Chen Ji
SUNetID:jjjjjc

Homework 4

Problem 1

The distortion plug-in has been designed with the following signal-flow block diagram:

Figure 1: Work-flow of Distortion processor

For the non-linearity part, we can do both Hard-clip and Soft-clip:

Hard Clip **Soft Clip**

$$\xi(x) = \begin{cases} 1, & x > 1, \\ x, & |x| \leq 1, \\ -1, & x < -1, \end{cases} \qquad \xi(x) = \frac{x}{1+|x|}.$$

Figure 2: Formula for hard-clip and soft-clip

(a) The spectrogram of input signal before distortion is shown as below:

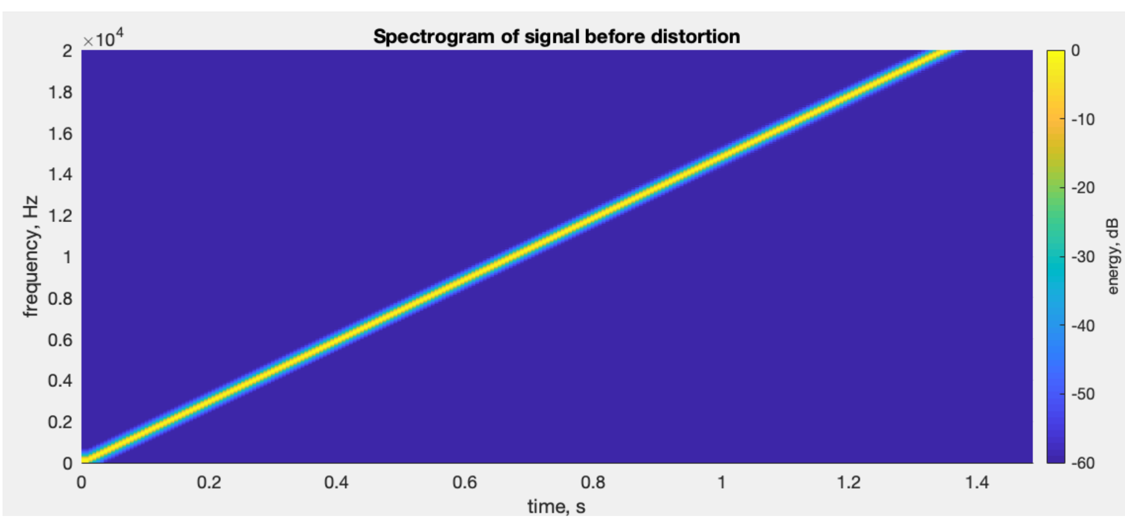

Figure 3: Spectrogram of input signal before distortion

After plugin the Distortion, we can get the spectrogram shown as below:

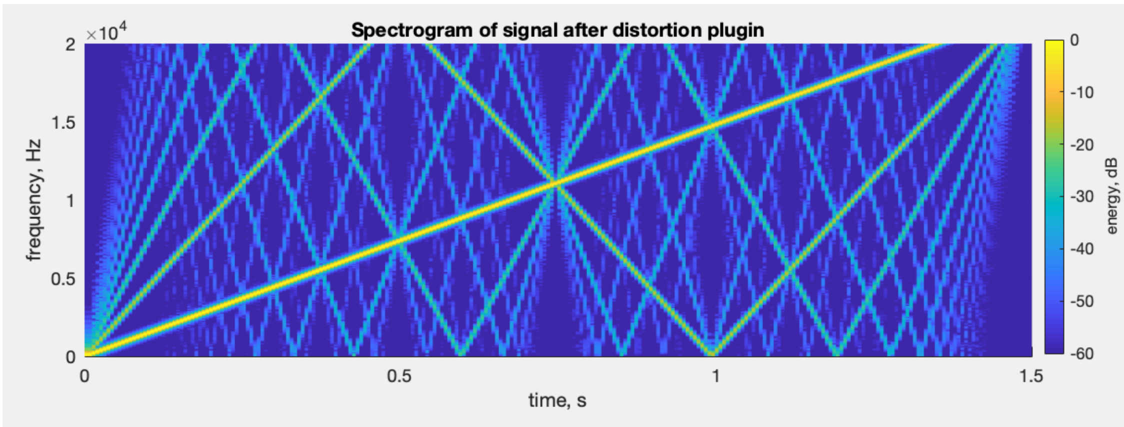

Figure 4: Spectrogram of input signal after distortion

After processing, aliasing is both audible in the sound and visible in the spectrogram (Figure 4). Without the plug-in, we only see and hear one ascending frequency sweep, but after processing, we can see and hear a series of other sweeps. The additional ascending sweeps are desirable harmonics, but the descending sweeps are due to aliasing.

(b) We need to design better anti-aliasing filter to reduce the aliasing.

```
% Filter parameters:
order = 6;
ripple = 2;
stopAtten = 100;
omega = 1/8 - 2/32;
% Design filter:
[b,a] = ellip(order, ripple, stopAtten, omega);
% Look at mag-spec
fvtool(b,a);
% Transform to bi-quads
s = tf2sos(b,a);
```

(c) By using the parametric section to compute the coefficients of the input and output filters from their center frequency, Q, and gain controls.

$$H(s) = \frac{(s/w_c)^2 + \frac{\gamma}{Q}(s/w_c) + 1}{(s/w_c)^2 + \frac{1}{Q}(s/w_c) + 1}$$

The code to achieve parametric section is shown below:

```
//TODO: design analog filter based on input gain, center
    frequency and Q
b0 = 1.0; b1 = 0.0; b2 = 0.0;
a0 = 1.0; a1 = 0.0; a2 = 0.0;
///////////////START/////////////////
center *= 2*pi;

// pre-warping
center = 2*fs*tan(center/(2*fs));

b0 = 1.0;
b1 = gain / ( center * qval );
b2 = 1.0 / ( center * center );


a0 = 1.0;
a1 = 1.0 / ( center * qval );
a2 = 1.0 / ( center * center );
```

Figure 5: S-plane

```
double T = 1/fs;
double Tsq = T*T;

// we need to normalize because the biquad struct assumes az0 = 1

az0 = ( a0*Tsq + 2*a1*T + 4*a2 );
az1 = ( 2*a0*Tsq - 8*a2 ) / az0;
az2 = ( a0*Tsq - 2*a1*T + 4*a2 ) / az0;

bz0 = ( b0*Tsq + 2*b1*T + 4*b2 ) / az0;
bz1 = ( 2*b0*Tsq - 8*b2 ) / az0;
bz2 = ( b0*Tsq - 2*b1*T + 4*b2 ) / az0;

az0 = 1;
```

Figure 6: z-plane

Problem 2

(a) The asymmetric saturator can be written as below:

$$\xi(x) = \frac{x + \gamma}{1 + |x + \gamma|}$$

```
% Constants:
freq = 440;
fs = 44100;
dur = 3;
T = 1/fs;
```

```
t = 0:T:dur-T;
N = length(t);
%gamma = 1;
% Signal:
x = cos(2*pi*440*t);
% Non-Linearity:
subplot(2,1,1);
x = (x + 0)./(1 + abs(x + 0));
% Create the window
w = blackman(512);
% Zero-pad the window
w = [w',zeros(1,N-512)];
% Plot spectrum:
X = fft(w.*x,16384);
plot(fftshift(20*log10(abs(X))));      % Spectral magnitude in dB
xlabel('frequency(Hz)');
ylabel('Amplitude');
title('Gamma = 0');
```

The two situations plot can be shown as below:

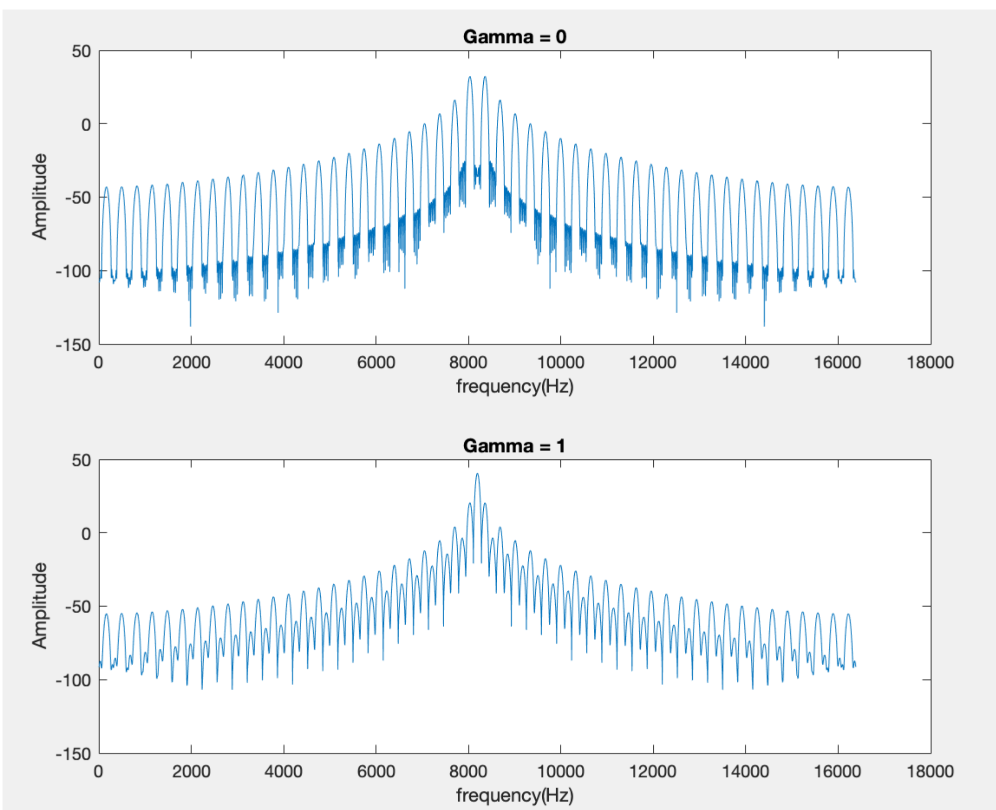

Figure 7: gamma=0 vs gamma=1

As we can see from the Figure 7, when gamma = 0, the spectrum is symmetry. It makes sense, when we add a DC offset (gamma = 1) to the input of the saturating non-linearity, it no longer maintains its odd symmetry.

(b) Design a second-order digital DC blocking filter by cascading two first-order high-pass filters of the form:

$$H(s) = \frac{s}{s+w_c}\frac{s}{s+w_c} = \frac{s^2}{s^2+2w_c s+w_c^2}$$

By using the bilinear Transform:

$$s = T/2\frac{1-z^{-1}}{1+z^{-1}}$$