Chen Ji
SUNetID:jjjjjc

Homework 2

Music424
April 19, 2019

Problem 1

(a) The gain could be calculated as below:

$$gain = l_{out} - l_{in} = (l_{in} - l_T)(1/\rho - 1)$$

Then the code can be modified as:

```
// above threshould
 if(threshould < level_estimate){
     float rho = comp_ratio;
     dbgainlevel = (log_level -dB(threshould))*((1/rho - 1));
 }
 //below threshould
 else{
     dbgainval = 0.0;//becomes 1 after dB2lin
 }
```

Figure 1: Modification code

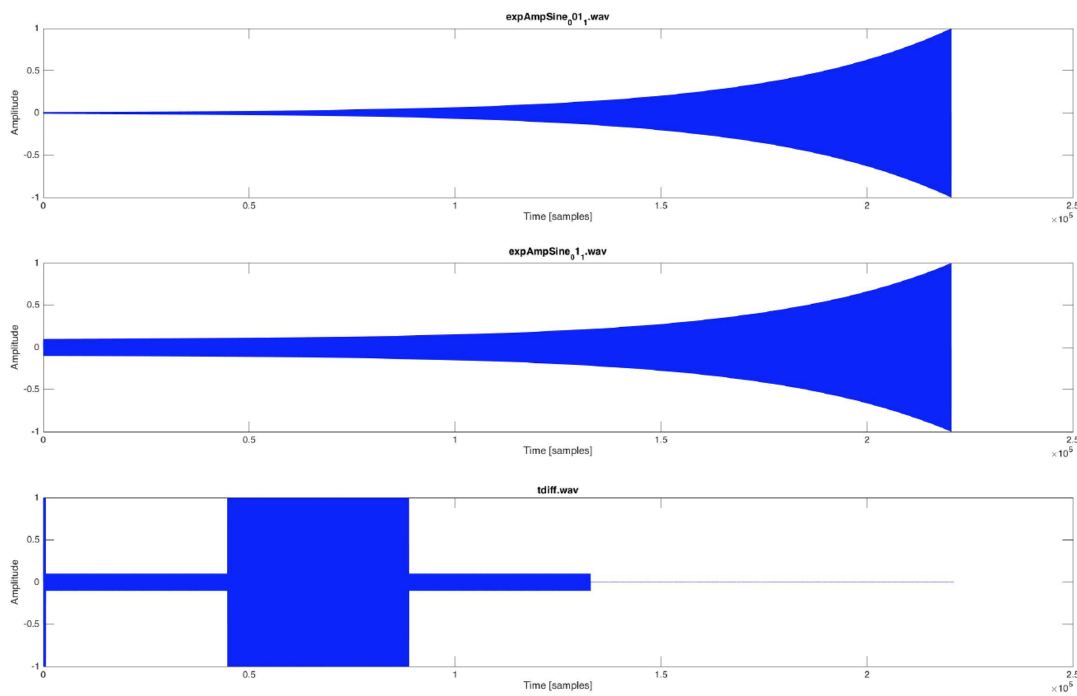

Figure 2: Test signals

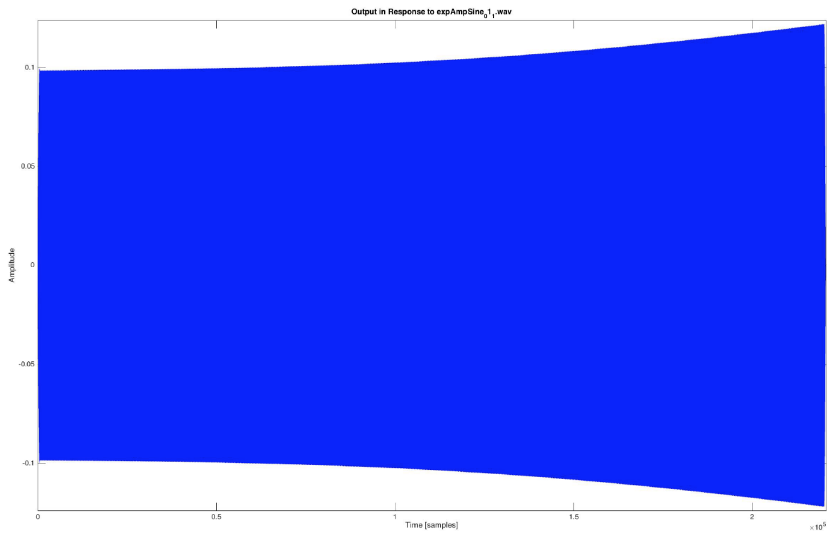

Figure 3: second test signal at ratio of 5, threshold of 0.1, a.t. 0.09 mS, r.t. 100 mS

Problem 2

(a) According to the formula given, the matlab code can be written as:

```
% Feed Forward Compressor Model:
R0 = 100e3; %100K
signalLevel = 1:0.01:100e3;
Rp = R0 .* ( signalLevel ) .^ (-0.75);
filterGain = 2 * Rp ./ ( R0 + Rp );
filterOutput = signalLevel .* filterGain;
plot(signalLevel , filterOutput);
xlim([0 100])
ylim([0 40])
xlabel('Input Level');
ylabel('Output Level');
title('Analog Compressor Models');
% Feedback Compressor Model:
hold on;
filterOutput2 = ((signalLevel + 1^(-0.75)* signalLevel .^ (1.75)) ./ 2);
plot(filterOutput2,signalLevel);
xlim([0 100])
ylim([0 40])
legend('Feed Fwd', 'Feedback');
```

Figure 4: Analogure compressor MATLAB code

(b) The plot can be shown as below:

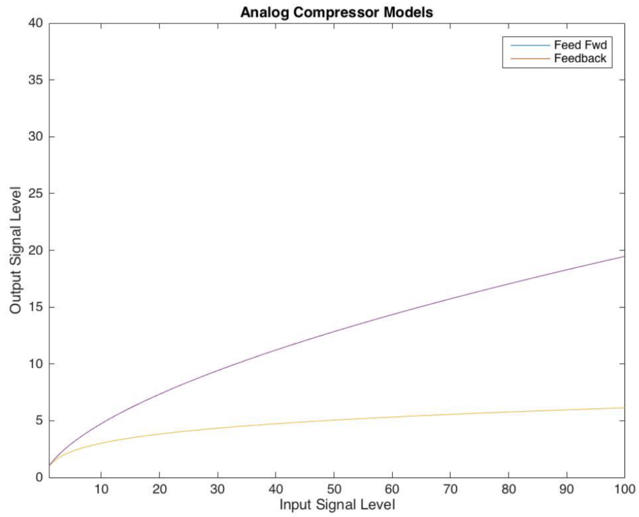

Figure 5: Analogure compressor: feedworward vs forward

Problem 3

(a) The code for "Release for treshould" can be written as:

```
void process (float input, float& output) {
    if ( fabs( input ) > levelEstimate ){
        //attack-state update equation
        levelEstimate += b0_a * ( fabs( input ) - levelEstimate );
    }
    else{
        //Release-state update equation(s)
        //Release to threshould
        levelEstimate += b0_r * ( threshould - levelEstimate );
    }
output = levelEstimate;
```

Figure 6: "Release for treshould" peak detector Code

Problem 4

(a) I created the test signal to test where is the treshould, so i create a step signal, where the step is 0.5, so that we can know where is the position it changes.
threshold is very close to 0.1, or -20 dB. It is certainly between 0.1 and 0.01.

(b) Figure8

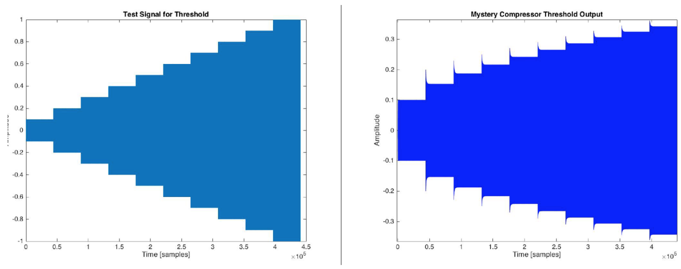

Figure 7: test

```
//feed-forward
if(input>levelEstimate){
    levelEstimate += b0_a*(input-levelEstimate);
}else{
    levelEstimate += b0_r*(input-levelEstimate);
}
//feed-back
if(input>levelEstimate){
    levelEstimate += b0_a*(output-levelEstimate);
}else{
    levelEstimate += b0_r*(output-levelEstimate);
}
```

Figure 8: feedforward or feedback topologies

(c) RMS or peak detection?
I created test signals in MATLAB: one full-scale sinusoid, and one full-scale square wave. If the compressor approaches a value of $1/\sqrt(2)$ for the sine wave, and $1/2$ for the square wave, then we know that this is an RMS detector.

(d) Approximate attack estimate: 5 mS (five cycles of a 1kHz test signal).
Approximate release estimate: 500 mS (25,000 samples / 44100 samples).

(e) Figure 9

```
void process(input, & output){
    if(fabs(input)>levelEstimate){
        levelEstimate += b0_a*(fabs(input)-levelEstimate);
    }else{
        levelEstimate += b0_r*(fabs(input)-levelEstimate);
    }
    output = levelEstimate;
}
```

Figure 9: signal estimator