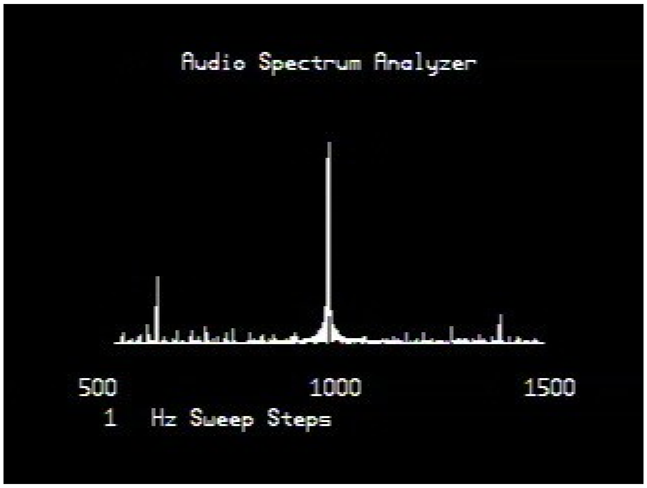
Propeller Spectrum Analyzer for Audio – DEMO



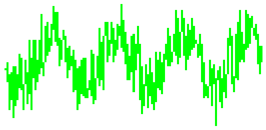
There are so many cool wow factors with this demo, it's hard to know where to start. First, I will begin by introducing a little bit of theory behind how and why the applied method works. The method used here was originally applied to radio where a signal might be there but it is so badly obscured by background noise and other junk that it makes it virtually impossible to distinguish the actual signal from the noise.

Consider a sine wave for a moment... If you take ANY two points on that sine wave that are 180 degrees out of phase from one another then whatever the values are for those two points, should have equal but opposite values right? We will call those two points a "Pole". The method that I am applying in software uses two of these "poles" interleaved by a phase difference of 90 deg to determine if a particular frequency is present.

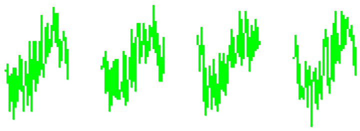
Now, here is actually the tricky part... "noise" as it were is indeterminate, meaning that if you are looking for a particular frequency at the same Pole interval described earlier, noise will show up as an irregular pattern sometimes being positive and other times being negative with no correlation to one another.

Now you ask, well how do you get rid of the "noise" or how can you see past the noise and retrieve the signal? The simple answer is integration… you must cut what you receive in exact size pieces and take the sum of those pieces. Like this…

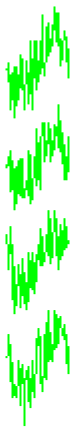
Suppose we have four periods of a wave:



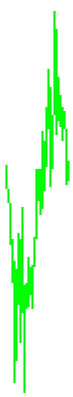
We cut these 4 periods away from each other:



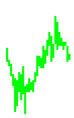
Then we put them one on top of the other and add them:



+



Finally we divide the result by four (for the purpose of this illustration):

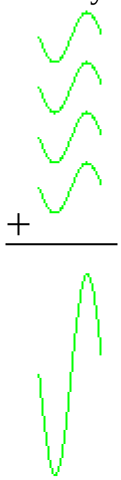


Now, the noise is half as strong, and the signal/noise ratio has increased by a factor of two!

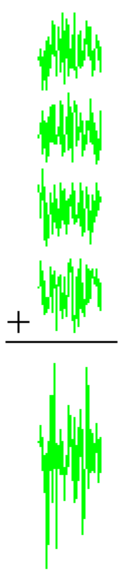
How come?

Here is the explanation:

When you take the sum of four sine periods, the result is a sine period four times larger.



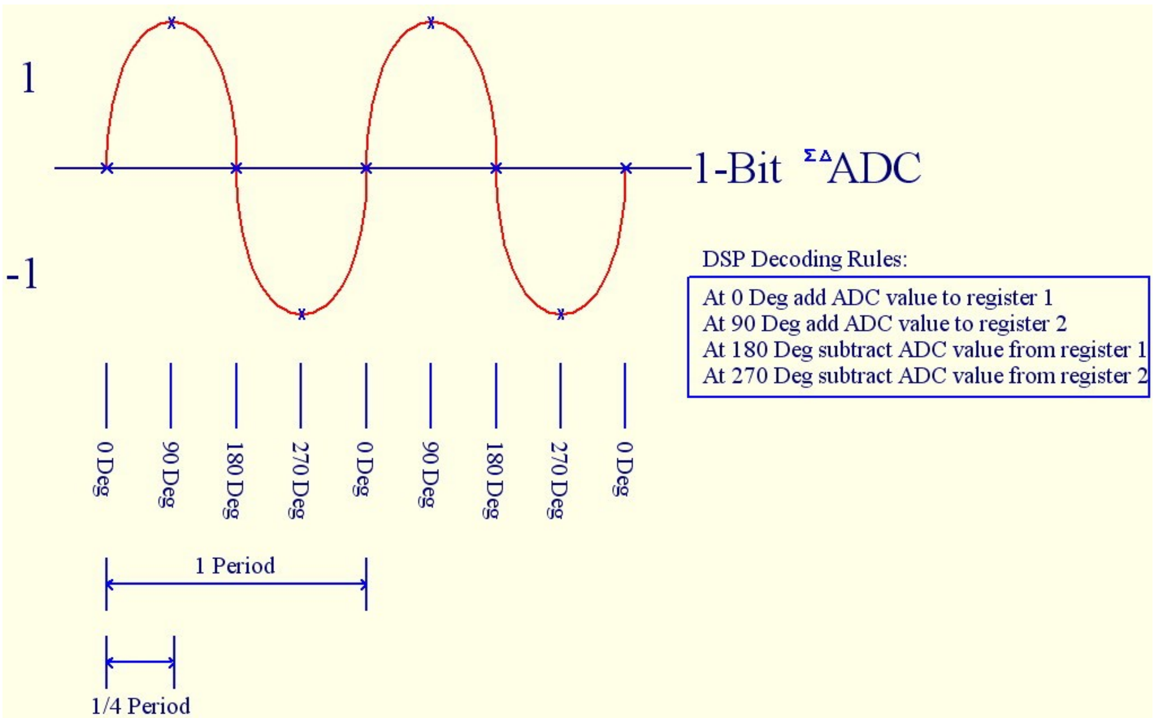
When you make the sum of four noise "periods", the result is only two times larger.



That's because the noise is sometimes positive, sometimes negative, at random. When you add random positive and negative numbers together, they have a way of "eating" each other up.

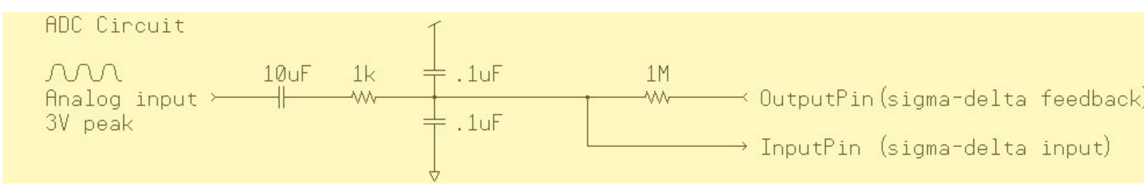
Thus, when we made the sum of four periods, the sine wave grew four times, but the noise grew only two times. The signal/noise ratio was thus increased by two.

Ok, but how do I apply this to a Propeller? It's actually pretty simple, there's only a few Basic elements that we need to consider. All we really care about is if the input signal is above or below zero, and what phase we are in. From this requirement, all we need is a 1-Bit ADC or threshold detector that tells us by a "1" or a "0" if the sampled input is above or below threshold. Instead of determining phase, we just control it by waiting a specific amount of time (equal to ¼ of the Period) before we take the next sample 90 Deg away. In the below illustration register1 relates to one "pole" while register2 relates to a second "pole".



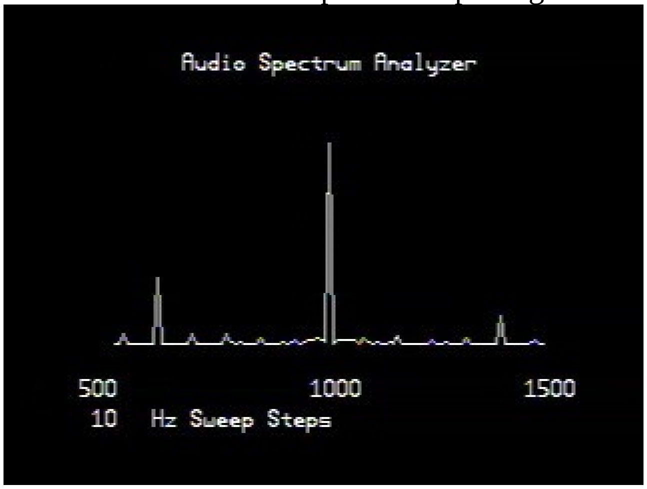
Ok, so I think I get it, this could be implemented in software. I bet it requires a pretty complex external circuit right?

Actually the answer is no, only 5 passive components and that's it!

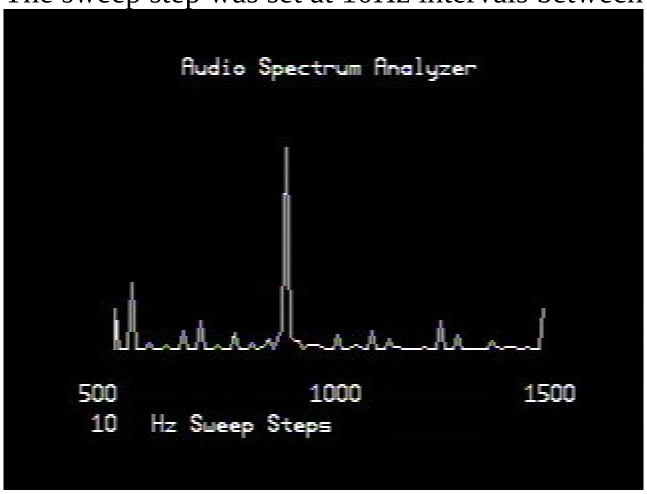


That's all there is to it… Actually, when I first stumbled upon this concept myself, I didn't believe it. The only drawback is that it can be slow, but it does work extremely well. This type of signal detection is used for inter-space communication where the signal may be sampled billions of times at much higher frequencies in order to determine if an actual signal is present or not. The detection bandwidth is very narrow between 5 hertz and 10 hertz which is exceptional.

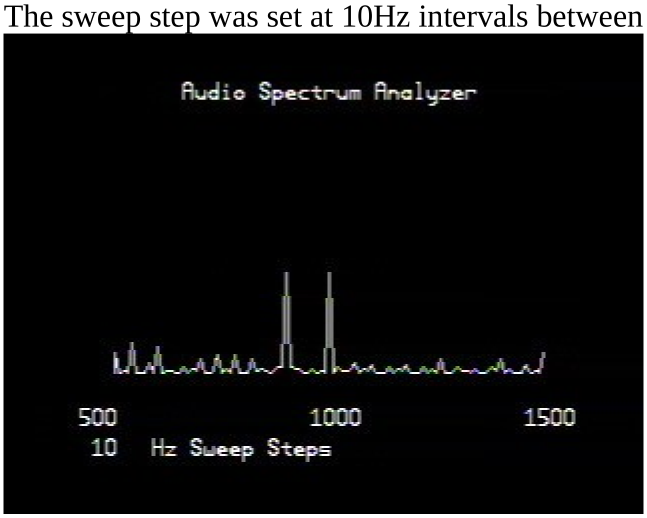
Here are a few test examples that I put together:



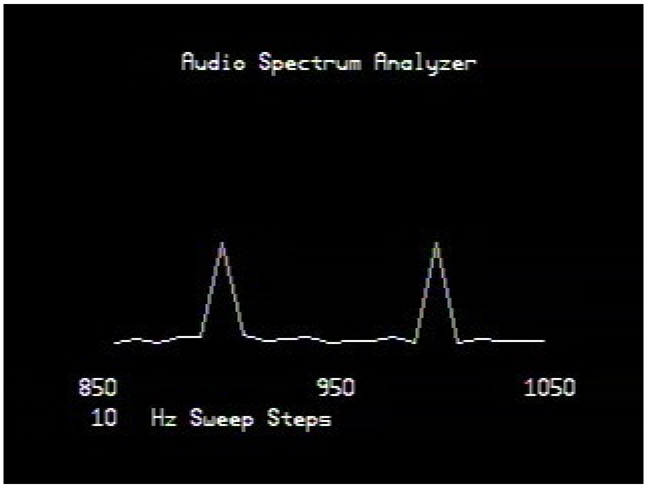
In this sample, the data was sampled for 100 Periods with an input frequency of 1000Hz. The sweep step was set at 10Hz intervals between 500Hz and 1500Hz



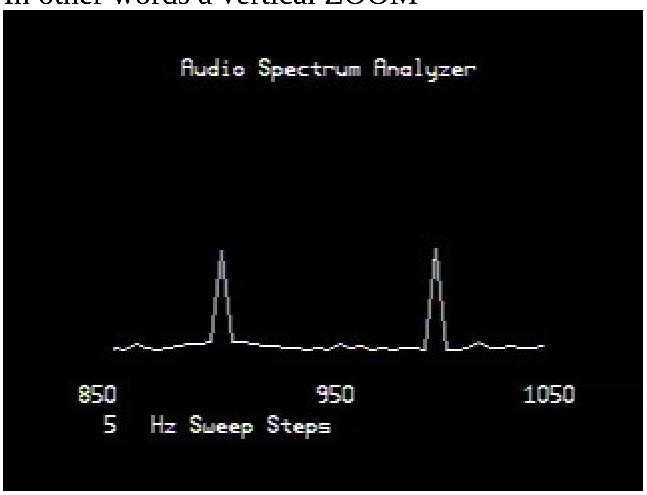
In this sample, the data was sampled for 100 Periods with an input frequency of 900Hz. The sweep step was set at 10Hz intervals between 500Hz and 1500Hz



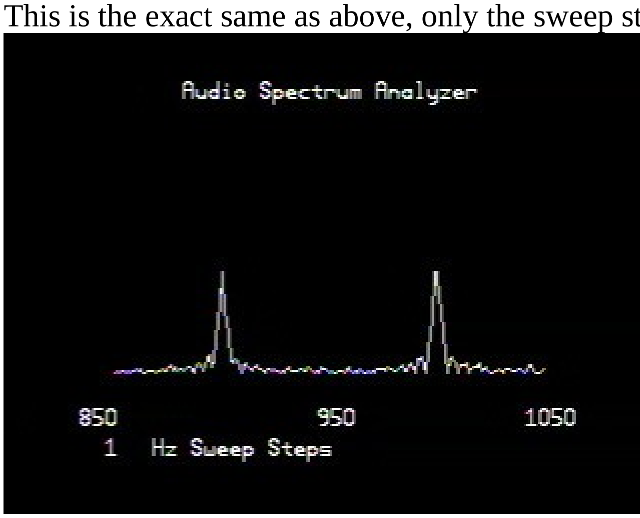
In this sample, the data was sampled for 100 Periods with an input frequency of 900Hz and 1000Hz. The sweep step was set at 10Hz intervals between 500Hz and 1500Hz



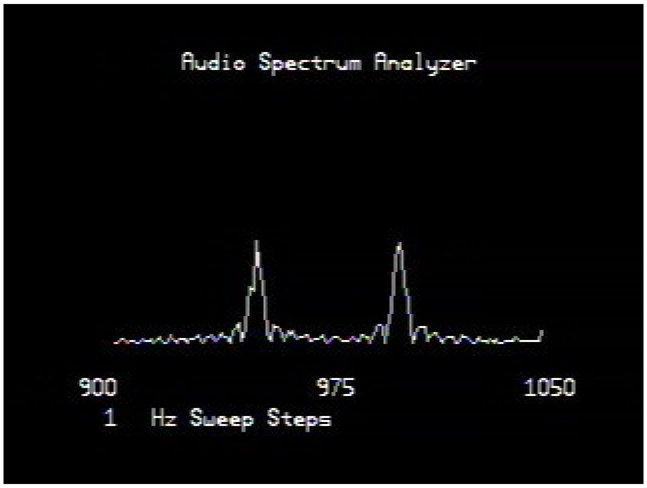
In this sample, the data was sampled for 100 Periods with an input frequency of 900Hz and 1000Hz. The sweep step was set at 10Hz intervals between 850Hz and 1050Hz In other words a vertical ZOOM



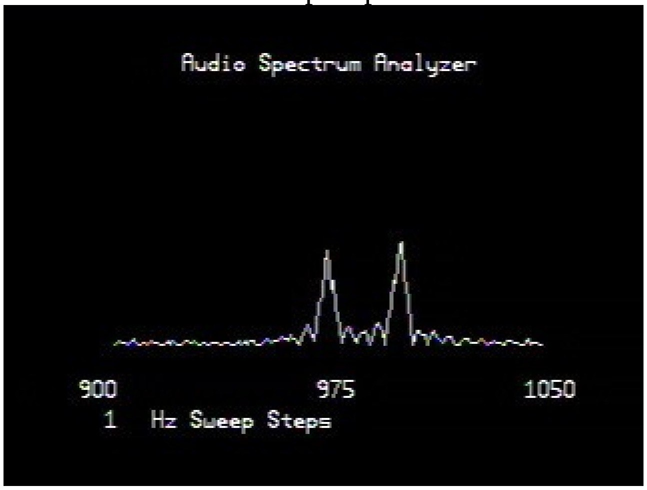
This is the exact same as above, only the sweep step is now 5Hz



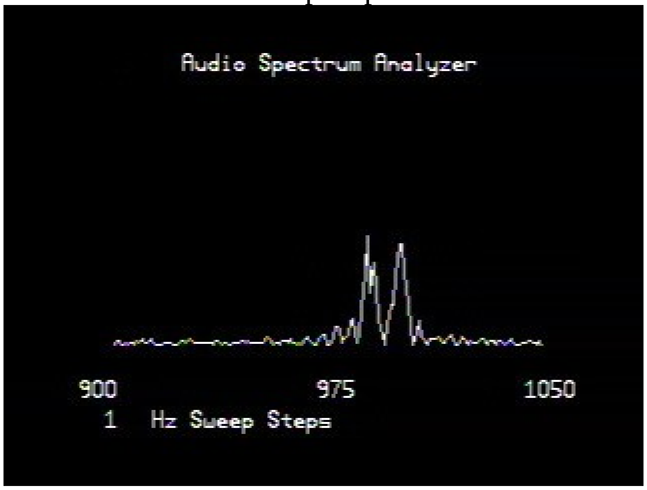
Again, this is the same as above, only the sweep step is now 1Hz



In this sample, the data was sampled for 100 Periods with an input frequency of 950Hz and 1000Hz. The sweep step was set at 1Hz intervals between 900Hz and 1050Hz

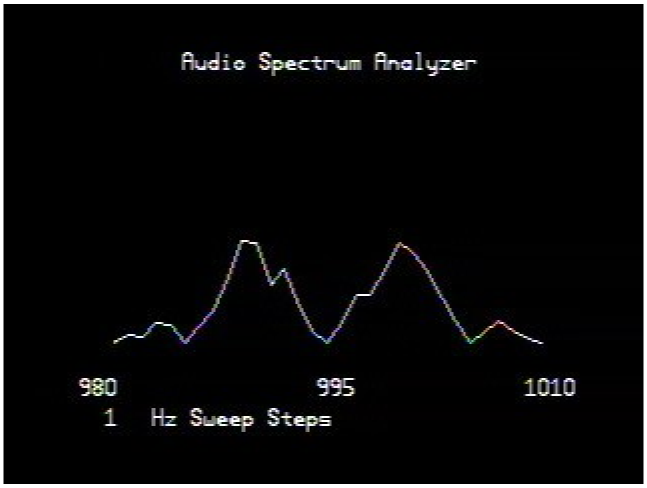


In this sample, the data was sampled for 100 Periods with an input frequency of 975Hz and 1000Hz. The sweep step was set at 1Hz intervals between 900Hz and 1050Hz



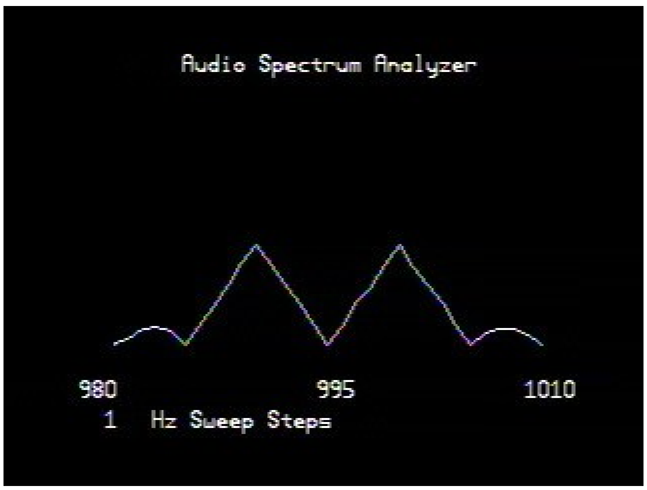
In this sample, the data was sampled for 100 Periods with an input frequency of 990Hz and 1000Hz. The sweep step was set at 1Hz intervals between 900Hz and 1050Hz

Notice, that even though the two frequencies are only 10Hz apart from one another, there is excellent narrow bandwidth discrimination between the two signals.

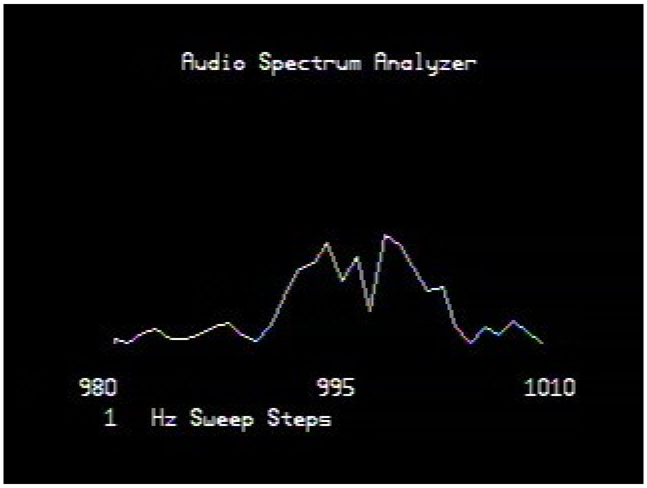


In this sample, the data was sampled for 100 Periods with an input frequency of 990Hz and 1000Hz. The sweep step was set at 1Hz intervals between 980Hz and 1010Hz

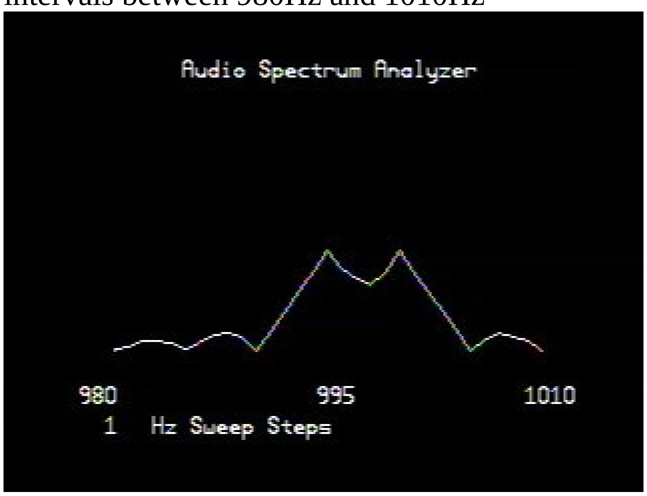
Another vertical ZOOM



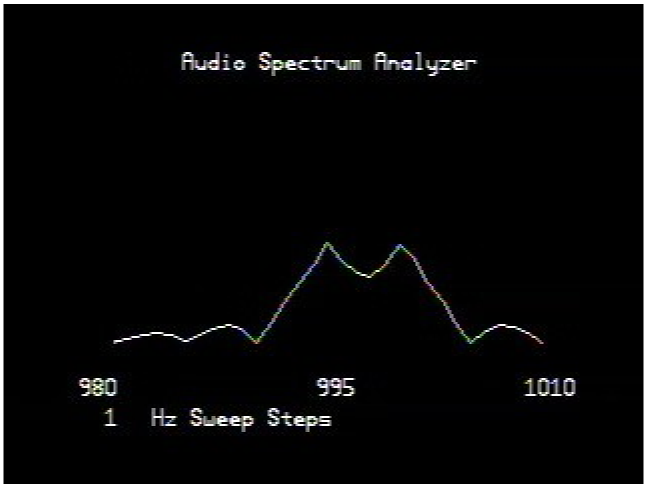
Just to show the effect, this image is exactly the same as above, only it was sampled for 1000 Periods instead of 100



This image illustrates the bandwidth limitation. Here, the data was sampled for 100 Periods with an input frequency of 995Hz and 1000Hz. The sweep step was set at 1Hz intervals between 980Hz and 1010Hz



In this image, the data was sampled for 1000 Periods with an input frequency of 995Hz and 1000Hz. The sweep step was set at 1Hz intervals between 980Hz and 1010Hz



And to be gross, in this image, the data was sampled for 10000 Periods with an input frequency of 995Hz and 1000Hz. The sweep step was set at 1Hz intervals between 980Hz and 1010Hz. As you can see, there is very little difference between the sample of 1000 Periods vs. 10000 Periods… In this situation, virtually all of the random noise has been filtered away, and the result is the influence of a controlled precision signal. This is the type of characteristic we look for when aiming a radio antenna at the stars. A regular "blip" from a precise frequency, any frequency we don't really care, hoping that someone or something other than us is out there among the stars.

Have fun with this one!!! – Beau Schwabe