Paul Vincent S. Nonat 2018-21366

Table of Contents

EE 274 Digital Signal Processing 1 Lab Activity 1

F. Audio File Formats

```
%The following exercise will demonstrate the effects of using
 quantization
%and sampling on audio signals.
%
% # Load music1.flac provided in UVLe folder. Can also be downloaded
 here
% # Using the MATLAB functions you have created in parts A-E,
 quantize, up/
%downsample using the following configurations:

[y,fs] = audioread ('Sample_BeeMoved_96kHz24bit.flac')
info =audioinfo('Sample_BeeMoved_96kHz24bit.flac')
t = 0:seconds(1/fs):seconds(info.Duration);
t = t(1:end-1);
figure
plot(t,y)
title('Original Audio')
xlabel('Time')
ylabel('Audio Signal')
Px=sum((y).^2)
```

X1 R=10, B=16, downsampled by 2

```
R=10
B=16
target_sampling1 =48000
y1= adc_uni(y,R,B) %quantize
x1=downsample(y1,fs/target_sampling1) % downsample

t1 = 0:seconds(1/(target_sampling1)):seconds(info.Duration);
t1 = t1(1:end-1);
```

```
plot(t1,x1) %generate plot
title('16-Bit ADC, Sampling Rate-48000')
xlabel('Time')
```

X2 R=10, B=8, downsampled by 2

```
R=10
B=8
target_sampling2 =48000
y2= adc_uni(y,R,B) %quantize
x2=downsample(y2,fs/target_sampling2) % downsample

t2 = 0:seconds(1/(target_sampling2)):seconds(info.Duration);
t2 = t2(1:end-1);

plot(t2,x2) %generate plot
title('8-Bit ADC, Sampling Rate-48000')
xlabel('Time')
```

X3 R=10, B=4, downsampled by 2

```
R=10
B=4
target_sampling3 =48000
y3= adc_uni(y,R,B) %quantize
x3=downsample(y3,fs/target_sampling3) % downsample

t3 = 0:seconds(1/(target_sampling3)):seconds(info.Duration);
t3 = t3(1:end-1);

plot(t3,x3) %generate plot
title('4-Bit ADC, Sampling Rate-48000')
xlabel('Time')
```

X4 R=10, B=16, downsampled by 6

```
R=10
B=16
target_sampling4 =16000
y4= adc_uni(y,R,B) %quantize
x4=downsample(y4,fs/target_sampling4) % downsample

t4 = 0:seconds(1/(target_sampling4)):seconds(info.Duration);
t4 = t4(1:end-1);

plot(t4,x4) %generate plot
title('16-Bit ADC, Sampling Rate-16000')
xlabel('Time')
```

X5 R=10, B=8, downsampled by 6

```
R=10
```

```
B=8
target_sampling5 =16000
y5= adc_uni(y,R,B) %quantize
x5=downsample(y5,fs/target_sampling5) % downsample

t5 = 0:seconds(1/(target_sampling5)):seconds(info.Duration);
t5 = t5(1:end-1);

plot(t5,x5) %generate plot
title('8-Bit ADC, Sampling Rate-16000')
xlabel('Time')
```

X6 R=10, B=4, downsampled by 6

```
R=10
B=4
target_sampling6 =16000
y6= adc_uni(y,R,B) %quantize
x6=downsample(y6,fs/target_sampling6) % downsample

t6 = 0:seconds(1/(target_sampling6)):seconds(info.Duration);
t6 = t6(1:end-1);

plot(t6,x6) %generate plot
title('4-Bit ADC, Sampling Rate-16000')
xlabel('Time')
```

X7 R=10, B=16, downsampled by 12

```
R=10
B=16
target_sampling7 =8000
y7= adc_uni(y,R,B) %quantize
x7=downsample(y7,fs/target_sampling7) % downsample

t7 = 0:seconds(1/(target_sampling7)):seconds(info.Duration);
t7 = t7(1:end-1);

plot(t7,x7) %generate plot
title('16-Bit ADC, Sampling Rate-8000')
xlabel('Time')
```

X8 R=10, B=8, downsampled by 12

```
R=10
B=8
target_sampling8 =8000
y8= adc_uni(y,R,B) %quantize
x8=downsample(y8,fs/target_sampling8) % downsample

t8 = 0:seconds(1/(target_sampling8)):seconds(info.Duration);
t8 = t8(1:end-1);
```

```
plot(t8,x8) %generate plot
title('8-Bit ADC, Sampling Rate-8000')
xlabel('Time')
```

X9 R=10, B=4, downsampled by 12

```
R=10
B=4
target_sampling9 =8000
y9= adc_uni(y,R,B) %quantize
x9=downsample(y9,fs/target_sampling9) % downsample

t9 = 0:seconds(1/(target_sampling9)):seconds(info.Duration);
t9 = t9(1:end-1);

plot(t9,x9) %generate plot
title('8-Bit ADC, Sampling Rate-8000')
xlabel('Time')

Pq1=sum((y1-y).^2);
SQNR1=10*log10(Px/Pq1)
soundsc(x1,target_sampling1)

Pq2=sum((y2-y).^2);
SQNR2=10*log10(Px/Pq2)
soundsc(x2,target_sampling2)

Pq3=sum((y3-y).^2);
SQNR3=10*log10(Px/Pq3)
soundsc(x3,target_sampling3)

Pq4=sum((y4-y).^2);
SQNR4=10*log10(Px/Pq4)
soundsc(x4,target_sampling4)

Pq5=sum((y5-y).^2);
SQNR5=10*log10(Px/Pq5)
soundsc(x5,target_sampling5)

Pq6=sum((y6-y).^2);
SQNR6=10*log10(Px/Pq6)
soundsc(x6,target_sampling6)

Pq7=sum((y7-y).^2);
SQNR7=10*log10(Px/Pq7)
soundsc(x7,target_sampling7)

Pq8=sum((y8-y).^2);
SQNR8=10*log10(Px/Pq8)
soundsc(x8,target_sampling8)

Pq9=sum((y9-y).^2);
SQNR9=10*log10(Px/Pq9)
soundsc(x9,target_sampling9)
```

```
%half bit resolution has more audible effect compared to using half
 the
%sampling rate. Because, the more samples that are taken, the more
 details
%about the audio is encoded. Hence, audio will become more audible.
%

function y = adc_uni(x, R, B)
level = [0:R/(2^B):R-R/(2^B)];
temp = [-Inf,(level(2:end)-R/(2^(B+1))),Inf];
y = zeros(1,length(x));
i=1
y=(x >= temp(i)).*(x < temp(i+1)).*level(i)
for i = 2:length(level)
    y = y + (x >= temp(i)).*(x < temp(i+1)).*level(i);
end
end
```

Published with MATLAB® R2020a