BELMONT UNIVERSITY, CURB COLLEGE OF ENTERTAINMENT AND MUSIC BUSINESS

AET 5420 Homework 1

Name : ______________________________

Due: February 1st, 2021

1 MID-SIDE DISTORTION

Mid-side processing is based on the decomposition of a conventional stereo signal into separate *mid* and *side* signals. After the signals have been encoded, engineers can process the *mid* signal independently of the *side* signal by using audio effects (EQ, Compression, Distortion, etc). Then a decoding process is performed to recover new *left* and *right* signals.

As one example, an engineer could add soft clipping to the *mid* channel and hard clipping to the *sides* channel.

1.1 PROBLEM

Create and save a **script** (m-file) in MATLAB that performs the following steps:

- Name the script: **msDistortion.m**
- Import the audio file: stereoDrums.wav
  - Next, separate the stereo signal into a mono, 'left' signal and a mono, 'right' signal
- Perform Mid-side Encoding
  - Refer to class lecture notes on the process
- Perform hard clipping distortion on the *sides* channel
  - Use the function created during class
- Perform soft clipping on the *mid* channel

- Use a function of one of the options covered in class
- Perform Mid-side Decoding
- Create a single variable to store the stereo audio signal
- Write this stereo audio signal to a new file - **distDrums.wav**

Remember to add comments to your code to explain what each command is accomplishing. For this problem, you will submit the script file - **msDistortion.m**, the distortion functions, and the sound file - **distDrums.wav**.

2 TRANSISTOR CLIPPING

The transistor is a circuit component used in many analog audio systems. It is one of the components in a circuit that can distort an in-coming signal, if the signal's amplitude is great enough. In many ways the transistor behaves like a hard-clipper. When the amplitude of the in-coming signal is relatively low, the transistor behaves linearly over this operating region. When the signal becomes too high, the transister clips the signal off abruptly.

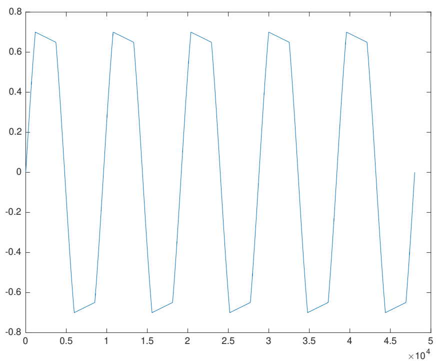

Figure 2.1: Wave of Signal with Transistor Clipping Effect

In many analog circuits, the transistor(s) show a unique behavior where the threshold of clipping is **not** constant. Instead, the threshold starts relatively high and then slowly decreases over time. This creates a gradual slope to the clipping effect, rather than a flat, constant level. This behavior is modeled in the AIR Distortion plug-in with the "Hard" setting. The waveform of a sine-wave signal after being distorted in this manner is shown in Figure 2.1.

To model this effect digitally, there are multiple parameters to control. First is the threshold, similar to the hard-clipping distortion. Second is the rate at which the threshold decreases as long as the signal stays in the distortion region. You can use a variable which decreases the threshold by a very small number whenever the amplitude of the input signal is greater than the threshold (or less than -threshold). When the amplitude of the signal is within the region of linear operation, the threshold level should reset to its original value.

2.1 PROBLEM

Two sound files are provided for you to use in this problem. One file is a bass DI track recorded without distortion. The second file is the distorted bass signal created by sending the DI track through an LA-3A with the Gain at 10 and Peak Reduction at 0. The goal of this problem is to create an effect which replicates this distortion.

Create a **function** and test **script** (m-files) in MATLAB that perform the following steps:

- Name the function: transistorClipping.m
- The function should have the following input/output variables:
  - in (input variable representing input signal)
  - thresh (input variable representing clipping amplitude)
  - out (output variable representing processed signal)
- Create a variable within the function to control the slope of the threshold
- Experiment with different values for this variable until the distortion resembles the example files
- Follow the recommend approach above for modifying the hard-clipping effect (loop with conditional statements) to achieve the transistor-type distortion
- Write your own test script for the function to demonstrate proper performance
  - Plot the waveform of a distorted sine wave to compare with Figure 2.1
  - Process the dry bass track and compare the result with the example sound file

Remember to add comments to your code to explain what each command is accomplishing. For this problem, you will submit the function file - **transistorClipping.m** and the **test script**.

3 PARALLEL DISTORTION

A common feature of some software distortion effects is the ability to blend between the "dry", unprocessed signal and the "wet", distorted signal. This feature can be created by using parallel distortion. The relative amplitude of each path is controlled by the gain scalars, g_1 and g_2. An example of the block diagram for this effect is shown in Fig. 3.1.

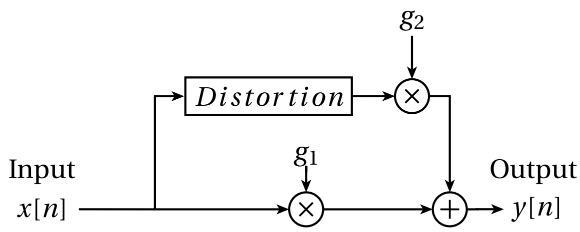

Figure 3.1: Block Diagram of Parallel Distortion

3.1 Waves Cobalt Saphira

The *Waves Cobalt Saphira* plug-in is a distortion effect with the ability to change the relative amount of even and odd harmonic distortion. The interface for the plug-in with independent controls for even harmonics and odd harmonics is shown in Fig. 3.2.

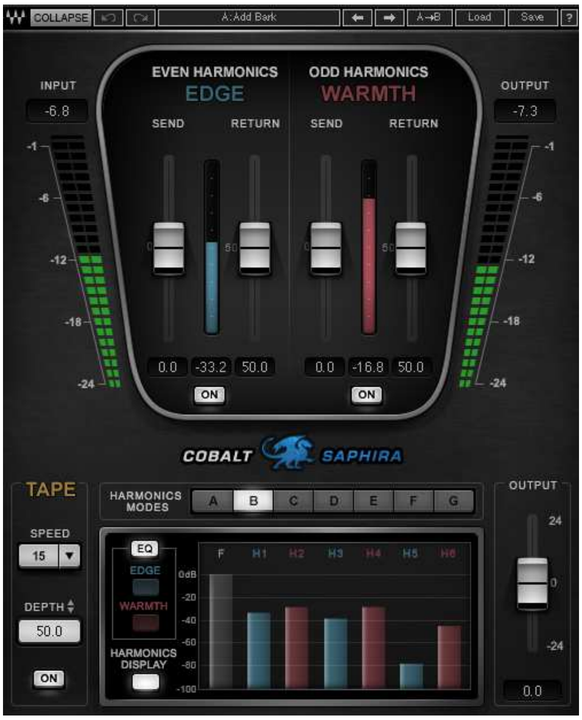

Figure 3.2: Waves Cobalt Saphira Interface

One approach to implementing this type of effect uses parallel distortion. As shown in class, full-wave rectification (i.e. absolute value function) creates even-order harmonic distortion. Additionally, arctangent soft clipping creates odd-order harmonic distortion. Each of these distortion effects can be performed in parallel for independent control of each type. A block diagram for this effect is shown in Fig. 3.3. The gain values (g_1 , g_2, and g_3) are used to change the relative amplitude of each parallel path.

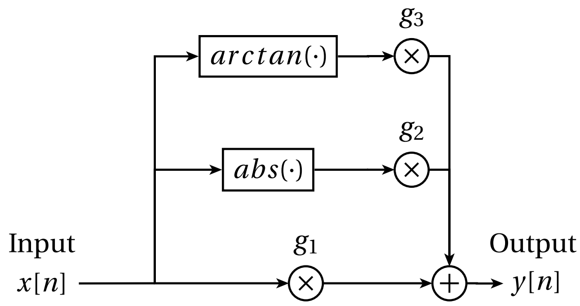

Figure 3.3: Block Diagram of Even and Odd Distortion

3.2 Problem

Create and save a **script** (m-file) in MATLAB that performs the following steps:

- Name the script: **parallelDist.m**
- Choose your own audio file to process
  - For the sake of processing time, use a clip shorter than 15 seconds
  - You will include this file in your homework submission
- Create 3 gain variables (g_1 , g_2, and g_3)
  - Experiment with different values for each variable between 0-1
- Perform the parallel distortion as described in Fig. 3.3.
  - Use full-wave rectification and arctangent clipping
  - Then scale the amplitude of the processed signals using g_1 , g_2, and g_3
- Combine all the different paths together to create a single output signal
- In addition to the sound file example, perform the following analyses
  - THD plot of entire effect for sine wave input
  - Characteristic curve plot of entire effect

Remember to add comments to your code to explain what each command is accomplishing. For this problem, you will submit the script file - **parallelDist.m** and the sound file you used.

4 Submission

To submit your homework, create a single zip file that contains your MATLAB functions, test scripts, and audio files. Name the zip file: xxxxx_AET5420_HW1.zip, where *xxxxx* is your last name. Email the zip file to: eric.tarr@belmont.edu before the start of class on February 1st.