Umutcan Uğuz

**1)** $x_c(t) = 4\sin(20000\pi t + \frac{\pi}{13})$ and $F_s = 3\,KHz$

$x[n] = x_c\left[\frac{n}{F_s}\right] = 4\sin\left(\frac{20\pi n}{3} + \frac{\pi}{13}\right)$

**a)**

sine function is periodic with $2\pi$

$\Rightarrow 4\sin\left(\frac{20\pi n}{3} + \frac{\pi}{13}\right) = 4\sin\left(\frac{2\pi(10000+\Delta f)n}{3000} + \frac{\pi}{13}\right)$

$$\Rightarrow \quad = 4\sin\left(\frac{20\pi n}{3} + \frac{\pi}{13} + \underbrace{2\pi\frac{\Delta f}{3000}n}_{\text{must be } 2\pi n}\right)$$

$\rightarrow \Delta f = 3000\ Hz$

$x_c(t)$ has frequency of 10kHz. 13 KHz, 16 KH

... would yield $x[n]$

$$x_{c,set}(t) = 4\sin\left(2\pi(10000 + 3000n)t + \frac{\pi}{13}\right)$$

where $n = -3, -2, -1, 0, 1, 2, \ldots$

**b)** $4\sin\left(\frac{20000\pi}{F_s} + \frac{\pi}{13}\right) = 4\sin\left(\frac{20\pi}{3} + \frac{\pi}{13} + 2\pi\right.$

$\Rightarrow \frac{20000}{F_s} = \frac{20}{3} + 2k \Rightarrow F_s = \frac{30000}{10+3k}$

which satisfies $F_s > 0$

$k \in \mathbb{Z}$

2) $N = \frac{2\pi k}{\omega_0}$ k must be smallest possible integer making N integer.

By intiution there must be $\pi$ for periodicity

$\rightarrow \sin(1{,}74\pi n + 3.1)$

$$N = \frac{k\,2\pi}{1{,}74\pi} = \frac{100k}{87} \qquad N=100$$

$\rightarrow \sin(1.74\pi n + 3.1\pi)$ N=100 again

$\rightarrow \cos\left(15{,}74\pi n + \frac{3\pi}{8}\right)$

$$N = \frac{k.2\pi}{15{,}74\pi} = \frac{100k}{787} \qquad N=100$$

$\rightarrow \cos(\sqrt{\pi}\, n)$

$N = \frac{k.2\pi}{\sqrt{\pi}}$ there is no k value making N integer. not periodic

$\rightarrow \cos(\pi\sqrt{\pi}\, n)$ not periodic

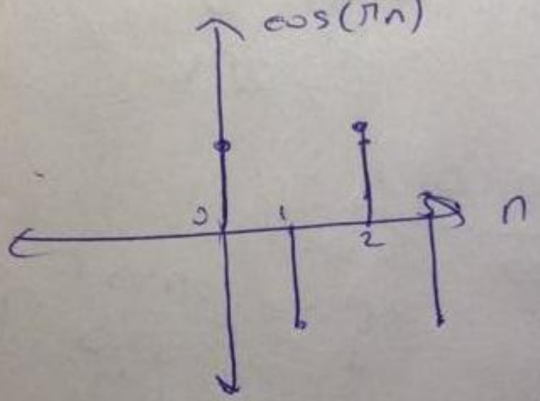
$\rightarrow \cos(\pi\sqrt{2}\, n)$

$$N = \frac{k\,2\pi}{\pi\sqrt{2}} = k\sqrt{2} \quad \text{not periodic}$$

3) it is $\pi$

$\rightarrow$ $\cos(\pi n)$ is the highest frequency sinusoidal

$$\cos(\pi n) = (-1)^n$$



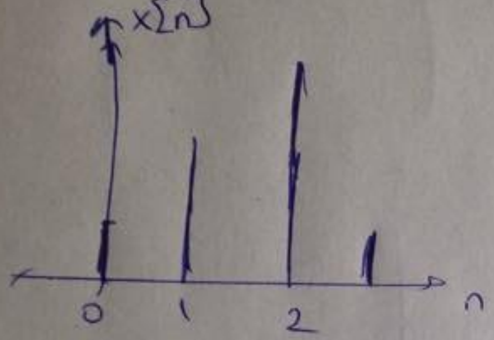

we don't take $\cos(2\pi n)$ into considera-tion as it is constant, not an actual signal

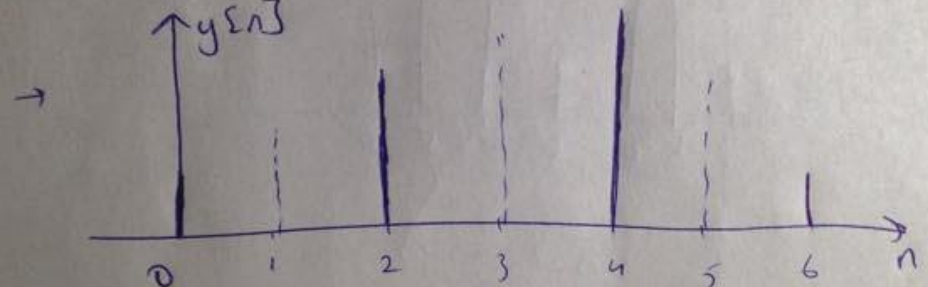
4) $y[0] = x[0]$ $\quad y[1] = \frac{x[0]+x[1]}{2}$

$y[2] = x[1]$ $\quad y[3] = x[1]+x[2]$

$y[4] = x[2]$

$y[6] = x[3]$




→ linearity $x_1 = a\,x[n] \rightarrow y_1[n] = a y_1$ linear

similarly $x_3 = a x_1 + b x_2 \rightarrow y_3[n] = a y_1 + b y_2$

→ it is easy to see the system is time-varying

$x_1 = x[n+1] \Rightarrow y_1[n] \neq y[n+1]$

This system makes time-scaling "up". It increase the number of samples. To fill the gaps it ~~adds~~ uses arithmetic mean. This can also be used to increase pixel numbers in ~~[illegible]~~ an image. It smoothes the signal

5) not LTI !

a) $y[n] = 2^{\delta[n+1]} + x[n-3]$

CASUAL, not depending on future inputs

$$y[n] = \begin{cases} x[n-3]+1 & , n \neq -1 \\ x[n-3]+2 & , n = -1 \end{cases}$$

for finite x inputs $y[n]$ is finite

STABLE

b) $$y[n] = \begin{cases} y[-\delta[n-1]] + x[n-3] & n>0 \\ 2^n x[n-3] & n \leq 0 \end{cases}$$

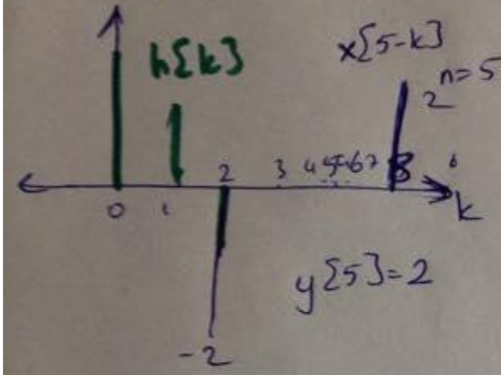
$$= \begin{cases} 2^n \cancel{x}[n-3] & n \leq 0 \\ y[-1] + x[-2] & n=1 \\ y[0] + x[n-3] & \text{otherwise} \end{cases}$$

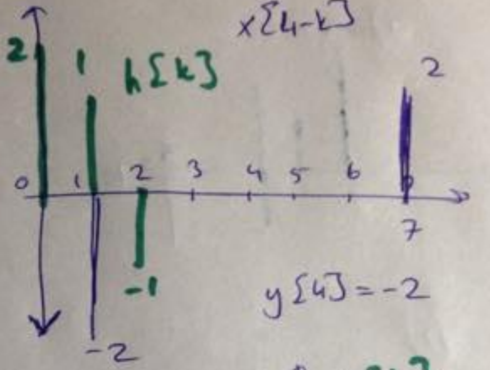
$$= \begin{cases} 2^n x[n-3] & n \leq 0 \\ \tfrac{1}{2} x[-4] & n=1 \\ \cancel{y}[-3] + x[n-3] & \text{otherwise} \end{cases}$$

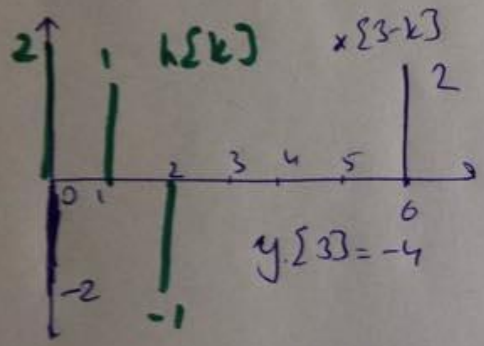
$y[n]$ is finite for finite $x[n]$ STABLE

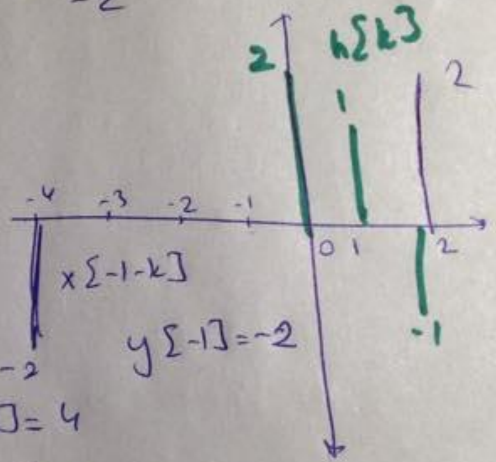
not depending on future inputs CASUAL

b) $y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k]$

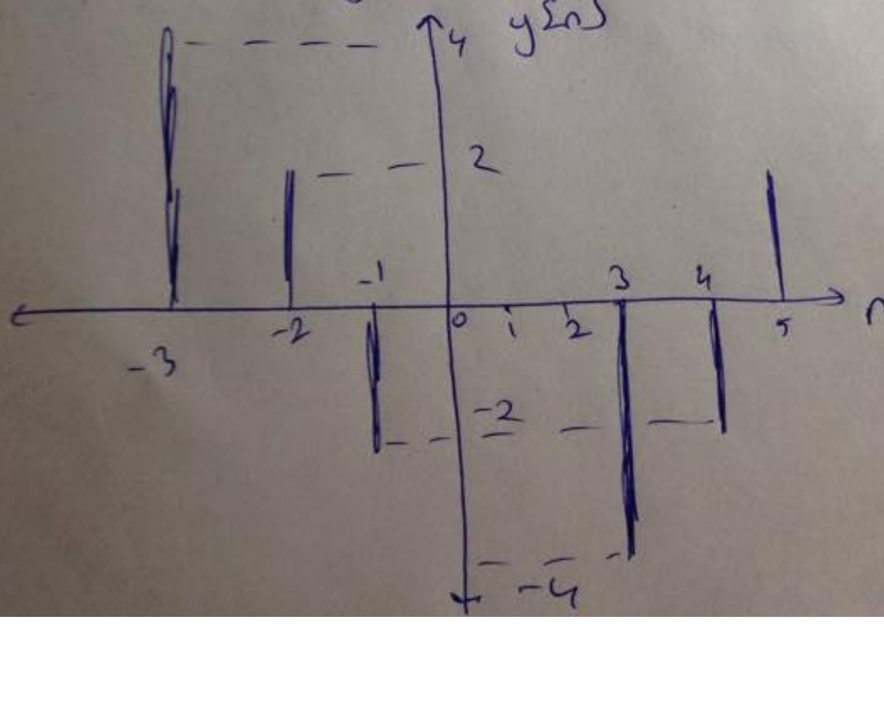



Similarly $y[-2] = 2$, $y[-3] = 4$
all other $y$ is zero



7) i) $x[-n] * h[-n]$

$$= \sum_{k=-\infty}^{\infty} x[k]\, h[-n-k] =$$

say $-n = a$

$$= \sum_{k=-\infty}^{\infty} x[k]\, h[a-k] = y[a] = y[-n]$$

✓

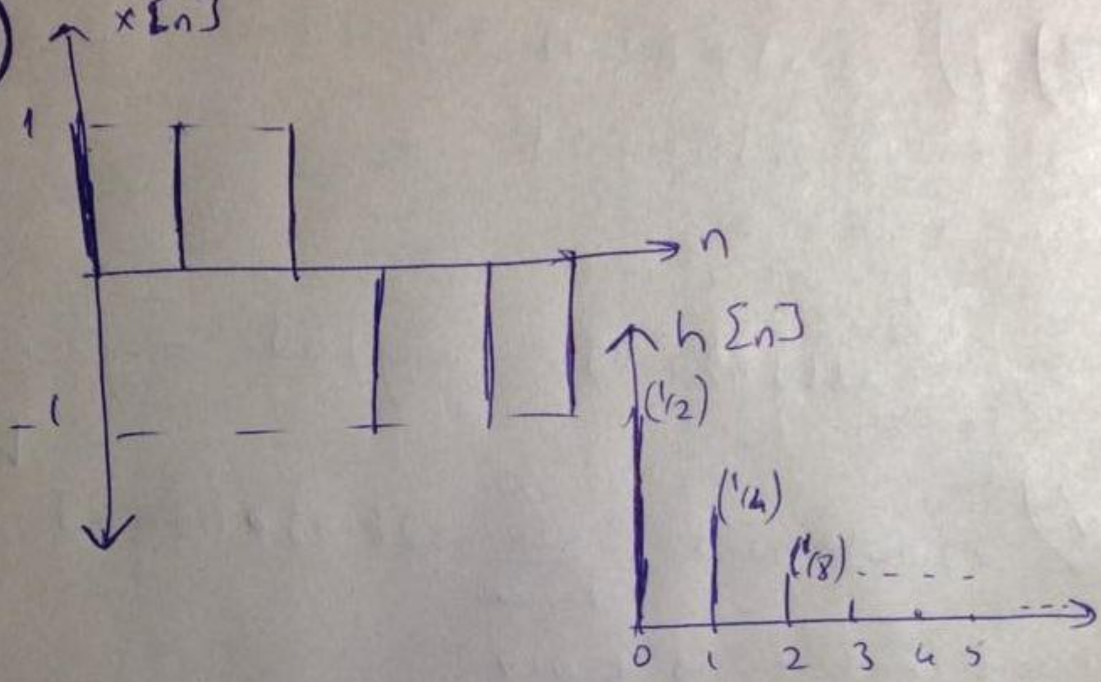
ii) $x[n-4] * h[n] = \sum_{k=-\infty}^{\infty} x[k-4] * h[n-k]$

say $a = k-4 \rightarrow k = a+4$

$$\sum_{a=-\infty}^{\infty} x[a] \cdot h[n-a-4] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k-4]$$

$$= y[n-4] = x[n] * h[n-4]$$

8)



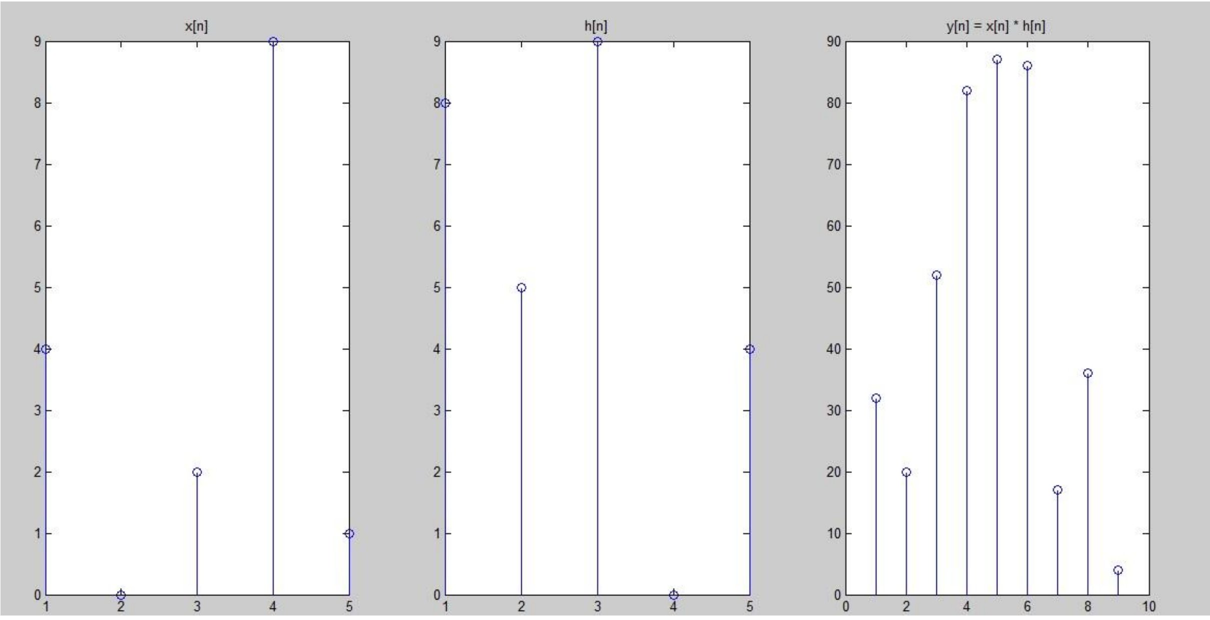

ii) $\sum_{k=-\infty}^{\infty} x[k]\,h[n-k]$

$= h[n] + h[n-1] + h[n-2] - h[n-3] - h[n-4] - h[n-5]$

# 9

---

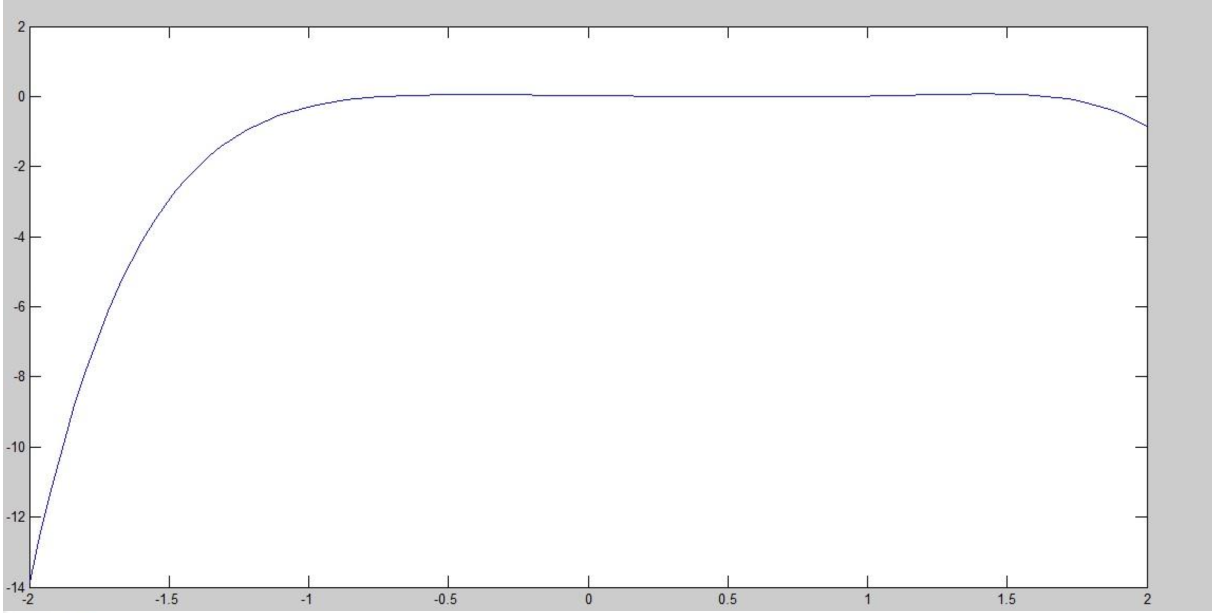
```
>> clear
x = floor(rand(1,5)*10)
h = floor(rand(1,5)*10)
y = conv(x,h)
subplot(1,3,1), stem(x), title('x[n]');
subplot(1,3,2), stem(h), title('h[n]');
subplot(1,3,3), stem(y), title('y[n] = x[n] * h[n]');

x =

     4     0     2     9     1

h =

     8     5     9     0     4


y =

    32    20    52    82    87    86    17    36     4
```




The length of the output signal is as expected which is (k+m-1), where k = 5 and m = 5.
As the both signals are positive valued, one may expect output signal to have higher values at middle n values.

# 10.a)

```
>> x1 = rand(1,3)-0.5;
x2 = rand(1,5)-0.5;
n = linspace(-2,2);
y = conv(x1,x2);
y = polyval(y,n);
plot(n,y)
>>
```




```
>> x1
x2
conv(x1,x2)

x1 =

   -0.2401    0.3001   -0.0686


x2 =

    0.4106   -0.3182   -0.2362   -0.3545   -0.3639


ans =

   -0.0986    0.1996   -0.0669    0.0361   -0.0028   -0.0849    0.0250
```

The convolution of the coefficients of the two polynomials gives us the coefficient vector of the multiplication polynomials which has the length of 5+3-1 = 7

# b) help deconv

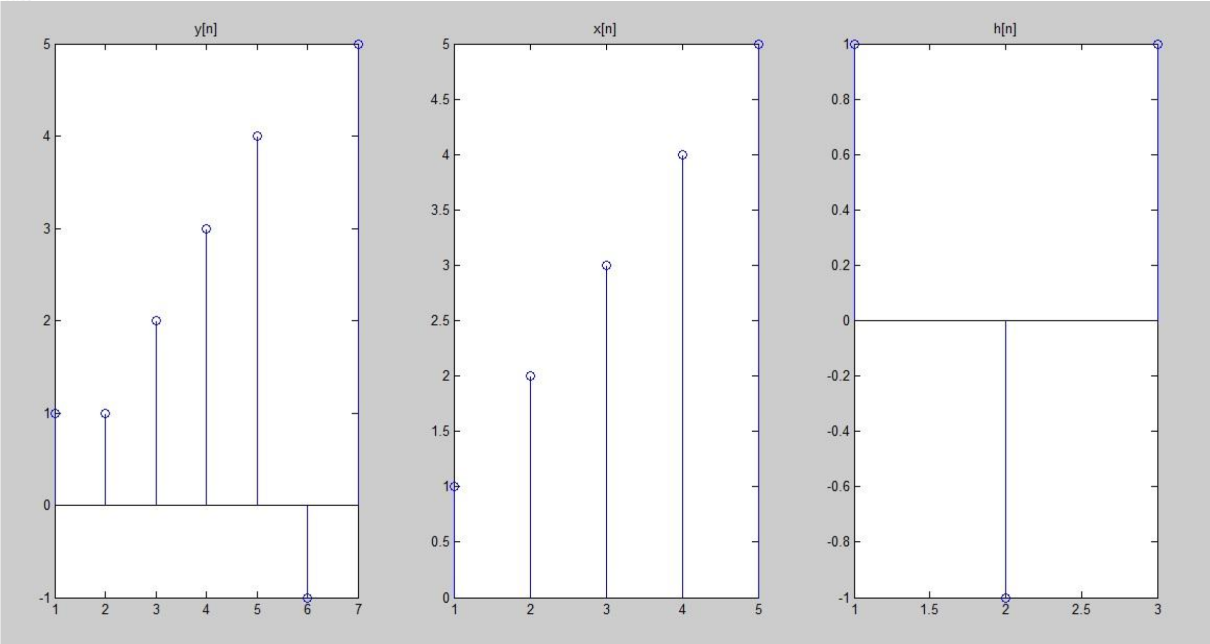
**c)** y[n] = {1,1,2,3,4,-1,5} ;
x[n] = {1,2,3,4,5};

If we consider them as polynomials by polynomial division we would find h[n]

(t^6 + t^5 + 2*t^4 + 3*t^3 + 4*t^2 - t + 5) / (t^4 + 2*t^3 + 3*t^2 + 4*t +5) = (t^2 –t + 1)

Then h[n] = {1,-1,1}

```
>> y = [1 1 2 3 4 -1 5];
x = [1 2 3 4 5];
h = deconv(y,x)

subplot(1,3,1), stem(y), title('y[n]');
subplot(1,3,2), stem(x), title('x[n]');
subplot(1,3,3), stem(h), title('h[n]');

h =

     1    -1     1

>>
```




**d)** y[n] = {1,2,2,3,4,-1,5} ;
x[n] = {1,2,3,4,5};

If we consider them as polynomials by polynomial division we would find h[n]

(t^6 + 2*t^5 + 2*t^4 + 3*t^3 + 4*t^2 - t + 5) / (t^4 + 2*t^3 + 3*t^2 + 4*t +5) = (t^2 + 1)

Then h[n] = {1,0,1}

```
>> y = [1 2 2 3 4 -1 5];
x = [1 2 3 4 5];
h = deconv(y,x)

subplot(1,3,1), stem(y), title('y[n]');
subplot(1,3,2), stem(x), title('x[n]');
subplot(1,3,3), stem(h), title('h[n]');

h =

     1     0    -1

>>
```