

Digital Signal Processing Lab

Demo 3 - Exercise 5 (Pyaudio, diff eq)

Saad Zubairi
shz2020

September 17th, 2025

Solution

For $h[n] = r^n \cos(\omega n) z^{-1} u(n)$, using formula:

$$H(z) = \frac{1 - r \cos \omega z^{-1}}{1 - 2r \cos \omega z^{-1} + r^2 z^{-2}}$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Implementing this in code, here is what we can add and change:

- Setting the denominator from some arbitrary values of ω and r .

```
1 r = 0.999
2 omega = 2*pi*400/Fs      # ~400 Hz tone
3 a1 = -2 * r * cos(omega)
4 a2 = r * r
```

- We can then add a numerator $B(z)$ with the values for b_0, b_1 , and b_2

```
1 b0 = 1.0
2 b1 = a1 / 2.0      # = -r cos(omega)
3 b2 = 0.0
```

- We of course also update the equation:

```
1 y0 = (b0*x0 + b1*x1 + b2*x2) - a1*y1 - a2*y2
```

- And the variables that need to be updated as well:

```
1 x2, x1 = x1, x0
2 y2, y1 = y1, y0
```

- Lastly, to keep the gain in check, we can reuse the same if statement used in the previous question

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

The final code is given as follows:

```
1 from math import cos, pi
2 import pyaudio, struct
3
4 Fs = 8000
5 T = 1
6 N = T * Fs
7
8 r = 0.999
9 omega = 2*pi*400/Fs      # ~400 Hz tone
10 a1 = -2 * r * cos(omega)
11 a2 = r * r
12
13 # h[n] = r^n cos(omega n) u[n]
14 b0 = 1.0
15 b1 = a1 / 2.0          # = -r cos(omega)
16 b2 = 0.0
17
18 # maintaing states
19 x1 = x2 = 0.0
20 y1 = y2 = 0.0
21
22 gain = 10000.0
23
24 p = pyaudio.PyAudio()
25 stream = p.open(format=pyaudio.paInt16, channels=1, rate=Fs, input=False, output=
    True)
26
27 for n in range(N):
28     # Use impulse as input signal
29     if n == 0:
30         x0 = 1.0
31     else:
32         x0 = 0.0
33
34
35     y0 = (b0*x0 + b1*x1 + b2*x2) - a1*y1 - a2*y2
36
37     # Delays
38     x2, x1 = x1, x0
39     y2, y1 = y1, y0
40
41     maxGain = ((2**15)/abs(y0)).__floor__() - 1
42     if(gain > maxGain):
43         print(y0, '/tmax gain:', maxGain)
44         gain = maxGain
45
46     output_value = gain * y0
47     output_string = struct.pack('h', int(output_value))
48     stream.write(output_string)
49
50 print("* Finished *")
51 stream.stop_stream(); stream.close(); p.terminate()
```