Proceedings of the 3rd International Conference on Mechatronics and Smart Systems
DOI: 10.54254/2755-2721/2025.30614

A Review of Convolution and Impulse Response and Their Applications in Audio Technology and Production

Chenrui Cai

Indian Springs School, Pelham, USA
chenrui.cai@indiansprings.org

Abstract. This essay is a summation of the author's personal survey on the topics of convolution, impulse response (IR), and linear time-invariant (LTI) systems. The essay briefly introduces the definition and properties of convolution, unit impulse functions and impulse response, and linear time-invariant systems; discusses further into the time complexity, optimized algorithms of convolution, and the applications of impulse response technology with LTI systems in audio production; and gives a mathematical proof on the linearity of a modeled non-LTI system. The essay addresses the importance of these applications and the impact on music industry, points out the limitation of applicable scenarios of these technologies due to the non-linearity of some systems, and concludes some possible directions for future development of convolution and IR technology in audio production.

Keywords: Signal processing, Impulse Response, Convolution, Linear Time-Invariant System, Audio Effect and Modeling

1. Introduction

With the advent of digital age, computers have provided audio engineers and musicians numerous ways to create and modify their sounds that could not be achieved with analog devices. As the computing power of electronic devices has gradually developed over the past few decades, many algorithms that were once impossible to put into practice are now widely used in many areas of signal processing and have greatly changed the way music is made and the entire music industry. One of these algorithms is convolution, which is a perfect example of implantation due to advancement in computers. In audio production, convolution is sometimes used with impulse responses sampled from spaces or devices (in other words, systems, such as rooms with good reverberation and guitar cabinets). An impulse response is a signal that can store the tonal characteristics of these systems, which can then be applied to any signal through convolution. This technology provides users with a convenient and cost-friendly way to reproduce certain sounds of certain equipment that they may not have or can't afford, thereby greatly reducing the cost and price of music and audio production, lowering the threshold for making music and opening opportunities for more people [1, 2]. However, this technology has its limitations and problems. Although convolution and IR technology are very mature and highly accurate, they are still relatively new and have only become popular and commercially available in the last 20 years due to the complex nature of computing convolutions and the lack to ability to calculate in short time [3]. Furthermore, due to the nature of the mathematics, impulse responses can only be used in limited situations in terms of the types of systems they can measure [4]. In this work, I outline definitions of convolution, impulse response, and related terms; explore the past and present issues that limit their application; and present some existing solutions addressing these issues.

2. Convolution

2.1. Definition

Convolution is one of the most widely used mathematical operations in signal processing with many applications [5, 6]. In the simplest way, convolution is a mathematical operation to blend two functions and produce the third one [6]. Its mathematical definition is defined as following:

$$(f*g)\,(t)=\int_{-\infty}^{\infty}f(\tau)g(t-\tau)d\tau \tag{1}$$

Note that convolution operation follows commutativity. Therefore, it can be also written as:

$$(f*g)\,(t)=\int_{-\infty}^{\infty}f(t-\tau)g(\tau)d\tau \tag{2}$$

From the definition, convolution is an integral of the product of two functions (usually in the time domain, thus they are of the variable t) over the dummy variable “ τ ”, evaluated for all values on the axis, and one of the functions is reversed (note that it is $-\tau$ in the function g) and shifted by a certain value t. Also, the asterisk symbol “*” is used as the operator for convolution for simplification.

Another commonly seen mathematical definition is as following:

$$\int_{0}^{t}f(\tau)g(t-\tau)d\tau \tag{3}$$

This is used when both functions are only defined from 0 to infinity, usually in the real-life situation where the functions are dependent on time, and there is a given upper limit to denote where the convolution should stop.

Also, for discrete-time signals, commonly seen in applications in computer science, convolution can be defined as following:

$$\sum_{k=0}^{n}f[k]g[n-k] \tag{4}$$

In which the integration sign is replaced by a summation, and the “n” represents the length of samples or the input size [5].

To explain how convolution works, it is reasonable to think that firstly, the -τ in function g reverse the function, and adding the time off-set t allows it to sweep through function f on the τ axis. Then, the integral renders the area under the curve of the product the two functions f and g, measuring the amount of the two functions overlap each other during the sweeping [6].

2.2. Convolution theorem

To introduce the convolution theorem, it is necessary to define what Fourier Transform is. Fourier transform is the mathematical operation that transform the signal from the time domain (relative to the variable t) to the frequency domain (relative to the variable ω , the angular frequency) [5]. Simply speaking, it will produce a curve that demonstrates the frequency contents of a complex signal. It is mathematically defined as:

$$\mathscr{F}\{g(t)\} = \int_{-\infty}^{\infty} g(t)e^{-i\omega t}dt = G(\omega),\ where\ \omega = 2\pi f \tag{5}$$

Also, there exists the Inverse Fourier Transform, which can convert the signal from the frequency domain back to the time domain:

$$\mathscr{F}^{-1}\{G(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} G(\omega)e^{i\omega t}dt, \tag{6}$$

The convolution theorem states that, the Fourier transform of the given convolution f*g is equal to the product of f and g after being Fourier transformed individually. In more general terms, it means that the convolution in the time domain equals the multiplication in the frequency domain [5, 6]. In mathematical terms, it can be written as following:

$$\mathscr{F}\{f(t) * g(t)\} = \mathscr{F}\{f(t)\} \cdot \mathscr{F}\{g(t)\} = F(\omega) \cdot G(\omega) \tag{7}$$

The converse case of this theorem is also a true statement:

$$\mathscr{F}^{-1}\{F(\omega) * F(\omega)\} = \mathscr{F}^{-1}\{F(\omega)\} \cdot \mathscr{F}^{-1}\{G(\omega)\} = f(t) \cdot g(t) \tag{8}$$

This means that the convolution in the frequency domain is also equal to the multiplication in time domain. But for the purpose of this survey essay, the first statement is relatively more important.

The convolution theorem allows people to take full advantage of its properties and apply it to many occasions. From a purely mathematical point of view, the theorem allows the calculation of the convolution result without having to perform the convolution and the calculation of the inner integral, but directly converting the two functions to the frequency domain with Fourier transform and then performing the multiplication. This is a very useful shortcut when the integral or summation of the convolution is too difficult or even impossible to calculate. Compared with the integral of the convolution, the integral of the Fourier transform is relatively easy to calculate. By applying this shortcut, the convolution can be solved more easily in many cases [5].

2.3. Time complexity of convolution

When implementing convolution in real life, the computation complexity or time complexity of the algorithm must be considered. As mentioned above, convolution is a very difficult operation to work with. When calculating convolution with computers, the signals must be first converted into discrete time, and using the discrete time convolution (Equation 4) to compute.

In computer science, the big O notation is commonly used to describe the time complexity of an algorithm. It demonstrates the number of steps needed to do the computation to the numbers of samples inputted and offers an easy way to classify different types of algorithms. Unfortunately, discrete-time convolution is an O(N^2) algorithm, where N is the input size [5]. This means that it grows quadratically as the number of input samples increases, which is a very fast rate of increase in step size [5]. While O(N^2) algorithms work well when the input size is limited to a small number of samples, if the input size gets larger, the number of computational steps will increase dramatically, eventually taking too much time and being unsuitable in certain situations [5].

Early computers often faced this dilemma. Although a certain algorithm had been discovered by mathematicians' decades or even hundreds of years ago, computer scientists were unable to put it into practice due to the high computational complexity required and the insufficient computing power of early computers. They could not complete the calculations within the time specified by the task to be solved. This is a very true situation to the convolution. While reverberations based on digital algorithms have been around since the 1970s (these algorithms require less computation at the expense of quality), the first real-time convolution reverb didn't appear until 1999. It was the Sony DRE S777, which finally had enough computational power and could calculate convolution in real-time [3, 7].

Figure 1. Sony DRE S777 [3]

Due to the limit of computational power how computers, all algorithms must be simplified to have actual implementation. To reduce the complexity of computing convolution, there are many improved algorithms developed over the years. One of them is the Fast Convolution. It exploits the convolution theorem (Equation 6), calculates the multiplication in the frequency domain [5], and implements the Fast Fourier Transform algorithm within itself to reduce the complexity [6].

The Fast Fourier Transform (FFT) is an algorithm optimized from the Discrete Fourier Transform (DFT) in 1965 by James Cooley and John Tukey (though it had been found but not published by Gauss two centuries ago) [8]. What this algorithm most known for is that it can significantly reduce the computational complexity of the traditional DFT algorithm from O(N^2) down to O(Nlog$_2$(N)), and therefore made possible for the computers with deficient computational power to run the Fourier Transforms for a large amount of data within a reasonable time [6]. Moreover, the reduction of complexity is the key to the development of Fast Convolution, which also replace the DFT with FFT to reduce the complexity of the traditional discrete time convolution:

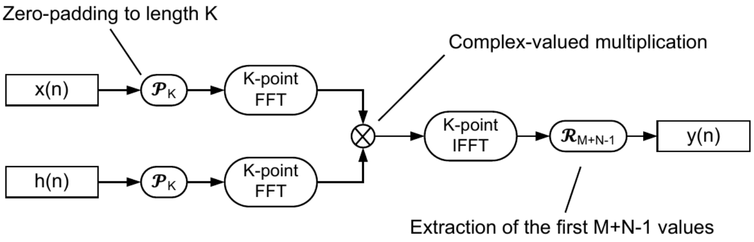

Figure 2. The flow chart of Fast Convolution [9]

As shown on Figure 2, the Fast Convolution algorithm will first convert the two discrete time signals x(n) and h(n) to the frequency domain with FFT, and multiply them directly, then apply the Inverse Fast Fourier Transform (IFFT) to convert the result back to the time domain, and eventually produce the result y(n). According to the convolution theorem, the convolution in time domain equals multiplication in frequency domain. Hence, the result y(t) is identical to the result if the convolution in time had been truly performed, and the FFT and IFFT algorithm would do the job converting back and forth from the two domains and reduce the complexity as well. As a result, the Fast Convolution can reduce the complexity from O(N^2) to O(Nlog(N)) [9].

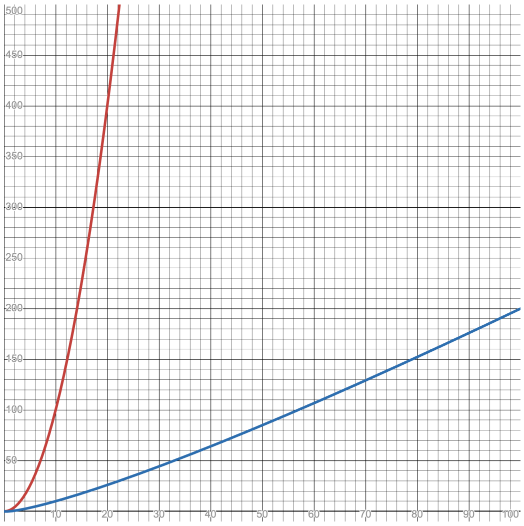

Figure 3. Comparison between O(Nlog(N)) (blue) and O(N^2) (red) complexities

In Figure 3, horizontal axis is the input size N, and the vertical axis is the number of operations needed. It shows that the O(Nlog(N)) can significantly reduce the needed steps from the original O(N^2) complexity. If the input size is 20 samples, O(N^2) needs 400 steps to perform the operation, whereas O(Nlog(N)) will take only about 30 steps. And this reduction of steps will become more significant as the input size increases. When the input size becomes 100 samples, O(Nlog(N)) requires roughly 200 steps, but the steps needed for O(N^2) will escalate to 10000. Therefore, the O(Nlog(N)) complexity is very useful dealing with a huge amount of data [10].

3. Impulse Response (IR)

3.1. Delta function / unit impulse function

The Dirac delta function, or unit impulse function, is a function of which value is zero everywhere on the axis, but infinity on the origin [11]. It is usually denoted by the Greek letter δ and is mathematically defined as:

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases} \tag{9}$$

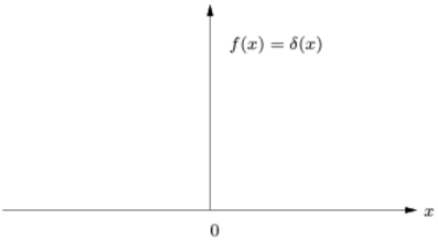

Figure 4. Delta function in time domain [11]

As shown on the figure, the delta function is essentially an infinitely thin line with infinitely high height, and it is defined to have the area of 1, meaning that the integration over itself is defined to be 1. Similarly, doing Fourier transform over itself yields the value of 1 [11]:

$$\int_{-\infty}^{\infty} \delta(x)dx = 1 \tag{10}$$

$$\mathscr{F}\{\delta(x)\} = \int_{-\infty}^{\infty} \delta(x)e^{-i\omega t}dx = 1 \tag{11}$$

This means that the delta function, or an impulse, will span over the frequency domain evenly, and with only the value of 1 [11].

Another important feature of delta function is the shifting property. It means that when convolve delta function with an arbitrary function f(x), the result will still be the function itself [11]. And if the delta function is shifted by some value a, the result of convolution will produce the same function f, but also shifted by the value a. Mathematically, it could be written as:

$$f(t) * \delta(t-a) = f(t-a) \tag{12}$$

3.2. Impulse response

The impulse response is the reaction or output of a system, a function, or a transformation f(x) - could be either physically tangible or mathematically constructed - when the unit impulse function □(t) is inputted, and it is usually denoted as $h(t)$. It could be mathematically defined as [12]:

$$h(t) = f(\delta(\mathrm{t})) \tag{13}$$

3.3. Linear Time-Invariant (LTI) systems

The property of linearity and time-invariance is pivotal to signal processing. A linear and time-invariant system is a system which has these properties, reacting to the input, and behaving accordingly. If a system is linear, it must follow the principle of superposition [12, 13].

The principle of superposition consists of law of additivity and law of homogeneity. Law of additivity means that, if the system f(x) is linear, and there exits two signals - $s_1(t)$ and $s_2(t)$ - that will be feed into a system, then the output of the system of the sum of $s_1(t)$ and $s_2(t)$ will be the same as the sum of the outputs of inputting the two signals individually [12, 13]. It could be written as:

$$f(s_1 + s_2) = f(s_1) + f(s_2) \tag{14}$$

Similarly, the law of homogeneity means that, if the system f(x) is linear, and the signal s(t) is multiplied by a scaler a before being feed into the system, the output will be the same as multiplying that scaler a with the output of feeding the signal s(t) directly into the system [12, 13].

$$f(as) = af(s) \tag{15}$$

Time invariant means that time t is not a variable of the system, meaning that the system does not change as time goes, and the output of the system will always be the same no matter when to input. Moreover, if a time-shifted signal is inputted to a time-invariant system, the time-shifting will be presented in the output signal [12, 13]:

$$f(s(t)) = y(t),\ and\ f(s(t-d)) = y(t-d) \tag{16}$$

One crucial property of LTI systems is that they can be completely described by their impulse response. And by performing a convolution between the IR h(t) of an LTI system f(t) and the input signal s(t), it will produce the same result with the output y(t) of the system as if s(t) has gone through the system [13]. It could be mathematically written as:

$$y(t) = \ (s * h)(t) = f(s(t)) \tag{17}$$

By exploiting this property, it is possible to get the wanted result just with the signal and the IR, and the system itself can be bypassed. This is highly efficient and beneficial when a system, especially a physical object, is not available due to many possible reasons - it is at somewhere else, operating it will cost a lot of effort, or it is simply too expensive to own. But by recording the IR of that system, people can store it digitally without taking up any physical space, carry it to other places or distribute it to others easily, and do the convolution to get the output with little effort [1, 2].

3.4. Applications of impulse responses of LTI systems

3.4.1. Convolution reverberation and guitar cabinet simulation

The most commonly application of such property of impulse responses and LTI systems stated in Equation (17) is the convolution reverberation. Reverberation is an acoustic phenomenon that exists in the real world and is one of the most important effects in audio production. It is the reflection plus the attenuation and amplifications of certain frequencies of the sound from a sound source in an arbitrary space, and thus giving the listener a sense of space. Sometimes, the reverberation can be recorded directly from a real space, a technique called chamber reverb, which is done playing the music through the loudspeaker on one end of a chamber and record the reverb with the microphone on the other, and eventually mix the recorded reverb with the music [7]. This technique could be very costly and not affordable to many, since it requires the user to build a real chamber. Therefore, artificial reverberation was developed. Before the first convolution reverb Sony DRE 777 came out in 1999, many artificial reverb techniques existed both in analog and in digital, such as plate and spring reverb, which captures the vibration of a real metal plate or spring to generate reverberation; the first digital reverberation was published in 1976, which is based on algorithms to calculate a series of differently weighted delay signals to simulate the early reflections and multiple filter, damping, and feedback circuits to simulate the later dense stage of reverberation ("the tail") [3, 7]. However, none of these artificially made reverberations sound natural enough, since they are merely virtual simulations.

Convolution reverberation is built differently – it is no longer a simulation but based on capturing the impulse response of real spaces, and it is highly accurate and natural on reproducing the spatial feeling comparing to others 7. As mentioned above, the impulse response of a linear time-invariant system can completely describe this system, and convolving the impulse response with a given signal will produce the very same result from that system. Also, the system can be either mathematically constructed formulas or physically tangible objects. Any space, whether it is a bathroom in a house, the hall of a church, a sport stadium that can hold thousands of people, or even an open space in the wild, can be seen as LTI systems. Any sound played in those spaces will get modified by the acoustic characteristics due to the material and the placement of the walls or any object: bumping into the walls, getting some frequencies filtered out or amplified, bouncing back, and repeating these steps many times [3]. This transforms the sound to reverberation, and it will be mixed with the original sound directly from the sound source, and eventually be heard by the audience in the same space. Relating back to the concept of LTI systems, the sound is the input of this system, and the reflections and modifications are the transformation, and the output is what the audience will hear. By replace the input with a unit impulse, it will produce the impulse response, and the space itself can be completely described by this IR signal, containing all the spatial characters. And convolve any audio signal with this IR will pass the spatial characters onto the signal, thus producing the reverberation [3].

Another popular application of impulse responses is guitar cabinet simulation. The concept is very similar to the convolution reverberation, except the system here is a guitar cabinet. Guitar cabinets are essentially loudspeakers that transduces the amplified guitar signals from the amplifier section to soundwaves. Their frequency responses are often intentionally altered, as opposed to monitor or public address (PA) speakers which advances for frequency curves to be as flat or as accurate as possible [14]. This feature makes guitar cabs to act like an equalizer, attenuating, amplifying, and even cutting out completely some frequencies of the raw signal, and thus giving the conventionally perceived guitar tone [14]. By playing an impulse through a guitar cabinet and capturing its impulse response with microphones, people may retain the frequency curve of that specific guitar cabinet in the IR signal and convolve the IR with raw guitar signals after the amplification stage to apply this curve onto it, thus producing the needed guitar tone [1, 2].

It is very worth noting that, in real situations, the impulse response does not only represent the system it measures, but also by the speaker playing it and the microphone capturing it, and the position they are placed in the space or the system. Also, impulse response only stores a system's linear feature but not the section that is non-linear. If any system contains a portion that does not present the characteristic of linearity and time-invariance, it is impossible for impulse response to represent the whole system.

3.4.2. The impact on the industry of music production

Since the characteristics of a complex system can be stored into a tiny impulse response signal in digital space, the impulse responses can be easily distributed online and made available to musicians around the world. Additionally, it means that producers no longer need to own a real reverb room or a set of 4x12 guitar amplifiers which can cost tens of thousands of dollars to get the desired sound but can simply use recorded impulse responses they downloaded online with just a personal computer, significantly reducing the cost and limitation of music production, inviting individual musicians to show their talents and diversifying the music market [1, 2].

3.5. Non-LTI systems and proof of linearity

Aside from rooms and guitar cabinets, there are many other systems that can be LTI, and the property of impulse response in Equation (17) can be exploited. However, in audio production, non-LTI systems are also very common. A non-LTI system could either be not following the principle of superposition (nonlinear), or simply changes relative to time (time-variant). More importantly, their impulse responses cannot fully characterize themselves, thus limiting the application of IR technology [4]. It is possible to mathematically test the principle of superposition and time-invariance to see whether some systems are LTI or not [15].

3.5.1. Example of non-LTI systems-distortion

One of the non-LTI systems is distortion, an audio effect that is almost ubiquitous in audio production. It shapes the waveform by manipulating the relationship between input and output, compressing the dynamic range of the signal, and generating new spectral information [4, 15]. One type of distortion is hard clipping, which allows the signal to pass under a certain threshold at some value, but "clips" the signal to a fixed value for the portions above the threshold. Therefore, for the output signal, the portion below the threshold will be the same to the input signal, but the portion above the threshold will be completely flatted to a line.

3.5.2. Mathematical proof of non-linearity of distortion

It is possible to mathematically model a transfer function of hard-clipping effect in many ways, one of them is by using a piece-wise function [15]. For example:

$$f\left(x\right)=\begin{cases}-5, x\leq-5\\ x,-5<x<5\\ 5,\ x\geq5\end{cases} \tag{18}$$

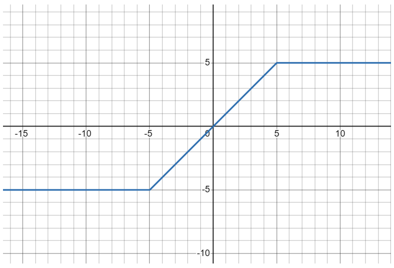

Figure 5. Graph of f(x)

On Figure 5, the horizontal axis is the value of input, and the transfer function f(x) will transform the input to the corresponding value on the vertical axis, which is the output. The transfer function f(x), of which the threshold is 5, means that any input signal above the numeric value 5 or below -5 will be hard clipped to 5 to the output, but any signal between -5 and 5 will be outputted linearly. This relationship is also shown on the graph, on which the x-axis represents the input, and the y-axis is the output.

With this modeling by math equations, it is possible to prove its principle of superposition mathematically. Also, note that law of additivity and homogeneity build on each other, and if either of the two laws is violated, linearity doesn’t hold on this system, and thus there is no need to test the other law. Therefore, to test the linearity the modeling of hard clipping mentioned above, the following is the demonstration testing the law of additivity with both mathematical equations and graphs produced by Desmos.

Suppose there exist two signals $s_1(t)$ and $s_2(t)$, which are defined as following:

$$s_1\left(t\right)=10\text{cos}(t)$$

$$s_2\left(t\right)=6\cos\left(t\right) \tag{19}$$

Their graphs plotted in respect to time are as following:

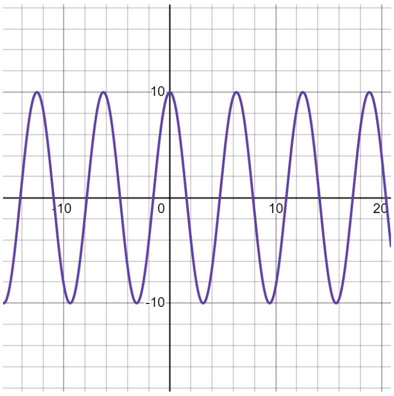

Figure 6. Graph of $s_1(t)$

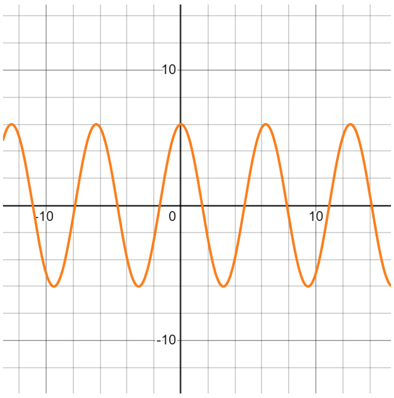

Figure 7. Graph of $s_2(t)$

For the simplicity of this demonstration, the two signals are all set to be simple cosine waves with the same phase and fundamental frequency but different amplitudes.

The law of additivity states that, the output of the system of the sum of the two signals will be the same as the sum of the outputs of the two signals run through the system individually. Therefore, the first step is to obtain the sum of the two signals:

$$s_{sum}(t) = s_1(t) + s_2(t) = 16\cos(t) \tag{20}$$

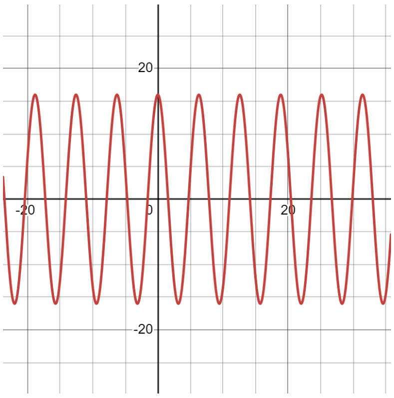

Figure 8. Graph of $s_{sum}(t)$

By adding two signals together, it obtains a cosine wave with the amplitude of 16.

The next step is to input the sum $s_{sum}(t)$ into the system f(x), which renders the result $y_{sum}(t)$:

$$y_{sum}(t) = f(s_{sum}(t)) = \begin{cases} -5, s_{sum}(t) \leq -5 \\ s_{sum}(t), -5 < s_{sum}(t) < 5 \\ 5,\ s_{sum}(t) \geq 5 \end{cases} \tag{21}$$

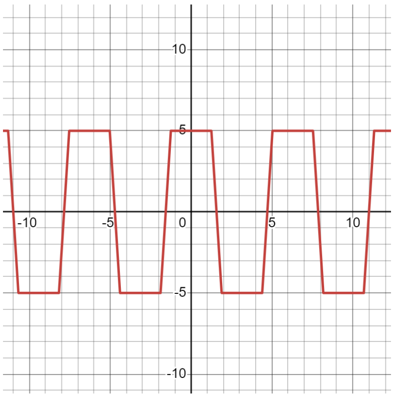

Figure 9. Graph of $y_{sum}(t)$

Since the transfer function f(x) is defined piece-wisely, by inputting $s_{sum}(t)$ as x into the system, the output $y_{sum}(t)$ is also piecewise.

To test the last half of law of additivity, the two signals will be first inputted into f(x) individually. For $s_1(t)$:

$$y_1\left(t\right)=f\left(s_1(t)\right)=\begin{cases}-5, s_1(t)\leq -5\\ s_2(t), -5<s_2(t)<5\\ 5,\ s_2(t)\geq 5\end{cases}\tag{22}$$

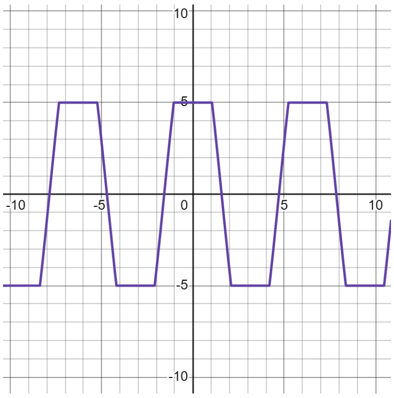

Figure 10. Graph of $y_1(t)$

For $s_2(t)$:

$$y_2\left(t\right)=f\left(s_2(t)\right)=\begin{cases}-5, s_2(t)\leq -5\\ s_2(t), -5<s_2(t)<5\\ 5,\ s_2(t)\geq 5\end{cases}\tag{23}$$

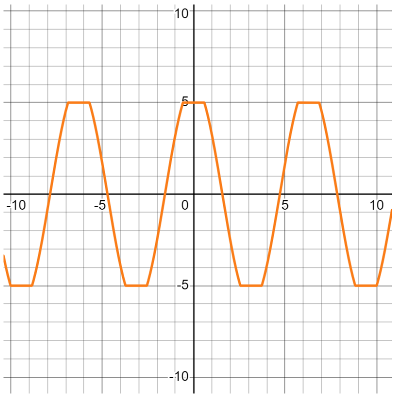

Figure 11. Graph of $y_2(t)$

After summing up the two outputs $y_1(t)$ and $y_2(t)$, it yields the result g(t) that:

$$g\left(t\right) = \begin{cases} -10,\ y_1\left(t\right)+y_2\left(t\right) \le -10 \\ y_1\left(t\right)+y_2\left(t\right),\ -10 < y_1\left(t\right)+y_2\left(t\right) < 10 \\ 10,\ y_1\left(t\right)+y_2\left(t\right) \ge 10 \end{cases} \tag{24}$$

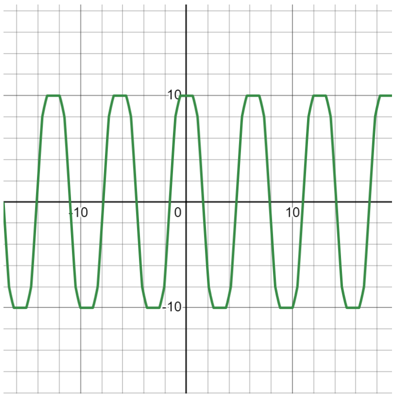

Figure 12. Graph of g(t)

The final step is to compare the output of the sum and the sum of the two outputs to see whether they are the same. Unfortunately, they do not equal, and thus the law of additivity is violated:

$$y_{sum}\left(t\right) \ \neq g\left(t\right),\ \ \therefore law\ of\ addtiviy\ violated \tag{25}$$

3.5.3. Discussion

Inspecting either Equation (21) and (23) or Figure 9 and Figure 12, the output of the sum $y_{sum}(t)$ and the sum of the two outputs g(t) is clearly not the same. Although the fundamental frequency and phase retailed same and unchanged, the amplitude and the shape are different. $y_{sum}(t)$, if seen as a continuous sinusoidal signal from the graph, has the amplitude of only 5 due to the hard-clipping effect. But g(t) is a summation of the outputs $y_1(t)$ and $y_2(t)$, meaning that their amplitude, both are 5, is added linearly, and thus has the value of 10.

Since the law of additivity has been proven to be false, it means that any system resembling hard clipping and distortion is not a linear system, and thus they do not belong to the category of LTI systems. This means that the impulse response of a distortion system cannot characterize itself, and the application of IR cannot be used on such systems. Therefore, this fact of distortion limits the applicable scenarios of using the IRs, when the system involves some components that do distortion to the signal. For example, although there exists countless IRs of guitar cabinets, IRs of guitar amplifiers have never been an actual thing, since the amplification process involves distorting the signal, and distortion is not an LTI system, thus recording the IR of a guitar amplifier is not a practical way attempting to recreate its sound [16].

4. Conclusion

Although convolution was once difficult to implement on computers, the development of optimized algorithms such as fast convolution and advances in computing power over the years have made it less of a problem and it is now widely used in music production in the form of convolution reverb or guitar amp simulations. These applications have revolutionized the music industry and community. From a mathematical point of view, the properties of LTI systems and impulse responses make these applications possible; yet, mathematics also limits the application of impulse response techniques, as non-LTI systems (such as distortion) cannot be characterized by their impulse response, so it is impossible to model a system or audio device (such as a guitar amplifier) involving non-LTI parts by recording its impulse response. For future works, regarding extending the application of convolution and impulse responses, there already exits some studies with innovated algorithms that allow the approximation of non-LTI systems such as distortion with non-linear convolution, Volterra series, and a summation of impulse responses of the harmonics in order generated by the system [4, 16]. Another promising direction is to develop algorithms other than convolution and impulse responses for modeling non-LTI systems [16]. But whichever route people take, there is no doubt that music production in the future will become more efficient and easier.

References

[1] Soundwoofer, “What is an Impulse response and how can it change the music industry?, ” Mynewsdesk, Jul. 14, 2022. https: //www.mynewsdesk.com/soundwoofer/pressreleases/what-is-an-impulse-response-and-how-can-it-change-the-music-industry-3194839 (accessed Sep. 12, 2024).

[2] A. Munoz, “Soundwoofer’s Free Impulse Response Library To Revolutionize Music Industry, ” muzique, Jun. 28, 2020. https: //muziquemagazine.com/soundwoofers-free-impulse-response-library-to-revolutionize-music-industry/ (accessed Sep. 11, 2024).

[3] P. White, “Choosing The Right Reverb, ” www.soundonsound.com, Mar. 2006. https: //www.soundonsound.com/techniques/choosing-right-reverb

[4] A. Farina, A. Bellini, and E. Armelloni, “Non-Linear Convolution: A New Approach for the Auralization of Distorting Systems, ” 2001. Accessed: Sep. 12, 2024. [Online]. Available: http: //pcfarina.eng.unipr.it/Public/Papers/154-AES110.PDF

[5] P. Karas and D. Svoboda, “Algorithms for Efficient Computation of Convolution, ” InTech eBooks, Jan. 2013, doi: https: //doi.org/10.5772/51942.

[6] N. Ghidirimschi, “Algorithms for Integer data types, ” 2021. Accessed: Sep. 12, 2024. [Online]. Available: https: //fse.studenttheses.ub.rug.nl/25184/1/bCS_2021_GhidirimschiN.pdf.pdf

[7] Computer Music, “A brief history of reverb, ” MusicRadar, Jun. 24, 2014. https: //www.musicradar.com/tuition/tech/a-brief-history-of-reverb-602421

[8] P. Duhamel and M. Vetterli, “Fast fourier transforms: A tutorial review and a state of the art, ” Signal Processing, vol. 19, no. 4, pp. 259–299, Apr. 1990, doi: https: //doi.org/10.1016/0165-1684(90)90158-u.

[9] “Fast Convolution: FFT-based, Overlap-Add, Overlap-Save, Partitioned - WolfSound, ” thewolfsound.com. https: //thewolfsound.com/fast-convolution-fft-based-overlap-add-overlap-save-partitioned/ [1]

[10] “The Algorithm That Transformed The World, ” www.youtube.com. https: //www.youtube.com/watch?v=nmgFG7PUHfo

[11] “Lecture 31 Fourier transforms and the Dirac delta function.” Available: https: //links.uwaterloo.ca/amath353docs/set11.pdf

[12] J. Cuff and S. Pauly, “Lecture 3 ELE 301: Signals and Systems, ” 2011. Available: https: //www.princeton.edu/~cuff/ele301/files/lecture3_2.pdf

[13] R. Baraniuk, “2.2: Linear Time Invariant Systems, ” Engineering LibreTexts, May 23, 2020. https: //eng.libretexts.org/Bookshelves/Electrical_Engineering/Signal_Processing_and_Modeling/Signals_and_Systems_(Baraniuk_et_al.)/02%3A_Introduction_to_Systems/2.02%3A_Linear_Time_Invariant_Systems

[14] D. H. published, “Guitar Essentials: What You Need to Know About Speaker Cabinets, ” Guitar Player, Apr. 16, 2020. https: //www.guitarplayer.com/gear/guitar-essentials-what-you-need-to-know-about-speaker-cabinets

[15] “Nonlinear Elements | Physical Audio Signal Processing, ” Dsprelated.com, 2024. https: //www.dsprelated.com/freebooks/pasp/Nonlinear_Elements.html (accessed Sep. 12, 2024).

[16] A. Bellini, G. Cibelli, E. Ugolotti, A. Farina and C. Morandi, "Non-linear digital audio processor for dedicated loudspeaker systems, " in IEEE Transactions on Consumer Electronics, vol. 44, no. 3, pp. 1024-1031, Aug. 1998, doi: 10.1109/30.713229.