Re: Best way to measure precise harmonics?

Source: http://sci.tech−archive.net/Archive/sci.electronics.design/2007−10/msg03710.html

---

- *From*: Tom Bruhns <k7itm@xxxxxxx>
- *Date*: Fri, 19 Oct 2007 11:29:10 −0700

On Oct 19, 9:00 am, Martin Brown <|||newspam...@xxxxxxxxxxxxxxxxxx>
wrote:

> On Oct 19, 4:32 pm, "Ken S. Tucker" <dynam...@xxxxxxxxxxxx> wrote:
>
>> On Oct 19, 1:37 am, Martin Brown <|||newspam...@xxxxxxxxxxxxxxxxxx>
>> wrote:
>>
>>> On Oct 18, 2:22 pm, eromlignod <eromlig...@xxxxxxx>
>>> wrote:
>>>
>>>> I need to find the component harmonic
>>>> frequencies of an AF wave and I
>>>> need for it to be pretty precise (+/− .001 Hz
>>>> or so). I have access
>>>> to a spectrum analyzer, but it just doesn't
>>>> seem to be precise enough
>>>> (or I'm using it wrong). It gives me peaks in
>>>> a frequency domain, but
>>>> they are not pinpoint lines, ostensibly due to
>>>> a limited−sample FFT.
>>>
>>> That goes with the territory. To obtain a +/− 0.001 Hz
>>> frequency
>>> resolution you would have to measure the signal for ~1000
>>> seconds
>>>
>>>> Are there any other devices or methods to
>>>> obtain accurate frequencies
>>>> of each harmonic to three decimal places?

> > > Thanks for any suggestions
> > > you might have.

> > FFT will do it if you can supply enough data in the time
> > domain
> > (choose a 2^N FFT). Practical implementations will require
> > potentially
> > anti−alias regridding and a few other tweaks to sort out
> > boundary
> > conditions.

> > Some hardware FFT based analysers can zoom in on a region
> > of interest,
> > but non of them can get around the uncertainty principle. A
> > short
> > burst of pure tone decaying in amplitude always contains a
> > range of
> > frequencies around the fundamental.

> Question: How good are todays A to D Converters?

Good enough for studio quality digital audio to have taken over from analogue.

> Will the conversion introduce serious artifacts?

Shouldn't do if it is done correctly. The most important thing is to have an analogue brick wall filter to ensure that no out of band frequencies reach the input to the ADC. Any timing phase jitter in the converter will also hurt.

> I'm thinking that once the wave form is in digital
> form it's just software to compare that to a perfect
> sine wave at various frequencies. I'm sure that's
> be done.

Just a SMOP... But for these volumes of data it requires some skill to obtain the optimum results for a high dynamic range spectrum containing a fundamental and a bunch of its near harmonics.

> FFT is just a quick way to decompose a signal into its frequency components. Classical slow DFT would be glacially slow on large datasets unless you were only looking at a handful of likely frequencies.
>
> Regards,
> Martin Brown

We've been building FFT analyzers for many years; I can assure you that ADCs that are very linear do a great job in these analyzers. The advent of delta−sigma converters made life a lot easier, and of course for audio they are pretty much universal now. Obviously, if you are looking for overtones in the spectrum of an excited string, and those overtones are very close to the frequency of harmonics of the fundamental, you'll want to know just how much harmonic distortion is being introduced in the signal path. It can come from the transducer that goes from acoustic to electrical, and in the amplifiers ahead of the ADC, and in the ADC itself. It's quite possible to get distortion in the electrical path lower than −100dBc in the audio range, but it's also pretty difficult (in my experience) to find hard specs on the distortion introduced by the acoustic to electrical transducer: microphone or other pickup. If the overtone and fundamental are both pure enough tones, and if the harmonics are enough different in frequency from the overtones, the analyzer can resolve them.

Given a stream of samples from a good audio "card" (or external USB audio port or whatever), the processor in a modern PC should have no trouble at all keeping up doing "zooming" and decimating. That makes the display of the results somewhat easier and the FFT processing can be done real−time, since you're doing FFTs on relatively small blocks of data at a slow data rate (after decimation). Does anyone sell software that actually does all this (nearly) real−time? We used to do it for audio−range analyzers using a custom ASIC chip set, but these days, there's certainly plenty of processing power in a typical PC. We still do it with an ASIC, but now much, much faster.

With respect to determining frequencies, _IF_ I know a priori that I'm dealing with a pure tone (and therefore stable in phase and amplitude), and the signal−to−noise ratio is good, I can determine the frequency to within 0.001Hz with well under 100 seconds of data. The reason is that I know exactly the response of each FFT point to any frequency, and the response of a set of FFT points to the waveform can only have happened with a particular input. It's equivalent, I guess, to fitting a sinusoid to the digitized points; if I am quite sure the input is a sinusoid with unknown frequency, phase, amplitude and perhaps DC offset, I don't need very many samples to nail down those four unknowns. Of course, the difficulty is that I almost never can be really SURE that my input is a pure sinusoid. I must also have enough data points to sufficiently average out whatever noise there is; thus, a really good SNR allows fewer points to determine the sinusoid.

You mentioned filtering to avoid aliasing. That's something else that has been aided a whole lot by the delta sigma converters, since the sample rate is much higher than the highest input frequency of interest. The analog filter can be relatively gentle, and the filtering becomes mainly a digital process; it can be linear phase FIR filters, which makes corrections somewhat easier, too.

Even when you don't know what the input waveform really is, or when you know it contains harmonics and overtones and the like, maybe even multiple "fundamentals," an FFT analyzer can give you a very good picture of what your signal looks like, spectrally. You do need to understand things like "windowing" and what happens if your input frequencies are not integer multiples of 1/(time record length) though.

Cheers,
Tom

.