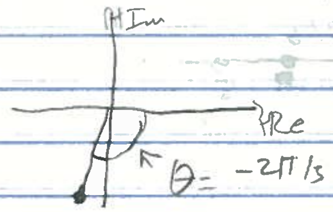
3

Dan Ports
6.003 PS1

1. a. $\frac{\sqrt{3}+j}{-\sqrt{3}+j} = \frac{(\sqrt{3}+j)(-\sqrt{3}-j)}{3+1} = \frac{-(-3-2\sqrt{3}j+1)}{4}$

$= -\frac{1}{2} - \frac{\sqrt{3}}{2}j = \boxed{1 \cdot e^{-j2\pi/3}}$ ✓

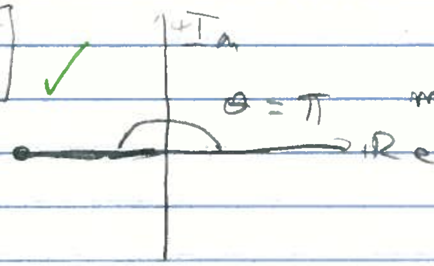


magnitude = 1

b. $(1-j)^7 e^{j3\pi/4} = (\sqrt{2}e^{-j\pi/4})^7 e^{j3\pi/4}$

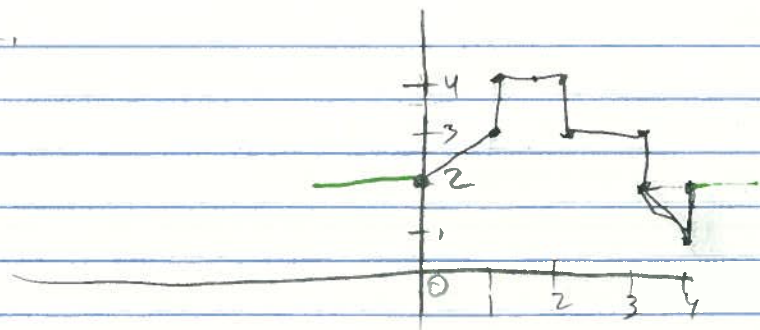
$= 8\sqrt{2}e^{-7\pi/4 j}e^{j3\pi/4} = 8\sqrt{2}e^{j-4\pi/4} = 8\sqrt{2}\cdot -1 = -8\sqrt{2}$

$= \boxed{8\sqrt{2}\cdot e^{j\pi}}$ ✓



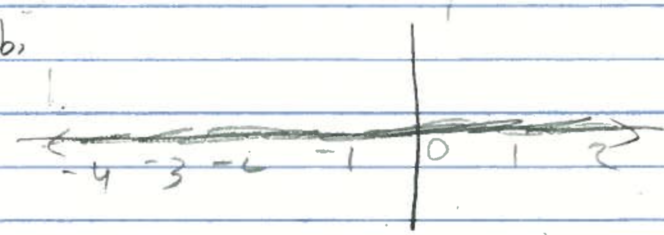
magnitude = $8\sqrt{2}$

2. a.



✓

b.

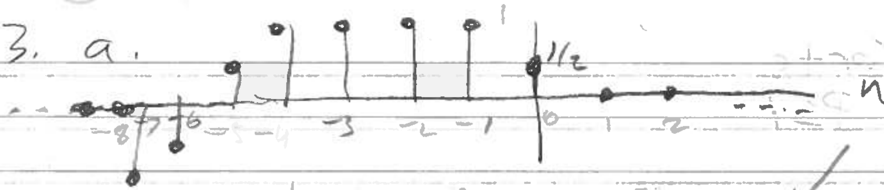


except 2 points
see solution

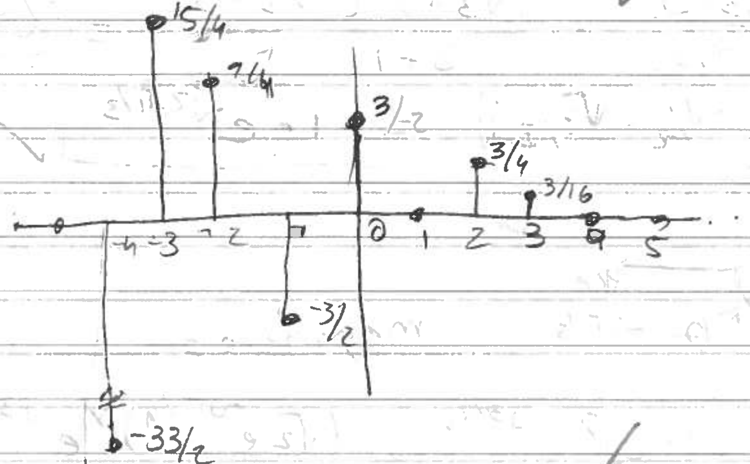
The signal is 0 for all t because the scaled, shifted signal is only non-zero for $t>0$, and v(t) is zero for $t>0$, and the impulses exist at points where the scaled, shifted signal is zero

3

3. a.



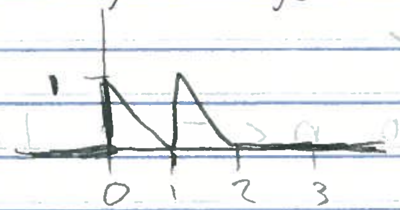
b.



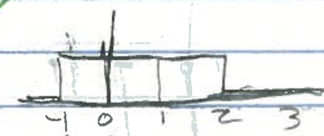
(not to scale)

4. a. True. Otherwise it would be non-linear. why?
b. False. It is not time-independent
c. True. Linear and time-independent
show some work

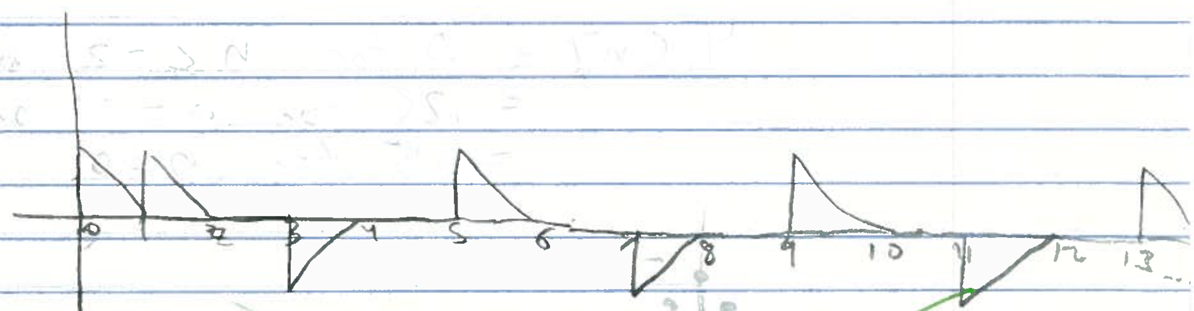
5. a $x'(t) = u(t+1) - u(t-3) = x(t) + x(t-1)$.
So $y'(t) = y(t) + y(t-1)$.



b. $x'(t) = u(t+1) - u(t-2)$.

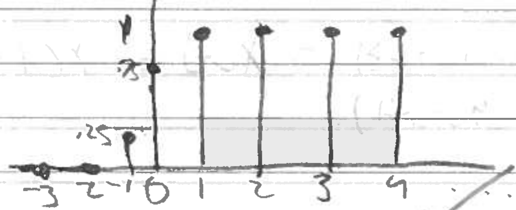


$= x(t) + x(t-1) - x(t-3) + x(t-5) - x(t-7) \ldots$
Therefore $y'(t) = y(t) + y(t-1) - y(t-3) + y(t-5) - x(t\ldots$



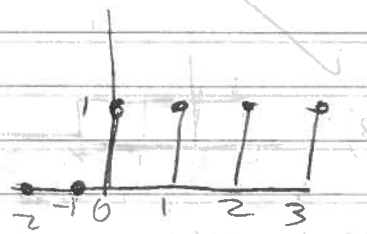
6. See attached.
The coefficients found were
0, 1.3, 0, 0.4, 0, 0.2

7. a. $y_1[n] = 0,\ n \le -2,$
$0.25,\ n=-1,\ 0.75,\ n=0$
$1,\ n \ge 1$



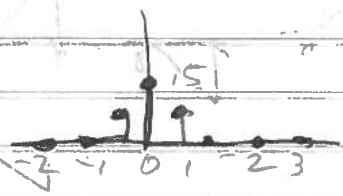
$y_1[n]$

$y_2[n] = 0,\ n < -1;\ 1,\ n \ge 0$

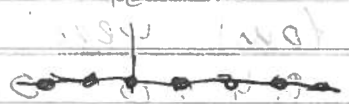


$y_2[n]$

b. $y_1[n] = 0$ for $n \le -2$ or $n \ge 2$
$= .25$ for $n = -1$ or $n = 1$
$= .5$ for $n = 0$



$y_2[n] = 0$ for all n



c. Both filters are time-invariant; time-shifting the input shifts the output accordingly. Only the first is linear. For y_1, we find that $y_1[x_1 + x_2]$
$= 0.25(x_1[n-1] + x_2[n-1]) + .5(x_1[n] + x_2[n])$
$+ .25(x_1[n+1] + x_2[n+1])$, which is what we expect from superposition. However, y_2 is non-linear. Consider $x_1 = u(t-1)$ and $x_2 = u(t+1)$. Then

$y_2[x_1] = y_2[x_2] = 0$ at $n=0$,
but $y_2[x_1+x_2] = 1$ at $n=0$; superposition does not hold.

d. Matlab function and printouts attached. The first filter reduced the prominence of the noise somewhat, but it was still present. The "noisy" pixels incorporated 25% of the value of each of their neighbors, but 50% still came from the noise. The artifacts were thus visible.
The second filter was more effective at hiding the noise. By taking the median of three neighboring pixels, sudden changes in value such as those resulting from black or white noise pixels were eliminated and replaced with a neighbor's value. This removed most of the noise. Also, edges were sharper in the second image than the first. We expect this because the second filter's unit sample response is zero; this is analogous to removing a single black noise point in a set of surrounding white pixels; the first only blurs the impulse. Also, the first filter's step response blurs the boundary between the zeros and ones, while the second's does not. This is reflected in the images.

good!

But some detail in the original image was lost when using the 2nd filter

```
function plotsin(a0,a1,a2,a3,a4,a5);

t=linspace(-pi,pi,100)
y=a0+a1*sin(t)+a2*sin(2*t)+a3*sin(3*t)+a4*sin(4*t)+a5*sin(5*t)
plot(t,y)
```

problem 6 - Matlab function
plotsin for plotting combinations of sines

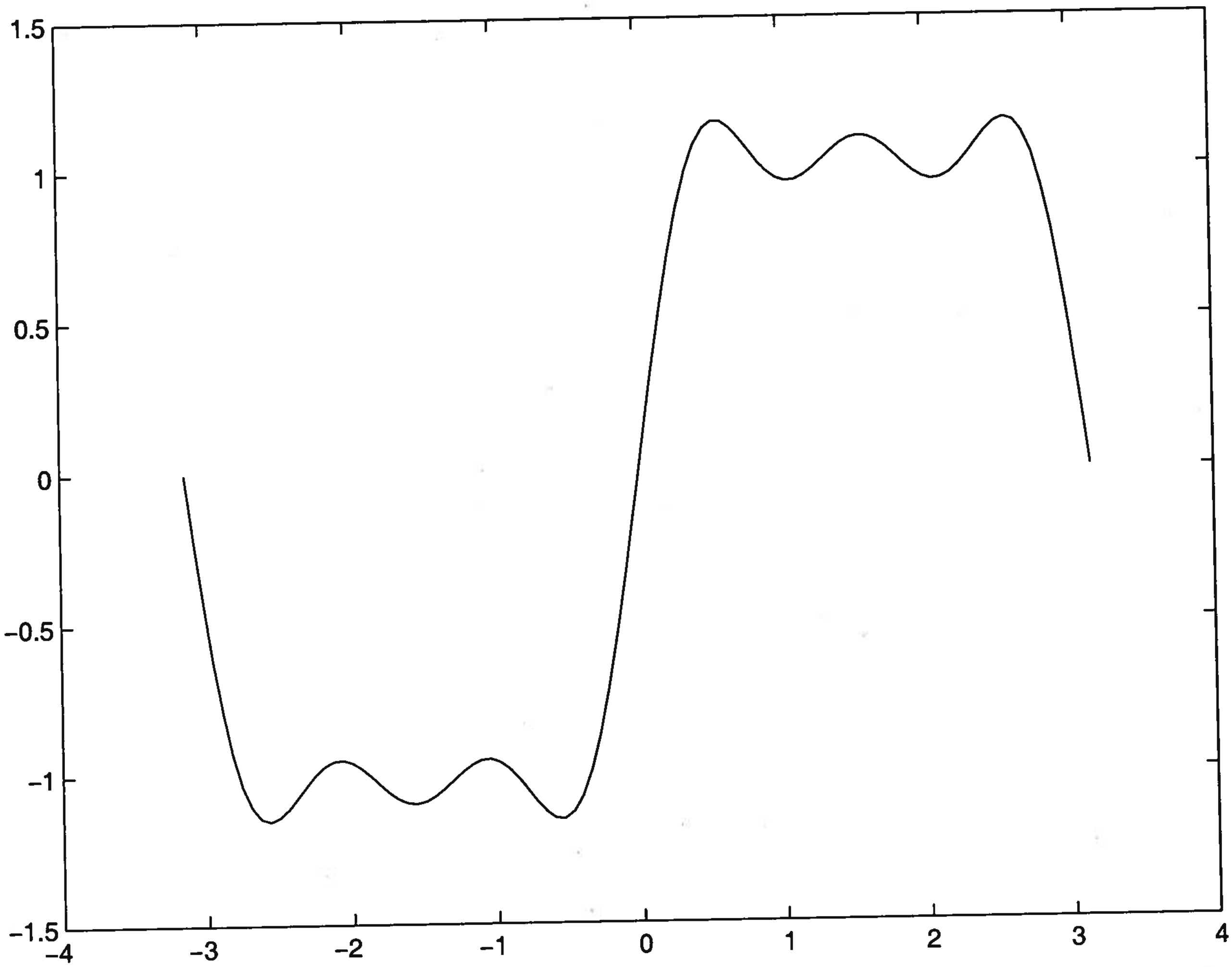
problem 6 – approximation of square wave

coefficients:

$a_0 = 0$ $\quad$ $a_3 = 0.4$

$a_1 = 1.3$ $\quad$ $a_4 = 0$

$a_2 = 0$ $\quad$ $a_5 = 0.2$



✓

```
function outputdata = filterII(inputdata)
[a,b] = size(inputdata);
inputdata=double(inputdata);


for x = 2:a-1,
for y = 1:b,
outputdata(x,y)=median([inputdata(x-1,y),inputdata(x,y),inputdata(x+1,y)]);
end
end

outputdata=uint8(outputdata)
```

problem 7 - Matlab function filterII

✓

problem 7 - output from filter I