

CW-Filter: Background and User Guide

Klaus von der Heide, DJ5HG

1. General Operation

CW-Filter is a digital bandpass filter running on a PC with soundcard. It is designed to support human reading of noisy CW-signals. Its Graphical User Interface (GUI) is shown in figure 1.

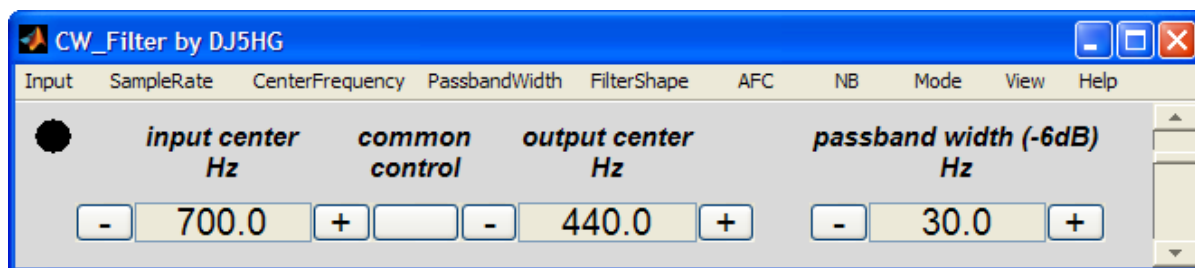


Figure 1. The Graphical User Interface (GUI) of CW-Filter. The main parameters can be edited directly or increased / decreased by pushbuttons (- and +). These are input center frequency, output center frequency, and passband width. Other parameters like sample frequency are changed via the menu.

2. Filter Shape

In a simplified view, the task of a passband filter is to maximize the signal-to-noise ratio (SNR) by stopping noise and other unwanted disturbances outside the signal spectrum while the signal can pass. This simple view is not appropriate for CW because the usual fast ON/OFF-keying produces a spectrum several hundreds of Hertz wide. Any cutoff by a narrow filter changes the fast ON/OFF-keying into a signal with slower raising and falling edges. Even more: If a filter with very steep cutoff is used then the signal edges cause a sound called "ringing". Figure 2 shows the impulse response and the response of such a filter to a Morse dot.

If the Morse signals are to be decoded by a digital algorithm then the optimal (matched) filter is realized by an integrator over the dotlength. Its frequency response is a sinc-function (figure 3). The sinc-function decays very slowly far away from the center frequency. Therefore, such a filter cannot stop QRM. If CW is decoded by a human listener the human himself is the integrator. So such a filter is of no use in this case.

The best approach to design a CW-filter to support human CW-decoding is to start with a demand to the impulse response:

- (a) It must be finite (and as short as possible)
- (b) The raising and the falling edges must be monotone (no ringing)

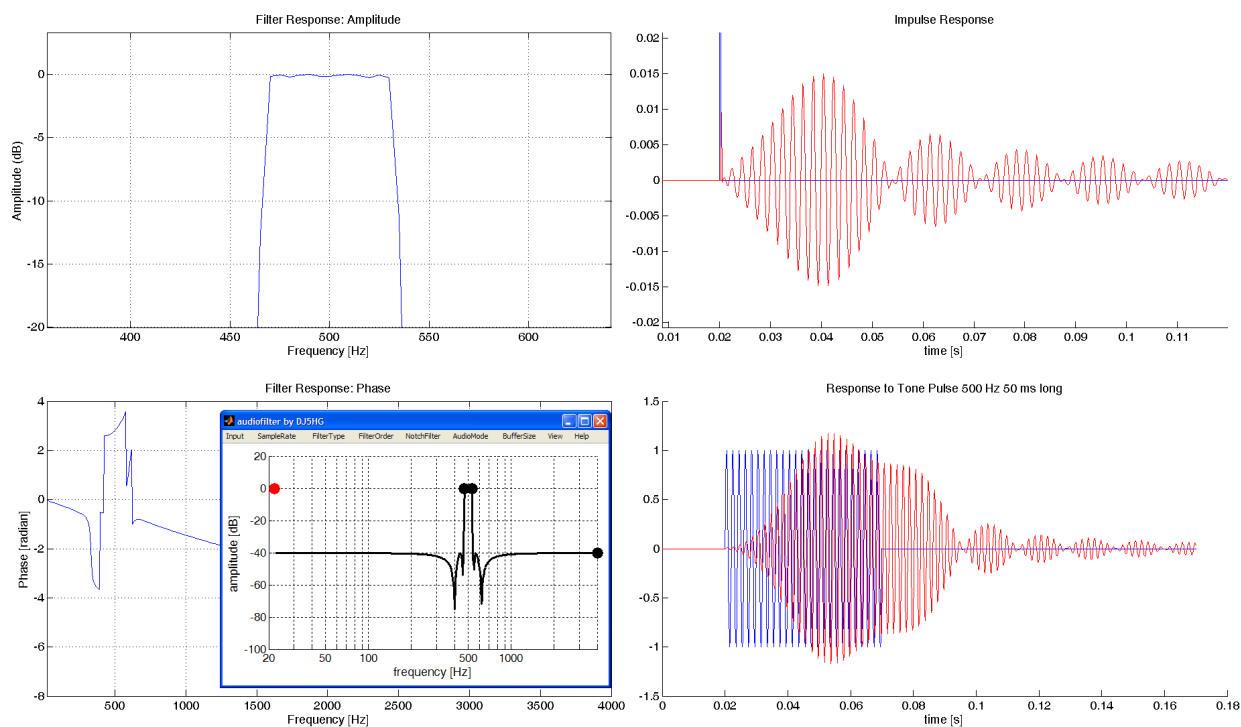


Figure 2. A sharp filter cutoff leads to an output signal that sounds like a decaying ringing bell or like a series of echoes. This is unusable for a CW-filter. This signal was generated using the program **audiofilter** of the author.

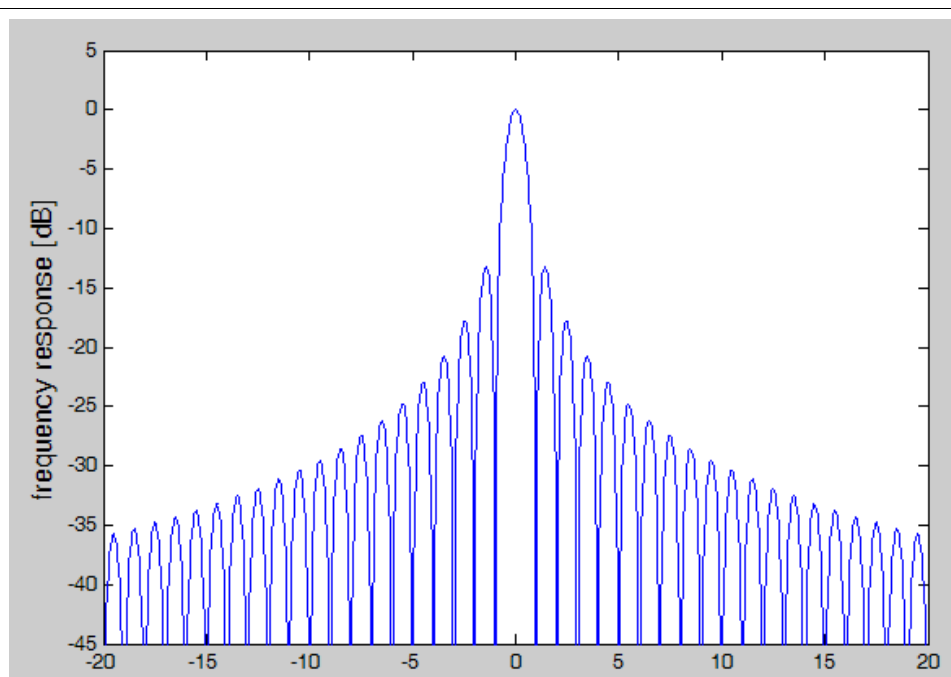


Figure 3. The frequency response of an integrator (the matched filter for a digital CW-decoder) is a sinc function. The integrator cannot sufficiently suppress QRM-signals.

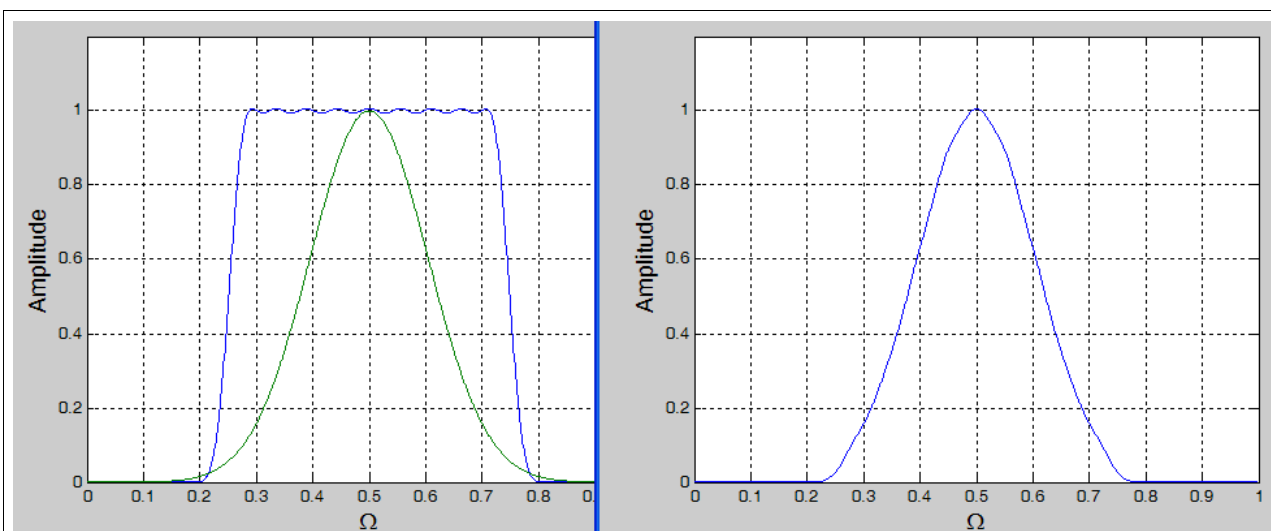


Figure 4. Frequency responses of a Gauss-filter plus a nearly rectangular filter in series (linear scale). Left: both individual filters; right: response of both filters in series.

In order to guarantee QRM-rejection, the same (a) and (b) should be valid for the frequency response of the filter, i.e. the same demands in time domain and frequency domain. Fortunately, there is a simple solution to the demands (b): The Gaussian bell function. Unfortunately, it extends infinitely wide to both sides. But fortunately its decay to zero is very fast. So the error is negligible if the Gaussian function is set to zero for argument values absolutely larger than an adopted limit.

Following this argumentation, CW-Filter uses a filter with nearly rectangular shape which defines this adopted limit and a filter with Gaussian shape in series. Figure 4 shows the responses of both filters (left side) and the response of the combined filter (right side).

The menu of CW-Filter has the option "view / show filter response". This generates four figures that characterise the behavior of the running filter. Figure 5 shows an example.

A further option is "view / show spectrum". This generates a real-time spectrum around the center frequency. It is ± 500 Hz wide, so it shows signals outside the filter passband. Figure 6 gives an example with an aurora signal on 144 MHz read with a filter width of 670 Hz.

A mouse click into the spectrum window toggles the frequency scale between ± 500 Hz or $\pm \text{bandwidth}/2$.

3. Realization

3.1. Filter Structure

The filter is realized as a multirate digital FIR-filter. This means that the computational effort is minimized by downsampling and upsampling. Input sampling rates 16000 and 32000 are reduced to 8000, rates 11025, 22050, 44100 are reduced to 5512.5, and rates 48000, 96000 are reduced to 6000. The resulting rates 8000, 5512.5, 6000 are further reduced in steps by 1/3 as far as is necessary to get the final bandwidth. These filters are implemented as bandpass filters with center frequencies $8000/4$, $5512.5/4$, $6000/4$. The reason is that nearly 2/3 of all filter coefficients of such filters are zero. A mixer shifts the input signal to this fixed IF, and a second mixer finally shifts the IF to the output center frequency.

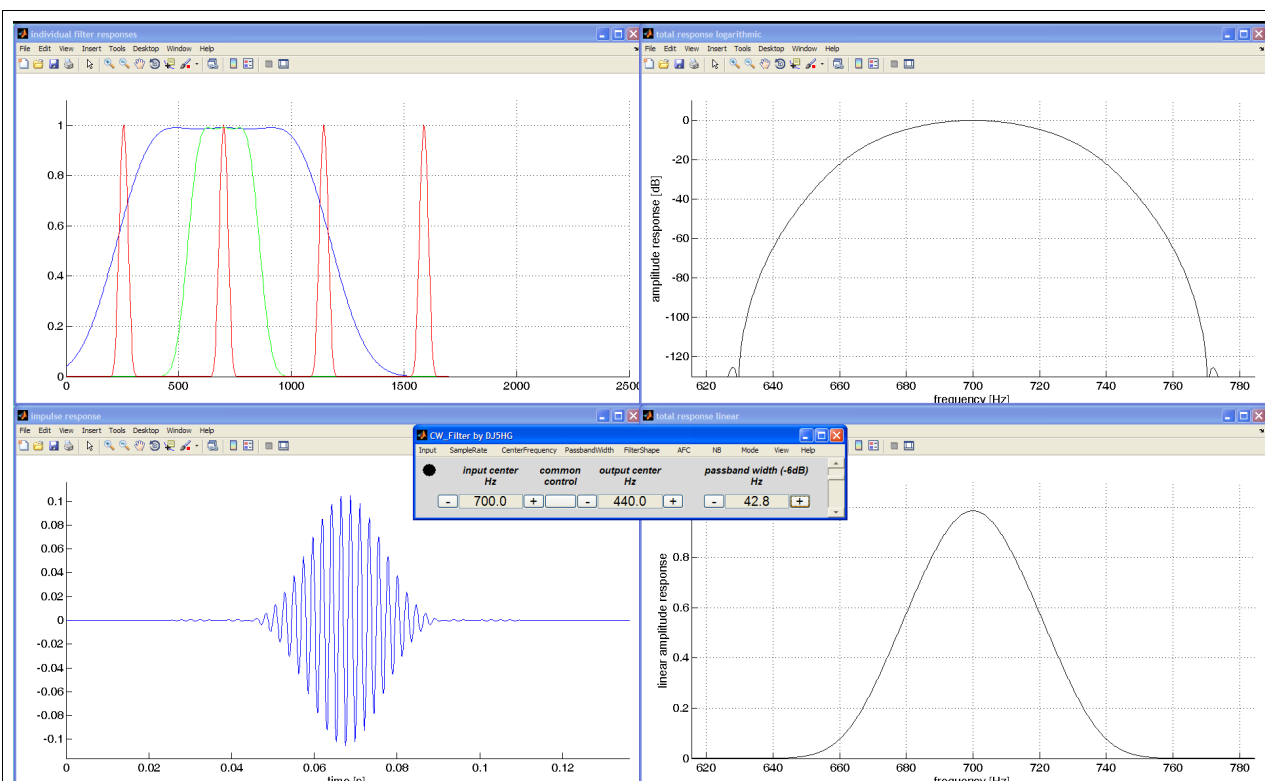


Figure 5. Behavior of an actual filter (output of the menu option "view / show filter response". Upper left: Individual frequency responses of all decimation filters and the kernel filter (linear). Upper right: Frequency response of the filter chain (in dB). Lower left: Impulse response of the filter. This shows a nice Gaussian-shaped pulse, no ringing. Lower right: Frequency response of the filter chain (in linear scale to show the Gaussian shape).

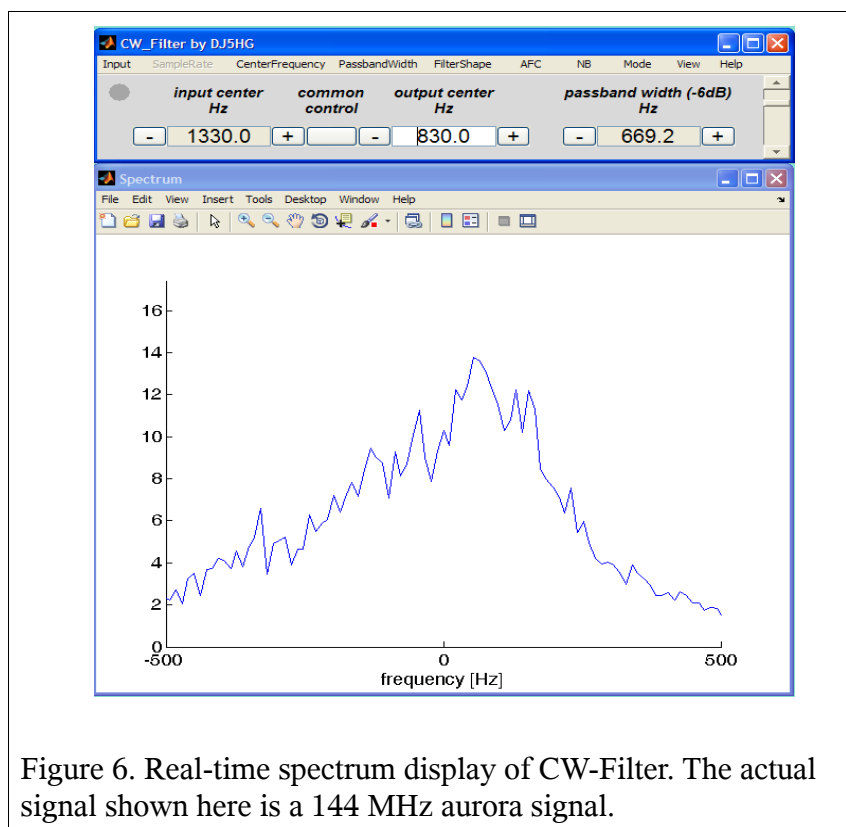


Figure 6. Real-time spectrum display of CW-Filter. The actual signal shown here is a 144 MHz aurora signal.

A filter actually may run with the following parameters:

sample rate	44100 samples per second
input center	600 Hz
output center	500 Hz
passband width 6dB	29.5 Hz

In this case the sample rates in the filter chain are are:

44100 22050 11025 5512.5 1837.5 612.5 204.1666 612.5 1837.5 5512.5 11025 22050 44100.

The corresponding center frequencies are

600 600 600 600 1378.25 1378.25 1378.25 1378.25 1378.25 500 500 500 500.

It should be noted that the center frequency 1378.25 Hz in all cases is larger than the Nyquist frequency. The sampling theorem is satisfied throughout the chain because the local filter bandwidth never is larger than the local sampling frequency (complex computation).

3.2. Delay

One of the drawbacks of a real-time digital filter is the inevitable delay between input and output. There are two independent sources of this delay:

- (a) A delay caused by the fact that the filter algorithm computes an output sample as a symmetric function of the last m samples (a simple example is the mean value). m usually is an odd number, so the delay is $(m-1)/2$. Analog filters also cause such a delay. But their filter order $m-1$ usually is very much smaller, and their impulse response is not symmetric (see figure 2).
- (b) A delay caused by buffering the acquired samples. This is necessary in a PC to cope with changing workloads and an operating system that was not designed as a real time system. The general structure of CW-Filter including these buffers is shown in figure 6.

In case of usual CW-filters, (a) is relatively small. An example is shown in figure 5. The impulse response (lower left image) has its maximum at 63 milliseconds after the input pulse. On the other hand, the necessary buffering (b) usually needs about 200 ... 500 milliseconds. Such a delay can be very irritating if signals are tuned by ear. Only in case of very narrow filters the delay (a) dominates.

3.3. Stopband Attenuation and Mirrors

The specification of the filter including the frequency shifts demands for an attenuation of -120 dB for all unwanted frequencies for passband widths up to 500 Hz and input center frequencies between $0 + 1.5 \cdot \text{passbandwidth}$ and $2500 - 1.5 \cdot \text{passbandwidth}$. Then the weakest element of the whole filter is the soundcard. The actual behavior of a chosen filter can be measured using the menu option "mode / scan" (see the next chapter 4).

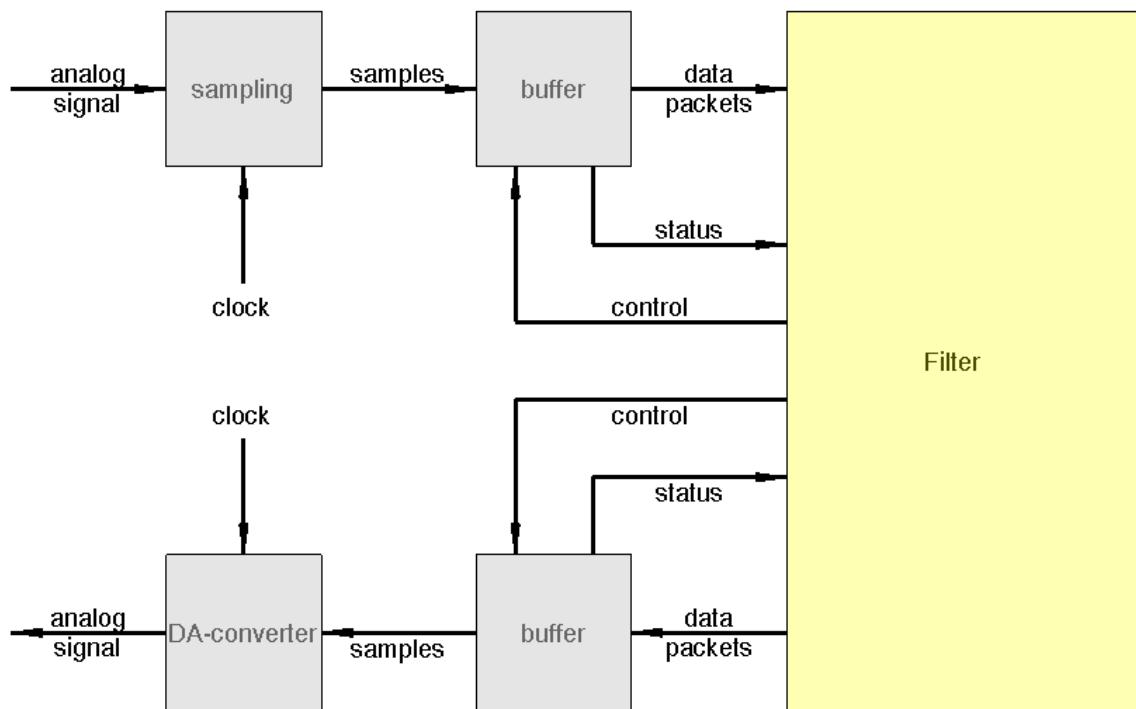


Figure 6. General structure of a real-time system using the soundcard on a PC. The DSP-application (Filter) gets the samples in data packets (1024 samples for example). The application is triggered when the sound engine has acquired a full packet. After computation, the application delivers a packet to the output buffer. The output buffer continuously delivers samples at the sample rate to the DA-converter. Therefore, the output buffer can run empty if the application does not deliver the packets in time. A late coming packet does not start the sound engine again. Sophisticated error handling is necessary to keep such a system running.

4. Filter Scan

CW-Filter offers two modes for measuring the filter properties (by the menu "mode"):

- (a) The internal scan
- (b) The external scan

The internal scan does not use any analog signal. A DDS generates data packets of a slow chirp and sends them directly to the filter. The output data are analysed by a FFT using a Kaiser window which suppresses the wings of a peak down to -120 dB. Since recognition of unwanted mirrors has priority, the frequency blurring effect of such a window is acceptable. Figure 7 shows the result of an internal scan. Figure 8 is the same result but turned with a mouse operation.

The difference of the external scan is that the DDS-generated signal is sent to the output of the sound card, and the filter also takes it's input from the soundcard (left channel). To get a meaningful scan, output and input of the soundcard must be connected by a usual audio cable. In this case, the behavior of the filter is measured with the analog-to-digital converter included. Non-linearities can be detected, and the noise of the quantization in the AD-converter is clearly visible. Figure 9 shows an example.

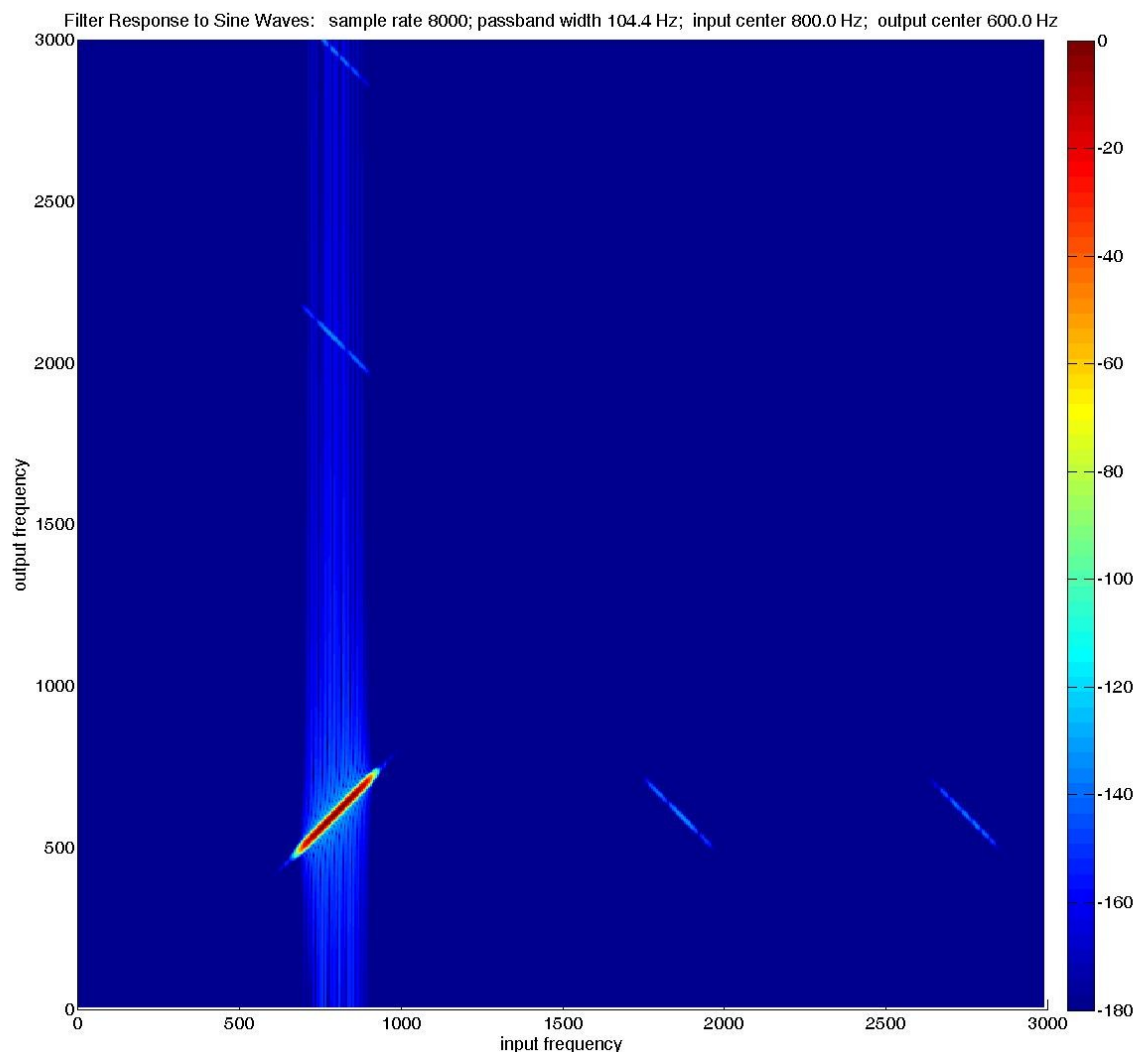


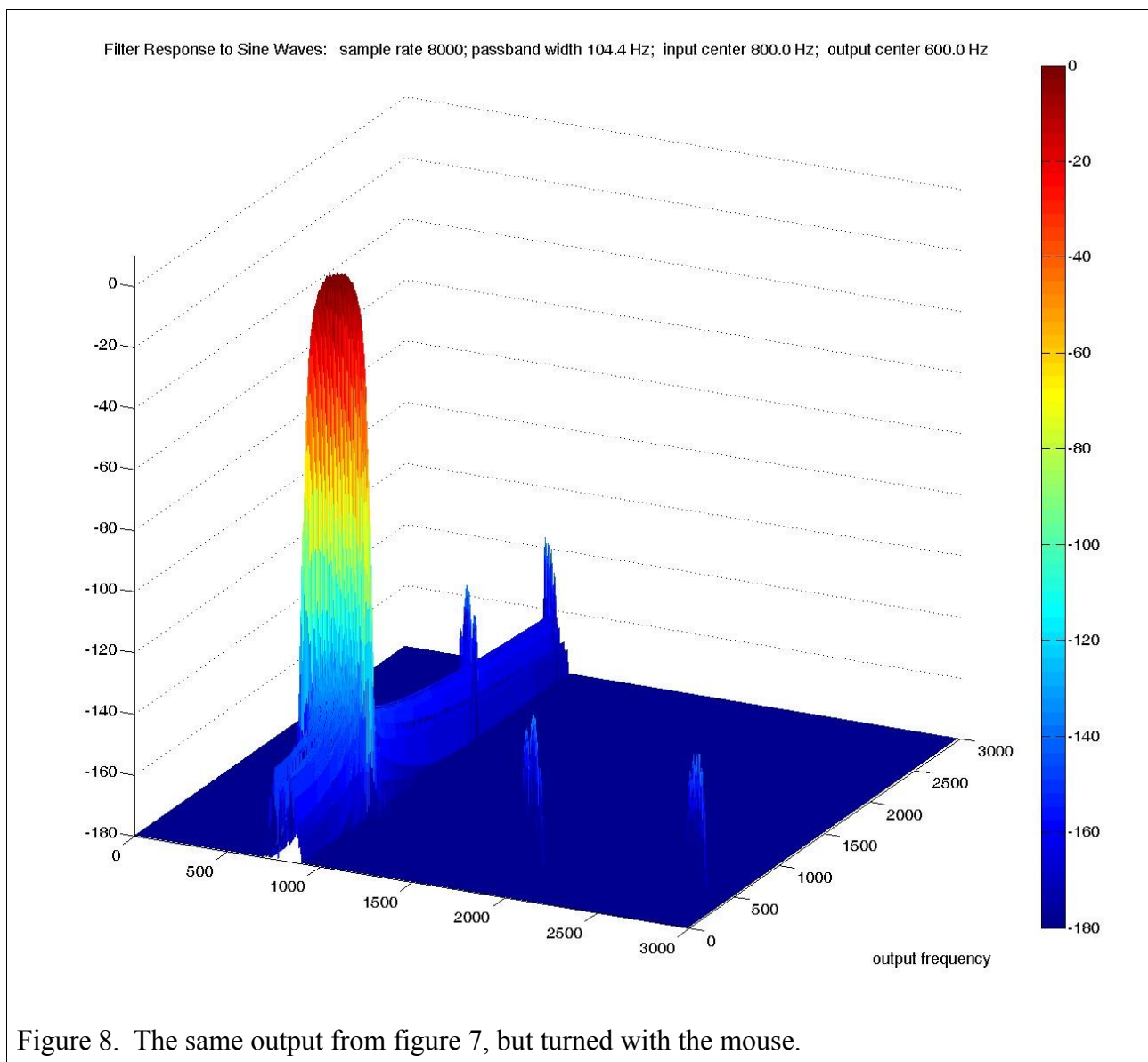
Figure 7. The scan mode of CW-Filter measures the spectrum of the output signal (vertical axis) for input sinewaves of frequencies from 0 to 3000 Hz (horizontal axis). The amplitude is color-coded [dB] with the colors of the bar. While the input center frequency is 800 Hz, the output center frequency is 600 Hz. Therefore, if you choose an input frequency of 750 Hz on the horizontal axis, the spectrum of the output is given by the colored vertical line above the 750 Hz. It shows a peak at 550 Hz on the vertical axis of the output frequency.

There are two mirrors visible at about 1800 Hz input and 2750 Hz input. This means that an input signal of 1800 Hz will lead to an output signal within the passband. Similar to analog systems, the mirrors are inverted: increasing the input frequency leads to a decreasing output frequency. The mirrors are slightly below -120 dB. So it is within the specification. Nevertheless, if a signal appears at 1800 Hz which is 120 dB above your weak signal at 800 Hz there would be a problem. The quantization noise of the soundcard usually hides this effect completely.

There are two mirrors above the input passband at output frequencies 2000 Hz and 3000 Hz. These are mirrors caused by the interpolation filters. There is no problem because the same input signal generates an output on 600 Hz and a weak interference at 2000 Hz and 3000 Hz.

The vertical blue bar around the input center frequency is an artifact of the FFT. A Kaiser window is used to press these artifacts down below -120 dB.

The graphical scan output is a 3D-image. It can be turned by the mouse resulting in figure 8.



The external scan is very sensitive to overloading of the soundcard input. Overloading causes clipping which is a non-linear operation that generates harmonics. Figure 10 shows a case with the harmonics visible.

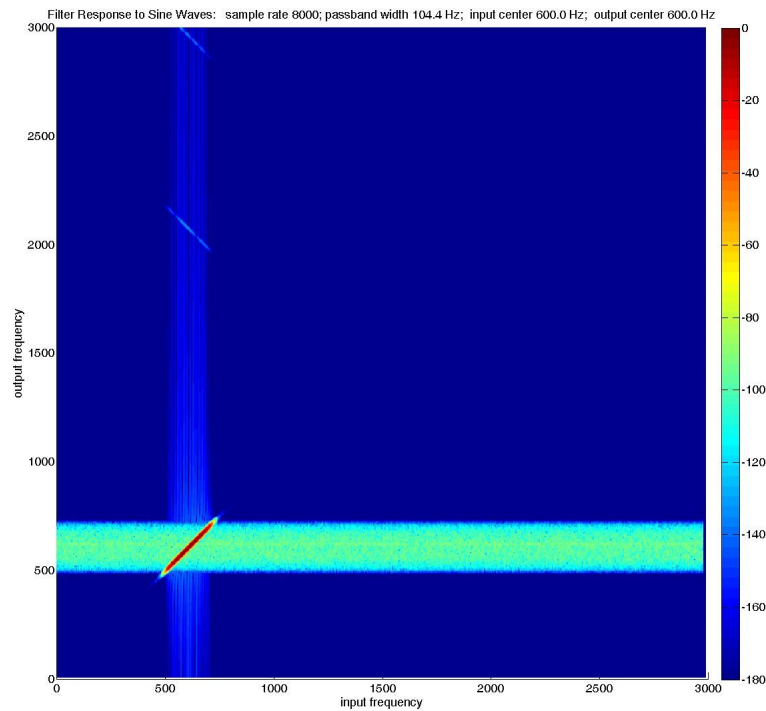


Figure 9. Result of an external scan. The quantization noise of the soundcard input is a broadband noise. As a part of the input signal, this noise is filtered. The filtered noise of course is independent of the generated input frequency resulting in a horizontal bar at about -90 dB.

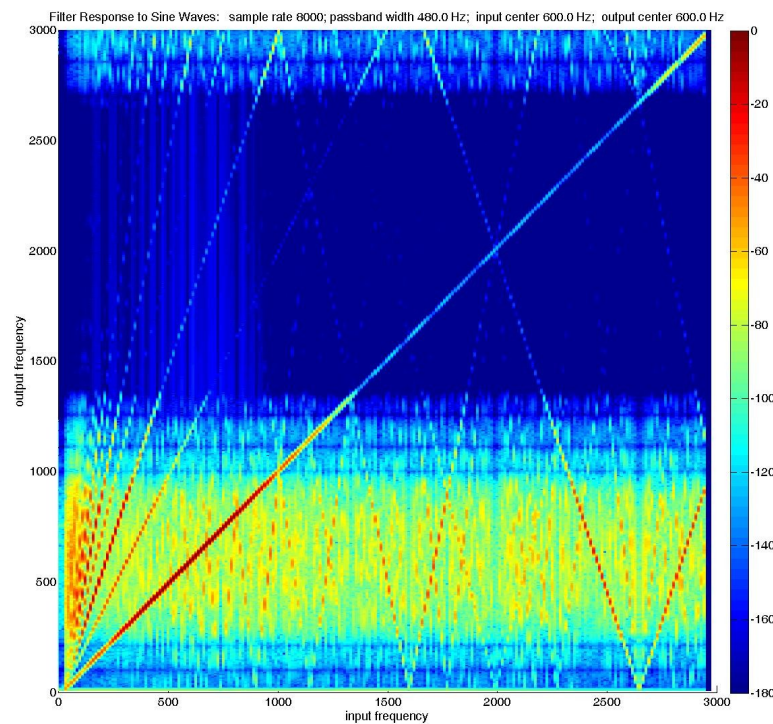


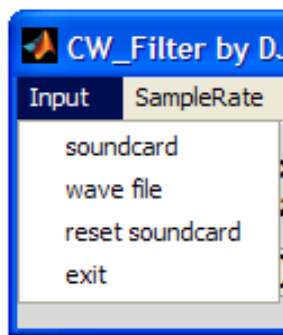
Figure 10. An overloaded external scan. For a given input frequency there is the same output frequency as the correct signal (centers of input and output equal). But there also is this frequency multiplied by 2, 3, 4, 5, These are the harmonics generated by overloading the input.

5. CW-Filter User Guide

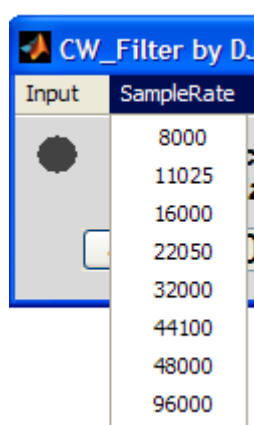
5.1. Direct Input into Edit Fields

As shown in figure 1, there are three edit fields to type in the values for "input center frequency", "output center frequency", and "passband width (-6dB)".

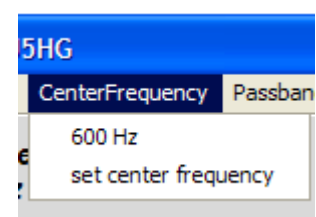
5.2. The Menu Bar is explained by the following Screenshots:



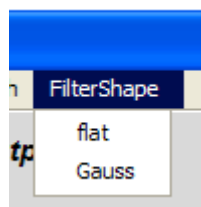
CW-Filter normally takes its input from the soundcard. But also a wave file can be chosen. In the last case, the samplerate is set to that of the wave file.



Menu to select a standard samplerate.



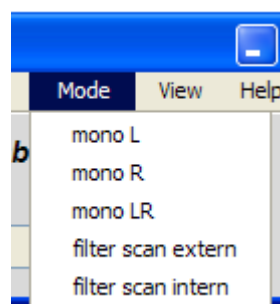
Additional to the direct input into the edit fields the menu offers the standard choice of 600 Hz for input and output center frequency.



There are two options for the shape of the passband:

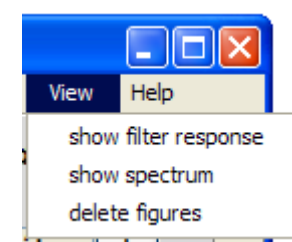
flat: the rectangular shape shown in figure 4

Gauss: the Gauss filter with edges defined by the rectangular filter as shown in figure 4 (this is for CW).



The filter input is the left audio channel (with the exception of the internal filter scan). The output can be chosen as L, R, or LR.

The internal scan uses the sound input for timing only. The external scan uses the left channel for input and output.



Show filter response computes four figures that characterize the actual filter.

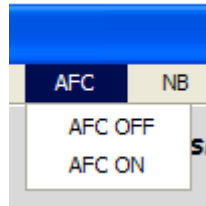
Show spectrum opens a window with a real-time display of the spectrum ± 500 Hz of the input center frequency.

Delete figures deletes all windows except from the main GUI.

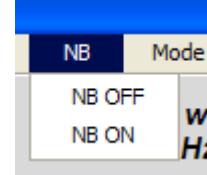
cy	PassbandWidth	
	953.2 Hz	
	811.2 Hz	
	669.2 Hz	
	486.7 Hz	
	385.3 Hz	
	317.7 Hz	
	270.4 Hz	
	223.1 Hz	
	162.2 Hz	
	128.4 Hz	
	105.9 Hz	
	90.1 Hz	
	74.4 Hz	
	54.1 Hz	
	42.8 Hz	
	35.3 Hz	
	30.0 Hz	
	24.8 Hz	
	18.0 Hz	
	14.3 Hz	
	11.8 Hz	
	10.0 Hz	
	8.3 Hz	
	6.0 Hz	
	4.8 Hz	
	3.9 Hz	
	3.3 Hz	
	2.8 Hz	
	2.0 Hz	
	1.6 Hz	
	1.3 Hz	
	1.1 Hz	
	0.9 Hz	
	0.7 Hz	
	0.5 Hz	

The passband width is quantized in 5 fixed width per triade. The values differ for different sample rates.

Please note that the filter only is specified for widths up to 500 Hz.



Automatic Frequency Control still is not implemented. AFC ON only opens a window showing the real-time mean phase of the signal within the passband.



A very simple Noise Blanker sets all samples to zero which absolutely exceed the value set by the slider at the right end of the CW_Filter-GUI.

5.3. Input Volume Display

The level of the sound input volume is color-coded in a circle below the input menu option. Black means very low input level, and white means maximum allowed level. In the case of overload, the circle will be red (the internal scan runs at exactly the limit causing a red circle which is ok).

5.4. Noise Blanker Level

At the right end of the main GUI there is a slider to set the threshold for the noise blanker. The maximum (upper) value means no blanking, the lowest value means total blanking.