Contents

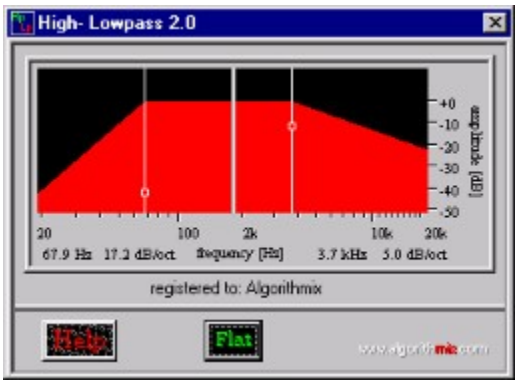

Fig.1: Screenshot of **High-Lowpass PlugIn** window.

Help file last modified: December 21, 1997.

Overview

A *highpass filter* (*low-cut filter*) sharply attenuates frequencies below a certain frequency (*cut-off frequency*). It is used to remove low-pitched noises such as studio rumble, microphone handling noises, vibration of microphone stands, microphone breath pops, and hum from guitar amplifiers. Another application is for cutting-off the table rumble of record players.

A *lowpass filter* (*high-cut filter*) sharply attenuates frequencies above a certain frequency (*cut-off frequency*) . It is used to reduce hiss-type noises such as tape noise from low-quality cassette recorders.

In a typical mastering *highpass* or *lowpass filter*, the attenuation rate (*slope*) can be adjusted to 6 dB per octave (*1st order filter*) or 12 dB per Octave (*2nd order filter*) and its *cut-off frequency* usually to some fixed values. In addition, a typical audio filter is a *non-linear phase filter*. It means, different frequencies are delayed differently when passing the filter. The phase shift may spread the attack edge of impulse-like signals (drums) and blur the sound.

It is practically impossible to design a *linear-phase filter* in analog technology. Initially, digital signal processing technology made it possible. But until now this type of filter has been used only in exotic high-end professional audio equipment. One reason for this is the high complexity of this filter; another is the actual studio praxis driven by the analog predecessors.

Algorithmix® brings high-end professional technology to the multimedia user. The **High-Lowpass PlugIn** represents a unique solution in the world of mastering audio filters. It contains two *linear-phase filters* (a *highpass* and a *lowpass*) with a *continuously adjustable slope* from 0 to 24 dB per octave. Even the most advanced filters used in the digital audio domain have only fixed slopes 6, 12, or 18 dB per octave. Slopes lower than 6 dB per octave and slopes like 9, 13, or 16 dB per octave are impossible with common filter architectures; slopes higher then 12 dB per octave (but still in 6 dB steps) are normally achievable only by cascading more filters. With the **Algorithmix® High-Lowpass PlugIn,** you receive a novel audio processing tool, allowing a filter with any slope you like between 0 and 24 dB per octave. A slope of 24 dB per octave corresponds to a classical filter of the *4th order*!

The *cut-off frequency* of the *highpass filter* can be continuously adjusted in the range from 20 to 200 Hz, the *cut-off frequency* for the *lowpass filter* in the range from 2 to 20 kHz.

Changing the parameter of the filters during playback causes no audible artifacts. Therefore you can safely start your wave-file player and look for the best sounding result while adjusting the filter parameters live.

Help file last modified: December 21, 1997.

The parameter set-up interface is implemented in an intuitive graphical form. The parameters of the highpass and lowpass function can be easily changed by clicking and moving the white square on the proper marker while holding down the left mouse button. For the technically oriented user, both the cut-off frequency and the slope for the current setting also appear in numerical form below the graphic window.

The range of the highpass cut-off frequency extends from 20 Hz to 200 Hz, the range of the lowpass cut-off frequency from 2 kHz to 20 kHz. The slope is adjustable for both filters in the range between 0 and 24 dB per octave.

The *Flat* button switches both *highpass* and *lowpass* filters to a linear characteristic that does not affect the input signal. This function is recommended as a starting position before setting up a new filter characteristic.

For a detailed technical evaluation of the filter characteristics and their influence on the input signal, we recommend using the **Analyzer PlugIn**.

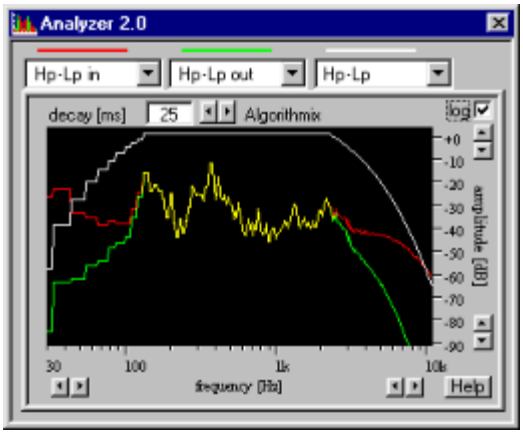

Fig.2: Analyzer with setting for the **High-Lowpass PlugIn**

By setting the **Analyzer PlugIn** to *HP-LP in* (red line), *HP-LP out* (green line), HP-LP (white line), and the upper limit of the amplitude scale to 0 dB, one can easily follow the effect of the *high/lowpass filtering* on the processed audio material.

Help file last modified: December 21 1997.

FFT Setting

To start on spectral modifications of the audio material to be processed, the signal has to be transformed from time domain to frequency domain. In the current version of Sound Laundry, this is achieved with a Pentium-optimized fast Fourier transform (FFT) using a modified radix-4 algorithm. For maximum flexibility, two FFT-specific parameters *Blocksize* and *Overlap* are available for the user. If working with an older PC compatible computer, the FFT process can be switched to mono mode with the *mono* button to save on computational power.

The FFT settings in *FFT Properties* window are valid for all frequency-domain PlugIns having been switched to active mode. The appropriate window opens automatically when loading the first frequency-domain PlugIn.

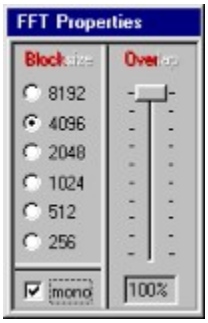

Fig.3: FFT Properties

The *blocksize* parameter (expressed in audio samples) controls the size of the audio blocks used for the Fourier transform. As the largest possible setting of the FFT *blocksize* depends on the I/O buffer size (FFT-blocksize <= I/O buffersize), an adjustment to the I/O buffer size in the I/O set-up dialog of the **PlugIn Station** may be necessary prior to set up large FFT buffer sizes.

The larger the FFT *blocksize*, the more frequency bands are available for carrying out filter functions in frequency domain. Thus a larger FFT *blocksize* usually improves the audio quality and a smaller one increases the risk of audible artifacts. Note that the larger *blocksize* also increases the CPU load.

The time resolution of the applied filter function, however, becomes worse when increasing *blocksize*. For a static filter like in the **High-Lowpass PlugIn,** this phenomenon is not critical. For filters dynamically changing their parameters according to the input signal (like in the **De-Noiser PlugIn**), a proper balance between *blocksize* and time resolution may be a quite important issue.

Using the window overlapping process is indispensable to avoid discontinuities of audio signal at the limits of the FFT blocks used for filtering in the frequency domain. The amount of overlapping is controlled by the *overlap* parameter. An *overlap* of 100 means only half of the audio data is new in the succeeding FFT blocks, thus each audio sample is transformed twice.

One of the most important components in the audio FFT filtering process regarding sound quality is the cross-fading operation which must be performed after inverse FFT (see Fig. 4). With **Algorithmix**® technology, the typical time-domain artifacts are successfully suppressed.

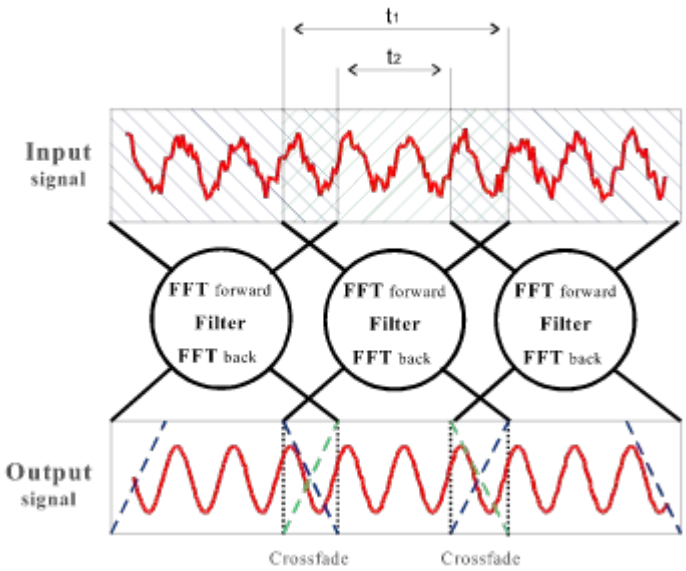

Fig. 4: Principle of the overlapping process used in the FFT in **SoundLaundry 2.0™**.

The process of overlapping used for the Fourier transform is illustrated in Fig. 4. The *overlap* parameter can be calculated from the times t1 and t2 defined in figure 4 as :

$$\text{overlap [\%]} = \frac{t1 - t2}{t1} \cdot 100$$

Increasing the *overlap* from 10% to 100% almost doubles the CPU load. Thus reducing the *overlap* is an effective way to save CPU computational power. For the final mix, however, *overlap* should be set to 100% in any case. 100% overlap prevents signal distortion and so guarantees the lowest THD+N factor.

For taking advantage of the **SoundLaundry™** real-time performance on slower systems, or if you're using mono files anyway, the FFT process can be switched to *mono* mode by selecting the appropriate checkbox in the *FFT properties* window.

Help file last modified: December 21 1997.

Application Tips

Some typical applications for the *highpass* and *lowpass* filters have already been mentioned in Overview. There are some other sound processing situations that make both filters very useful.

Recordings from old gramophone records can be made pleasant again by cutting turntable rumble, low frequency resonances in the pick-up (highpass function), and hiss (lowpass function).

Sometimes audio material recorded with low-quality equipment sounds either too shrill or too dull. This is often due to the wrong tone balance between high and low frequencies from the psychoacoustical point of view. In this case it helps to cut the proper end of the frequency bandwidth.

Background music normally heard in stores or public places is not hi-fi quality. It is cut off at the low and high end of the frequency spectrum to make it less pushy and more stimulating.

The *highpass* and *lowpass* filters can also be used for special effects like the simulation of telephone call quality.

When using these filters as a technical mastering tool, remember that sometimes noise you want to remove has spectral components in the range of the useful program content. In this case you have to be very careful adjusting the cut-off frequencies and slopes. A typical example for that is the compromise between hiss and higher frequencies of the audio signal. If it is not necessary, do not use very steep attenuation; gradual sloping is usually more “musical”.

We do not recommend chaining the **High-Lowpass PlugIn** before the **De-Scratcher PlugIn** or **De-Noiser PlugIn**. To achieve the best performance, it is better if those PlugIns get the unprocessed sound material. However, after *de-scratching* and/or *de-noising*, an additional signal make-up (frequency range correction) with the **High-Lowpass PlugIn** can be very useful.

For further information concerning the PlugIns and our new products, visit us on the Internet at:

http:// www.algorithmix.com

or send e-mail to:

support@algorithmix.com

- if you need any information about installation and performance of this product.

info@algorithmix.com

- if you have general suggestions and questions concerning our product line.

Help file last modified: December 21, 1997.