distCONV: A REVOLUTIONARY TIME-VARIANT, DYNAMIC CONVOLUTION/DISTORTION PROCESSOR!

Daniel John Natoli (SID: 460032124)

Final Written Review for Digital Audio Systems, DESC9115, 2016.
Graduate Program in Audio and Acoustics
Faculty of Architecture, Design and Planning, University of Sydney.

Abstract:

distCONV is a revolutionary dynamic convolution/distortion processor that allows a user to convolve two audio files (.wav) whilst applying time-augmentation and distortion processing to the convolved signal over time. The distCONV function offers multiple, interchangeable distortion algorithms (from subtle saturation to buzz-saw distortion!), time stretching, impulse reversal and FFT-based convolution in two variable systems. The entire process has been streamlined for computational efficiency, and is controlled through a graphical user interface (GUI), offering complete control over the system's parameters.

Introduction & Problem Description:

Most, if not all, commercial DSP convolution algorithms focus on the concept of reverberation, applying the impulse response of a physical space (or hardware unit such as the EMT 140) to an input signal. Some units, notably LiquidSonics' *Reverberate*, offer delay-modulation effects between stereo channels (LiquidSonics, 2009a) but lack other effects-based processing.

distCONV is the first convolution-based algorithm to offer distortion as an audio effect. Whilst the distCONV algorithm still behaves as a fully-functioning convolution reverb, the algorithm can be used in extreme circumstances as a sound design tool; convolving two completely-unrelated audio signals for interesting and gloriously-overdriven results!

This review aims to discuss the technical specifications of the current distCONV model, with physical performance assessments and evaluations. The review will also suggest improvements to the current system, as well as discussing new features for future release as a stand-alone DSP sound design engine.

Technical Specifications:

1. Mono/Stereo Sorting:

The distCONV system is compatible with any combination of mono and stereo audio (.wav) files. To do this, the algorithm works in parallel, offering the following outputs based on the following input combinations:

Fig. 1(a) Mono Input + Mono Impulse Response = Mono Output

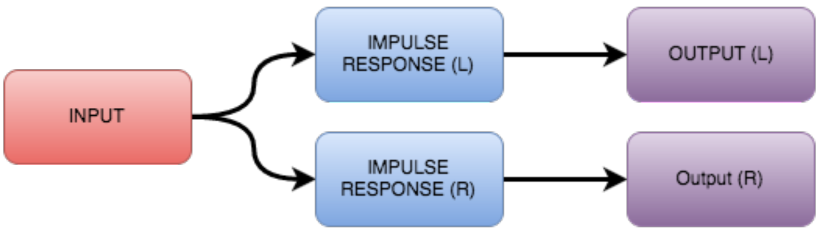


Fig. 1(b) Mono Input + Stereo Impulse Response = Stereo Output

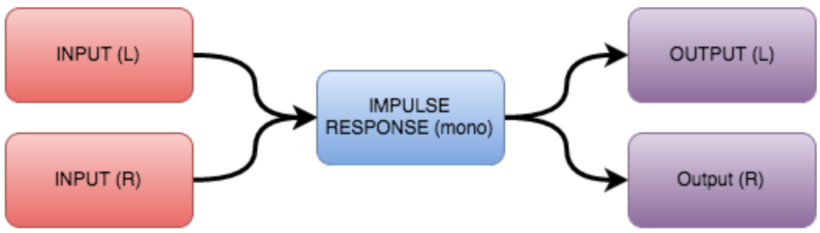


Fig. 1(c) Stereo Input + Mono Impulse Response = Stereo Output

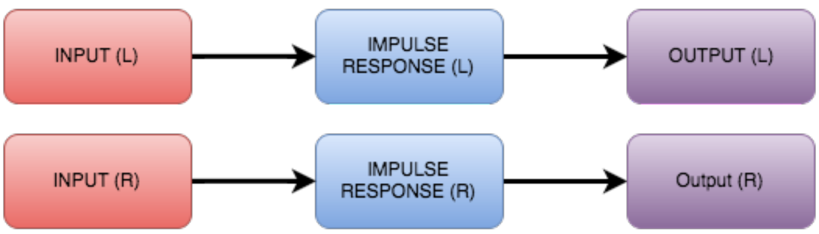


Fig. 1(d) Stereo Input + Stereo Impulse Response = Stereo Output

As a consideration for future release, a "true stereo" model (typically found in high-end algorithmic reverbs) will be implemented where the left and right input components are cross-convolved with their IR counterparts, then summed to stereo (LiquidSonics, 2009b).

2. Distortion Functions

Rather than applying static non-linear distortion to a signal, distCONV applies a *variable* state of distortion *over time*. In essence, the convolved signal will get progressively more distorted over time. distCONV currently offers three distortion functions, each with a unique sonic characteristic:

tanh: "Returns the hyperbolic tangent of each element in an array" (MathWorks, 2016). In this case the *distortion value* parameter relates directly to the curvature of the hyperbolic tangent. "At lower values (< 20) the distortion behaves like a soft-limiter overdrive, whereas at extreme values (> 1000) it inherits bit-crushed, almost buzz saw-like characteristics" (Natoli, 2016, p. 3).

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x}-1}{e^{2x}+1}$$

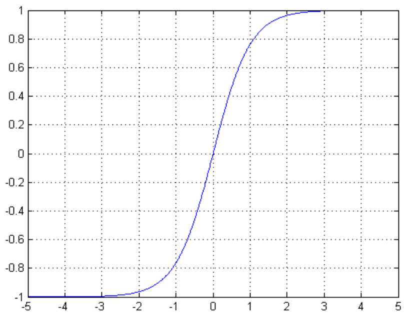


Fig. 2 The hyperbolic tangent function over the domain $-5 \leq x \leq 5$ (MathWorks, 2016).

expdist: A soft-curve distortion simulation based on an exponential function. The simulation is given by the following equation: $f(x) = sgn(x)\left(1 - e^{-|x|}\right)$ (Dutilleux, Holters, Disch, & Zölzer, 2011, p. 127). Produces a subtle overdrive at lower distortion values, with a more "vintage" overdrive aesthetic when pushed.

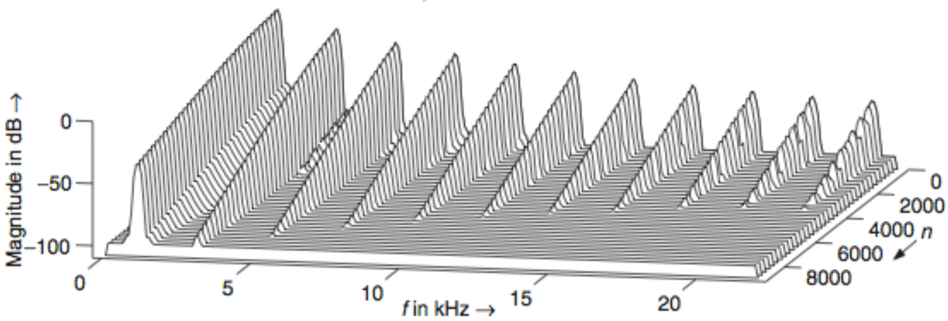


Fig. 3 Short-time FFTs (waterfall representation) of exponential distortion (Dutilleux et al., 2011, p. 128).

symclip: A soft-saturation, nonlinear, symmetrical clipping overdrive. The symclip function provides a midpoint between the harsh hyperbolic tangential distortion and the soft-curve exponential curve.

$$f(x) = \begin{cases} 2x & \text{for} \quad 0 \leq x \leq 1/3 \\ \frac{3-(2-3x)^2}{3} & \text{for} \quad 1/3 \leq x \leq 2/3 \\ 1 & \text{for} \quad 2/3 \leq x \leq 1. \end{cases}$$

"Up to the threshold of 1/3 the input is multiplied by two and the characteristic curve is in its linear region. Between input values of 1/3 up to 2/3, the characteristic curve produces a soft compression described by the middle term of the equation. Above input values of 2/3 the output value is set to one" (Dutilleux et al., 2011, p. 125).

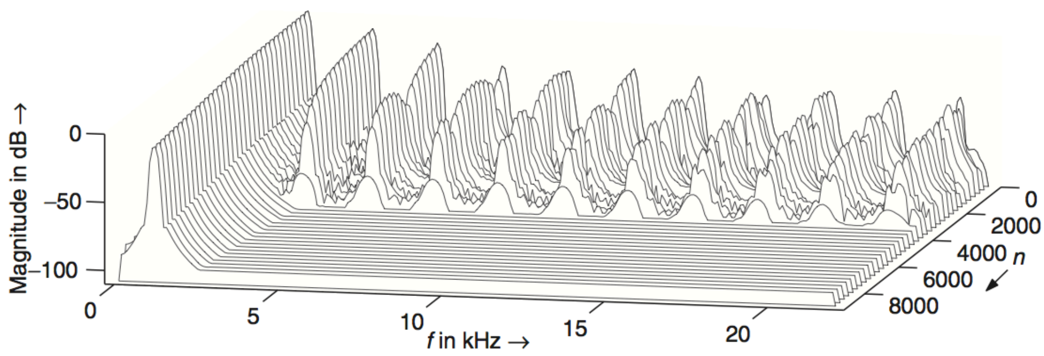


Fig. 4 Short-time FFTs (waterfall representation) of symmetrical hard clipping for a decaying sinusoid of 1kHz (Dutilleux et al., 2011, p. 127).

distCONV's time-variant distortion function revolves around the *my_decay_function* algorithm. A two-dimensional decay curve is created based on the length of the audio file and a variable *decayfactor* parameter. The inverse of this curve is then applied to the convolved audio – as the clean audio signal decays the distorted audio signal increases in level. The *decayfactor* parameter varies the slope of this curve, affecting the rate at which distortion is introduced.

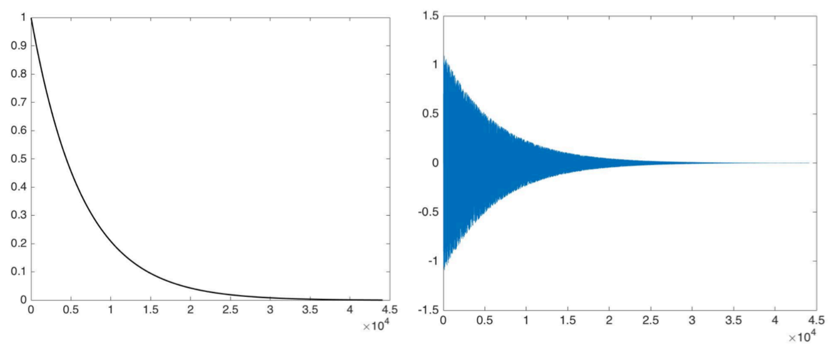


Fig. 5
Left: A 1-second exponential decay curve with a factor of 60dB.
Right: Noise signal multiplied sample-by-sample by the above decay curve.
(Manor & Martens, 2016, p. 2).

3. Reverse Impulse Response:

A common feature in convolution reverb is the option to reverse an impulse response. This feature is achieved by simply flipping the impulse response vector about the horizontal axis. See **Example 6 - Appendix C (p. vii)** for an example.

4. Time Stretching Functionality

distCONV offers a time-stretching algorithm based on the phase-locked system discussed in Laroche and Dolson's paper, *'Improved phase vocoder time-scale modification of audio'* (1999). Alternative time-stretching models (Dutilleux et al., 2011, pp. 249 – 255) may be more computationally efficient, but exhibit phase unwrapping between different bins, meaning that two successive, identical sounds in a LTI system may not be processed he same. Laroche and Dolson's model aligns the phase of each spectral bin to the phase of a singular spectral peak. When stretching to an integer multiple (e.g. 300%, 400%) phase unwrapping to 2π is no longer necessary (Dutilleux et al., 2011, pp. 255 – 257). For use in a convolution algorithm, non-integer multiples are preferred, hence the use of a phase-locked vocoder.

5. System A/B Mode

As discussed in *'Implementing a Time-Variant, Non-Linear Distortion Circuit in a Signal Convolution Algorithm'* the location of the distortion function within the algorithm will dramatically alter the sonic characteristic of the unit (Natoli, 2016). As a result, two system functions have been mapped to distCONV:

> **System A (default):** Distortion is applied *after* the input and impulse response signals are convolved. This can be described as *post-convolution* distortion.
>
> **System B:** Distortion is applied to the impulse response *pre-convolution*. The input signal is then convolved with the distorted version of the impulse response. Note: The distortion curve is still active during System B convolution, resulting in a much more "subtle" distortion effect when compared to System A.

*See **Appendix B (p. iii)** and **Appendix C (p. iv)** for the corresponding signal flow diagrams.*

6. FFT Convolution

For computational efficiency, distCONV utilises FFT-based convolution. "The input signal is transformed into the frequency domain using DFT, multiplied by the frequency response of the filter, then transformed back into the time domain using the Inverse DFT" (Smith, 1999, p. 312).

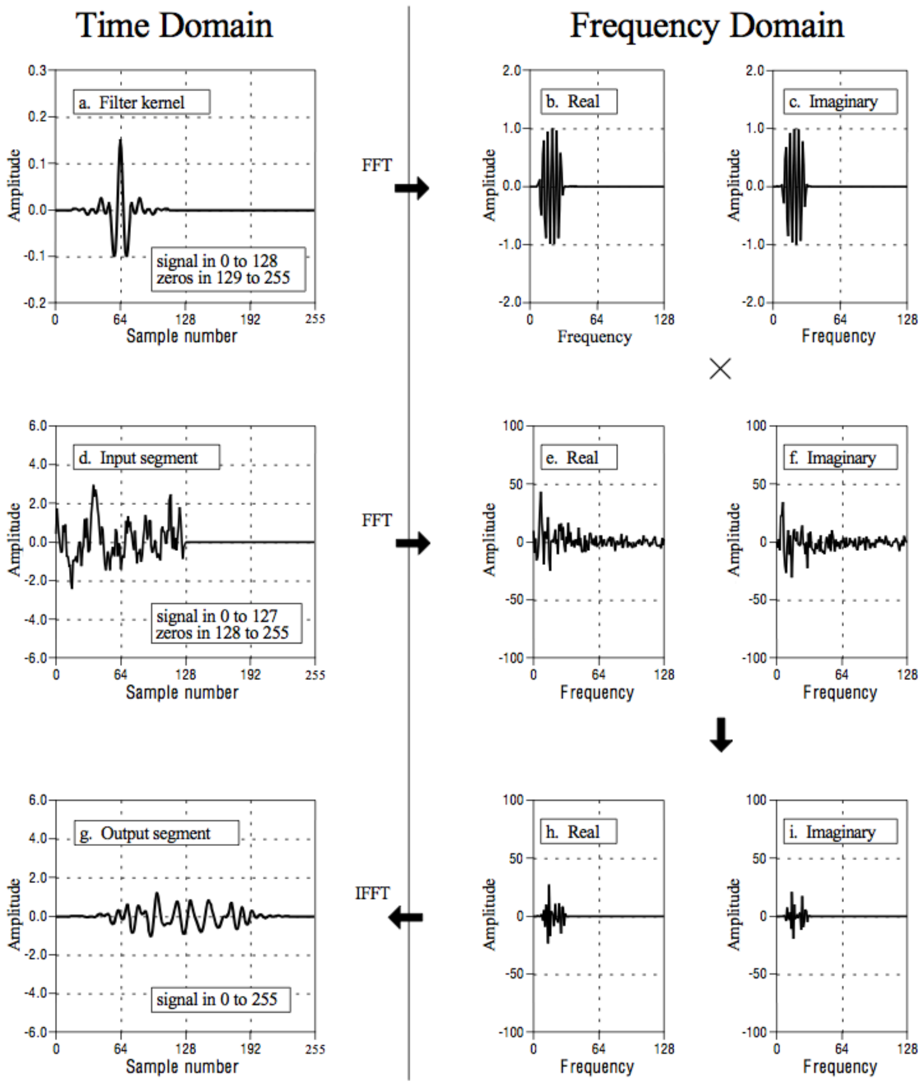


Fig. 6 The process of FFT convolution (Smith, 1999, p. 315).

Numerous tests were conducted using MATLAB R2014a to compare time-domain convolution to frequency-domain convolution. On average, a 4,500% increase in computational efficiency was observed when convolving two single-channel audio signals using FFT convolution (Natoli, 2016, p. 2).

Implementation (UX and UI)

The current build of distCONV offers a simple, graphical user interface for interaction with the system. The user is prompted to select their input and IR files (.wav) then faced with a dialogue box to adjust the algorithm's input variables.

Distortion Type: Select between **tanhdist, expdist** and **symclip** distortion functions.

Decayfactor: Determines the curvature of the distortion function in dB. Default "1" = linear.

Input/IR Mix: Blend between the original input signal, and convolved/distorted IR.

Distortion Wet/Dry Mix: Blend between the clean IR and distorted IR signals.

Distortion Value: The "gain" of the selected distortion function. Experiment with values!

Reverse Impulse Response: Self explanatory.

Sampling Rate: Adjust the sampling rate. Default = 44.1kHz

Time Stretch: Prompts the time-stretch dialogue box, where a **stretch ratio** and **window size** can be set.

System Type: Switch between **System A** and **System B** as discussed above.

Fig. 7 Input dialogue box for distCONV, with default settings (MATLAB R2014a on OSX 10.7.5).

Pressing "OK" allows the system to run the assigned variables. An output signal is auditioned and a plot of the input/IR/output files is displayed. The user is then asked whether they would like to export the output file to disk (as a .wav file at the designated sampling rate).

No messy code… No complicated interface…
Clean, efficient processing for absolutely filthy distortion!

Note: Setting the "Distortion Wet/Dry Mix" parameter to 0% allows the algorithm to behave like a traditional convolution reverb!

Performance Assessment and Evaluation

The distCONV algorithm is extremely stable and has a number of built-in error-correction functions that deal with incorrect variables and system switching. The use of frequency-domain processing greatly reduces computation time, and allows offline processing (sub-real-time) to occur. The algorithm accepts a wide range of usable values and promotes interesting and creative results, every time.

Considering that distCONV is the first algorithm of its kind, no accurate performance comparison with other convolution platforms can be made. Instead, as a performance assessment, a number of raw audio files and convolution examples have been supplied to demonstrate the range of possible outputs. The user should select **only two** raw audio files from that list, then use the distCONV algorithm in as many possible combinations; testing the different input slots (input = file 1, IR = file 2 *and* input = file 2, IR = file 1), systems (A/B) and distortion functions (with IR reversal and time-stretching functionality on/off) to output as many different results as possible! The results are, quite literally, infinite.

Examples 1 – 4 compare the different distortion functions using the same audio files and default parameters, so as to accurately compare the sonic characteristics of each function. All output files have been plotted against their input and IR counterparts, with the appropriate input variables documented in order to re-create the result. These files can be found in **Appendix D (pp. v - vii)**.

Conclusion

With variable distortion functions, decay curves, time stretching, impulse response reversal *and* high-speed FFT convolution, distCONV is the world's most advanced dynamic, time-variant convolution/distortion processor.

With the groundwork laid for a fully functional variable system, producing a stand-alone application based on the distCONV algorithm is a simple matter of interfacing and porting. The rest is up to you!

Bibliography:

Dutilleux, P., Holters, M., Disch, S., & Zölzer, U. (2011). *DAFX: Digital Audio Effects* (Second ed.). John Wiley & Sons Ltd., West Sussex, UK.

Laroche, J., & Dolson, M. (1999) Improved phase vocoder time-scale modification of audio. *IEEE Trans. Speech Audio Process.*, 7(3): 323–332.

LiquidSonics. (2009a). *Reverberate.* Retrieved May 28, 2016, from http://www.liquidsonics.com/software_reverberate.htm

LiquidSonics. (2009b). *Reverberate: True Stereo.* Retrieved May 28, 2016, from http://www.liquidsonics.com/software_reverberate_true_stereo.htm

MathWorks. (2016). *tanh [Help Documentation].* Retrieved April 20, 2016, from http://au.mathworks.com/help/matlab/ref/tanh.html

Manor, E., & Martens, W. L. (2016). *Week 4 – Convolution and Simple Reverb Simulation* [Tutorial PowerPoint Slides]. The University of Sydney.

Natoli, D. (2016). *Implementing a Time-Variant, Non-Linear Distortion Circuit in a Signal Convolution Algorithm.* Unpublished manuscript. The University of Sydney.

Smith, S. W. (1999). *The Scientist and Engineer's Guide to Digital Signal Processing* (Second ed.) United States of America: California Technical Publishing.

Appendix A:

Files included in distCONV package:

distconv.m (Top-level script. Run this file).

Functions_for_distconv

- indlgbox.m
- stretchtest.m
- princarg.m
- playback.m
- my_decay_function.m
- mix_percentage_check.m
- fft_conv.m
- distconv_systemA.m
- distconv_systemB.m
- distconv_function.m
- expdist.m
- tanhdist.m
- symclip.m

Convolution_Examples

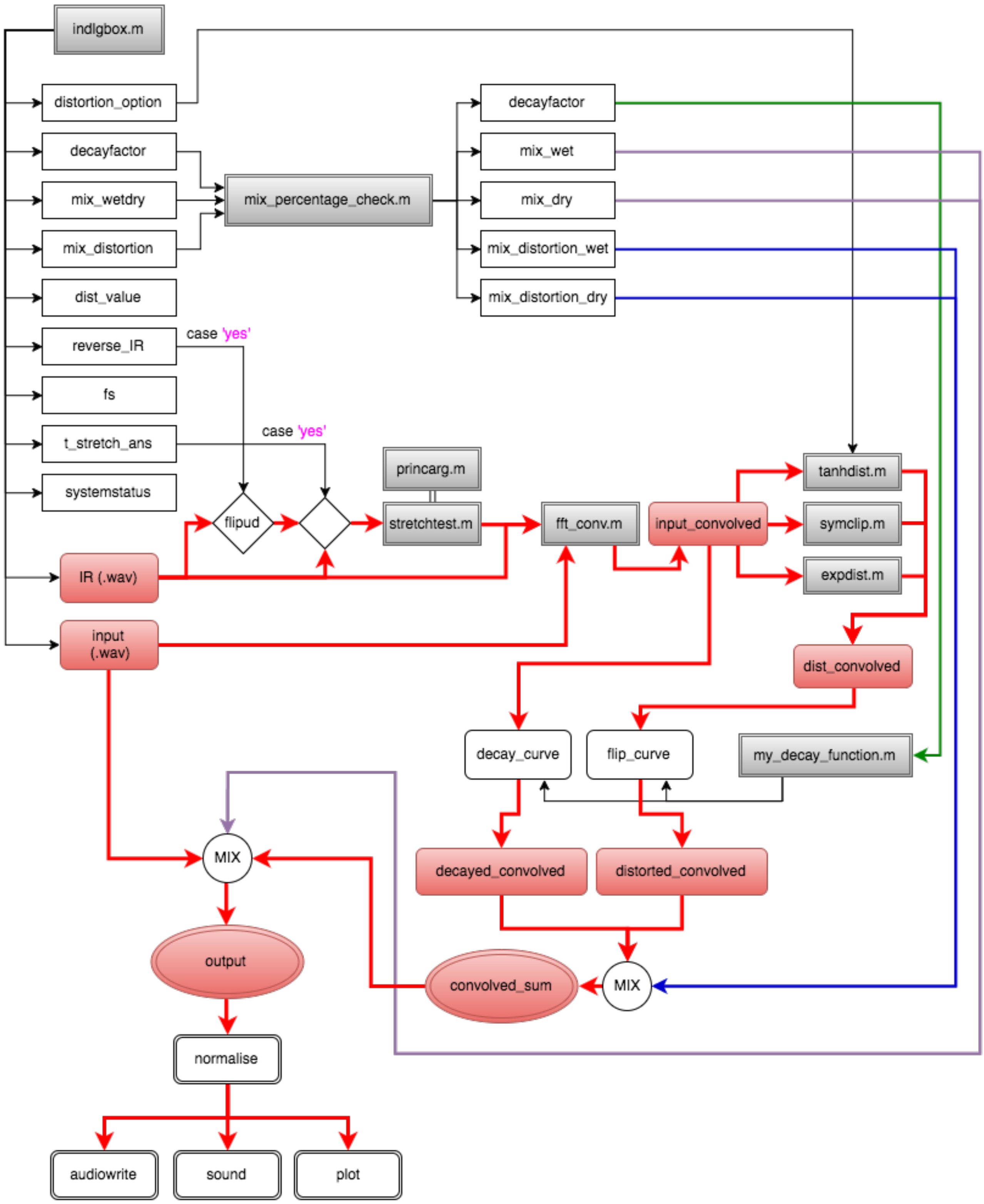
- Example1_la_drumloopA.wav
- Example2_la_drumloopB.wav
- Example3_la_drumloop_expdist.wav
- Example4_la_drumloop_symclip.wav
- Example5_drumloop_TCM5000_flambeat.wav
- Example6_dryspeech_480L.wav

Raw_Audio_Files

- 480L_Large_Hall.wav
- anechoic_voice.wav
- drumloop.wav
- dryspeech.wav
- la.wav
- space_echo_35.wav
- space_echo_114.wav
- TCM5000_flambeat.wav
- Xylophone.wav

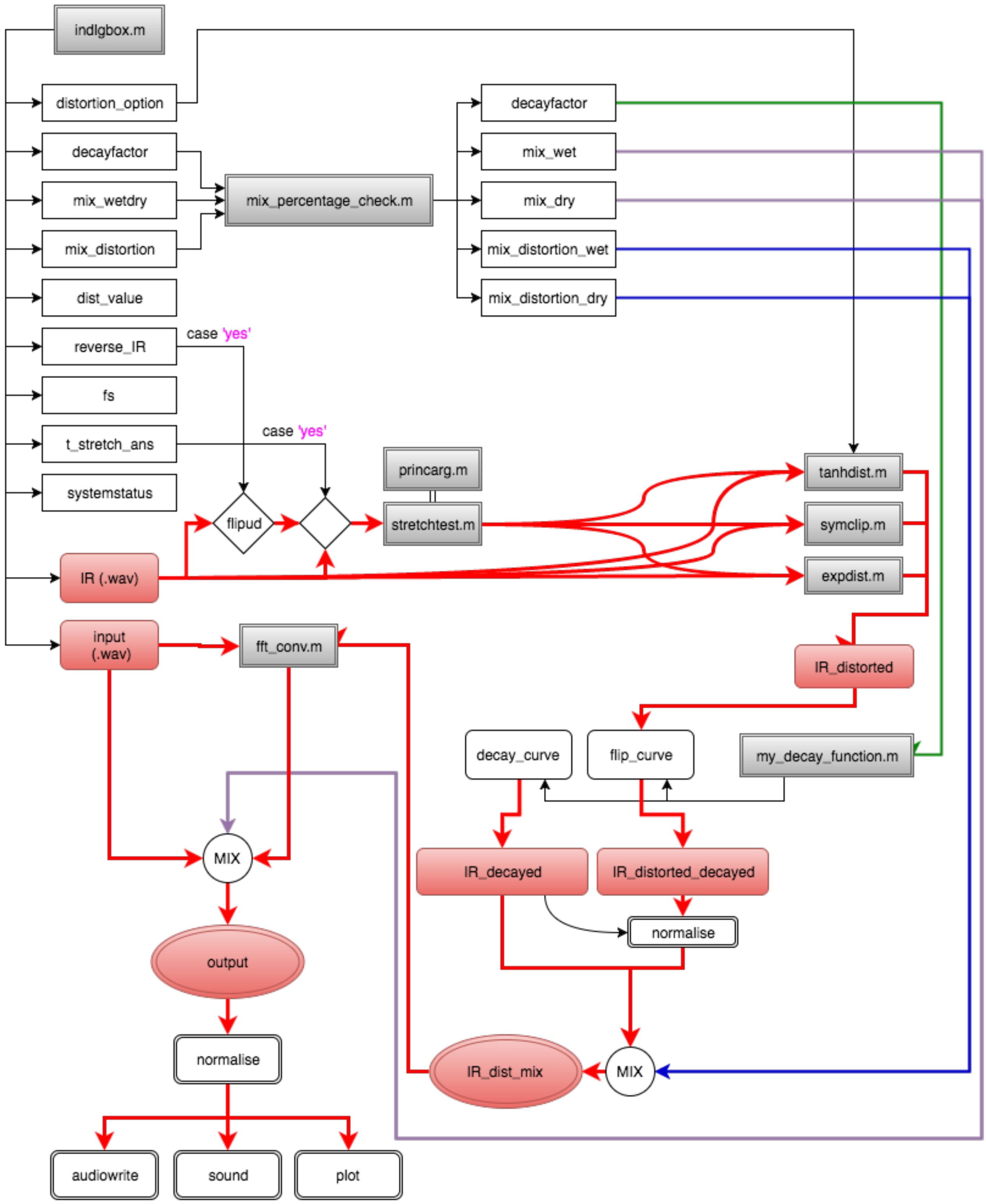
Appendix B:

Signal Flow diagram of “System A” in mono/mono mode.



Appendix C:

Signal Flow diagram of "System B" in mono/mono mode.



Appendix D:

Performance Assessment & Sound Examples (all output files have been normalised):

Example 1: Default settings, System A.

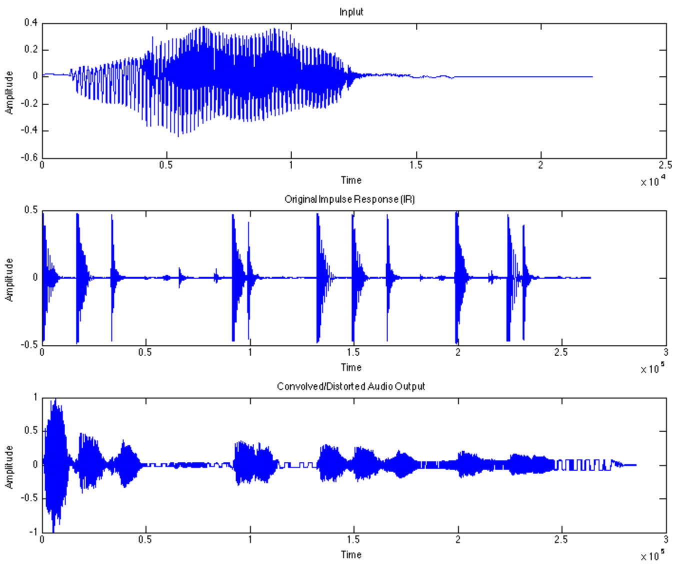
Filename: **Example1_la_drumloopA.wav**

Input: **la.wav**

Impulse Response: **drumloop.wav**

Elapsed Time: **0.138322 seconds**

Distortion Type:	**1**
Decayfactor:	**1**
Input/IR Mix	**50**
Distortion Wet/Dry Mix:	**50**
Distortion Value:	**1000**
Reverse Impulse Response:	**'No'**
Sampling Rate:	**44100**
Time Stretch:	**0**
System Type:	**'A'**



Example 2: Default Settings, System B.

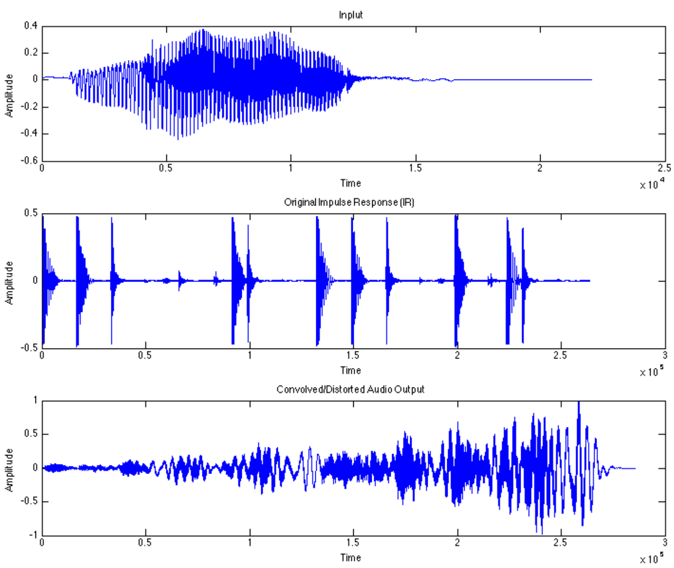
Filename: **Example2_la_drumloopB.wav**

Input: **la.wav**

Impulse Response: **drumloop.wav**

Elapsed Time: **0.443922 seconds**

Distortion Type:	**1**
Decayfactor:	**1**
Input/IR Mix	**50**
Distortion Wet/Dry Mix:	**50**
Distortion Value:	**1000**
Reverse Impulse Response:	**'No'**
Sampling Rate:	**44100**
Time Stretch:	**0**
System Type:	**'B'**



Example 3: Default settings, System A (expdist.m distortion)

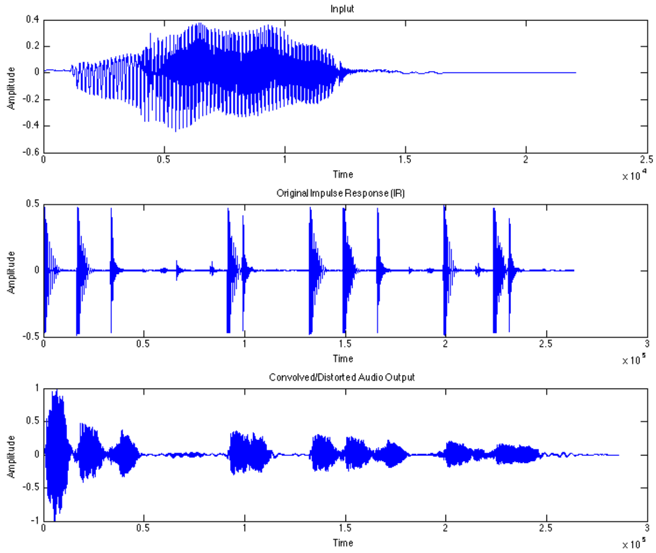
Filename: **Example3_la_drumloop_expdist.wav**

Input: **la.wav**

Impulse Response: **drumloop.wav**

Elapsed Time: **0.156312 seconds**.

Distortion Type:	**2**
Decayfactor:	**1**
Input/IR Mix	**50**
Distortion Wet/Dry Mix:	**50**
Distortion Value:	**1000**
Reverse Impulse Response:	**'No'**
Sampling Rate:	**44100**
Time Stretch:	**0**
System Type:	**'A'**



Example 4: Default settings, System A (symclip.m distortion)

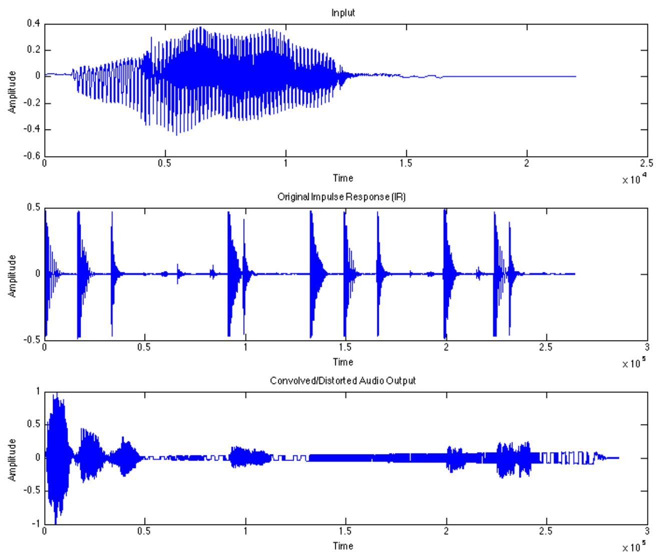
Filename: **Example4_la_drumloop_symclip.wav**

Input: **la.wav**

Impulse Response: **drumloop.wav**

Elapsed Time: **0.147445 seconds.**

Distortion Type:	**3**
Decayfactor:	**1**
Input/IR Mix	**50**
Distortion Wet/Dry Mix:	**50**
Distortion Value:	**1000**
Reverse Impulse Response:	**'No'**
Sampling Rate:	**44100**
Time Stretch:	**0**
System Type:	**'A'**



Example 5: Symmetrical Clipping, "bit-crush" distortion effect. Higher mix percentage.

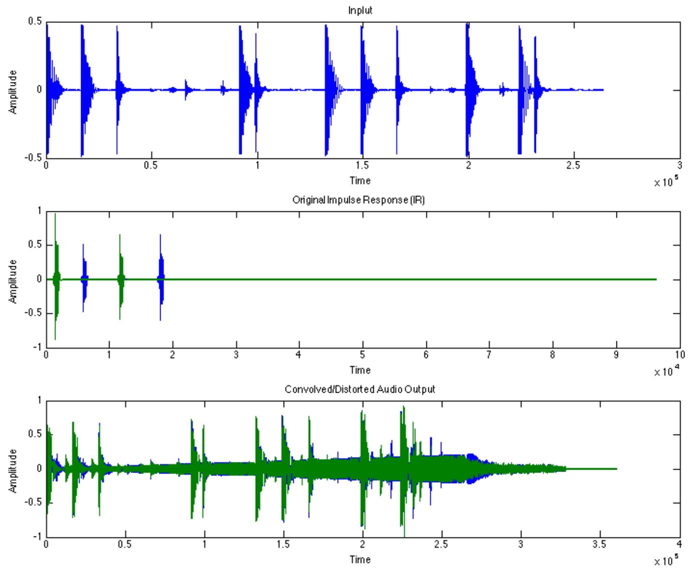
Filename: **Example5_drumloop_TCM5000_flambeat.wav**

Input: **drumloop.wav**

Impulse Response: **TCM500_flambeat.wav**

Elapsed Time: **0.522572 seconds**.

Distortion Type:	**3**
Decayfactor:	**1**
Input/IR Mix	**65**
Distortion Wet/Dry Mix:	**80**
Distortion Value:	**500**
Reverse Impulse Response:	**'No'**
Sampling Rate:	**44100**
Time Stretch:	**0**
System Type:	**'A'**



Example 6: Extreme system usage; time stretching & impulse response reversal (watch your volume!).

Filename: **Example6_dryspeech_480L.wav**

Input: **dryspeech.wav**

Impulse Response: **480L_Large_Hall.wav**

Elapsed Time: **24.244790 seconds.**

Distortion Type:	**2**
Decayfactor:	**1**
Input/IR Mix	**100**
Distortion Wet/Dry Mix:	**80**
Distortion Value:	**2000**
Reverse Impulse Response:	**'Yes'**
Sampling Rate:	**44100**
Time Stretch:	**1** (x6)
System Type:	**'A'**