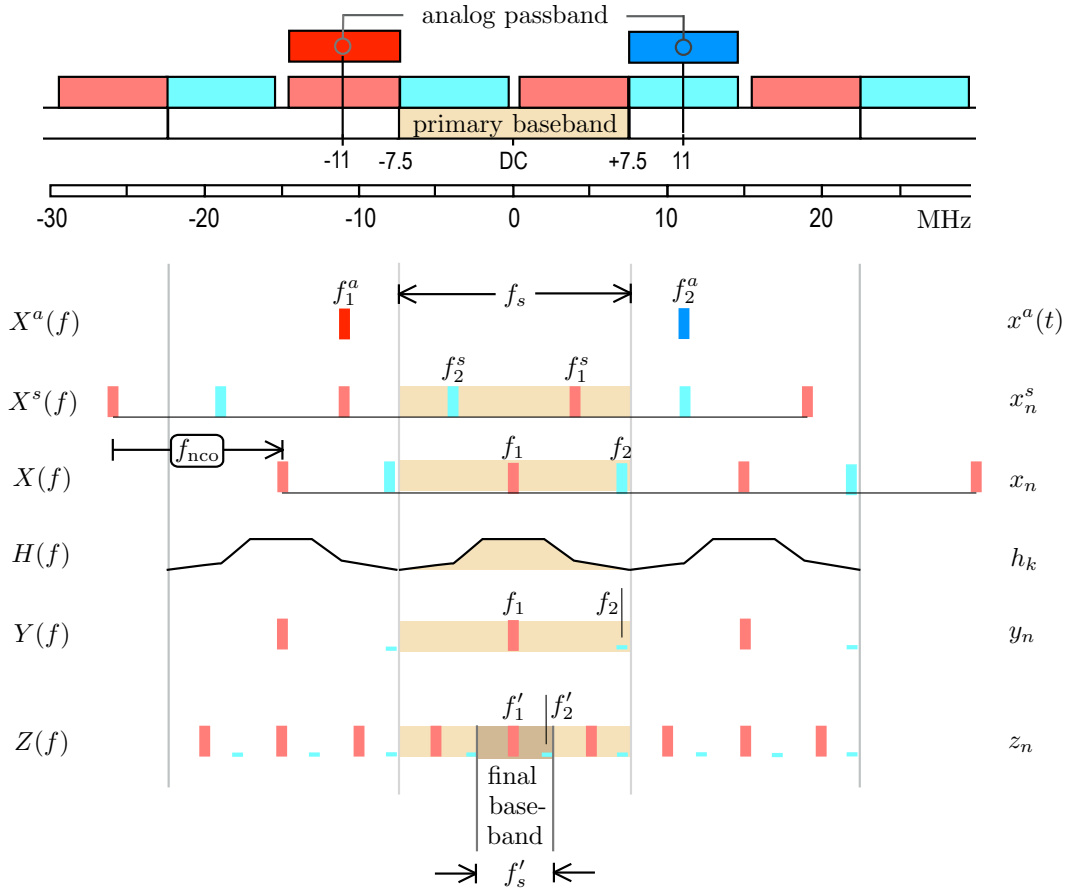

Figure 8: Frequency mapping on an EISCAT digital channel. The analogue passband is centred at (about) 11 MHz, and has 3 dB width (about) 7 MHz. There figure has been drawn as if these where the exact values. Both positive and negative frequency parts are marked in the figure. In sampling, the passband becomes periodic by the 15 MHz sampling frequency. The bottom part of the figure shows how the spectrum of a narrow band analogue signal is transformed when traversing the receiver stages shown in Fig. 6. In the drawing, the analogue signal is at the centre frequency 11 MHz of the passband so that the analogue spectrum $X^a(f)$ has support only near $f_2^a = +11$ MHz and $f_1^a = -11$ MHz. In sampling, X^a becomes periodic by the sampling frequency f_s and becomes the spectrum X^s. The two components of the periodic function $X^s(f)$ that fall within the Nyquist zone around DC, marked as the primary baseband in the figure, have frequencies f_1^s and f_2^s. The sampled signal is translated by the amount f_{nco} in the channel's NCOM unit, and the spectrum becomes the function $X(f)$. In this example, the NCO value is chosen so as to move a negative frequency component to zero frequency, so $f_{\text{nco}} = 11$ MHz, $f_1 = 0$ MHz, and $f_2 = 7.0$ MHz, near the edge of the primary baseband. Next, the signal is filtered in the PDF module, which amounts to multiplication of $X(f)$ with the filter transfer function $H(f)$, giving the spectrum $Y(f)$. Ideally, the high frequency component at f_2 should be completely extinguished, but in practise this succeeds only to a degree. Finally, in the decimator M, the spectrum $Y(f)$ is periodically replicated by the decimation factor ($M = 3$ in this drawing), resulting in the frequencies $f_1' = 0$ and $f_2' = 2.0$ MHz in the final baseband. The final baseband has width $f_s' = f_s/M$, 5 MHz in this drawing.

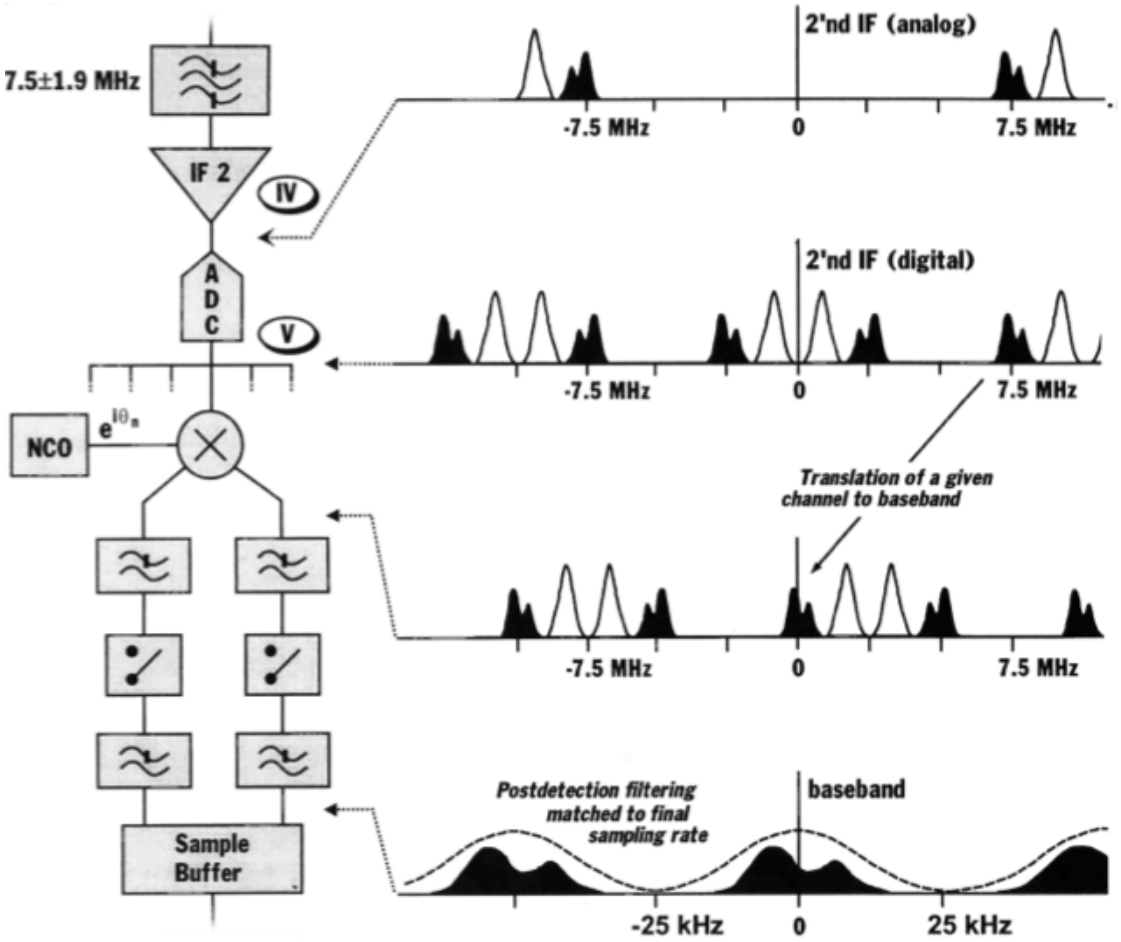

Figure 9: The original EISCAT Svalbard Radar digital channel. The figure is a partial reproduction of Figure 6 in the 1997 Radio Science paper by Wannberg et al. on the brand new ESR system. The original caption labels the Figure as being a "functional block diagram of the ESR receiver". In the initial ESR system, the primary A/D sampling rate was 10 MHz, and the roughly 4 MHz wide anti-aliasing filter was centred at 7.5 MHz. The actual digital channel hardware after the A/D was essentially the same as in all present-day EISCAT systems. Assuming that Fig. 6 and Fig. 7 of the present article are correct, can you suggest any improvements to the Figure? Answer: The dashed "filter line" from the bottom right panel should be removed, and a second decimator should be added to the bottom left diagram, between the lowpass filter and the sample buffer.

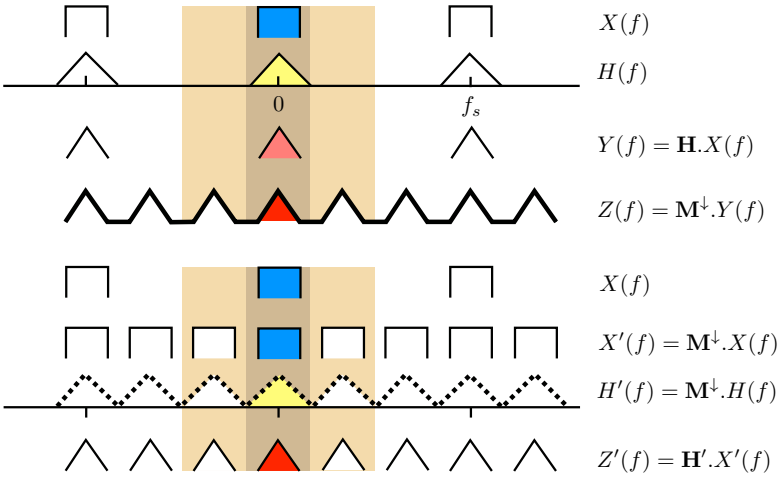

Figure 10: Processing a strongly band limited signal by a narrow filter. Here both the filter $H(f)$ and the signal $X(f)$ have bandwidth less than the final sampling frequency, so that their support in the primary baseband (light-colorer stripe) stays strictly within the final baseband (dark-coloured stripe). In this case, it is possible to envisage a filter, $H'(f)$, that is periodic by the final sampling frequency f_s' (=$f_s/3$ in this illustration), and produces the correct final spectrum $Z(f)$ in the final baseband. The filter $H'(f)$ is the periodic extension of $H(f)$, with period f_s', as in Eq. (18). Here we have denoted by $\mathbf{M}^{\downarrow}$ the "decimation operator". Its effect in spectral domain is to perform the finite periodic summation. The filter H' operates to a fictitious signal x_n', which has the spectrum $X'(f)$ and could be the decimated version of the actual input signal x_n. Because X is strictly band-limited, decimation does not distort it. In essence, the artificial scheme, the bottom panel of the Figure, begins with the decimation and does filtering afterwards, while the actual system in the top panel performs these operations in the opposite order. These systems are not equivalent in general, but are equivalent for strongly band-limited input signals and narrow filters. The filter-related dashed line in the bottom right panel of Fig. 9 could correspond to $|H'(f)|^2$. So, to get the [W97] drawing, we need to pick the $H'(f)$ from the bottom panel of this figure, while still can take the signal $Z(f)$ from the top panel. With hindsight, this seems rather contrived way of doing things. If one necessarily wants to draw some filter shape to the figure, one should just draw $|H(f)|^2$, as in the top panel of this figure.

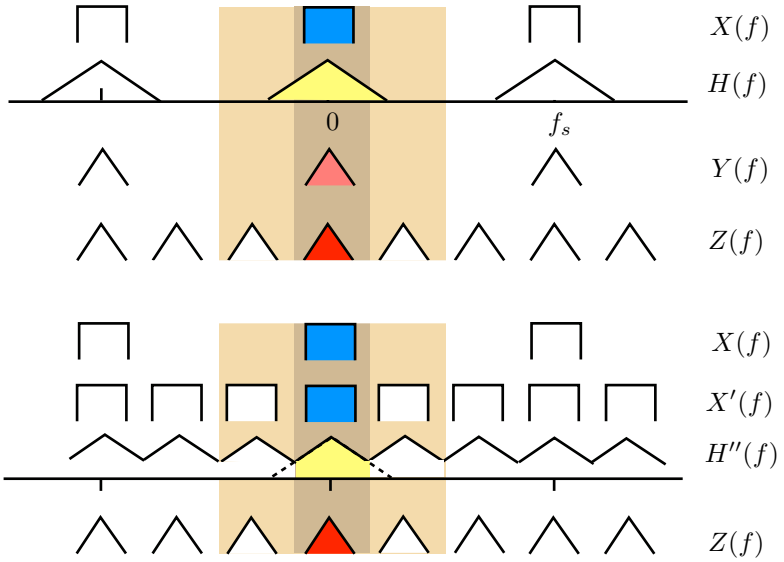

Figure 11: Processing a strongly band limited signal by a wide filter. Even if the filter is not strictly band limited to the final baseband, the signal should typically be. In this case one can consider the dashed line of [W97] to represent the power response $|H''(f)|^2$ where the filter $H''(f)$ is found by setting $H(f)$ to zero outside the final baseband, and then extending the result periodically by the final sampling frequency. The filter $H''(f)$ operates on the fictitious samples x'_n of Fig. 10, which already represent the final sampling rate but have the spectrum $X'(f)$, not to the original primary samples, which have the spectrum $X(f)$. In essence, the artificial scheme begins with the decimation and does filtering afterwards, while the actual system performs these operations in the opposite order. These systems are not equivalent in general, but are equivalent for strongly band-limited input signals. The artificial scheme is depicted in the lower panel of the figure, while the upper panel illustrates the actual transformations performed by the filter and the decimator in the EISCAT receiver.

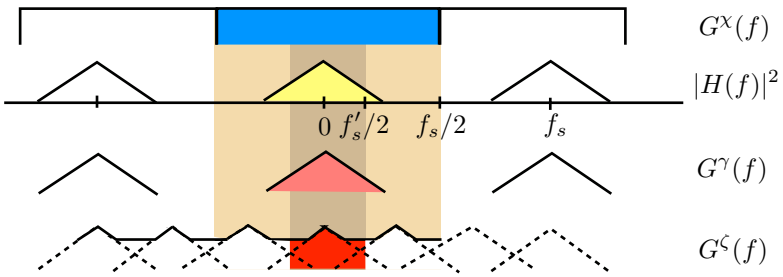

Figure 12: Processing a wide-band signal by a wide filter. If neither the filter nor the filter input are strictly band limited to the final baseband, the only sensible strategy to compute the spectrum in the final baseband is to explicitly take into account both the filter and the decimator. Here, we have assumed that the filter input is white noise, so that the initial power spectral density $G^{\chi}(f)$ is constant. In the filtering, the density is multiplied by the filter's power response $|H(f)|^2$, to produce the power spectral density $G^{\gamma}(f)$. Then in decimation, $G^{\gamma}(f)$ is periodically extended, with period equal to the final sampling rate f_s'. When the filter $H(f)$ is wider than the final sampling rate, aliasing of spectral density to the final baseband takes places, resulting in the spectrum $G^{\zeta}(f)$ which is periodic by the final sampling frequency, as it must be. The shape of the final spectrum needs not to have much in common with the filter's power response shape. For instance, for "ideally matched" boxcar filters such as the filter b2300d3 (Fig. 32), for which the impulse response length is equal to the sampling interval (more precisely, the decimation M is equal to the number of filter taps), $G^{\zeta}(f)$ is a constant, while $|H(f)|^2$ is of type $\left[\frac{\sin(\pi M \tau_s f)}{M \sin(\pi \tau_s f)}\right]^2$. On the other hand, for some other, more strictly band limited filters (Fig. 31), there is only a small amount of noise aliasing, and then the shape of final noise spectrum $G^{\zeta}(f)$ can be quite similar to the dashed filter-shape line in [W97].

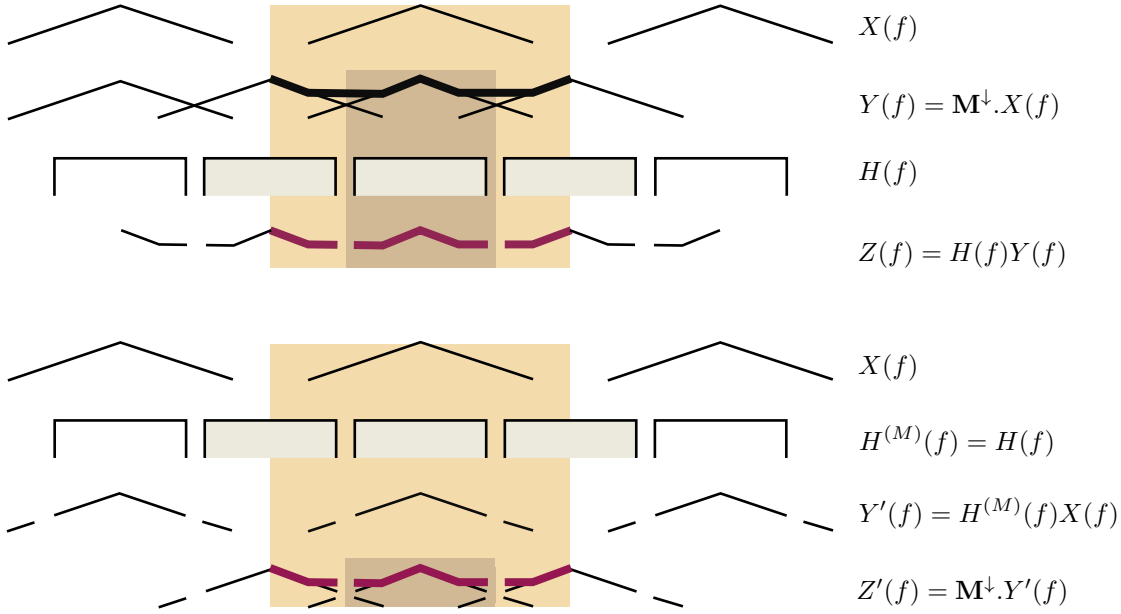

Figure 13: Decimator→Filter order change rule in frequency domain. For any input spectrum $X(f)$, the system decimator followed by filter, [M $\rightarrow$ H], the top panel, will give the same output as the system [$\mathrm{H}^{(\mathrm{M})} \rightarrow$ M], the bottom panel, so that $Z'(f) = Z(f)$. The filter $\mathrm{H}^{(\mathrm{M})}$ is the "interpolated" filter which has $M-1$ zeros between each pair of the taps of H. Note that $\mathrm{H}^{(\mathrm{M})}$ operates to samples x_n with the higher sampling rate f_s, while H operates to the samples y_n with the lower rate $f_s' = f_s/M$, and therefore the two filters have the same transfer function as function of f. The basic reason to this rule is that the interpolated filter has the same form as the original filter, so that is periodic by f_s' instead of only by f_s. Due to this, it is able to modify all the Nyquist components of the input spectrum that will ultimately contribute to the final baseband in the decimation. And due to the linearity of the aliasing phenomenon, it does not matter whether the aliasing components are modified in filtering before the aliasing takes place (as in bottom panel) or after the aliasing (as in the top panel). In the drawing, decimation factor $M = 2$, and the light-coloured yellowish stripe represent the primary baseband, the the dark yellowish colour the final baseband. The final baseband is marked only for those spectra which correspond to the sampling frequency f_s'.

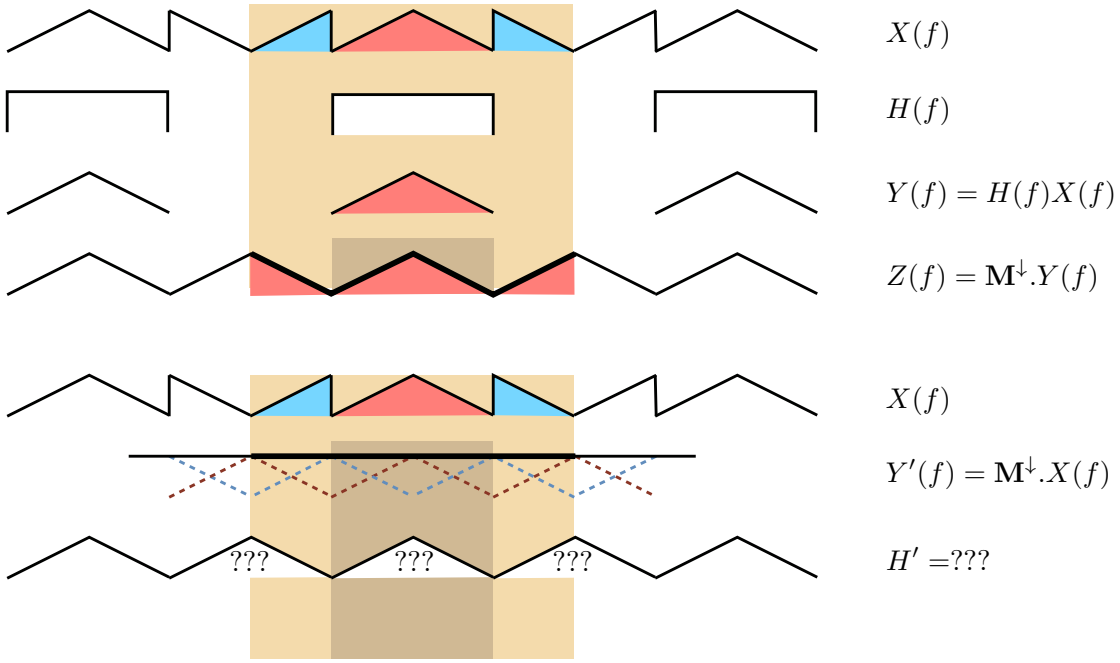

Figure 14: Filter→Decimator no-swap rule in the frequency domain. It is not, in general, possible to move a decimator that is after a filter to the front of the filter, even if the filter would be modified at the same time, and still get an equivalent system. That is, given the system $\langle H \rightarrow M \rangle$, it is not possible to find an equivalent system of the form $\langle M \rightarrow H' \rangle$. In spectral terms, this is because when the decimator is after the filter, as in the top panel, the filter "protects" the decimator so that aliasing will not occur in the decimation. If decimation is done first, as in the bottom panel, aliasing can happen, and can lead to irreversible loss of information. In the bottom panel, in order to get the same final output $Z(f)$ as in the top panel, one would need to tailor the filter H' separately for each specific input $X(f)$. But for an equivalent configuration, the selection of H' must not depend on the input, it can only depend on M and H.

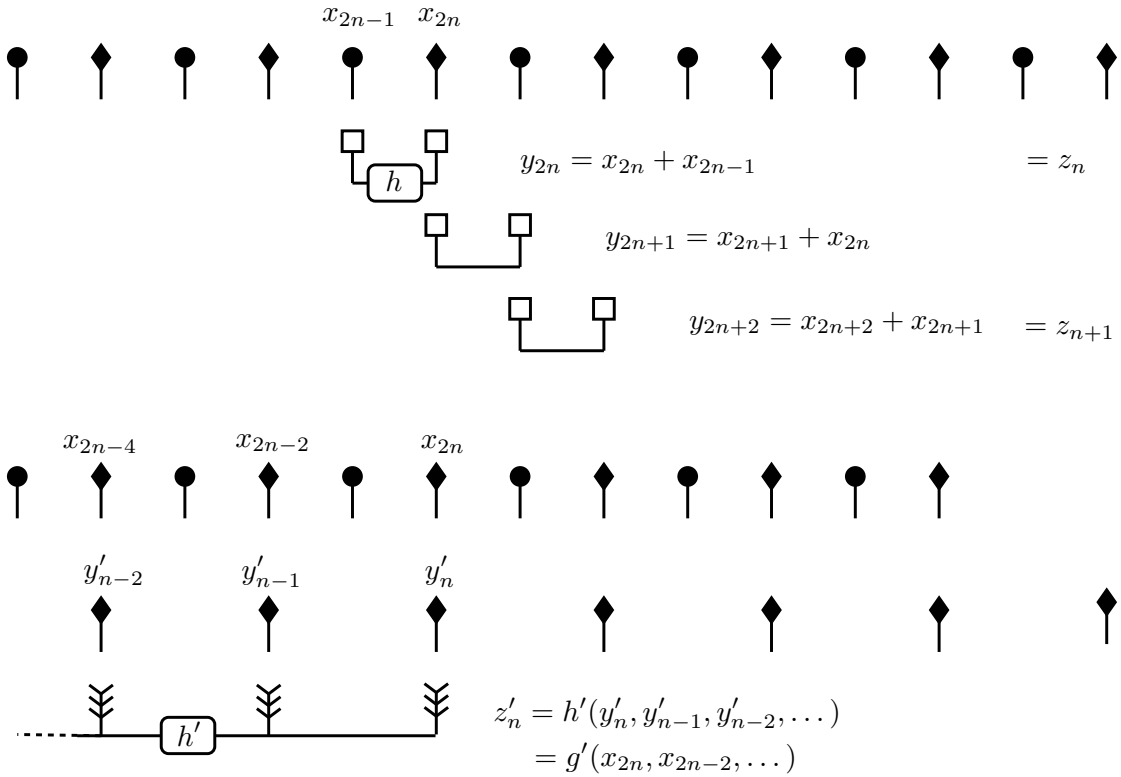

Figure 15: Filter-Decimator no-swap rule in time domain. In terms of the samples, the rule that given a system $\langle H \to M \rangle$, one cannot find an equivalent system in the form $\langle M \to H' \rangle$, is due to the loss of degrees of freedom if the decimation is done first, as in the bottom panel. In the bottom panel, the decimated sequence y'_n depends only of the primary samples x_m where the index m is an even number. Therefore, no matter what the filter h' would be, the final samples z'_n also can depend only on the even-indexed x_m. Instead, in the top panel, where filtering is done before decimation, the final samples z_n depend both on even-indexed x_m and odd-indexed x_m, $z_n = g(x_{2n}, x_{2n-1})$. Therefore the linear functionals g and g' that define the input-output relation in the two systems cannot be made equal, no matter how the taps of h' are chosen. In the drawing, the filter H has two taps, both equal to unity, and the decimation factor M also is two. The top panel also shows how the filtering is done by sliding the filter along the input sequence.

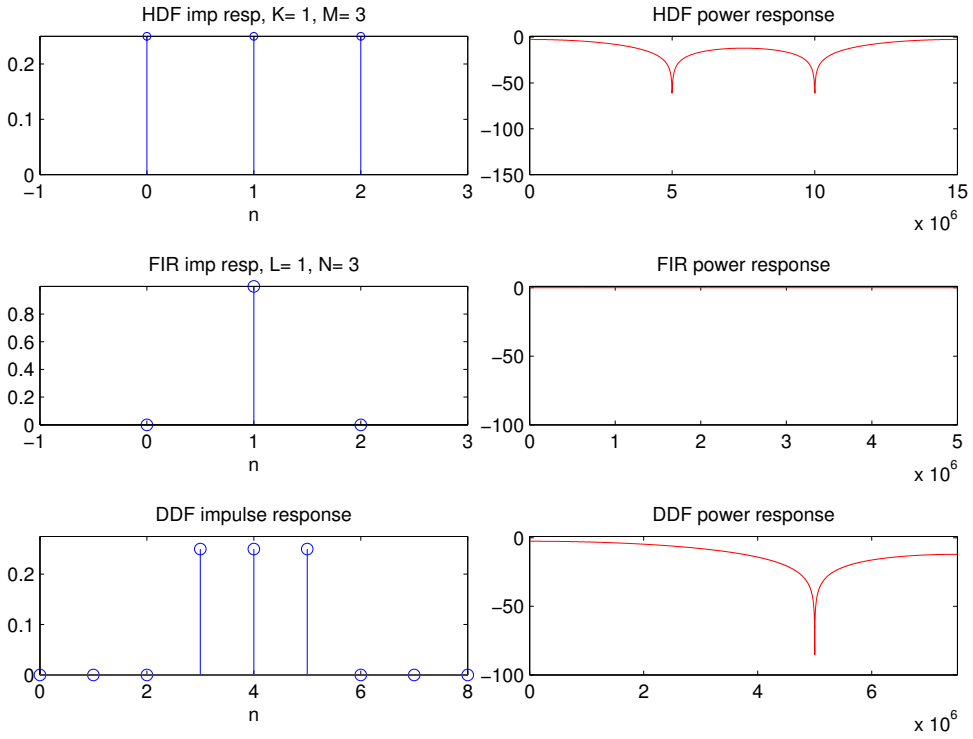

Figure 16: Filter b2300d3, standard ddfplan graphics, overview plot. The top row of panel gives the impulse response and the power response $|H_{\mathrm{CIC}}(f)|^2$ of the high-order decimation filter of HSP43220. The middle panels give the impulse response and the power response $|H_{\mathrm{FDF}}(f)|^2$ of the fir decimation filter of HSP43320. The bottom panel gives the impulse response and the power response $|H_{\mathrm{DDF}}(f)|^2$ of the equivalent filter of the whole system. For b2300d, the FDF is arranged, in effect, to be in all-pass state. This is an exception to the normal ddfplan.m algorithm for the FDF design as a cascade of boxcar sections. All the power responses are defined in the 15 MHz primary baseband $[-7.5, 7.5]$ MHz; various parts of it are shown in these plots. The DDF has a null at 5.0 MHz, but except in its immediate vicinity, the filter does to offer sufficient stop band attenuation. From the log scale plot, that is not conspicuously obvious, though.

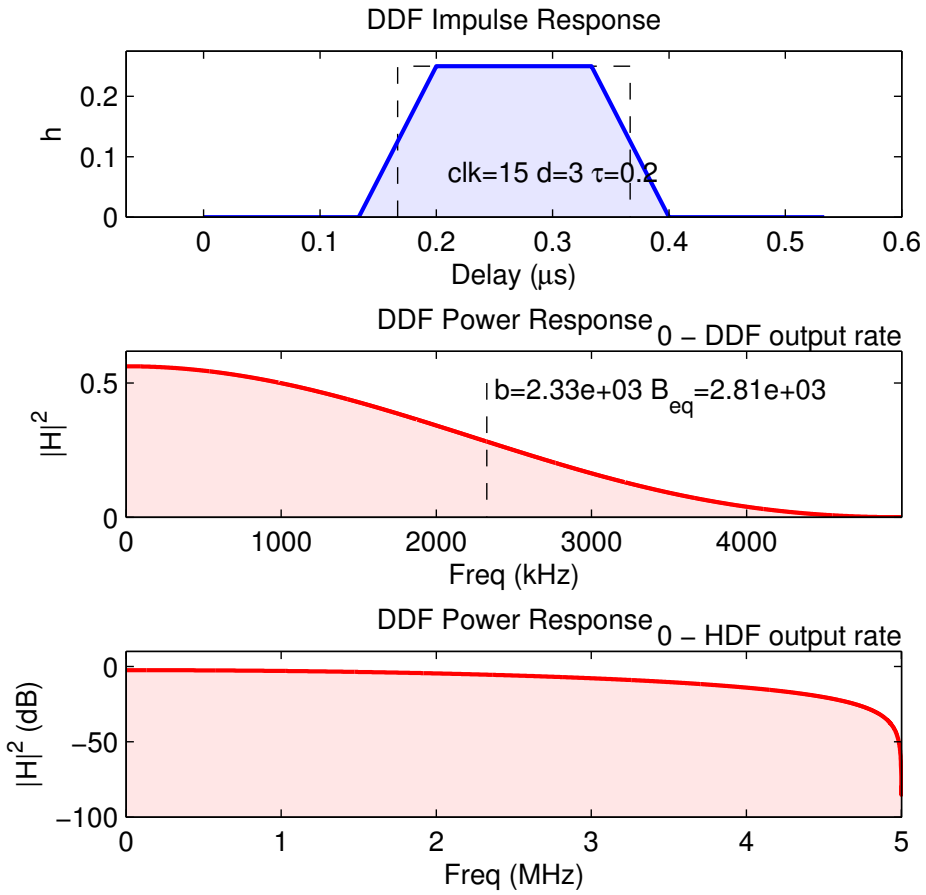

Figure 17: Filter b23od3, ddfplan graphics, standard plot 2. The filter frequency response in panels 2 and 3 is shown only for frequencies up to 5.0 MHz instead the whole way up to 7.5 MHz, the edge of the primary baseband. In addition of providing a somewhat "arbitrary" end point for the frequency axis (neither the primary nor the final baseband), this choice also unfortunately results in an overly optimistic impression of the filter behaviour for this anomalously wide filter. As an unrelated problem, in the top panel, it is questionable how much sense it makes to draw the three-tap discrete filter impulse response using the (or any) continuous curve. This could potentially be a problem if the filter were used in standard incoherent scatter analysis, where the impulse response curve is the sole source of information about a channel's filter.

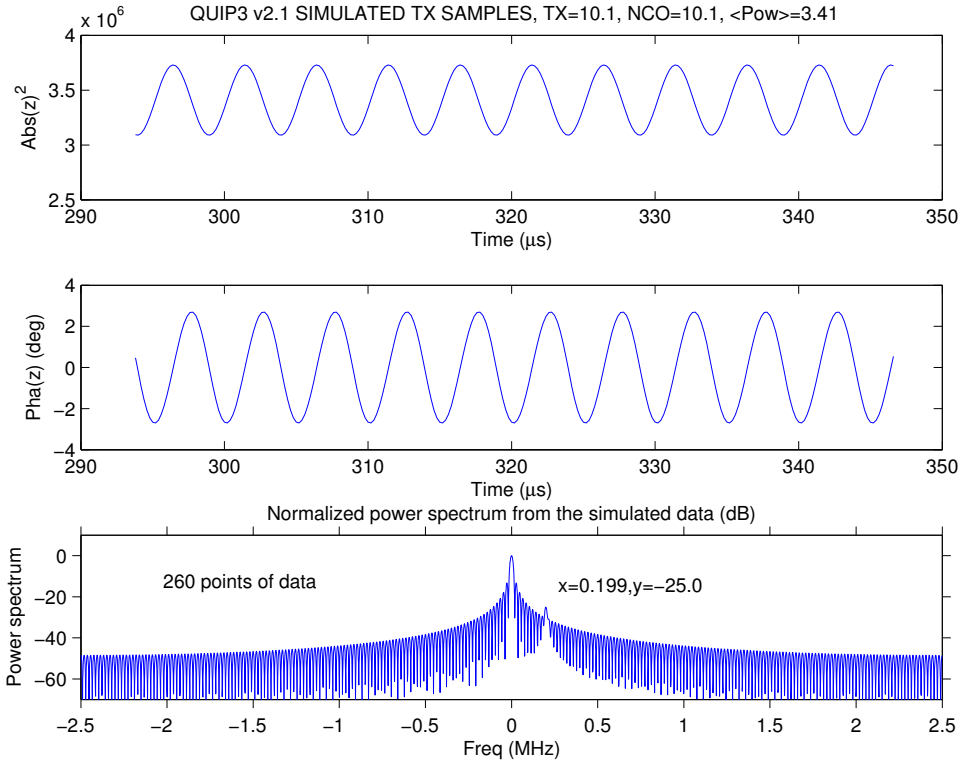

Figure 18: Simulation of the 5 MHz channel in quip. The corresponding measured data are shown in Fig. 3. In the simulation, one generates samples of a 10.1 MHz real-valued sinusoidal, noise-free, signal to represent the 15 MHz primary A/D samples, then shifts those to baseband by multiplication with a 10.1 MHz complex sinusoid, and finally filters and decimates using the b200d3 filter. The level of the simulated signal is selected so that the mean value of the power data in the top panel equals to the mean power in the measured data in Fig. 3. The effect of the filter is simulated by directly computing the convolution of the samples and the filter's inputs response. This was how we initially verified that our understanding of the "oscillations" in Fig. 3 was on a right track. For more general use, the script ddfsimu.m now provides similar plots, such as in Fig. 21. Those are computed in a different way, though.

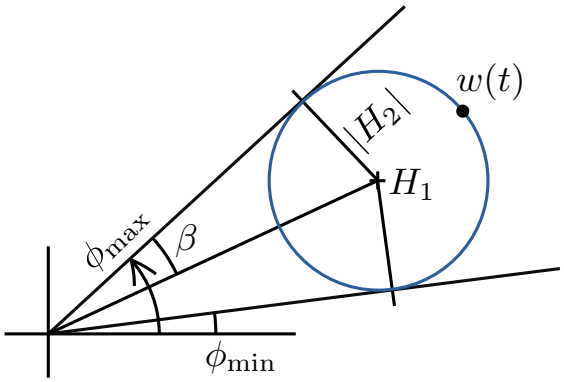

Figure 19: Geometry of the beat in the complex plain. With the primary Doppler $f_{RX} - f_{TX}$ removed, the noise-free complex signal samples $w_n = w(t_n)$ in the final baseband trace a circle centred on the complex number $H_1 = H(f_1)$ and having radius of $|H_2| = |H(f_2)|$, where f_1 is the wanted, near-DC, component of the original analog spectrum after it is translated to the primary baseband by bandpass sampling and multiplication by the NCO frequency, and f_2 is the unwanted, higher-frequency spectral component (deriving from the positive frequency component of the analog spectrum), which ideally should be entirely absent due to post-detection filtering. With $|H_2| > 0$, both the magnitude and phase of the samples vary periodically, giving rise to the observed beat, the oscillations in both power and phase. The figure suggests that the peak-to-peak variation in $|w(t)|^2$ equals to $(|H_1| + |H_2|)^2 - (|H_1| - |H_2|)^2 = 4|H_1||H_2|$. The peak-to-peak phase variation of $w(t)$ corresponds to twice the angle β of the figure, solvable from $\sin(\beta) = |H_2|/|H_1|$.

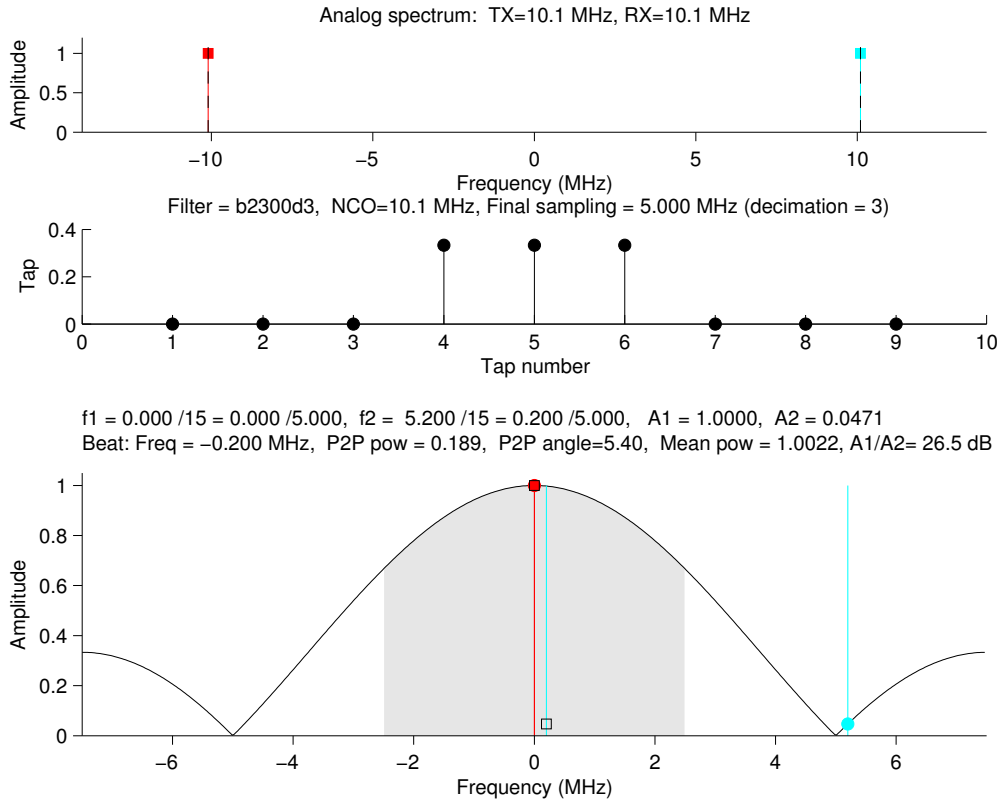

Figure 20: Beat analysis of the quip 5 MHz channel, case 1. This is a standard ddfsimu analysis plot for the 5 MHz decimating filter b2300d3. Input is a sinusoidal signal at 10.1 MHz, and also the NCO is set to 10.1 MHz. The corresponding simulation plot is in Fig. 21. Another case is shown in Fig. 22, and that figure also provides a more detailed caption about the plot contents.

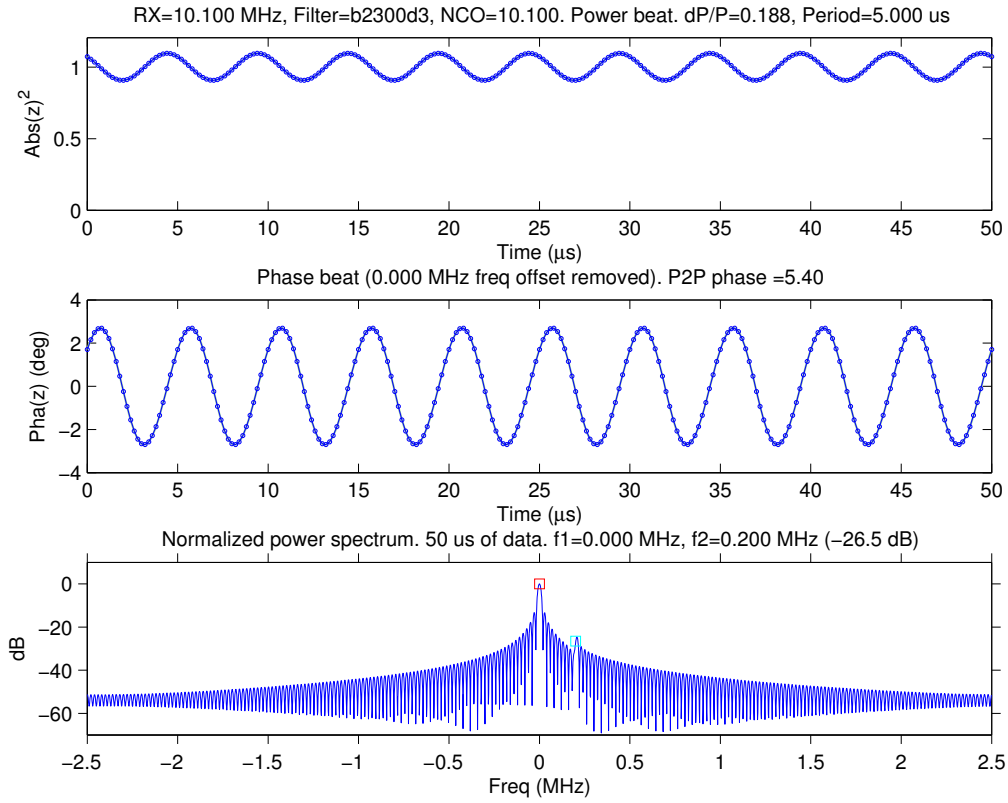

Figure 21: Beat simulation on the quip 5 MHz channel, case 1. This is a standard ddfsimu simulation plot for the decimating filter b2300d3, used in quip for 5 MHz sampling. Input is a sinusoidal signal at 10.1 MHz, and also the NCO is set to 10.1 MHz. The corresponding analysis plot is in Fig. 20. Three other beat simulations are shown in Fig. 23, Fig. 24 and Fig. 24.

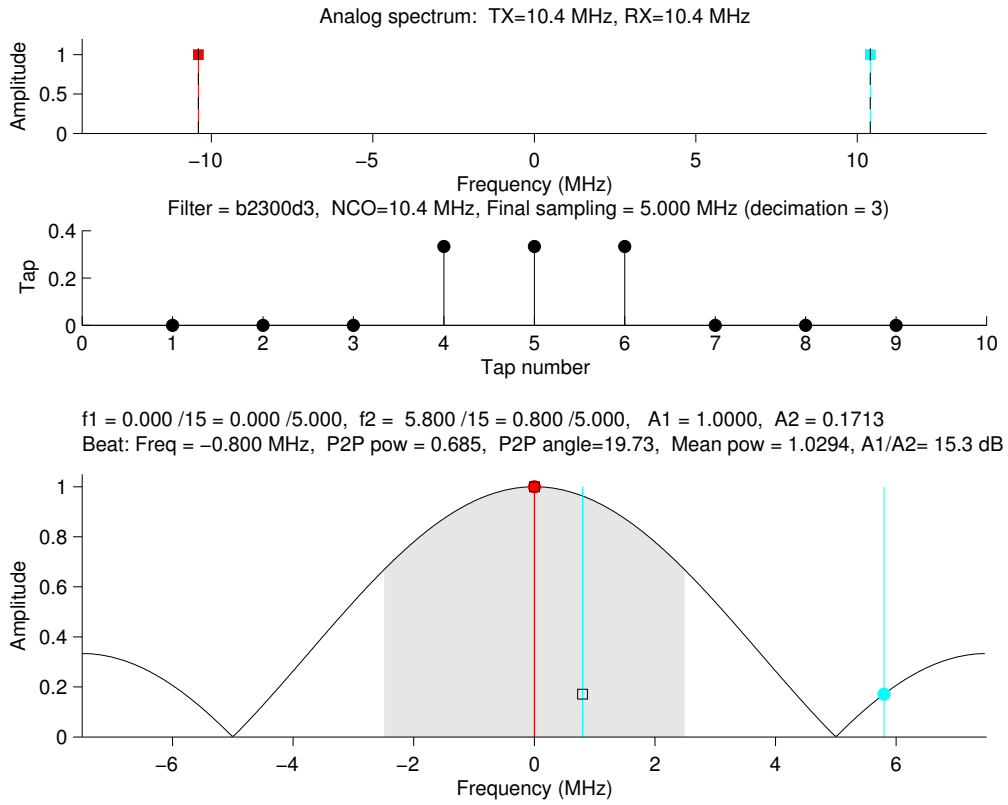

Figure 22: Beat analysis of the quip 5 MHz channel, case 2. The top panel shows the delta-function-like spectrum of the real sinusoidal input at 10.4 MHz. Also the NCO is set to 10.4 MHz. The middle panel shows the impulse response of the b2300d3 filter. The bottom panel shows the magnitude $|H(f)|$ of the filter's transfer function, plotted on linear scale, over the whole primary baseband from −7.5 MHz to +7.5 MHz. The final baseband $[-2.5, +2.5]$ MHz is indicated by the grey-colorer area. Locations of the spectral components both in the primary baseband (solid circles) and the final baseband (open squares) are indicated. The header of the bottom panel gives quantitatively the frequencies both at the primary and the final baseband; the values A_1 and A_2 of $|H(f)|$ at the locations of the spectral components on the primary baseband (in the main text these are denoted by $|H_1|$ and $|H_2|$); the beat frequency; peak-to-peak power variation, peak-to-peak phase variation; mean power; and the amplitude ratio of the two spectral components. The plot was generated with ddfsimu.m. The corresponding simulation plot is in Fig. 23.

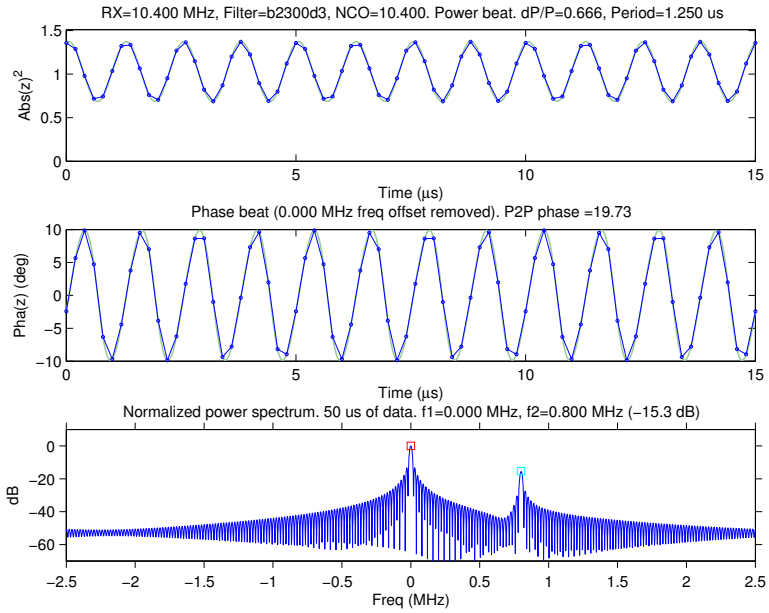

Figure 23: Simulation on a b2300d3 channel with RX=TX=F12.

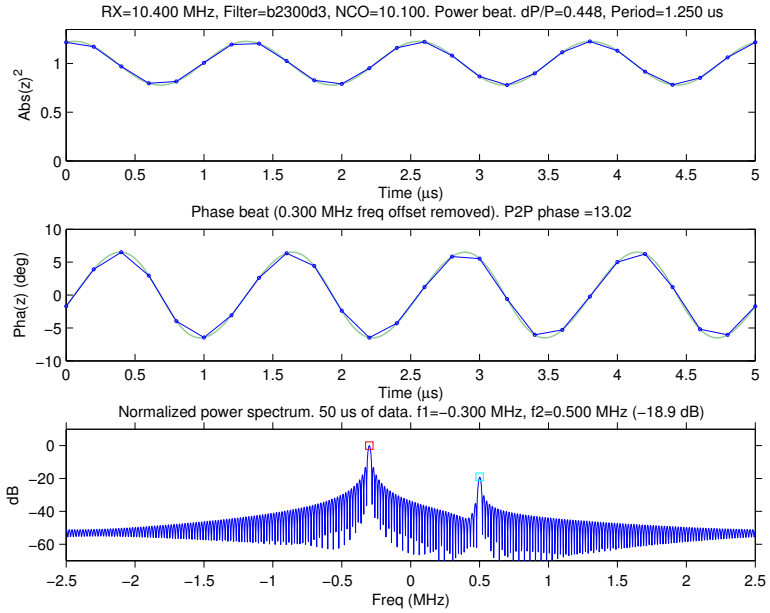

Figure 24: Beat aliasing, plot A. See the caption of Fig. 25.

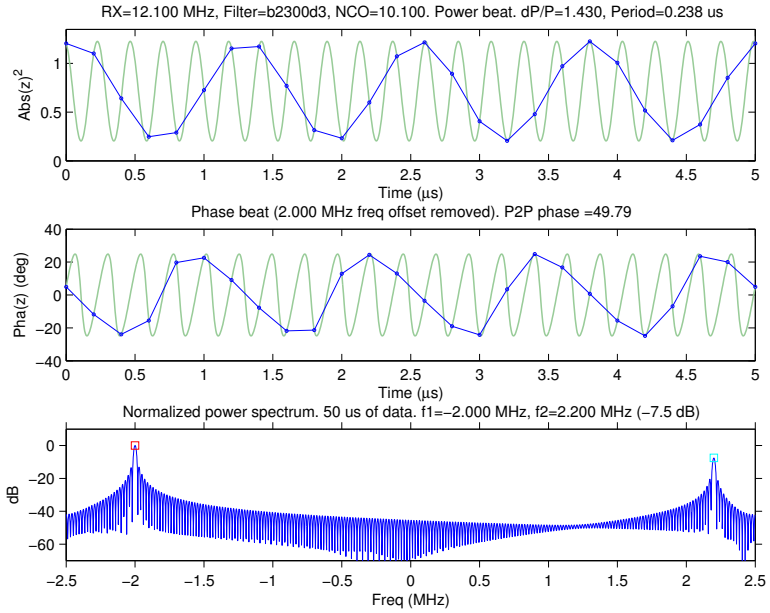

Figure 25: Beat aliasing, plot B. This figure together with Fig. 24 illustrates beat aliasing where two very different, correctly sampled signals give the same apparent beat frequency, 800 kHz in this case (the blue-colorer time-series curves). Note that the beat curves here and in Fig. 24 are not identical, though, and the underlying continuous-time beat, the solid green curves, is of course entirely different. These figures were generated with ddfsimu.m.

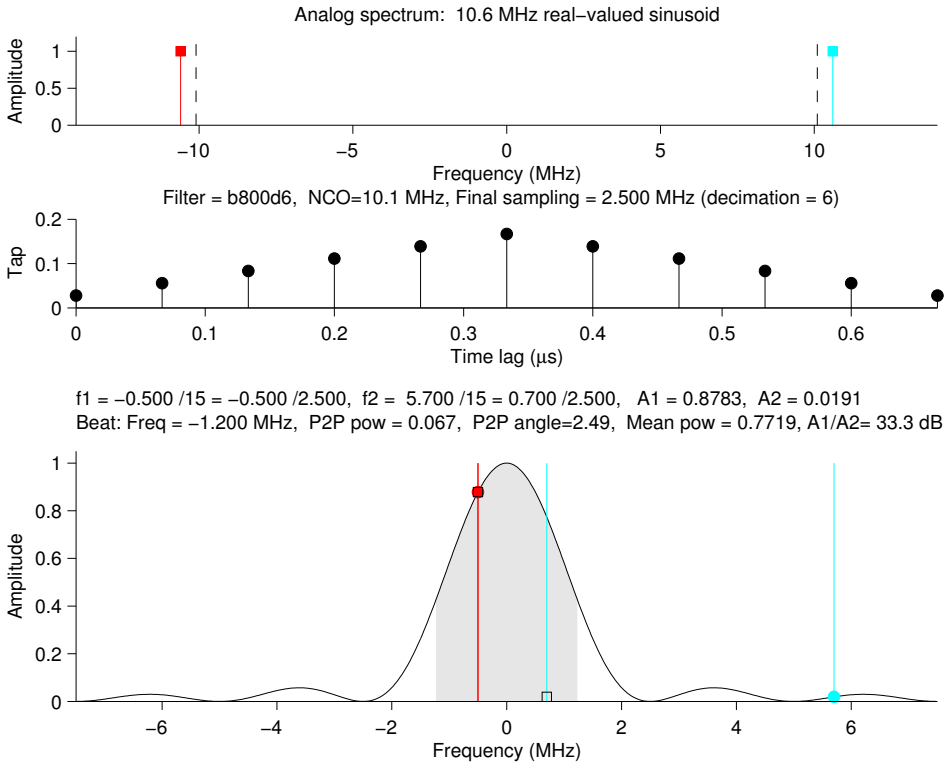

Figure 26: Plot 1/2 of filter b800d6 simulation with f_{RX} = 10.6, f_{NCO} = 10.1 MHz. The channel is tuned to 10.1 MHz (UHF F13). There is a 0.5 MHz "Doppler-shift" in the input signal. Final sampling rate is 2.5 MHz. The unwanted detection product (blue) is 33.3 dB down from the wanted one (red). This is sufficient to cause a 8.7% peak-to-peak variation in power, and a 2.49° peak-to-peak variation in phase (on top of the phase variation due to the Doppler).

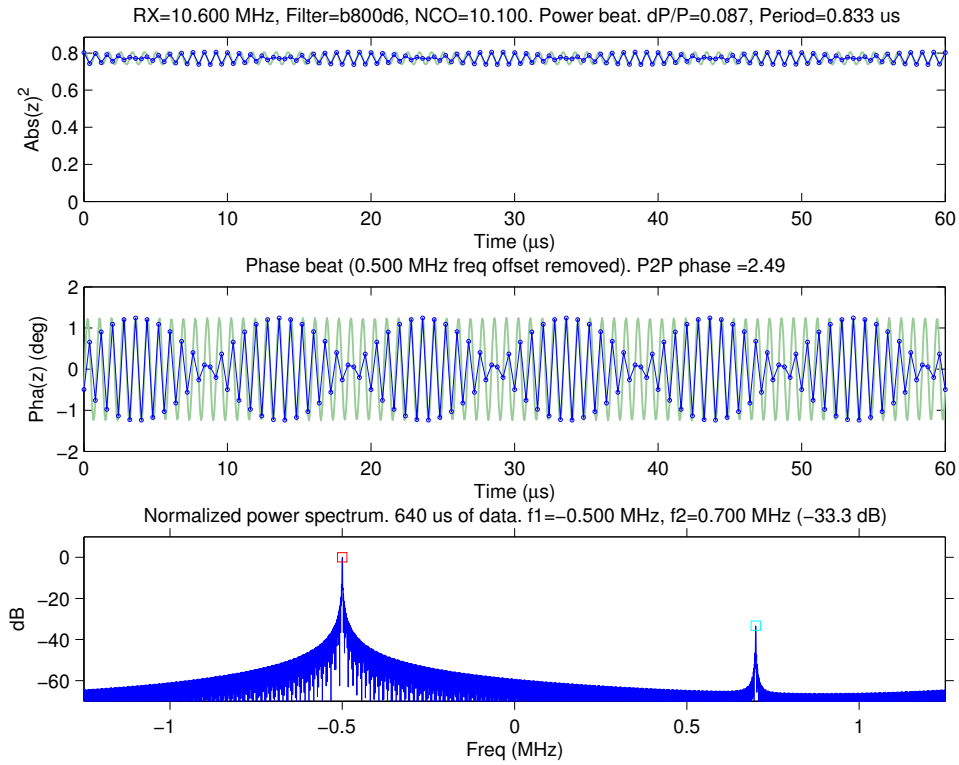

Figure 27: Plot 2/2 of filter b800d6 simulation with f_{RX} = 10.6, f_{NCO} = 10.1 MHz. The beat frequency is −1.2 MHz, the period is 0.833 μs. The period is only slightly longer than two times the 0.4 μs sampling interval, giving a most typical beat pattern of samples.

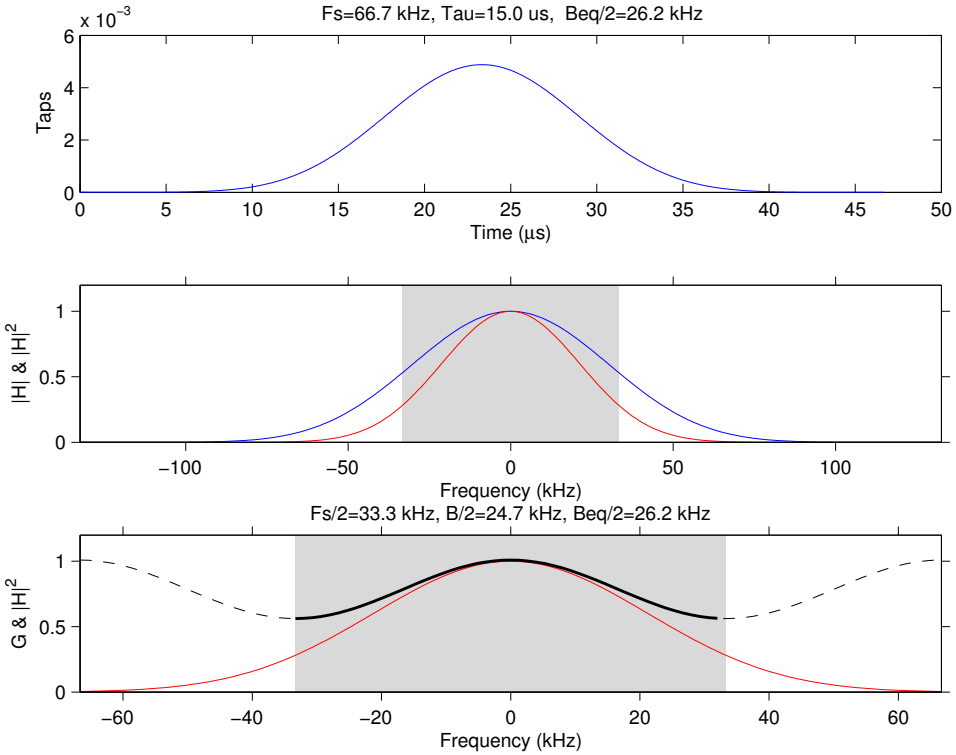

Figure 28: The decimating filter b25d255. This decimating digital filter is for 15 μs final sampling interval, $f_s' = 66.7$ kHz final sampling rate. The top panel shows the impulse response. There are 701 taps in this single-filter, single-decimator equivalent representation of the HSP43220 operation. The middle panel shows the magnitude (blue) and squared magnitude (red) of the equivalent filter's transfer function $H(f)$ in the interval $[-2f_s', 2f_s']$. The grey area shows the final baseband $[-f_s'/2, f_s'/2]$. The bottom panel shows the noise response $G^{\zeta}(f)$ of the filter (black curve) in the interval $[-f_s', f_s']$. The noise response is computed from Eq. (104), which assumes white noise input. The red curve in the bottom panel is the squared magnitude of the transfer function, the same as in the middle panel, plotted here for comparison. The greyed area indicates the final baseband. The filter's noise equivalent bandwidth is 52.3 kHz and the 3 dB bandwidth is 49.4 kHz. This filter is currently used in the ion-line channels of the UHF bella experiment of IH.

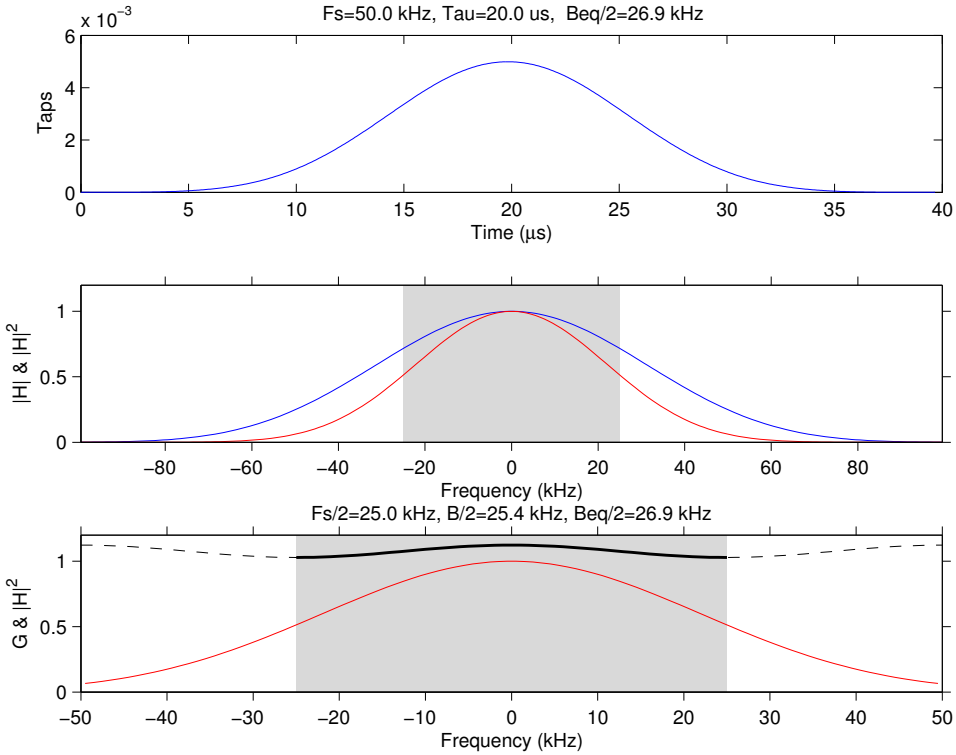

Figure 29: The decimating filter b25d300. This decimating digital filter is for 20 μs final sampling interval, $f_s' = 50.0$ kHz final sampling rate. The top panel shows the filter's impulse response. There are 596 taps in this single filter, single decimator equivalent representation of the HSP43220 operation. The middle panel shows the magnitude (blue) and squared magnitude (red) of the equivalent filter's transfer function $H(f)$ in the interval $[-2f_s', 2f_s']$. The grey area indicates the final baseband $[-f_s'/2, f_s'/2]$. The bottom panel shows the noise response $G^{\zeta}(f)$ of the filter (black curve) in the interval $[-f_s', +f_s']$. The noise response is computed from Eq. (104). White noise input to the filter is assumed. The red curve is the squared magnitude of the transfer function as in the middle panel, plotted for comparison. The greyed area indicates the final baseband. The noise equivalent bandwidth is 53.8 kHz, and the 3 dB bandwidth is 50.8 kHz. This filter is currently used on the ion-line channels in the UHF beata experiment of IH.

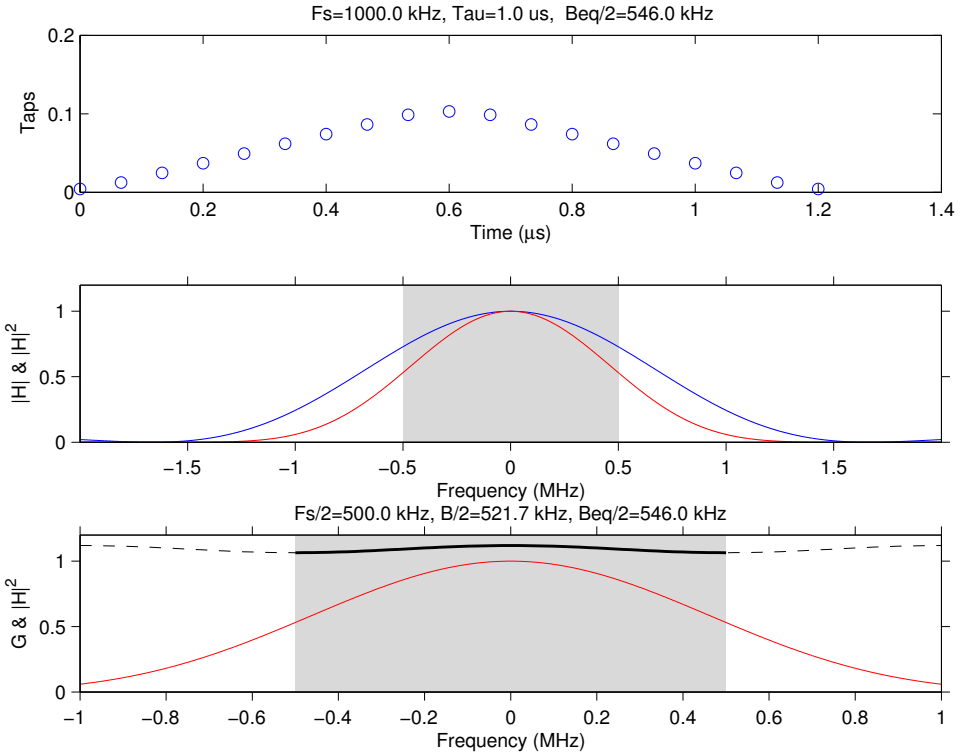

Figure 30: The decimating filter b500d15. This decimating digital filter is for 1.0 μs final sampling interval, $f_s' = 1.0$ MHz final sampling rate. The top panel shows the filter's impulse response. There are 19 taps in the single-filter, single-decimator equivalent representation of the HSP43220 operation. The middle panel shows the magnitude (blue) and squared magnitude (red) of the filter's transfer function $H(f)$ in the interval $[-2f_s', 2f_s']$. The grey area indicates the final baseband $[-f_s'/2, f_s'/2]$. The bottom panel shows the periodic-by-f_s' noise response $G^{\zeta}(f)$ of the filter (black curve) in the interval $[-f_s', +f_s']$, computed from Eq. (104). White noise input was assumed. The red curve in the bottom panel is the squared magnitude of the transfer function, same as in the middle panel, plotted here for comparison. Again, the greyed area indicates the final baseband. The noise equivalent bandwidth is 1.09 MHz and the 3 dB bandwidth is 1.03 MHz. The filter had earlier been used in my satellite measurement experiment leo. The filter was now used in the quip experiment in May 2014 on the channel 1, which collected ion-line voltage-level data (Fig. 1).

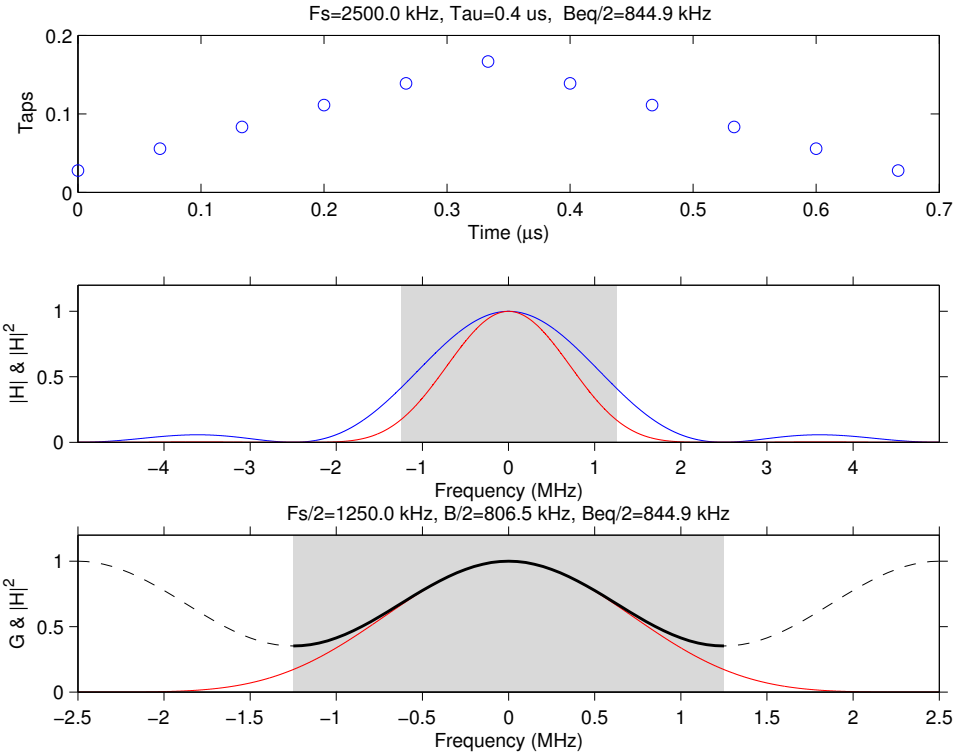

Figure 31: The decimating filter b80od6. This decimating digital filter is for 0.4 μs final sampling interval, $f_s' = 2.5$ MHz final sampling rate. The top panel shows the filter's impulse response. There are 11 taps in the single-filter, single-decimator equivalent representation of the HSP43220 operation. The middle panel shows the magnitude (blue) and squared magnitude (red) of the equivalent filter's transfer function $H(f)$ in the interval $[-2f_s', 2f_s']$. The grey area indicates the final baseband $[-f_s'/2, f_s'/2]$. The bottom panel shows the noise response $G^{\zeta}(f)$ of the filter (black curve) in the interval $[-f_s', +f_s']$. The noise response is computed from Eq. (104). White noise input to the filter is assumed. The red curve is the squared magnitude of the transfer function, the same as in the middle panel, plotted here for comparison. The greyed area again indicates the final baseband. The noise equivalent bandwidth is 1.69 MHz, and the 3 dB bandwidth is 1.61 MHz. This filter is currently used for the plasma-line channels in the UHF beata experiment of IH.

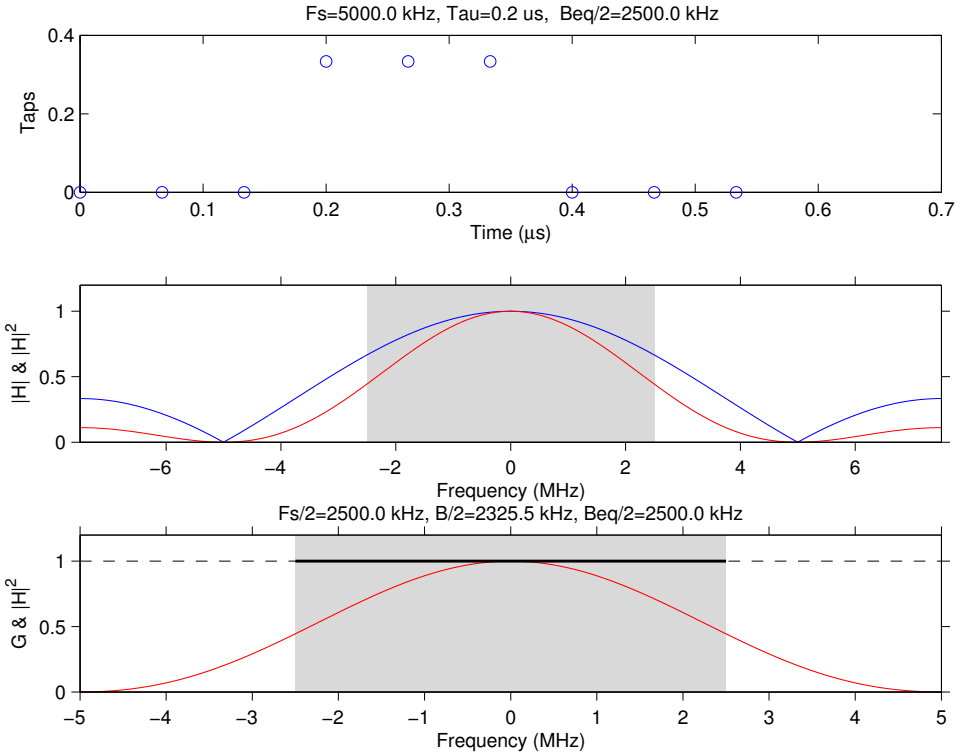

Figure 32: The decimating filter b2300d3. This decimating digital filter is for 0.2 μs final sampling interval, $f_s' = 5.0$ MHz final sampling rate. The top panel shows the filter's impulse response. There are only three non-zero taps in this single-filter, single-decimator equivalent representation of the HSP43220 operation. The middle panel shows the magnitude (blue) and squared magnitude (red) of the equivalent filter's transfer function $H(f)$ in the whole primary baseband $[-7.5, 7.5]$ MHz. The grey area indicates the 5 MHz wide final baseband $[-2.5, 2.5]$ MHz. The bottom panel shows the—now constant—noise response $G^{\zeta}(f)$ of the filter (black curve) in the interval $[-f_s', +f_s']$. The noise response is computed from Eq. (104). White noise input to the filter is assumed. The red curve in the bottom panel is the squared magnitude of the transfer function, the same as in the middle panel, plotted here for comparison. The greyed area indicates the final baseband. The noise equivalent bandwidth of the filter is 5.0 MHz, and the 3 dB bandwidth is 4.66 MHz. This filter was for the first time used in the quip experiment in May 2014.

9 Matlab routines

Header sections, as of September 5th, 2014, of the three Matlab routines referred to in this document are listed below. The code itself is available in the EISCAT HQ software repository, in the subdirectory ddf of the CVS module fir. Use "cvs get fir/ddf" to get a copy of it.

Listing 1: Routine ddfplan.m to design filters for EISCAT digital receiver.

```
% DDFPLAN (TYPE,HDEC,HSEC,FDEC,FSEC,FTAPS)
%   Design routine for EISCAT Digital Filters.
%
%   The function calculates and plots impulse response, power
%   response (square of filter transfer function), and
%   ''white noise response'', given:
%
%   TYPE:    Must be either 'b' for 15 MHz sampling clock,
%            or 'w' for 10 MHz sampling clock.
%   HDEC:    Decimation factor of the high order filter (1...1024).
%            Parameter ignored in bypass mode (see HSEC).
%   HSEC:    Number of sections in the high order filter (1..5).
%            Set HSEC to zero to bypass the high order filter.
%   FDEC:    Decimation factor of the FIR (1..16).
%            Parameter ignored in bybass mode (see FSEC).
%   FSEC:    Number of sections in modelling the FIR with boxcar
%            sections.
%            Set FSEC to 0 to bypass the low order FIR section (but
%            see next param).
%   FTAPS:   Number of taps in one boxcar section in the FIR model.
%            If FSEC == 0 and FTAPS == 3, we use FIR = [0 1 0],
%            otherwise with FSEC == 0, this parameter is ignored.
%
%   Filter parameters are written to file firpar.fir. For production
%   use, the file must be manually renamed to the format
%   <x><kkk>d<mmm>.fir, where <xis the TYPE, <kkkk> is 3dB point in kHz,
%   and <mmm> is total decim.
%
%   The graphics is saved to files firpar.eps, firpar.ps, and firpar.png.
%
%   (c) EISCAT Scientific Association 1998-2014

% Nov-26-1998 Jm: using fs = 15 MHz for Pentek testing
% Nov-27-1998 Jm: back to 10 MHz.
% Nov-15-1999 Jm: location of filter poles added
% Nov-25-1999 Jm: fs = 15 MHz
% 18-Feb-2000 Jm
%  1-Apr-2004 Jm: Code somewhat restructured, plots improved,
%                 more info put into firpar file header.
% 15-Mar-2007 Jm: Compute noise equivalent bandwidth.
%                 Provide firpar.png.
%  3-Oct-2011 Jm: Check for minimum Hdec added.
% 21-Apr-2014 Jm: The Hdec min check changed from error to warning.
% 23-Apr-2014 Jm: Check for the number of FIR taps added.
% 27-Apr-2014 Jm: HDF and FIR bypass modes implemented. Specify Hsec or
%                 Fsec as zero to achieve this.
%   -Aug-2014 Jm: Generate noise plot also. This is still experimental,
%                 as it slows things down quite a lot.
% 24-Aug-2014 Jm: The term "pole" was used wrongly---filter zeros were
%                 called poles (!)
```

Listing 2: Routine ddfsimu.m to simulate filter output for a sinusoidal input. The routine was especially written to analyse the beat behaviour of the wideband filters.

```
% DDFSIMU
% Inspect an EISCAT digital channel, especially for the beat behaviour.
%
% The routine simulates channel output assuming a noiseless sinusoidal
% input to the A/D, with adjustable frequency, and adjustable
% receiver NCO setting.
%   Two plots are generated: 1) an analysis plot which shows the analog
% spectrum, the filter inpulse response, and the filter tranfer function
% with the frequency mapping, and 2) a simulation plot that shows time
% series of power, time series of phase (Doppler-removed), and the power
% spectrum of the simulated data.
%
% INPUT
%   1. FIRFILE (required)       .fir file name,
%                               e.g. '/kst/dsp/fir/b80od6.fir'
%   3. FRX_HZ (optional)        RX frequency at IF2 (before A/D) in Hz,
%                               default = 10.1E6.
%   2. FNCO_HZ (optional)       NCO frequency in Hz,  Default = FRX_HZ.
%   4. WINLEN_SEC (optional)    Window length for time series plotting,
%                               default = 50E-6.
%   5. FFTLEN_SEC (optional)    Data length for FFT, in seconds,
%                               default = WINLEN_SEC.
%
% OUTPUT is two structures, OUT and IN, with the following fields.
%   OUT.
%       f1_pri      Frequency of the neq freq component of the real analog
%                   input mapped to primary baseband.
%       f2_pri      Frequency of the pos freq component of the real analog
%                   input in mapped to primary baseband.
%       f1_fin      Frequency of the neq freq component of the real analog
%                   input mapped to final baseband.
%       f2_fin      Frequency of the pos freq component of the real analog
%                   input mapped to final baseband.
%       f_beat      Beat frequency (the primary one; not the alised one)
%       h1_dB       Power of f1 after filtering = |H(f1_pri)|^2.
%       h2_dB       Power of f2 after filtering  = |H(f2_pri)|^2.
%       pow_p2p     Peak to peak beat power variation.
%       pow_ave     Mean power.
%       pha_p2p     Peak to peak beat phase variation in degr.
%   IN.
%       firname     Name of the filter
%       frx_Hz      RX frequency at IF2
%       fnco_Hz     NCO frequency
%       fs_pri      Primary sampling frequency
%       decim       Decimation factor
%       fs_fin      Final sampling frequency
%       tau_fin     Final sampling interval
%
% EXAMPLE
%   [out,in] = ddfsimu ('/kst/dsp/fir/b80od6.fir', 10.4E6, 10.1E6, 60E-6,
    1000E-6)
%
% (c) EISCAT Scientific Association 2014-

% 24-Aug-2014 Jm    Initial version.
```

Listing 3: Routine get_impresp.m to extract filter parameters from a .fir file. This routine can be used to fetch the filter impulse response from the filter definition file produced by ddfplan.m.

```
% GET_IMPRESP - Get impulse response from an EISCAT filter definition file.
% INPUT
%       FIRFILE     .fir file path.
%       P_DTAU      Time step in miscrosec of the returned continuous-
%                   time h(t).
%       PLOT_FLAG   (Optional) If nonzero, plot the impulse response of
%                   the HDF, FDF, and DDF.
% OUTPUT
%       IMPRESP     The continuous-time (pchip-interpolated) version of
%                   impulse response. The IMPRESP is normalized to unit
%                   area.
%       TSTART      Start time of the interpolated IMPRESP.
%       DDF         Taps of the equivalent filter.
%       DECIM       Decimation factor of the equivalent filter.
%
% EXAMPLE
%       >> [impresp, to, taps, decim] = get_impresp('b25d300.fir',1);
%
% (c) EISCAT Scientific Association 1998-

% 24-Jul-1998 Jm
% 23-Feb-2001 Jm: Handling of odd number of taps changed.
% 13-May-2014 Jm: Plotting made more precise espcially for high-speed
%                 filters. Also note that for those, the returned
%                 "interpolated" impulse response can be a pretty bad
%                 approximation. Also return the normalized taps of
%                 the equivalent filter h = conv(h1,h2^(M1)),
%                 where h1 is the HDF and h2^(M1) is zero-stuffed FDF,
%                 with M1-1 zeros between each pair of taps of the FDF.
%  6-Aug-2014 Jm: Also return the total decimation factor M = M1*M2.
```