Scalpel
a comprehensive digital audio processing utility

*Sum & Difference Matrix*

Desc 9115: Digital Audio Systems
Architecture, Design & Planning

*Peter Hüttenmeister*

450077807

Friday, 12 June 2015
“My mix sounds flat and lifeless.” “The soundstage is narrow and cramped.” “The foundational elements of the mix are undefined.” “I wish I could increase the ambience without affecting the vocals.” “Oh well, the mastering engineer can fix all that.”

These are the kinds of attitudes every mastering engineer loathes to hear. With the rapid rise of the home studio, less of an emphasis is placed on achieving a high quality mix. Overcoming these potential issues at the mix stage is preferable as it allows the mastering engineer greater ability to focus their efforts in extending the potential of a track as opposed to addressing severe deficiencies. The problem that arises, therefore, is the access of the mix engineer to fundamental tools used in the sculpting of a clear, clean cut mix requiring less attention in the mastering phase to surgical enhancement.

Enter Scalpel. Scalpel is designed as a modular shell processor. Its fundamental function is to allow the user the ability to derive the sum and difference information of a stereo input signal and ultimately achieve the processing flexibility previously perceived as a magical mastering tool. Whilst it is possible to set up a sum and difference matrix within a digital audio workstation, a certain level of complexity is involved along with the severe potential for failure if exact principles are not properly adhered. Scalpel cuts out these difficulties and places an unprecedented level of control in the hands of the mix engineer.

Sum and difference has its roots in analogue audio processing. Originally designed as a decoder for stereo microphone techniques, a similar mathematical application can be used to process stereo audio signals. The simplest way of conceptualising how a sum and difference matrix applies to stereo audio is in the form of a ratio. Consider a stereo input signal:

\[
x = [x_L, x_R]
\]

From this input we can then derive the sum and difference information:

\[
y_a = \frac{x_L + x_R}{2}
\]

\[
y_b = \frac{x_L - x_R}{2}
\]

Where \(y_a\) is the sum and \(y_b\) the difference. It seems fairly straightforward, doesn’t it? And you’re quite right. Intuitively, the sum (\(y_a\)) of a stereo input signal is the combination of the left and right channels, otherwise known as ‘summing to mono’. The difference (\(y_b\)), therefore, must be the left channel minus the right channel, or vice versa. In effect, what this subtraction does is cancel all the correlated data between the two channels leaving only the difference information. In an analogue situation, physically panning the left and right channels to the centre and inverting the phase of one of the channels achieves the same outcome. Continuing this analogy, if we subsequently have a mono summed copy of the input signal alongside the phase inverted difference signal we then have direct control over both of these elements of the mix (Evers 1991). In a digital world Scalpel achieves this, and more.
Whilst designed as a shell utility processor, Scalpel also has its own processing ability. If we consider sum and difference as a ratio we compare:

\[ y_a : y_b \]

Let’s think about this for a moment. A ratio of 1:1 would result in the output being equal to the input. It is not until we begin to manipulate the ratio that we see and hear changes. Take a moment to listen to the supplied audio samples ‘sample_1.wav’ and ‘sample_2.wav’. Can you hear a difference? As a standalone signal processor, Scalpel gives the user control over this ratio \( y_a : y_b \) allowing non-destructive shaping of the soundstage width. Can you match the sample to the graph below?

As can be seen, the algorithm behind this processor is not overly complex; however, for the purposes of allowing both mono and stereo processing it is important to extrapolate this algorithm slightly to better suit the desired outcome. Scalpel constructs the sum and difference information in either a mono or stereo environment allowing application of the user’s own DSP on either the sum or difference channels separately. (see ‘sample_mono.wav’ and ‘sample_stereo.wav’ for examples of operational modes.) Scalpel then decodes the sum and difference information back to a stereo output signal \( y \) by reversing the previous mathematical formula:

\[
\begin{align*}
y_L &= y_a + y_b \\
y_R &= y_a - y_b \\
y &= [y_L, y_R]
\end{align*}
\]

Let’s consider a real world example. You have a two channel input signal consisting of a complex mix with several elements. There are issues with the level of low frequencies in the centre of the mix and the soundstage is quite narrow and a little lifeless. What can you do? Well, you can process the signal using Scalpel. Scalpel will encode the input signal into its sum and difference channels ready for further processing. In order to fix up the low frequency issues a high-pass filter, some other
equalisation module or even compression can be applied to the sum channel. By doing this we avoid any further narrowing of the soundstage by isolating the sum information and not applying the same processing to the difference information. Furthermore, in relation to the issues in the stereo field we can slightly increase the level of the difference channel over the sum and add a high-pass filtered reverberation module. Both of these processes will help to create greater differences between the left and right channels and therefore increase apparent source width without drenching the foundational elements in the centre of the track in additional ambience. Scalpel will then decode the processed sum and difference channels back to a two-channel stereo output signal as explained above.

It all seems fairly simple, doesn't it? Well, my question is, why are there so few examples of this kind of plugin available in the current market? Some companies such as Brainworx rely heavily on the use of sum and difference matrices for their plugins. They even offer a free download of a tool very much similar to Scalpel (Brainworx Music 2015). However, before you read the word ‘free’ and wipe your hands of this proposal, take into consideration that this free tool has only limited usability giving control only over width and a method through which to audition the different encoded channels. The remainder of their plugins have this function built in, however Scalpel has a direct advantage as it allows the user to insert whatever plugin they wish over the sum and difference channels without being limited to those of the product designer. This functionality opens a vast number of creative pathways through which sum and difference processing may be utilised as an integral tool in the palette of the engineer.

Through this document it should be clear; from this point all that is left to achieve is the real-time implementation, and development of a user interface that allows the end user to insert their own DSP solution across the sum and difference information. Further development into generating simple preset modules designed to address common uses of such a processor and their incorporation into Scalpel to create greater standalone functionality is also an area of great interest. We feel that at its current level and the exciting potential for further growth Scalpel has the potential to be an irreplaceable weapon in the arsenal of the modern digital studio.

