

Digital Signal Processing Lab

Demo 3 - Exercise 4 (Pyaudio, clipping)

Saad Zubairi
shz2020

September 17th, 2025

Solution

The solution for this was rather simple of course. We can simply calculate the maximum possible value of gain for the specific frame being processed, check it against the user set gain, and if the set gain exceeds the said calculated maximum gain, it should revert to the maxGain value that we calculated.

```
1
2     maxGain = ((2**15)/abs(y0))
3     if(gain > maxGain):
4         print(y0, '/tmax gain:', maxGain)
5         gain = maxGain.__floor__() - 1
6 .
```

Snippet 1: Snippet of the if function to enable clipping

A small point to note, when calculating the maxGain value, we must use the absolute value of y_0 as certain times, low values may lead to integer overflows resulting in unusually high gain values.

Following is the listing for the entire Solution.py file:

```
1     # filter_16.py
2 #
3 # Implement the second-order recursive difference equation
4 #  $y(n) = x(n) - a_1 y(n-1) - a_2 y(n-2)$ 
5 #
6 # 16 bit/sample
7
8 from math import cos, pi
9 import pyaudio
10 import struct
11
12
13 # Fs : Sampling frequency (samples/second)
14 Fs = 8000
15 # Also try other values of 'Fs'. What happens? Why?
16
17 T = 1          # T : Duration of audio to play (seconds)
18 N = T*Fs       # N : Number of samples to play
19
20 # Difference equation coefficients
21 a1 = -1.9
22 a2 = 0.998
23
24 # Initialization
25 y1 = 0.0
26 y2 = 0.0
27 gain = 10178.0
28 # Also try other values of 'gain'. What is the effect?
29 # gain = 20000.0
30
31 # Create an audio object and open an audio stream for output
32 p = pyaudio.PyAudio()
33 stream = p.open(format = pyaudio.paInt16,
34                 channels = 1,
35                 rate = Fs,
36                 input = False,
37                 output = True)
38
39 # paInt16 is 16 bits/sample
```

```

40
41 # Run difference equation
42 for n in range(0, N):
43
44     # Use impulse as input signal
45     if n == 0:
46         x0 = 1.0
47     else:
48         x0 = 0.0
49
50     # Difference equation
51     y0 = x0 - a1 * y1 - a2 * y2
52
53     # Delays
54     y2 = y1
55     y1 = y0
56
57     maxGain = ((2**15)/abs(y0))
58     if(gain > maxGain):
59         print(y0, '/tmax gain:', maxGain)
60         gain = maxGain.__floor__() - 1
61
62     # Output
63     output_value = gain * y0
64     output_string = struct.pack('h', int(output_value))    # 'h' for 16 bits
65     stream.write(output_string)
66
67 print("* Finished *")
68
69 stream.stop_stream()
70 stream.close()
71 p.terminate()

```

Snippet 2: Snippet of the if function to enable clipping