

ECEN 220 Lab 1

Niels Clayton
300437590

Part 1: Signal Periodicity

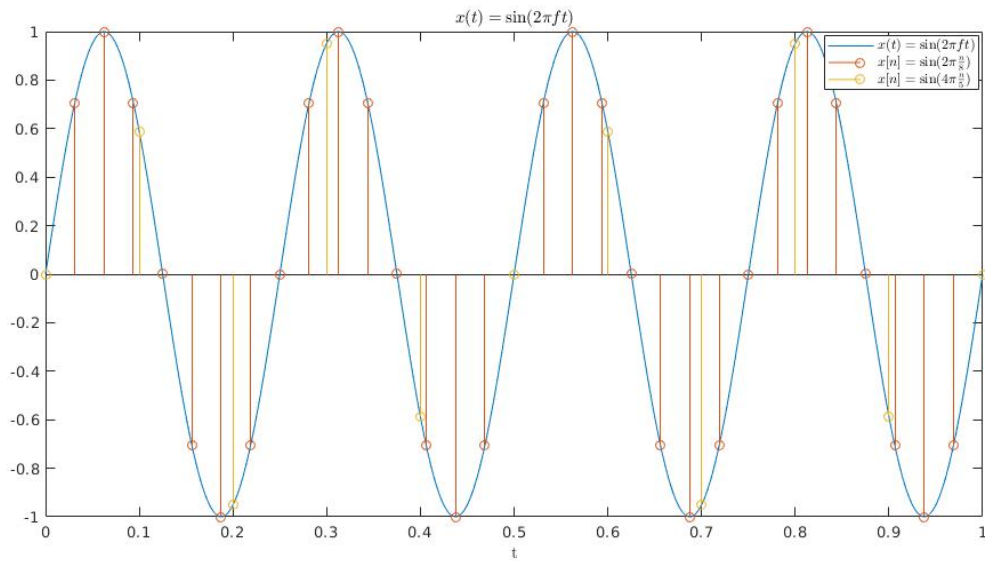


Figure 1: continuous and discrete time plots

The period of a continuous time signal can be found using $\frac{1}{f_0} = \frac{1}{4}$.

The period of a discrete time signal can be found using $\frac{2\pi}{\omega_0} = \frac{N}{M}$, Where N is the period as a number of samples, and M is the number of samples that fit into one cycle of the discrete signal. The period of one sample is one over the sampling frequency.

Using this we can find that:

$x(t)$ has a period of $\frac{1}{4}$ or $0.25s$

$x_1[n]$ has a period of 8 samples

$x_2[n]$ has a period of 5 samples

Using the sampling frequencies of x_1 and x_2 respectively, we can calculate the time period in terms of seconds.

$$x_1[n] : 8 \times \frac{1}{32} = 0.25s$$

$$x_2[n] : 5 \times \frac{1}{10} = 0.5s$$

From this we can see that x_1 has the same period as $x(t)$, this can be seen in figure 1 above. It can also be seen that x_2 has a period twice that of $x(t)$, meaning that it repeats ever 2 periods of $x(t)$

Part 2: Linearity

If a system $f(x)$ is linear, The following rule will be true:

$$f(x_1 + x_2) = f(x_1) + f(x_2)$$

When this test was applied to the system $y[n] = 2^{x[n]}$ the two different outputs were plotted against each other in figure 2.

Since the two discrete time plots do not match all values, it can be concluded that the system $y[n] = 2^{x[n]}$ is not linear.

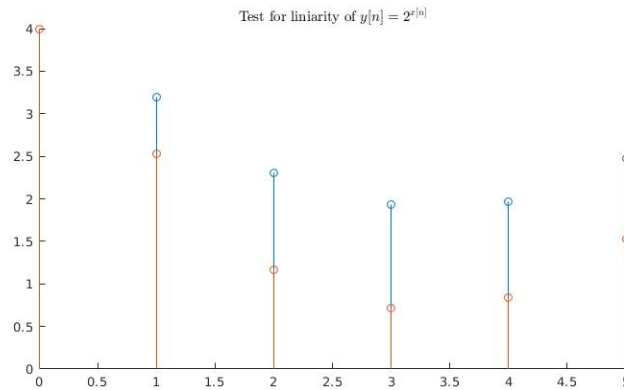


Figure 2: $f(x_1 + x_2) = f(x_1) + f(x_2)$

When this test was applied to the system $y[n] = nx[n]$ the two different outputs were plotted against each other in figure 3.

Since the two discrete time plots match entirely, it can be concluded that the system $y[n] = nx[n]$ is linear.

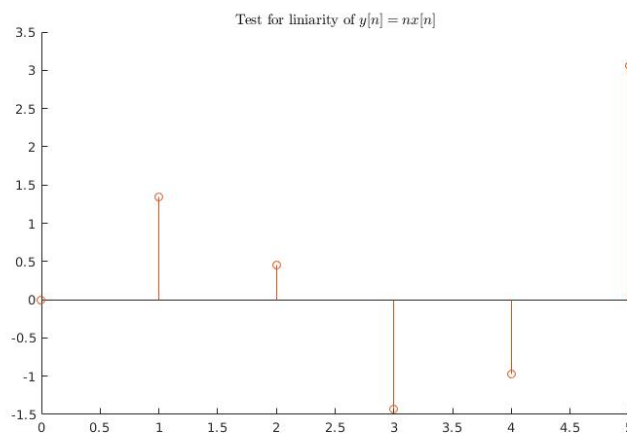


Figure 3: $f(x_1 + x_2) = f(x_1) + f(x_2)$

Convolution

The functions $h[n] = 0.7^n$ and $x[n] = u[n] - u[n - 4]$ were convolved in matlab, and by hand, to produce the function $y[n]$.

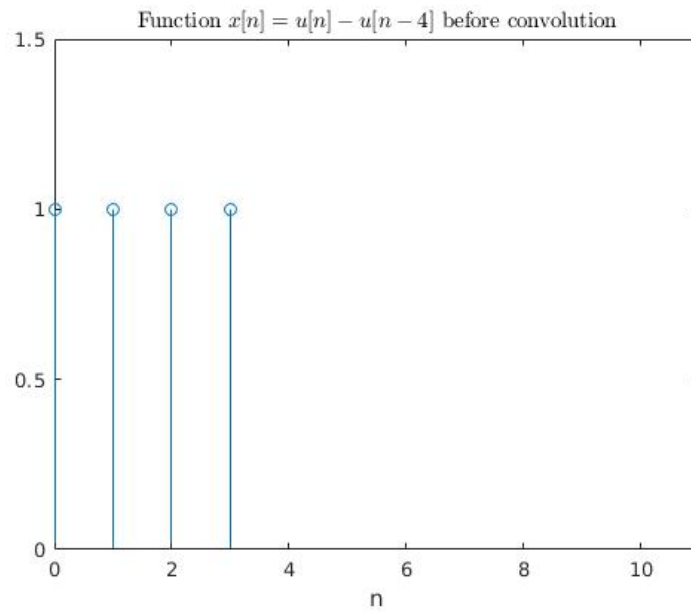


Figure 4: $h[n] = 0.7^n$

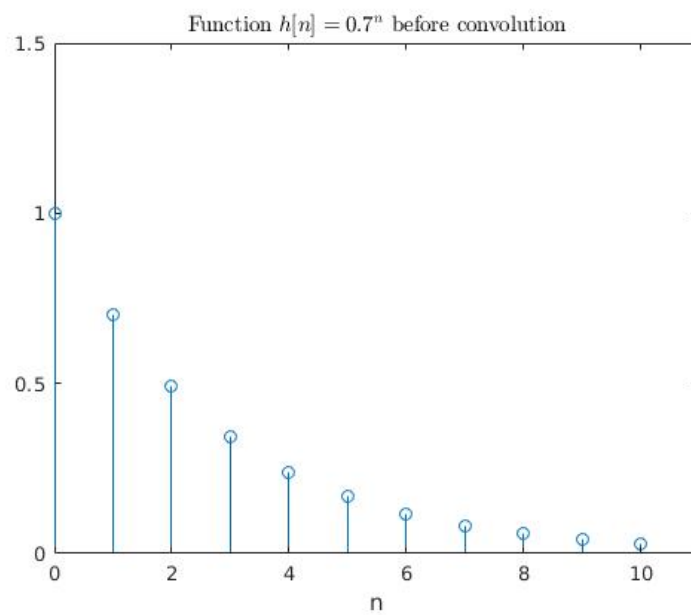


Figure 5: $x[n] = u[n] - u[n - 4]$

The output of this convolution done in matlab can be seen in figure 6.

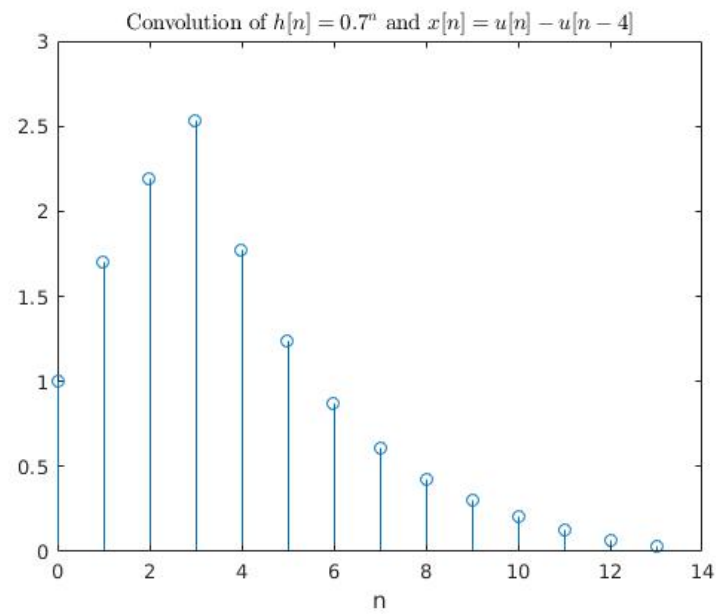


Figure 6: Convolution of $x[n]$ with $h[n]$

The output of the convolution done by hand can be seen to be exactly the same in figure 7, with the working below in figure 8.

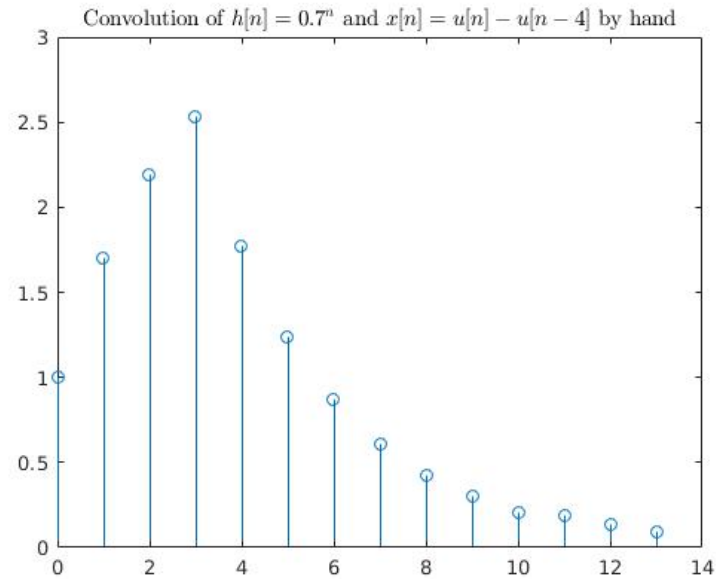


Figure 7: Convolution of $x[n]$ with $h[n]$ done by hand

$$h[n] = 0.7^n, \quad x[n] = u[n] - u[n-4]$$

Region 1 : $n < 0$: $y[n] = 0$

Region 2 : $0 \leq n < 3$

$$y[n] = \sum_{k=0}^n 0.7^k = \frac{1 - 0.7^{n+1}}{1 - 0.7}$$

Region 3 : $3 \leq n \leq 10$

$$y[n] = \sum_{k=n-3}^n 0.7^k = \frac{0.7^{n-3} - 0.7^{n+1}}{1 - 0.7}$$

Region 4 : $10 \leq n \leq 13$

$$y[n] = \sum_{k=n-3}^{10} 0.7^k = \frac{0.7^{n-3} - 0.7^{11}}{1 - 0.7}$$

Figure 8: Convolution of $x[n]$ with $h[n]$ working

Matlab Code

```

1 %% Question 1
2 clear variables; clc;
3
4 hold on
5 t = linspace(0, 1, 1000); % Continuous time sampling
6 f_0 = 4; % Frequency
7 x = sin(2*pi*f_0*t); % Function
8
9 figure(1);
10 plot(t,x)
11 title('$x(t)=\sin(2\pi f_0 t)$','interpreter','latex')
12 xlabel('t','interpreter','latex')
13
14 n = 0:1:32; % Discrete time sampling

```

```

15 x_1 = sin(2*pi*n/8);
16 stem(n/32, x_1);
17
18 n = 0:1:10; % Discrete time sampling
19 x_1 = sin(4*pi*n/5);
20 stem(n/10, x_1);
21 legend('x(t)=\sin(2\pi ft)', '$x[n]=\sin(2\pi \frac{n}{8})$', '$x[n]$'
        '=\sin(4\pi \frac{n}{5})$', 'interpreter', 'latex')
22 hold off
23
24
25 %% Question 2
26 clear variables; clc;
27
28 n = 0:1:5
29 x_1 = power(0.8, n);
30 x_2 = cos(n);
31 y_1 = power(2, x_1);
32 y_2 = power(2, x_2);
33 y_out_1 = y_1+y_2;
34
35 x_3 = x_1+x_2;
36 y_out_2 = power(2, x_3);
37
38 figure(2);
39 hold on
40 title('Test for liniarity of $y[n]=2^{\{x[n]\}}$', 'interpreter', 'latex')
41 stem(n, y_out_1)
42 stem(n, y_out_2)
43 hold off
44
45 y_1 = n.*x_1;
46 y_2 = n.*x_2;
47 y_out_1 = y_1+y_2;
48 y_out_2 = n.*x_3;
49 figure(3)
50 hold on
51 title('Test for liniarity of $y[n]=nx[n]$', 'interpreter', 'latex')
52 stem(n, y_out_1)
53 stem(n, y_out_2)
54 hold off
55
56 %% Question 3
57
58 clear variables; clc;
59
60 n = [0:1:10];
61 n2 = [0:1:3];
62 n3 = [0:1:13];
63 h = power(0.7, n)
64 x = (n2>=0) - ((n2-4)>=0);
65

```

```

66 y = conv(h,x);
67
68 figure(4)
69 stem(n2,x)
70 xlabel("n")
71 axis([0 11 0 1.5])
72 title('Function  $x[n] = u[n] - u[n-4]$  before convolution', '
       interpreter','latex')
73
74 figure(5)
75 stem(n,h)
76 xlabel("n")
77 axis([0 11 0 1.5])
78 title('Function  $h[n] = 0.7^n$  before convolution','interpreter','
       latex')
79
80
81 figure(6)
82 stem(n3,y)
83 title('Convolution of  $h[n] = 0.7^n$  and  $x[n] = u[n] - u[n-4]$ ','
       interpreter','latex')
84 xlabel('n')
85
86 figure(7)
87
88
89 syms n_2
90 y_hand = piecewise(0<=n_2<=3, (1-0.7.^(n_2+1))/0.3, 3<n_2<=10,
       ((0.7.^(n_2-3))-0.7.^(n_2+1))/0.3, 10<n_2<=13,((0.7.^(n_2-3))
       -0.7.^(n_2+11))/0.3)
91
92
93 n_3 = 0:1:13
94 y_hand_2 = subs(y_hand, n_2, n_3)
95 stem(n_3, y_hand_2)
96 title('Convolution of  $h[n] = 0.7^n$  and  $x[n] = u[n] - u[n-4]$  by
       hand','interpreter','latex')

```