SIMULATION OF DISCRETE TIME SYSTEMS

LAB # 07

CSE402L Digital Signal Processing Lab

Submitted by: **Shah Raza**

Registration No: **18PWCSE1658**

Class Section: **B**

"On my honor, as a student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: ______________

Submitted to: **Engr. Faiz Ullah**

Tuesday, February 4th, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Lab Objectives:

Objectives of this lab are as follows:

- Learn about discrete time systems.

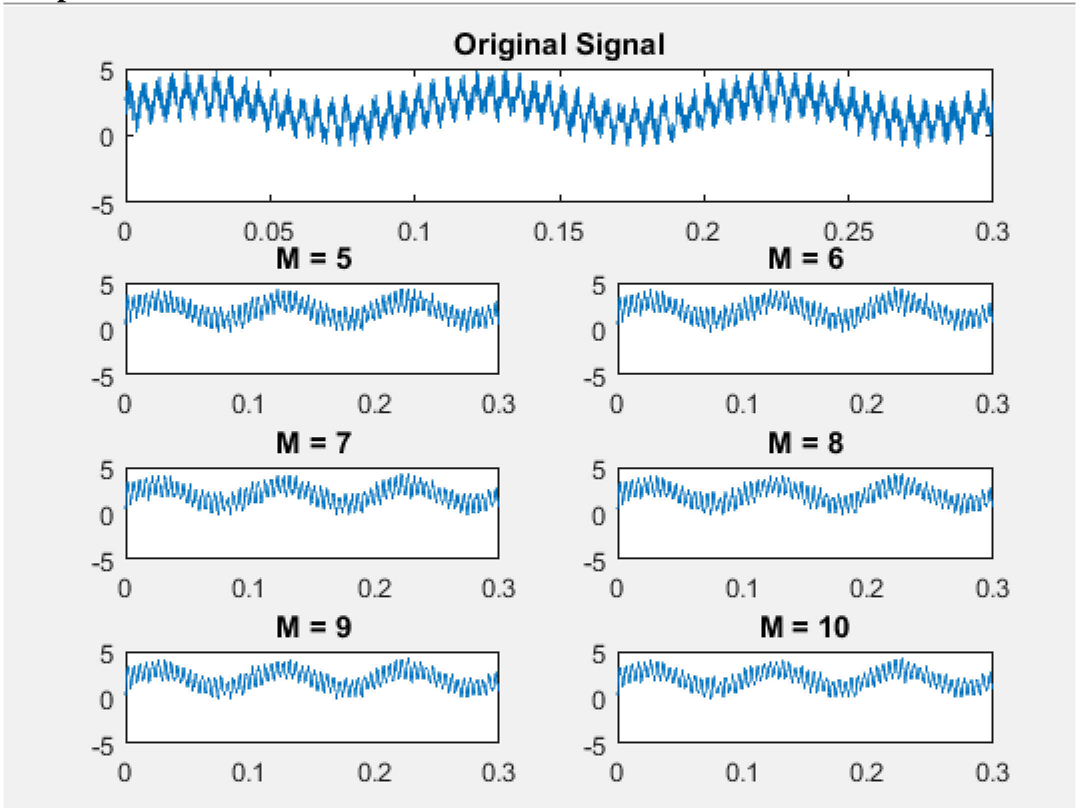
Practical:

1. Simulate a Moving Average System using in[n] equal to a sum of two sinusoid of frequency 10 Hz and 200 Hz, sampled at 8000 Hz, and vary M from 5 to 10. Plot out[n].

   Code:

```
clc
clear all
close all
f1 = 10;
f2 = 200;
fs = 8000;
t = 0:1/fs:0.3;
noise = 1+2*rand(size(t));
x1 = sin(2*pi*f1*t);
x2 = sin(2*pi*f2*t);
x = (x1+x2)+noise;
subplot(4,2,[1,2]);
plot(t,x);
title('Original Signal');
i=3;
for m=5:10
    b = (1/m)*ones(1,m);
    y = filter(b,1,x);
    subplot(4,2,i)
    plot(t,y);
    title(['M = ',num2str(m)]);
    i=i+1;
end
```

Output:



2. Find the impulse response of the system y[n] = 2.2403x[n] + 2.4908x[n-1] + 2.2403x[n-2] + 0.4y[n-1] - 0.75y[n-2] . Find whether the system given by the above equation is stable, time variant or linear using the first 50 samples of the output. Take x[n] the sum of two sinusoids of frequency 10 and 20 Hz. Plot y[n], both in MATLAB and VisualDSP++ 4.5.

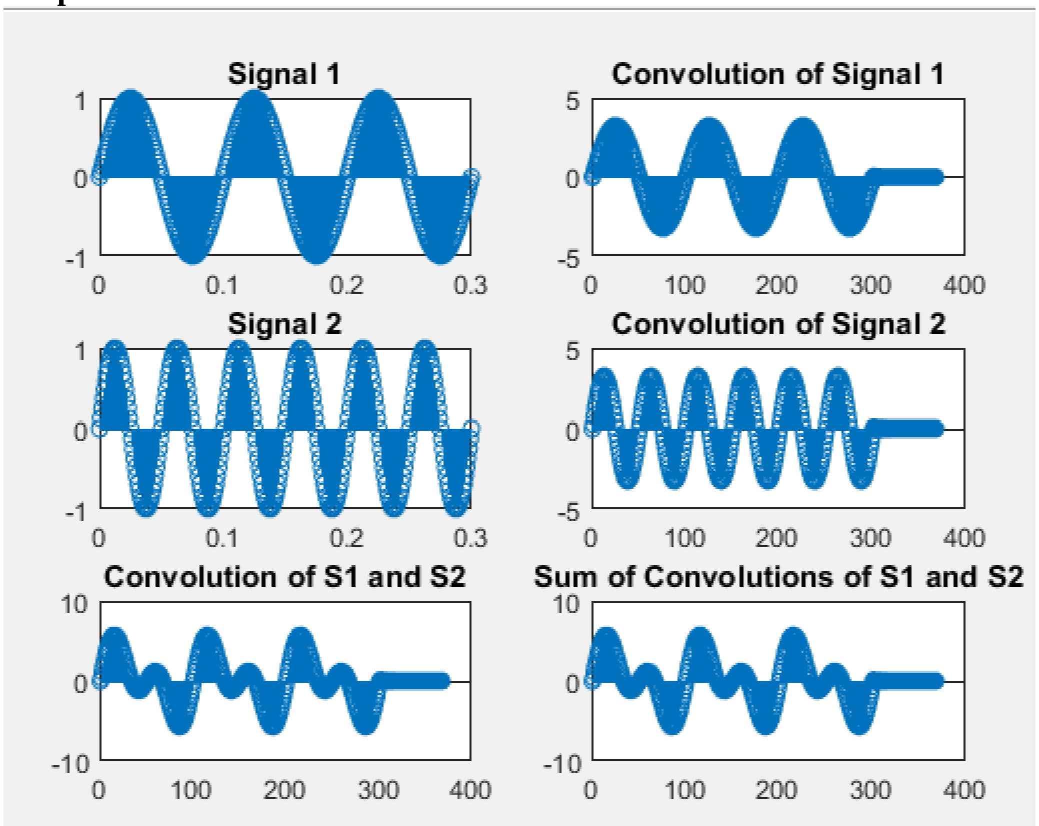
Code:

```
clc
clear all
close all
f1 = 10;
f2 = 20;
fs = 1000;
n = 0:1/fs:0.30;
bx = [2.2403 2.4908 2.2403];
by = [1 0.4 0.75];
h = impz(bx,by);
x1 = sin(2*pi*f1*n);
x2 = sin(2*pi*f2*n);
x = x1+x2;
y1 = conv(x,h);
subplot(3,2,1)
stem(n,x1);
title('Signal 1');
subplot(3,2,3)
stem(n,x2);
title('Signal 2');
```

```
subplot(3,2,5)
stem(y1);
title('Convolution of S1 and S2');

yx1 = conv(x1,h);
yx2 = conv(x2,h);
y2 = yx1+yx2;
subplot(3,2,2)
stem(yx1);
title('Convolution of Signal 1');
subplot(3,2,4)
stem(yx2);
title('Convolution of Signal 2');
subplot(3,2,6)
stem(y2);
title('Sum of Convolutions of S1 and S2');
```

Output:



Discussion:
Since the output is same so the system given by the above equation is linear.

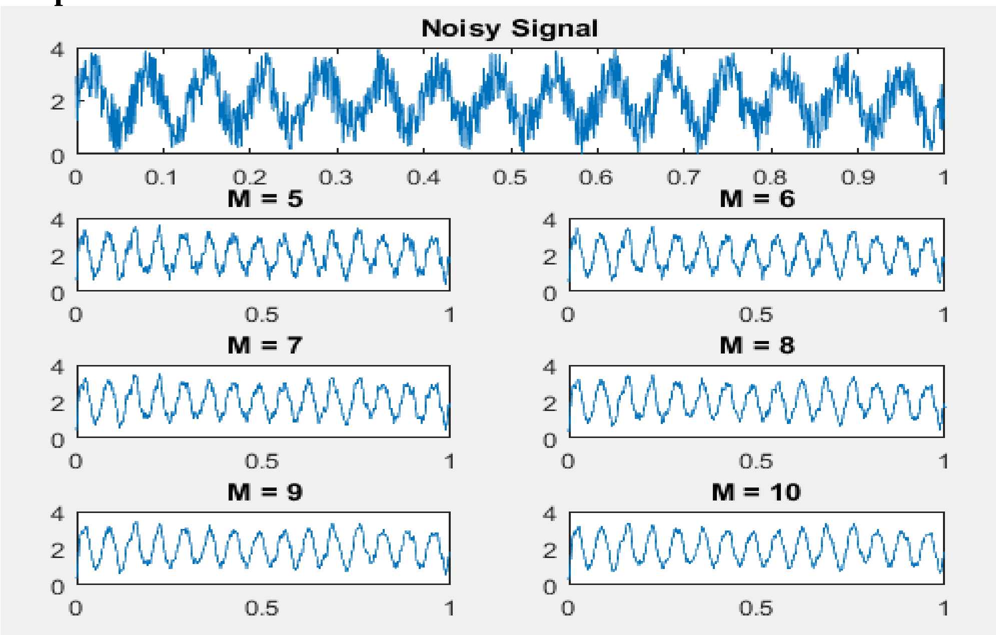
Questions:

1. **Replace in[n] in the practical by a sinusoid of frequency 15 Hz and contaminated it with noise. Perform the moving average. What value of M removes most of the noise? What is the drawback of the moving-averaging filter?**

Code:

```
clc
clear all
close all
f = 15;
fs = 1000;
t = 0:1/fs:1;
noise = 1+2*rand(size(t));
signal = sin(2*pi*f*t);
x = signal+noise;
subplot(4,2,[1,2]);
plot(t,x);
title('Noisy Signal');
i=3;
for m=5:10
    b = (1/m)*ones(1,m);
    y = filter(b,1,x);
    subplot(4,2,i)
    plot(t,y);
    title(['M = ',num2str(m)]);
    i=i+1;
end
```

Output:



Discussion:
The value of 10 for M removes most of the noise in above example. The drawback of moving average filter is that after a certain value of M, we may lose some data if we increase the value of M.

2. **What is the value of M which must not be taken for a moving average in the above question?**

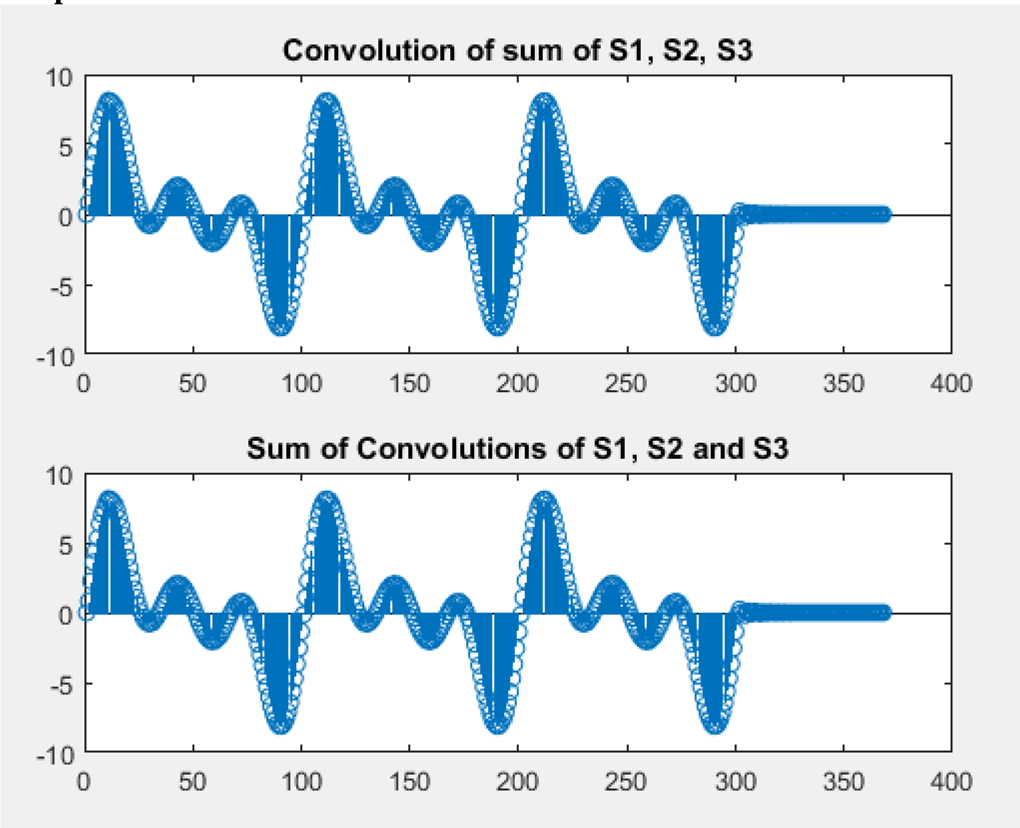
Answer:
I think we should avoid taking the value of M above 10 for the above example as we may lose data if we do so.

3. **Change x[n] to sum of three sinusoids with the third of frequency 30 Hz and repeat the procedure. Does the properties of the system change?**

Code:
```
clc
clear all
close all
f1 = 10;
f2 = 20;
f3 = 30;
fs = 1000;
n = 0:1/fs:0.30;
bx = [2.2403 2.4908 2.2403];
by = [1 0.4 0.75];
h = impz(bx,by);
x1 = sin(2*pi*f1*n);
x2 = sin(2*pi*f2*n);
x3 = sin(2*pi*f3*n);
x = x1+x2+x3;
y1 = conv(x,h);
subplot(2,1,1)
stem(y1);
title('Convolution of sum of S1, S2, S3');

yx1 = conv(x1,h);
yx2 = conv(x2,h);
yx3 = conv(x3,h);
y2 = yx1+yx2+yx3;
subplot(2,1,2)
stem(y2);
title('Sum of Convolutions of S1, S2 and S3');
```

Output:



Discussion:

Again the output is same so, the properties of the system does not change.