Tunable Virtual Bass Enhancement (May 2009)

Brian T. Gerstle

***Abstract*—This paper will discuss the design and implementation of a tunable virtual bass enhancement Audio Unit (AU) plug-in. Background theory will also be given on the missing fundamental theory and harmonic generation using nonlinear devices, since they are crucial to the operation of this plug-in.**

Keywords -- virtual bass, missing fundamental, nonlinear devices, harmonic generation

I. Introduction

ALTHOUGH the theory and techniques discussed in this paper have been in use for years, the author aims to show that this implementation is unique because it allows users to tune the algorithm in real-time on a limitless variety of sources. This flexibility will allow for a better understanding of how virtual bass can be used in different settings, and hopefully, to give mixing engineers another tool for adding more low frequencies into their mixes.

II. Prior Implementations

For the past several years, virtual bass has manifested itself on a wide variety of platforms. Currently, media technology companies will implement a particular form of this algorithm to enhance the perceived bass reproduction of the playback system on their consumer electronic devices. For example, several LCD TV, laptop, portable media player, mobile phone, and even automobile manufacturers will utilize this technology (often in tandem with stereo widening or dialog enhancement) to enhance their products' audio performance. Since virtual bass is already well established in consumer electronics and media technologies, the author deemed that this implementation should target the professional audio industry by adapting virtual bass for the digital audio workstation (DAW). However, this implementation of virtual bass can also be used to facilitate the tuning of the algorithm for a wide variety of applications as a result of its time-varying properties (see *IV. Plug-in Design*) [1-7].

III. Background Theory

A. Missing Fundamental

In order to truly understand why virtual bass is so effective at boosting low frequencies, one must at least be aware of the missing fundamental phenomenon. This phenomenon is a psychoacoustic illusion whereby humans perceive the lowest, fundamental frequency in a series of harmonics at a higher volume, even if the fundamental might not even be physically present [1].

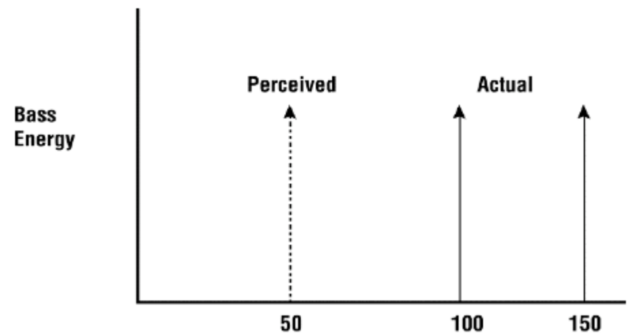

Figure 1: Perception of non-existent fundamental frequency.

In the case of a complex tone consisting of multiple frequencies, this phenomenon can be perceived as a bass boost [2], which is one of the reasons it is so widely exploited on a variety of platforms.

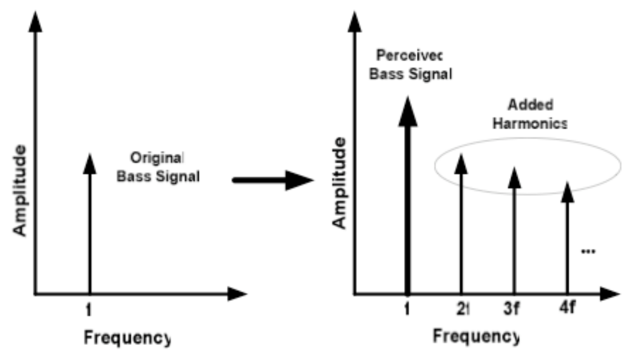

Figure 2: Pitch perception of a complex tone (Missing fundamental effect).

In order to avoid estimating what might be several different fundamental frequencies in a complex signal, nonlinear devices are commonly used to generate the harmonics needed to exploit the missing fundamental effect.

B. Nonlinear Devices

There are a wide variety of methods for implementing nonlinear devices (NLDs), and there are an even larger number of nonlinear devices to choose from. In general, a nonlinear device is an algorithm that alters an input signal in the time domain in order to produce a certain set of harmonics of the input signal. Subsequently, nonlinear devices are usually measured by the amount of which harmonics they produce. For example, when a single sine tone is fed into the

B. T. Gerstle is a graduate of University of Miami's Music Engineering Technology program and also has a minor in Electrical Engineering. He has no plans for employment after graduation and would greatly appreciate any assistance you can provide in this area!

NLD, the Discrete Fourier Transform (DFT) is taken to show which harmonics were produced and at what amplitude [3]:

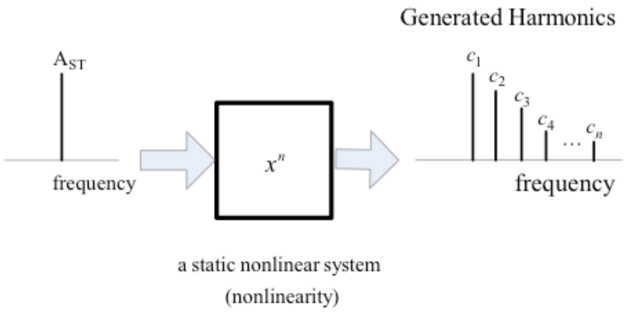

Figure 3: NLD measurement by introducing a single sinusoid into the system.

It is easily inferable why NLDs are so popular for implementing virtual bass algorithms. However, as with most audio effects, NLDs are not without their faults. In particular, when the input to the NLD is a complex signal with a bandwidth between 20 – 200 Hz (as is usually the case for virtual bass systems), Intermodulation distortion can occur. Intermodulation distortion is defined as an additional component of the output signal that is formed by adding or subtracting two or more harmonics [3].

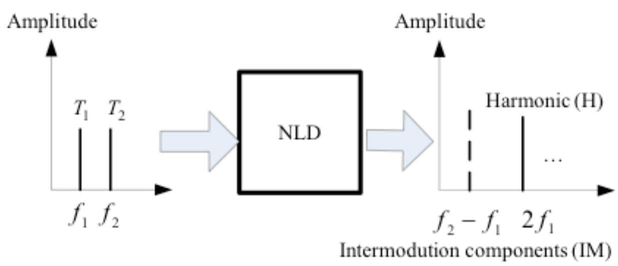

Figure 4: Intermodulation distortion caused by a nonlinear device (NLD).

In order to prevent this, the input to the NLD in a virtual bass system is usually low-pass filtered with the cut-off being close to the natural cut-off of the desired playback system (e.g. 80-100 Hz). In addition to the initial low-pass filtering, band-pass filtering is applied to the output as an attempt to eliminate intermodulated components of the signal above what might be considered the low-frequency range or below the physical range of the playback system (e.g. between 80-300 Hz). The band-passed output of the NLD is then summed with a high-passed version of the input to form the final output signal. Once again, the output from the overall algorithm consists of a high-passed version of the original input combined with the band-passed output of the NLD. Such a filtering scheme can be found in most virtual bass algorithms [1-4 **CHECK THIS**] and as a result, the following design or similar can be seen in virtual bass implementations:

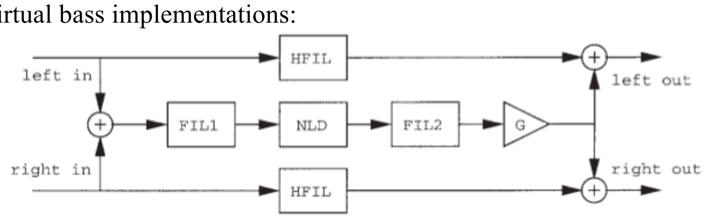

Figure 5: Example of a virtual bass algorithm.

Where HFIL corresponds to the high-pass filter, FIL1 to the low-pass filter, and FIL2 to the band-pass filter. As previously mentioned, there are a wide variety of methods that accomplish nonlinear distortion, each of which was explored before the implementation of the tunable virtual bass plug-in.

One of the first techniques explored during the development of the tunable virtual bass plug-in was the full-wave rectifier method. This approach employs the previously mentioned filtering scheme with a full-wave rectifier as the NLD. The full-wave rectifier generates the even harmonics of the input signal and its transfer function can be expressed as simply [3]:

$$y = |x| \quad (1)$$

After experimenting with this method on a few different sources, the author determined that the results were clearly noticeable. However, it was later deemed that this method caused too much intermodulation distortion when compared with the other methods.

While there is also the possibility of using a half-wave rectifier as a NLD, the author decided to experiment with the full-wave integration method instead. In addition to a full-wave rectifier, this technique performs integration on the output of the rectifier and then resets the rectified signal to zero at every zero crossing with a positive slope, which ensures that the output has the same fundamental period as the original signal.

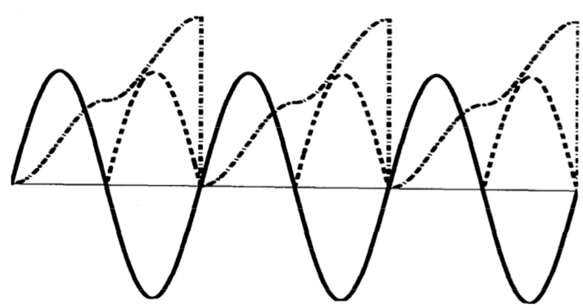

Figure 6: Output of a full-wave integrator at different stages. The dashed line is obtained by full-wave rectification, the dashed-dotted line by full-wave integration.

Upon comparison of the full-wave rectifier and the full-wave integrator, the author noted that the full-wave integrator produced a sensation of deeper bass. This can be attributed to the difference in generated harmonics between the two methods. While the full-wave rectifier generates only even harmonics, the full-wave integrator generates both even and odd harmonics (as pictured below) [4].

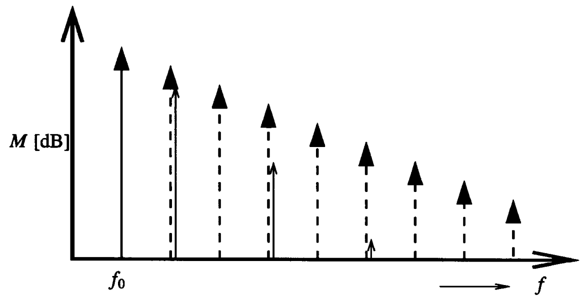

Figure 7: Spectrum of the output of both a full-wave rectifier and full-wave integrator. The solid arrow at f_0 represents the input spectrum, arrows at 2 f_0, $4f_0$,, etc., represent the output spectrum for the full-wave rectifier and dashed arrows at $2f_0$, $3f_0$,, etc., represent the output spectrum for the full-wave integrator.

Following experimentation with the full-wave integrator, soft clipping distortion was chosen as the next and final method to explore. This method uses nonlinear equations, usually polynomial equations that vary in order, to generate a series of harmonics [3]. Pictured below are some examples of polynomial transfer functions of varying order with equations similar to:

$$y = h_1 x + h_2 x^2 + h_3 x^3 + \ldots + h_6 x^6 \quad (2)$$

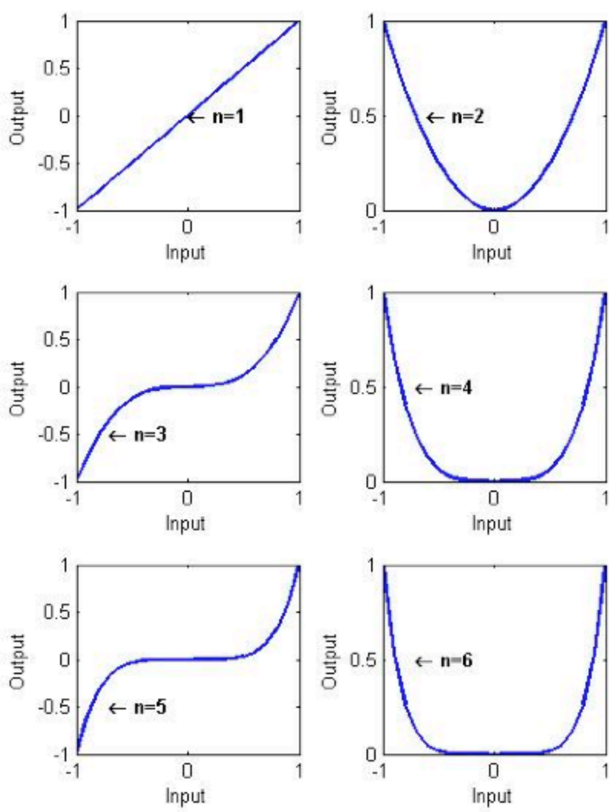

Figure 8: Linear and nonlinear system transfer function plots from n = 1 to n = 6, where n is the order of nonlinearity.

By implementing one of these equations, nonlinear distortion can be applied to the input signal, which will generate the harmonics needed for the virtual bass effect.

Now that the varying methods and theory behind virtual bass have been explained, the author will now elaborate on the design of the tunable virtual bass algorithm.

IV. Plug-in Design

Digital signal processing for the algorithm was initially developed in MATLAB. Initially designing the algorithm allowed for the easy implementation and fine-tuning of the filters needed for the virtual bass algorithm (see Figure 5). Once the design was finalized, the algorithm was ported to C++ as an Audio Unit plug-in for use with digital audio workstations (DAWs).

The overall design of the plug-in was adapted from the common virtual bass topology (see Figure 5), but with an additional gain before the NLD as well as a control for the hardness of the knee in the NLD, all of which are variable.

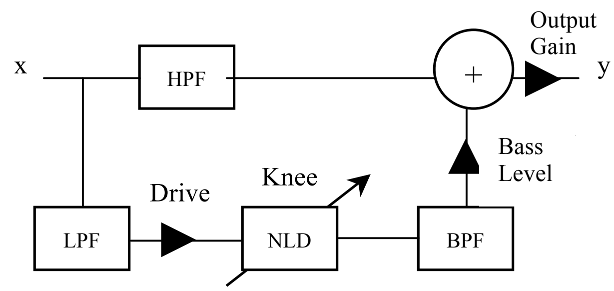

Figure 9: Tunable virtual bass plug-in design.

A. Filtering

Filters were designed in MATLAB as fourth-order elliptical filters with 1dB of pass-band ripple and 80 dB of stop-band attenuation. Cut-off frequencies were set to 100 Hz for the high- and low-pass filters, while the band-pass filter's bandwidth ranged from 80-200 Hz. It was deemed necessary to have such high attenuation in the stop-band in order to prevent leakage between the filters. This ensured that only low frequencies would be passed through the NLD, which reduces the amount of intermodulation distortion.

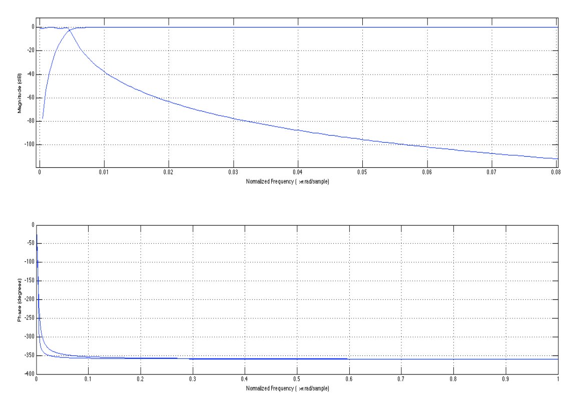

Figure 10: High- and low-pass filter frequency and phase responses.

Implementing these filters in C++ required that each had its own array of coefficients as well as corresponding sample buffers. Since the filters had an infinite impulse response, each filter required two buffers: one for previous input samples and another for previous output samples.

B. Nonlinear Device

Soft clipping was chosen as the NLD for the final implementation of the plug-in. This decision was made because soft clipping was subjectively determined to be the best sounding of the NLDs tested during experimentation in MATLAB (i.e. full-wave rectifier, full-wave integrator, and soft clipping). Instead of using one of the polynomial equations discussed in the background theory, the author was recommended another equation during a conversation with James Johnston [5]. This equation allows for linear operation at low signal levels, but gradually applies much more attenuation at high signal levels. Essentially, this equation was described as being smoother than its polynomial counterparts.

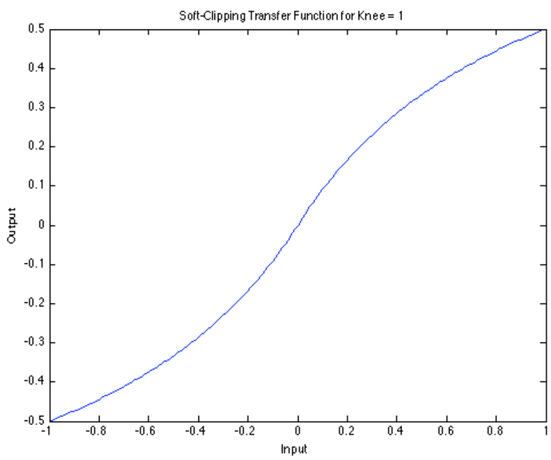

Figure 11: Soft clipping transfer function for K = 1, note the smoothness.

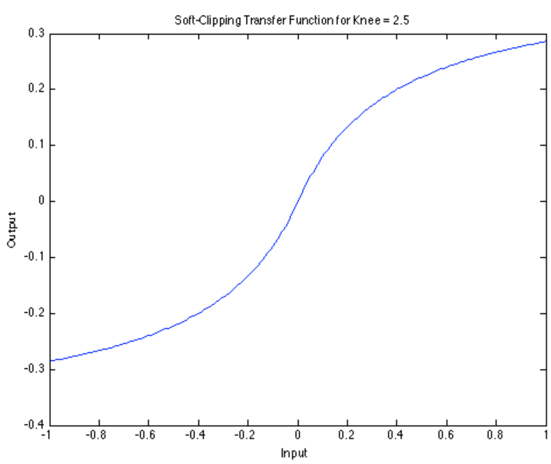

Figure 12: Soft clipping for K = 2.5, note how the curve flattens much faster than the previous transfer function.

$$y = \frac{x}{K|x| + 1} \quad (3)$$

Where K represents the knee of the distortion in the NLD (see Figure 9). Commonly found in effects such as compressors, this knee allows the user to gradually alter the effect from having a soft (K = 1) to hard knee (K = 2.5). In order to illustrate the difference between the two settings, note the shape of the transfer function curve as well as its maximum and minimum values. By examining the input versus output spectrum for a sinusoidal input, one can better observe the direct effect the knee has on harmonic generation.

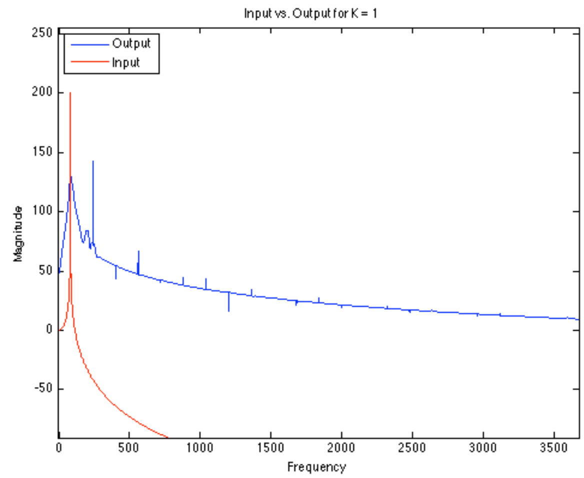

Figure 13: Input versus output for K = 1

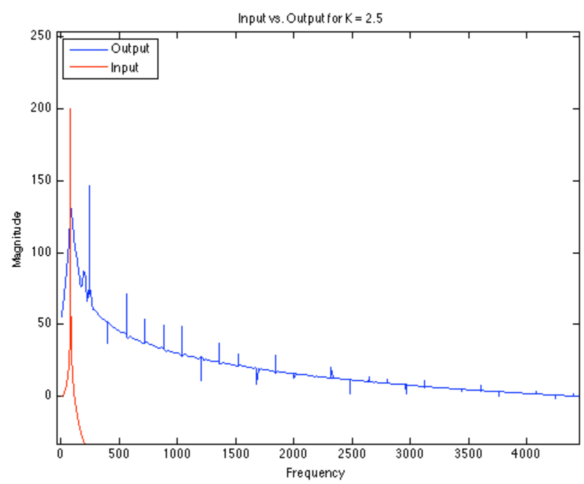

Figure 14: Input versus output for K = 2.5

The other adjustable parameters associated with the low-frequency section of the virtual bass algorithm are Drive, Bass Level, and Output Gain. The concept for the drive parameter was adapted from the same parameter commonly found in distortion pedals along with the headroom concept from Minnaar's paper [6]. Raising the Drive parameter causes the user to send a louder input signal into the NLD, subsequently causing more distortion by raising the volume of the generated harmonics (which also depends on the knee). Bass Level simply boosts or attenuates the output of the low-frequency section of the algorithm and combined with Output Gain, acts as a way for users drive the NLD with a very loud input signal and cause a lot of distortion without over-saturating the output.

After implementing the NLD and filters, the parameters were all incorporated into the Audio Unit plug-in architecture. As such, the Audio Unit API generates sliders for each parameter with values that are defined along with the parameter's maximum and minimum. As a result, the user-interface for the plug-in will appear as follows in most DAWs (pictured in Logic Pro 8):

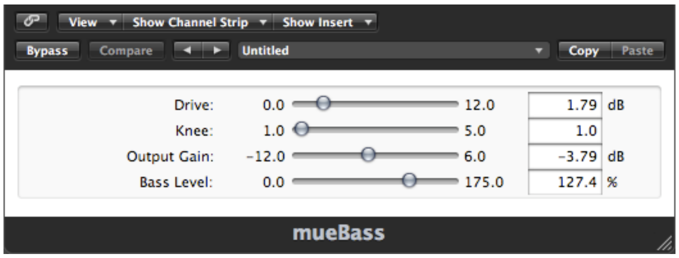

Figure 15: mueBass plug-in user interface.

V. Conclusion

Even though the plug-in functions properly and up to the author's expectations, there were still some features that were not implemented due to time constraints. The original plug-in design also incorporated a menu that would allow the user to choose between a full-wave rectifier, full-wave integrator, or soft clipping distortion as the NLD that the plug-in would utilize for harmonic generation. In addition, the author was not able to experiment with a phase vocoder implementation of the virtual bass algorithm [7], which could have been another viable option for the NLD stage. The filtering stage could have also been improved by giving the filter cut-off frequencies the ability to be adjusted in real-time by the user. This feature could vastly improve flexibility because the plug-in could be tuned to a wider variety of input sources as well as playback systems. However, implementing this feature would require that the author perform additional research on digital filter design (a deep topic unto itself) or find a C++ library that designs and implements time-varying elliptical filters.

In spite of the lack of these features, the plug-in still performs well and is a robust, reliable, and effective manifestation of virtual base on the DAW platform.

Acknowledgment

The author would like to sincerely thank his professor Dr. Colby Leider for his assistance and guidance with the design of the tunable virtual bass algorithm. In addition, Dr. Corey Cheng and James Johnston helped with the design of the filtering algorithm and nonlinear device respectively. Many thanks go to the University of Miami Music Engineering Technology program for providing these challenging opportunities for undergraduate students.

References

[1] H. Moon et. al, "Enhanced Bass Reinforcement Algorithm for a Small-sized Transducer," May 2007. [Online]. Available: AES E-Library http://www.aes.org.iiiprxy.library.miami.edu/e-lib/ [Accessed: Mar. 31, 2009].

[2] M. Arora et. al, "Low Complexity Virtual Bass Enhancement Algorithm for Portable Multimedia Device," Sep. 2006. [Online]. Available: AES E-Library http://www.aes.org.iiiprxy.library.miami.edu/e-lib/ [Accessed: Mar. 31, 2009].

[3] N. Oo, "Harmonic and Intermodulation Analysis of Nonlinear Devices Used in Virtual Bass Systems," May 2008. [Online]. Available: AES E-Library http://www.aes.org.iiiprxy.library.miami.edu/e-lib/ [Accessed: Mar. 31, 2009]

[4] E. Larsen and R. M. Aarts, "Perceiving Low Pitch through Small Loudspeakers," Feb. 2000. [Online]. Available: AES E-Library http://www.aes.org.iiiprxy.library.miami.edu/e-lib/ [Accessed: Mar. 31, 2009].

[5] "In a personal conversation with James Johnston (currently Chief Scientist at Neural Audio of DTS Inc.), he recommended the soft clipping equation used by the author in his nonlinear device"

[6] P. Minnaar, "Non-Linear Processing for Low Frequency Enhancement," Aug. 2008. [Online]. Available: AES E-Library http://www.aes.org.iiiprxy.library.miami.edu/e-lib/ [Accessed: Mar. 31, 2009].

[7] R. B. Mingsian, and L. Wan-Chi, "Synthesis and Implementation of Virtual Bass System with a Phase-Vocoder Approach," Nov. 2006. [Online]. Available: AES E-Library http://www.aes.org.iiiprxy.library.miami.edu/e-lib/ [Accessed: Mar. 31, 2009].