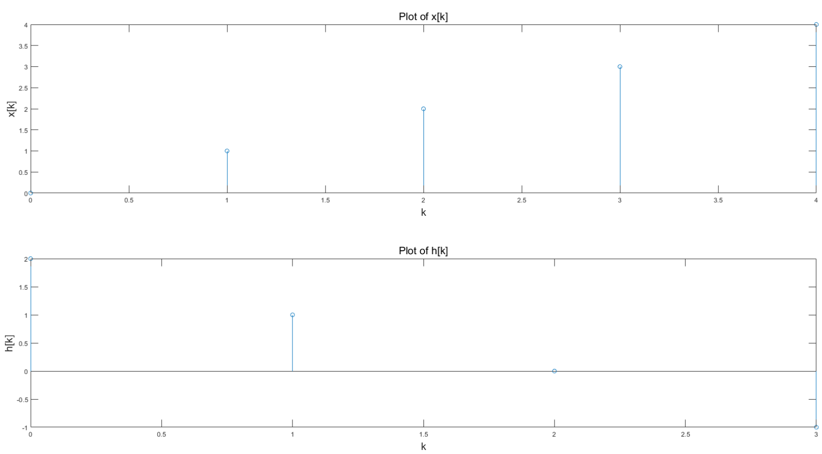
Lab 2

Chengxuan Li
1631060
Section 801
Oct 19th, 2021

1 Convolution

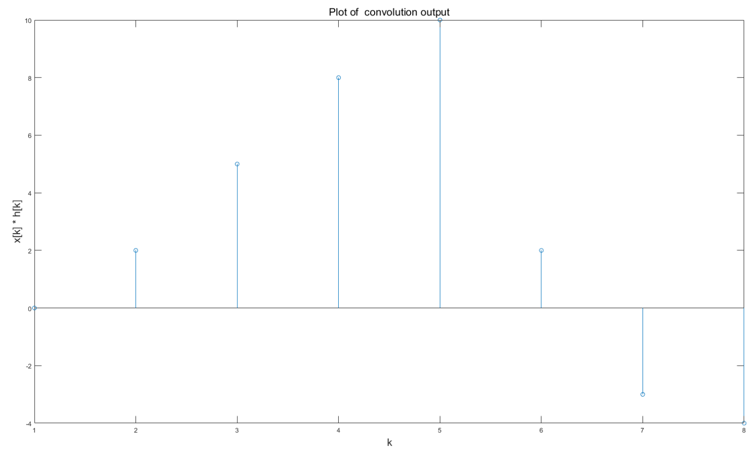
Q1(a) Plot of two given function and Matlab script.

```matlab
k = (0:1:4);
x = k;
h = zeros(1,4) + 2 - k(1:4);

tiledlayout(2,1);
ax1 = nexttile;
stem(ax1, k, x);
title('Plot of x[k]', 'Fontsize', 16);
xlabel('k', 'Fontsize', 16);
ylabel('x[k]', 'Fontsize', 16);

ax2 = nexttile;
stem(ax2, k(1:4), h);
title('Plot of h[k]', 'Fontsize', 16);
xlabel('k', 'Fontsize', 16);
ylabel('h[k]', 'Fontsize', 16);
```



Q1(b) Plot of the convolution results and MATLAB script.

```matlab
xh = conv(h, x);
tiledlayout(1,1);
stem(xh);
title('Plot of  convolution output', 'Fontsize', 16);
xlabel('k', 'Fontsize', 16);
ylabel('x[k] * h[k]', 'Fontsize', 16);
```



Q1(d) Verify the results of part c using sliding tap method in Excel.

n	k	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	y[n]
	x[k]	0	0	0	0	0	1	2	3	4	0	0	0	0	
	h[k]	0	0	0	0	2	1	0	-1	0	0	0	0	0	
0	phi[k]	0	-1	0	1	2	0	0	0	0	0	0	0	0	0
1	phi[k-1]	0	0	-1	0	1	2	0	0	0	0	0	0	0	2
2	phi[k-2]	0	0	0	-1	0	1	2	0	0	0	0	0	0	5
3	phi[k-3]	0	0	0	0	-1	0	1	2	0	0	0	0	0	8
4	phi[k-4]	0	0	0	0	0	-1	0	1	2	0	0	0	0	10
5	phi[k-5]	0	0	0	0	0	0	-1	0	1	2	0	0	0	2
6	phi[k-6]	0	0	0	0	0	0	0	-1	0	1	2	0	0	-3
7	phi[k-7]	0	0	0	0	0	0	0	0	-1	0	1	2	0	-4
8	phi[k-8]	0	0	0	0	0	0	0	0	0	-1	0	1	2	0

xh

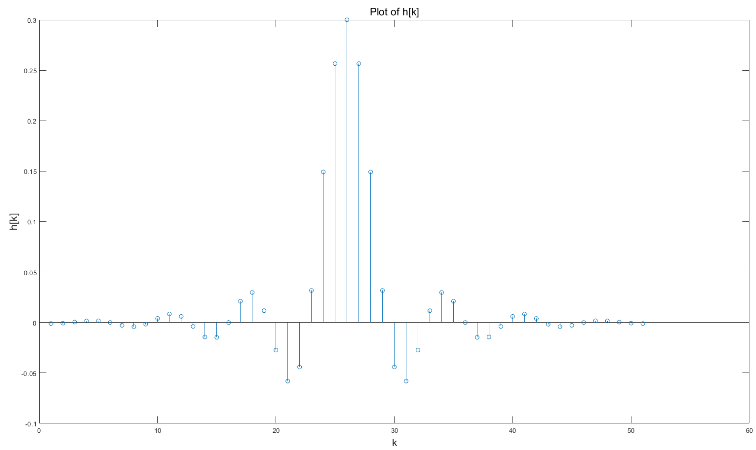
1x8 double

	1	2	3	4	5	6	7	8
1	0	2	5	8	10	2	-3	-4

2 Convolution

Q2(b) Plot of the h[k] and MTLAB script.

```
k = (0:1:50);
sincPart = 0.3 .* sinc(0.3 .* (k - 25));
cosPart = 0.54 - 0.46 .* cos(2 .* pi .* k ./ 50 );
h = sincPart .* cosPart;

stem(h);
title('Plot of h[k]', 'Fontsize', 16);
xlabel('k', 'Fontsize', 16);
ylabel('h[k]', 'Fontsize', 16);
```



Q2(c) MATLAB script for calculating the convolution results

```
[x3, Fs] = audioread("baila.wav");
x3h = conv(x3, h);
```

Q2(d) MATLAB script for store the convolution output as **baila_filtered.wav**

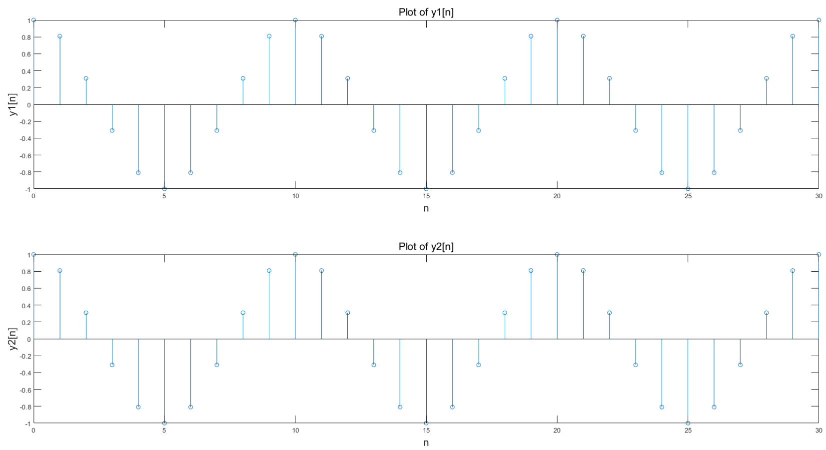
```
audiowrite("baila_filtered.wav", x3h, Fs);
```

Q2(e) The audio quality of the filtered version was distincly lower than the original version. Specially, the instrucment, which creates medium to relative high frequency (pitch) sound, became less clear and more dull in the audio due to sink function in $h[k]$.

3 Aliasing effects on 1D sinusoidal signals

Q3(a) Plot of y1[n] and y2[n], and MATLAB script. The sample version of both sinusoids were identical.

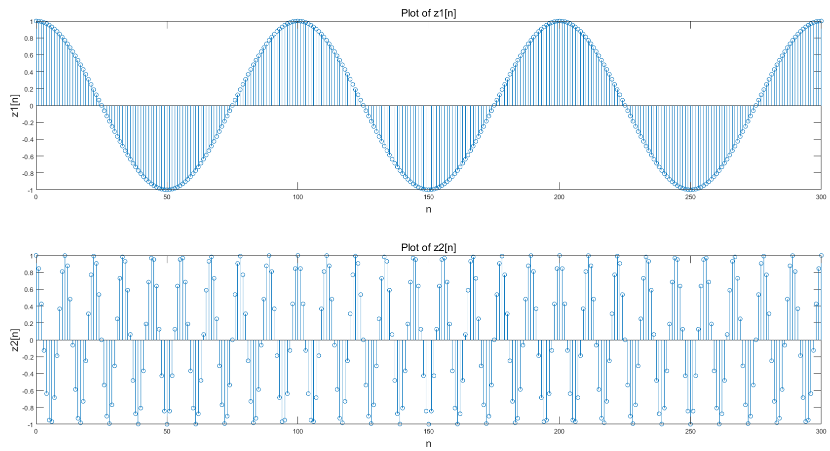
```
n1 = (0:1:30);
fs1 = 100;
y1 = cos(20 .* pi .* n1 ./ fs1);
y2 = cos(180 .* pi .* n1 ./ fs1);

tiledlayout(2,1);
ax1 = nexttile;
stem(ax1, n1, y1);
title('Plot of y1[n]', 'Fontsize', 16);
xlabel('n', 'Fontsize', 16);
ylabel('y1[n]', 'Fontsize', 16);
ax2 = nexttile;
stem(ax2, n1, y2);
title('Plot of y2[n]', 'Fontsize', 16);
xlabel('n', 'Fontsize', 16);
ylabel('y2[n]', 'Fontsize', 16);
```

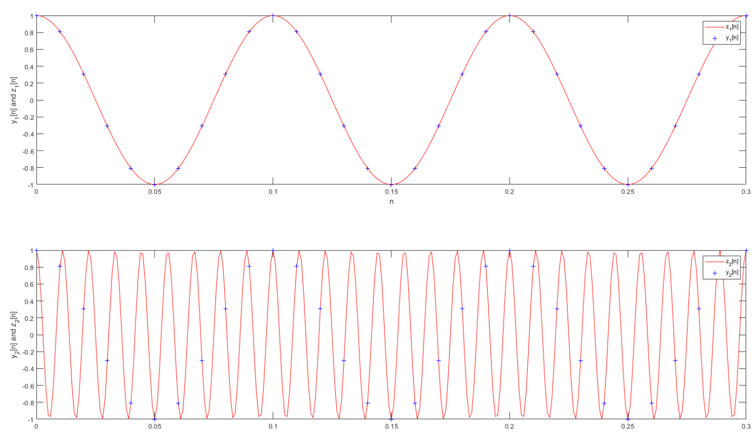


Q3(b) Plot of z1[n] and z2[n], and MATLAB script. The plot of z1[n] and z2[n] were clearly distinguishable after increasing the sampling frequency. By increasing the sample frequency, which were two times larger than the frequency of the x1[n] and x2[n],the number of samples increased. Thus, after converting these samples back to continuous time signals, which corresponds to z1[n] and z2[n] respectively, z1[n] and z2[n] were more accurate and close to what x1[n] and x2[n]. Compared to the sample frequency used for y1[n] and y2[n], y1[n] and and z1[n] were both accurate because the sample frequency was high enough to take more samples, which prevent that one sample and its aliasing become indistinguishable to each other. For y2[n], the sample frequency was smaller than the frequency of x2[n]. Thus, it took less samples, and it created inaccuracy reconstruction because of overlapping with one sample and its aliasing.

```
n2 = (0:1:300);
fs2 = 1000;
z1 = cos(20 .* pi .* n2 ./ fs2);
z2 = cos(180 .* pi .* n2 ./ fs2);

tiledlayout(2,1);
ax1 = nexttile;
stem(ax1, n2, z1);
title('Plot of z1[n]', 'Fontsize', 16);
xlabel('n', 'Fontsize', 16);
ylabel('z1[n]', 'Fontsize', 16);
ax2 = nexttile;
stem(ax2, n2, z2);
title('Plot of z2[n]', 'Fontsize', 16);
xlabel('n', 'Fontsize', 16);
ylabel('z2[n]', 'Fontsize', 16);

subplot(2,1,1);
plot(n2/fs2,z1,'r-', n1/fs1,y1,'b+');
xlabel('n'); ylabel('y_1[n] and z_1[n]');
```

```
legend('z_1[n]','y_1[n]');
subplot(2,1,2);
plot(n2/fs2,z2,'r-', n1/fs1,y2,'b+');
xlabel('n'); ylabel('y_2[n] and z_2[n]');
legend('z_2[n]','y_2[n]');
```

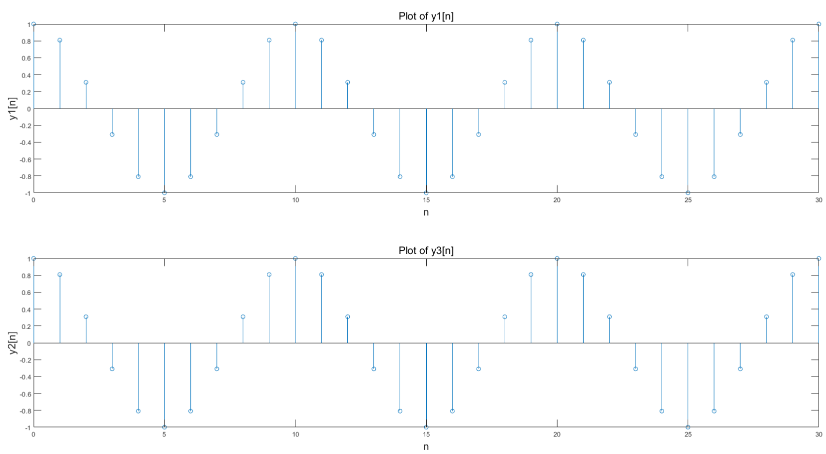




Q3(c) $x_3 = cos(220\pi t)$. Plot of y1[n] and y3[n], and Matlab script.

```
n1 = (0:1:30);
fs1 = 100;
y1 = cos(20 .* pi .* n1 ./ fs1);
y3 = cos(220 .* pi .* n1 ./ fs1);

tiledlayout(2,1);
ax1 = nexttile;
stem(ax1, n1, y1);
title('Plot of y1[n]', 'Fontsize', 16);
```

```
xlabel('n', 'Fontsize', 16);
ylabel('y1[n]', 'Fontsize', 16);
ax2 = nexttile;
stem(ax2, n1, y3);
title('Plot of y3[n]', 'Fontsize', 16);
xlabel('n', 'Fontsize', 16);
ylabel('y2[n]', 'Fontsize', 16);
```



4 The effects of aliasing in 2D images

Q4(a) Matlab script for reading ‘barbaraLarge.jpg’ into an array.

```
Img = imread('barbaraLarge.jpg');
```

Q4(b) Plot of the image and Matlab script

```
Img = imread('barbaraLarge.jpg');
figure,  imshow(Img); title('Original Barbara Image'),
    colorbar;
```

Q4(c) Matlab script. When the image was scaled by a factor of 0.9, there was no too much difference between using anti-aliasing and not using anti-aliasing. Also, the image still kept relatively enough details / information of the original image.

When the image was scaled by a factor of 0.7, the regions with a lot of fine lines or edges (pants, scarf, tablecloth and the chair) started to lose its details. Distortion also started to appear in that region. There was a little bit of difference between using anti-aliasing and not using anti-aliasing. Particularly, anti-aliasing was applied around woman's knee and lower part of the pants. The pixels around those region were become brighter after some black pixels were removed.

When the image was scaled by a factor of 0.5, the image lost a lots of detail and the high frequency region distore significantly. Anti-aliasing was applied mainly around the scarf, the pants, tablecloth and the chair. Those high frequency region looked closer to low frequency region, i.e, the pixels, with darker brightness compared to the surrounding pixels, becomre brighter.

```
% Running in a debugger, put breakpoint on imshow() function
    to stop and compare results
```

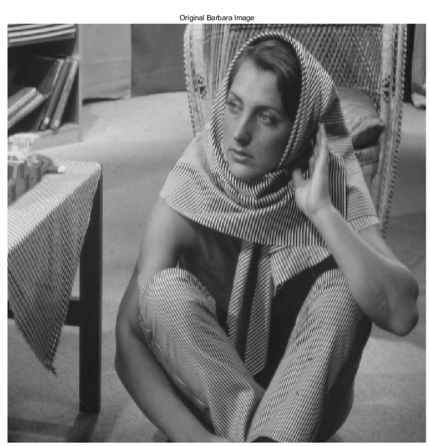
```
resize_factor = [0.9, 0.7, 0.5];
for i = resize_factor(1:3)
    resizeImg = imresize(Img, i, 'nearest', 'antialiasing', 0);
    resizeImgAA = imresize(Img, i, 'nearest', 'antialiasing', 1);
    str = sprintf('Barbara Image resized by %f of original size without anti-aliasing', i);
    strA = sprintf('Barbara Image resized by %f of original size with anti-aliasing', i);
    figure, imshow(resizeImg); title(str), colorbar;
    figure, imshow(resizeImgAA); title(strA), colorbar;
end
```

Q4(d) When the image was reszied by a factor of 0.9, the high frequency region, particularly, the pants started showing aliasing. Also, the shadows of the table legs also showed small degree of alias. As resizing factor decreased, aliasing at the high frequency region (pants, scarf and tablecloth) became more obvious.

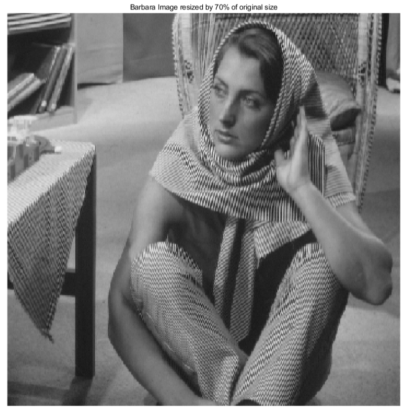
In 2D images, the number of pixels and the brightness pixels contain the information of the images. Thus, they can be called as the samples of the image. Reszing the image to smaller resolution was equivalently reducing the number of pixels from the original image, which also meant that the number of samples was reduced, and the sampling freuqency was also reduced. Consequently, this generated aliasing in the high frequency region due to which the sampling frequency was lower than the frequency in high frequency region, as well as the reduce in number of samples.

The regions marked by white shape are the regions that produce aliasing as the sample rate decrease

Q4(e) test3.m demonstration



Barbara Image resized by 70% of original size



Low pass filter applied before Resizing to 70% of original size