FlexRadio Systems®
Software Defined Radios

White Paper

TRACKING NOTCH FILTER™

History, Theory and Application

October 2011

Austin, Texas

Stephen Hicks, N5AC, AAR6AM, VP Engineering, FlexRadio Systems
Robert McGwier, Ph.D., N4HY, Professor ECE Virginia Tech

HISTORY AND THE PROBLEM

Most radio amateurs are familiar with the concept of a notch filter. The basic notch filter is designed to remove a carrier from the receive passband when listening to sideband signals. The notch filter eliminates the single frequency carriers, which are generally AM carriers, from broadcast stations, CW signals that are in the passband or interference from either internal (inside the radio) or external signals.

The first notch filters came out in amateur radios in the late-1970's (in the Kenwood TS-830, for example) and notched the IF with a fixed notch width. Most notches could be adjusted within the radio passband by moving a control. In some radios, the notch dial would have to be turned one way for LSB and the other way for USB to increase the notch frequency. For operators suffering from a tone in the audio passband, these controls were a welcome addition.

As digital signal processing (DSP) became available in HF radios, notch controls became more sophisticated including the addition of automatic notch filters. Automatic notch filters find large carriers in the passband and notch them automatically. They are typically based on the *Least Mean Square* (**LMS**) or *Widrow Algorithm* where the control of the action of the filter is left up to an algorithm. These LMS filters are very popular in all modern HF transceivers and are widely used to suppress tones and other interference sources where there is significant tone like content that is sustained over a time period that is longer than the adaptation time constants which control the operation of the filter.

One benefit of an automatic notch is that it will follow the offending carrier as the signal moves. Signals move in the intermediate frequency (IF) when either the signal itself moves or when the radio frequency dial is turned and the IF center frequency is adjusted. Since the automatic notch is based on DSP that detects the offending signal, as the dial is turned some time is required to detect the new signal so it is heard briefly as the dial is turned.

While these filters are very popular, they are often not appropriate to the user or the mode of operation. For example, it is not appropriate to have an automatic tone removal process running when the mode of operation is CW. In this case it much more advantageous to allow the user to control the filter operation. A typical filter is shown in Figure 1. This filter is tuned to 400 Hz. As you can see the filter was set up to suppress a very fairly narrow range of frequencies and to suppress in the best case around -50 dB.

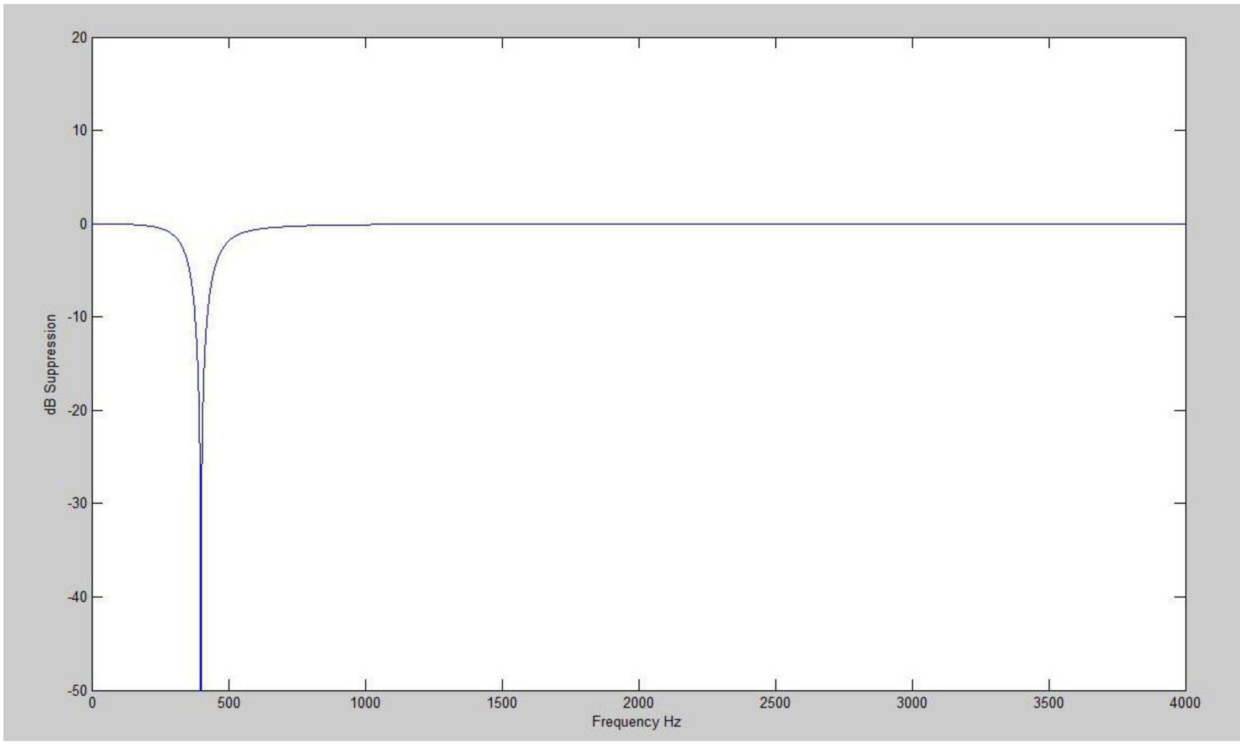

Figure 1: Frequency Response of 400 Hz Notch

Most modern transceivers have at least one manual notch, sometimes two, and an automatic notch filter. Manual notch filters still notch out a specific place in the IF, however, and as the radio frequency is moved, the offending signal is again revealed, but at a different *audio* frequency. This means that a manual notch has good utility if the operator is to remain on a particular frequency, but that its utility is significantly reduced if the radio is frequently tuned. This was the first problem we identified with the modern transceiver manual notch filter.

In addition, many operators encounter interference that persists for long durations. It is not uncommon to hear operators state that they have a "spur on that frequency" which is a known frequency where a spur persists and makes communication on a specific frequency more difficult. There is no way in a modern transceiver to notch out all of these interferers and remember the list of them over time. To some extent this problem is one of user interface — how would a radio with a combination numerical / fixed indicator display indicate where all of the notch frequencies are and how would the operator know if he was on one of them? Further, if he notched a frequency unintentionally how would he later detect this fact and correct the problem? This second problem, we surmised, could also be addressed with a new kind of notch filter.

In addition to these problems, the operator is also typically limited to one filter profile — a specific notch depth and width — which restricts the flexibility that the operator has to notch a signal in cases where the signal is wider than a single carrier or is perhaps a weak signal that doesn't demand an extraordinarily deep notch.

TRACKING NOTCH FILTER™ CONCEPT

The Tracking Notch Filter™ or TNF™ was designed to solve these three problems described above:

1. The notch needs to follow or track the RF frequency of the offending signal in order to keep it notched as the operator changes frequency
2. The notch needs to have the capability of being persistent over time and automatically be enabled as the operator passes by the offending signal later (as it enters the receive passband)
3. The notch should have a variable width and depth for maximum operational effectiveness

Since FlexRadio Systems' products are based on software and digital signal processing, changing the IF frequency of the notch as the radio is tuned in real time is a simple matter of elementary math to calculate the notch frequency and adjust it in real time. This equation would always land on the offending signal since all of a FlexRadio Systems transceiver's frequency calculations and adjustments are made in the digital domain, away from any analog component variations that would reduce the repeatability of this capability in another transceiver. Persistence and variable parameters are also easily solved in software.

TNF™ USER INTERFACE IMPLEMENTATION

Since we were going to introduce the concept of a notch filter that could move in and out of the receive passband, be persistent, and have flexible depth and width, we needed the capability of indicating and controlling these characteristics in a visual manner that would assist the user in quickly comprehending the settings of any individual notch. The interface needed to be intuitive and easy to "read" at a glance. In addition, the user needed to be able to quickly add, move and adjust the interface to meet operational needs.

Without laboring through the iterations of user interface design that we tested, the interface components settled upon are:

1. Notch presence on a frequency would be indicated with a vertical cross-hatched bar in the panadapter
2. Notch persistence would be indicated with color, yellow-green for temporary and green for permanent (remembered between launches of PowerSDR™ in a database)
3. Notch width would be indicated by the width of the vertical bar
4. Notch depth would be indicated by the density of the cross-hatch
5. All notches could be enabled and disabled quickly with a single button (TNF) on the console
6. A single notch could be added with a single button (+TNF)
7. Width could be increased by click-drag from the center of the notch either up or down with a visual indication in real-time of the notch

8. Depth could be changed with a right-click selection on the panadapter and the depth could be “x1, x2 or x3”
9. Persistence could be enabled or disabled with the same right click and a selection for “remember”

On HF, there are some extremely large signals and we would like to suppress them so we can hear really weak ones near them. So our notch filters can be stacked or “cascaded” in filter design language to provide for greater rejection. The notch is applied at the IF, *before* the automatic gain control (AGC) system. This means that a large interferer on CW (say) can be eliminated and then the AGC will lift the weak signal up to audible levels. As we tune for best response to our ears for the signal of interest, we automatically retune the filter to track our tuning.

The display in figure 2 shows four TNF notches, two outside of the current passband (depicted by the light grey vertical band) and two inside the passband. The light green TNF notches are permanent and will stay after PowerSDR is shut down while the yellow-green TNF notches will disappear after PowerSDR is shut down—they are temporary. In this example the two outside of the passband are showing that these notches are in the database, but they are not “active” meaning that the DSP is not actively processing them since they are not in the passband.

Figure 2: TNF Display

TNF™ DSP IMPLEMENTATION

We are looking to discover for our purposes a mathematical description of a notch filter in a digital signal processing system so we can compute the parameters needed to implement it in code. We have chosen a fairly traditional approach which is to determine our filter(s) by breaking them up into quadratic sections. For our purposes, this will mean we are looking to average three input samples and use this result in an average with two previous filter outputs. This is called an infinite impulse response or IIR filter. It provides very good frequency response but has a very short description so the code to implement it is short.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$H(z)$ is the mathematical description of our quadratic section description. This results in a filter that is implemented by the formula

$$y_n = b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} - a_2 y_{n-1} - a_3 y_{n-2}$$

The $y's$ are the filter outputs. The $x's$ are the sample inputs. The formula says the filter output at sample time n is equal to a weighted average (where the $b's$ are the weights) add to a weighted average (where the $-a's$ are the weights) of the previous two filter outputs (those at time $n-1$ and $n-2$. The formula is simple and easy to implement in computer code. The difficulty is in knowing how to determine the $a's$ and the $b's$.

A notch filter is a well-known entity and was described in mathematical terms long before there was digital signal processing. There are continuous time analog versions and their responses have been known for a long time. The mathematical description of their frequency response has been studied by engineering students forever by using what is called a Laplace transform. We are going to simply write down the Laplace transform of notch filter and then state how we transform that into a digital or z-transform version given by our formula for $H(z)$. We will do this without derivation but we will give the important thing for us and that is how to compute the $a's$ and the $b's$. The response or transfer function of an analog, continuous time notch filter is given by

$$H(s) = \frac{s^2 + 1}{s^2 + s/Q + 1}$$

Here Q is the familiar Q to radio amateurs and engineers known as the quality factor and is the center frequency f_0 of the notch divided by the 3 dB power bandwidth $\Delta f_{3\text{dB}}$ or

$$Q = f_0 / \Delta f_{3\text{dB}}$$

We can undertake a fairly complex mathematical transformation of $H(s)$ and produce $H(z)$ using what is called the bilinear transformation, using a series of trigonometric identities and finally arriving at our formulae for the $a's$ and the $b's$. We are going to spare the reader and simply give the results after giving your these hints and leaving the steps as an "exercise for the reader". But the big idea is to convert continuous time into sampled time. And for this we will now always mark time in samples with a known sampling frequency F_s .

Given our desired notch frequency f_0 and the desired bandwidth BW or alternatively the Q, we derive the following set of formulae. We will work with frequency in radians per sample ω_0 given by

$$\omega_0 = {2\pi f_0}/{F_s}$$

We need to compute some regularly used terms for our formulae to help make them more readable.

$$\alpha = {\sin(\omega_0)}/{2Q}$$

when we are using Q. If we are using 3dB bandwidth the formula is

$$\alpha = \sin(\omega_0)\sinh\left(\ln(2)/2 + BW + {\omega_0}/{\sin(\omega_0)}\right)$$

Notice that for constant Q, the bandwidth increases with increasing frequency and inversely, for constant BW, the Q increases with increasing frequency. These are important consequences to remember or the mathematics. The more complicated formula showing this is given by

$$Q = \left(2\sinh\left(\frac{\ln(2)}{2}\mathrm{BW}\frac{\omega_0}{\sin(\omega_0)}\right)\right)^{-1}$$

We are now prepared to just state the formulae for the $a's$ and the $b's$.

$$\begin{aligned} b_0 &= 1 \\ b_1 &= -2\cos(\omega_0) \\ b_2 &= 1 \\ a_1 &= {-2\cos(\omega_0)}/{1+\alpha} \\ a_2 &= {1-\alpha}/{1+\alpha} \end{aligned}$$

This probably seems anticlimactic given the buildup but there are lots of trigonometric simplification steps left out to get the formulae we needed.

The control software for the notch automatically ignores TNF notches that are not in the current passband so as not to run unnecessary computations that would be filtered by the passband filter. When a notch is moved into the passband, the code for this notch is automatically inserted into the filter chain with the correct frequency. We initially allowed up to three notches in the passband at a single time with up to a depth of x3 each (stacked notches) which results in a total of nine notches being computed in real time.

TNF™ IN BETA TESTING

During Beta testing we discovered that it could be difficult to place a narrow notch on a narrow carrier when the panadapter is zoomed out. As we tried to reproduce the steps to properly place a TNF notch, we discovered that it was typically too many steps to be a comfortable procedure. The operator would have to zoom in, recenter the panadapter, drop the notch, adjust it and then zoom back out. To remedy this problem, we introduced a new element, auto-zoom. With auto-zoom, after a notch is placed the operator can click on the notch without moving the cursor for one second and the panadapter will reveal a sub-display of the notch that is zoomed and the drag operation to center the notch will be reduced to make fine tuning easier. When the mouse button is released, the display reverts to the prior display.

We also found that some users actually found uses for more than three simultaneous notches in the passband so we expanded the number that could be simultaneously computed to nine. With a depth of three each, this could result in up to 27 notches being computed in real time. Some additional CPU time is required, but the tests showed that even this many notches are not a problem for most modern computers.

CONCLUSIONS

FlexRadio Systems' exclusive Tracking Notch Filter™ dramatically redefines the way amateur radio operators are thinking about and actually removing noise sources. The feature-rich implementation includes the ability to vary operational parameters of any number of notches as well as preserve these notches for continued noise-reduction capabilities. Operationally, TNF represents a leap in capability and technology that allows operators to completely remove not just a couple of individual static noise sources, but all noise sources that are static even when each noise source has different characteristics.

FOR MORE INFORMATION

For more information on FlexRadio Systems visit www.flexradio.com or contact us at sales@flexradio.