CENG 384 - Signals and Systems for Computer Engineers
Spring 2023
Homework 2

Meyvecioğlu, Hasan Ege
e2449783@ceng.metu.edu.tr

Şanlı, Enes
e2375749@ceng.metu.edu.tr

April 19, 2023

---

1. (a) The output of the adder is $y'(t)$, which is equal to $x(t) - 5 * y(t)$.If we integrate $y'(t)$, we get the output signal $y(t)$. Hence:

$$y'(t) = x(t) - 5y(t) \to y'(t) + 5y(t) = x(t) \tag{1}$$

(b) If $x(t) = (e^{-t} + e^{-3t})u(t) = e^{-t}u(t) + e^{-3t}u(t)$ and the system is initially at rest, then:

$$y(0) = y'(0) = 0$$

We can use Laplace transform to solve this ODE:

$$\mathscr{L}\{y'(t)\} + 5 * \mathscr{L}\{y(t)\} = \mathscr{L}\{e^{-3t} * u(t)\} + \mathscr{L}\{e^{-t} * u(t)\}$$

$$s * Y(s) - \cancel{y(0)} + 5 * Y(s) = \frac{1}{s+1} + \frac{1}{s+3}$$

$$(s+5) * Y(s) = \frac{1}{s+1} + \frac{1}{s+3}$$

$$Y(s) = \frac{2s+4}{(s+1)*(s+3)*(s+5)}$$

$$Y(s) = \frac{A}{s+1} + \frac{B}{s+3} + \frac{C}{s+5}$$

$$As^2 + 8A * s + 15A + Bs^2 + 6sB + 5B + Cs^2 + 4Cs + 3C = 2s + 4$$

$$A + B + C = 0$$

$$8A + 6B + 4C = 2$$

$$15A + 5B + 3C = 4$$

Those equations gives:

$$A = \frac{1}{4}, B = \frac{1}{2}, C = \frac{-3}{4}$$

Then, $Y(s)$ becomes:

$$Y(s) = \frac{1}{4*s+1} + \frac{1}{2*(s+3)} + \frac{-3}{4*(s+5)}$$

Remember that we are trying to find the general solution $y(t)$ and we can find it by applying inverse Laplace transform on $Y(s)$:

$$\mathscr{L}^{-}\{Y(s)\} = \mathscr{L}^{-}\{\frac{1}{4*(s+1)}\} + \mathscr{L}^{-}\{\frac{1}{2*(s+3)}\} + \mathscr{L}^{-}\{\frac{-3}{4*(s+5)}\}$$

$$y(t) = \frac{e^{-t}}{4} + \frac{e^{-3t}}{2} - \frac{3*e^{-5t}}{4} \tag{2}$$

2. (a)

$$x[n] = x_1[n] + x_2[n]$$

$$x_1[n] = 2\delta(n)$$

$$x_2[n] = \delta[n+1]$$

$$h[n] = h_1[n] + h_2[n]$$

$$h_1[n] = \delta[n-1]$$

$$h_2[n] = 2\delta[n+1]$$
$$x[n] * h[n] = (x_1[n] + x_2[n]) * (h_1[n] + h_2[n])$$
$$x[n] * h[n] = x_1[n]h_1[n] + x_1[n]h_2[n] + x_2[n]h_1[n] + x_2[n]h_2[n]$$

Result

$$x[n] * h[n] = 2\delta[n-1] + 4\delta[n+1] + \delta[n] + 2\delta[n+2] \tag{3}$$

(b)

$$\frac{dx(t)}{dt} = \delta(t-1) + \delta(t+1) = a(t)$$
$$y(t) = \int_{-\infty}^{\infty} a(t-\tau) * h(t) d\tau$$
$$y(t) = \int_{-\infty}^{\infty} \delta(t-\tau-1) * h(t) d\tau + \int_{-\infty}^{\infty} \delta(t-\tau+1) * h(t) d\tau$$
$$y(t) = h(t-1) + h(t+1)$$
$$= e^{-t+1} * sin(t-1) * u(t-1) + e^{-t-1} * sin(t+1) * u(t+1) \tag{4}$$

3. (a)

$$y(t) = \int_{-\infty}^{\infty} x(\tau) * h(t-\tau)\, d\tau$$
$$= \int_{-\infty}^{\infty} e^{-\tau} * u(\tau) * e^{-2*(t-\tau)} * u(t-\tau)\, d\tau$$
$$= e^{-2*t} * \int_{0}^{t} e^{\tau} *\ d\tau$$
$$y(t) = e^{-2*t} * (e^t - e^0) = e^{-2*t} * (e^t - 1) = e^{-t} - e^{-2*t} \tag{5}$$

(b)

$$y(t) = \int_{-\infty}^{\infty} x(t-\tau) * h(\tau)\, d\tau$$
$$y(t) = \int_{-\infty}^{\infty} e^{3\tau} * (u(t-\tau) - u(t-1-\tau)) d\tau$$
$$y(t) = \int_{0}^{t} e^{3\tau} * u(t-\tau) d\tau - \int_{0}^{t-1} e^{3\tau} * u(t-1-\tau) d\tau$$
$$= \frac{e^{3t}}{3} - \frac{1}{3} - (\frac{e^{3(t-1)}}{3} - \frac{1}{3})$$
$$= \frac{e^{3t} * (1 - e^{-3})}{3}$$

4. (a) Let say $y[n] = a * r^n$

$$a * r^n - a * r^{n-1} - a * r^{n-2} = 0$$
$$1 - r^{-1} - r^{-2} = 0$$
$$r^2 - r - 1 = 0$$
$$b^2 - 4ac = (-1)^2 - 4 * 1 * -1 = 5$$
$$r = \frac{1 \pm \sqrt{5}}{2}$$

Then we should find constant "a_1 and a_2", give n = 2

$$y[n] = a_1 * (\frac{1+\sqrt{5}}{2})^n + a_2 * (\frac{1-\sqrt{5}}{2})^n$$
$$y[0] = 1, a_1 + a_2 = 1$$
$$y[1] = 1, a_1 * \frac{1+\sqrt{5}}{2} + a_2 * \frac{1-\sqrt{5}}{2} = 1$$
$$a_1 = \frac{5+\sqrt{5}}{10}, a_1 = \frac{5-\sqrt{5}}{10}$$

So y[n] will be

$$y[n] = (\frac{5+\sqrt{5}}{10} * (\frac{1+\sqrt{5}}{2})^n) + (\frac{5-\sqrt{5}}{10} * (\frac{1-\sqrt{5}}{2})^n) \tag{6}$$

(b) This is linear homogeneous differential equation, so let say

$$y(t) = e^{\lambda t}, y'(t) = \lambda e^{\lambda t}, y''(t) = \lambda^2 e^{\lambda t}, y'''(t) = \lambda^3 e^{\lambda t}$$
$$\lambda^3 - 6\lambda^2 + 13\lambda - 10 = 0$$
$$(\lambda - 2)(\lambda^2 - 4\lambda + 5) = 0$$
$$\lambda = 2, (2 - i), (2 + i)$$
$$y(t) = c_1 * e^{2t} + e^{2t}(c_2 cos(t) + c_3 sin(t))$$
$$y(0) = c_1 + c_2 = 1$$
$$y'(t) = 2c_1 e^{2t} + 2e^{2t}(c_2 cos(t) + c_3 sin(t)) + e^{2t}(-c_2 sin(t) + c_3 cos(t))$$
$$y'(0) = 2c_1 + 2c_2 + c3 = 3/2$$
$$y''(t) = 4c_1 e^{2t} + 4e^{2t}(c_2 cos(t) + c_3 sin(t)) + 2e^{2t}(-c_2 sin(t) + c_3 cos(t)) + 2e^{2t}(-c_2 sin(t) + c_3 cos(t)) + e^{2t}(-c_2 cos(t) - c_3 sin(t))$$
$$y''(0) = 4c_1 + 4c_2 + 2c_3 + 2c_3 - c_2 = 4c_1 + 3c_2 + 4c_3 = 3$$

Then,

$$c_1 = 2, c_2 = -1, c_3 = -1/2$$
$$y(t) = 2e^{2t} - e^{2t} * (cos(t) + \frac{sin(t)}{2}) \tag{7}$$

5. (a) Since $x(t) = cos(5t)$, we can propose a particular solution in the form of:

$$y_p(t) = K_1 * cos(5t) + K_2 * sin(5t)$$

Then the derivatives become

$$y_p'(t) = -5 * K_1 * sin(5t) + 5 * K_2 * cos(5t)$$
$$y_p''(t) = -25 * K_1 * cos(5t) - 25 * K_2 * sin(5t)$$

Putting those values to equation:

$$(-25*K_1*cos(5t) - 25*K_2*sin(5t)) + 5*(-5*K_1*sin(5t) + 5*K_2*cos(5t)) + 6*(K_1*cos(5t) + K_2*sin(5t)) = cos(5t)$$
$$(-19 * K_1 + 25 * K_2) * cos(5t) + (-25 * K_1 - 19 * K_2) * cos(5t) = cos(5t)$$

This gives us the following system of equations:

$$(-19 * K_1 + 25 * K_2) = 1$$
$$(-25 * K_1 - 19 * K_2) = 0$$

If we solve this system, we get:

$$\begin{pmatrix} K_1 \\ K_2 \end{pmatrix} = \begin{pmatrix} -\frac{19}{986} \\ \frac{25}{986} \end{pmatrix}$$

Then the particular solution becomes:

$$y_p(t) = \frac{-19}{986} * cos(5t) + \frac{25}{986} * sin(5t) \tag{8}$$

(b) We can use the method of undetermined coefficients for finding $y_h(t)$

$$\lambda^2 + 5 * \lambda + 6 = 0 \to \lambda_1 = -3, \lambda_2 = -2$$

Hence the homogeneous solution is in this form:

$$y_h(t) = C_1 * e^{-3*t} + C_2 * e^{-2*t} \tag{9}$$

(c) If the system is initially at rest, then the followings are valid:

$$y(0) = 0, y'(0) = 0, y''(0) = 0$$

We can find the C_1 and C_2 coefficients of the homogeneous using these.

$$y_g(t) = y_h(t) + y_g(t) = C_1 * e^{-3*t} + C_2 * e^{-2*t} + \frac{-19}{986} * cos(5t) + \frac{25}{986} * sin(5t)$$
$$y_g(0) = y_h(0) + y_g(0) = C_1 + C_2 + \frac{-19}{986} + 0 = 0$$

$$y_g'(t) = y_h'(t) + y_g'(t) = -3 * C_1 * e^{-3*t} - 2 * C_2 * e^{-2*t} + \frac{19 * 5}{986} * sin(5t) + \frac{25 * 5}{986} * cos(5t)$$

$$y_g'(0) = y_h'(0) + y_g'(0) = -3 * C_1 - 2 * C_2 + 0 + \frac{25 * 5}{986} = 0$$

We can solve for C_1 and C_2 using those equations:

$$\begin{pmatrix} C_1 \\ \\ C_2 \end{pmatrix} = \begin{pmatrix} \frac{3}{34} \\ \\ \frac{-2}{29} \end{pmatrix}$$

The general equation for the system that is initially at rest is:

$$y_g(t) = y_h(t) + y_g(t) = \frac{3}{34} * e^{-3*t} - \frac{2}{29} * e^{-2*t} - \frac{19}{986} * cos(5t) + \frac{25}{986} * sin(5t) \tag{10}$$

6. (a)

$$w[n] = x[n] + \frac{1}{2} w[n-1]$$

Choose the impulse as input

$$x[n] = \delta[n]$$

to get inpulse response:

$$h_0[n] = \delta[n] + \frac{1}{2} h_0[n-1]$$

This recursive function can be solved using the fact that the system is initially at rest.

$$h_0[n] = 0 \text{ ,} n < 0$$

$$h_0[0] = \delta[0] + \frac{1}{2} h_0[-1] = 1 + 0 = 1$$

$$h_0[1] = \delta[1] + \frac{1}{2} h_0[0] = 0 + \frac{1}{2} = \frac{1}{2}$$

$$h_0[2] = \delta[2] + \frac{1}{2} h_0[1] = 0 + \frac{1}{2^2} = \frac{1}{2^2}$$

$$\vdots$$

$$h_0[n] = \frac{1}{2^n} u[n] \tag{11}$$

(b) The overall impulse response is the convolution the two h_0 responses.

$$h[n] = h_0[n] * h_0[n]$$

$$h[n] = \sum_{k=-\infty}^{\infty} h_0[k] h_0[n-k]$$

$$h[n] = \sum_{k=-\infty}^{\infty} \frac{1}{2^k} u[k] \frac{1}{2^{n-k}} u[n-k]$$

$$h[n] = \sum_{k=0}^{n} \frac{1}{2^k} \times \frac{1}{2^{n-k}} = \sum_{k=0}^{n} \frac{1}{2^n}$$

$$h[n] = \frac{n+1}{2^n} * u[n] \tag{12}$$

(c)

$$w[n] - \frac{1}{2} w[n-1] = x[n]$$

$$y[n] - \frac{1}{2} y[n-1] = w[n]$$

$$y[n] - \frac{1}{2} y[n-1] - \frac{1}{2}(y[n-1] - \frac{1}{2} y[n-2]) = x[n]$$

$$y[n] - y[n-1] + \frac{1}{4} y[n-2] = x[n] \tag{13}$$

7. This is the python code to convolve two functions. It is used for both parts by adjusting titles.

```
import matplotlib.pyplot as plt

def convolution(x0,s0,x1,s1):

    dif = abs(s0)-abs(s1)
    if dif<0:
        for i in range(-dif):
            x0.insert(0,0)
    elif dif>0:
        for i in range(dif):
            x1.insert(0,0)


    #convolution operation
    n0 = len(x0)
    n1 = len(x1)
    conv = [0]*(n0+n1-1)
    for i in range(s0,s0+n0):
        for j in range(s1,s1+n1):
            conv[i+j-s0-s1] += x0[i-s0]*x1[j-s1]

    #plot the convolution
    fig1, (ax1, ax2,ax3) = plt.subplots(3, 1, figsize=(16, 8))
    fig1.subplots_adjust(hspace=0.5, wspace=0.5)
    fig1.set_size_inches(8.27, 5.5)
    ax1.stem(list(range(s0, s0+n0)),x0,markerfmt=' ')
    ax1.set_title("x[n]")
    ax2.stem(list(range(s1, s1+n1)),x1,markerfmt=' ')
    ax2.set_title("h[n]")
    newlist = list(range(min(s0,s1),min(s0,s1) + len(conv)))
    for q in range(len(newlist)):
        newlist[q] += s0
    ax3.stem(newlist,conv,markerfmt=' ')
    ax3.set_title("x[n] * h[n]")
    plt.xlabel('n')
    plt.ylabel('Amplitude')
    fig1.suptitle('Convolution of two functions')

    plt.show()




if __name__ == "__main__":
    with open("hw2_signal.csv", 'r') as f:
        data = f.read().split(",")
        s0 = int(data[0])
        #print(s0)
        data0 = [float(i) for i in data[1:]]

    #Part a
    data1 = [0.]*len(data0)
    data1[5-s0] = 1.
    s1=s0
    convolution(data0,s0,data1,s1)

    # #Part b
    h = lambda n, N: 1/N if (n>=0 and n<=N-1) else 0

    for N in [3, 5, 10 ,20]:
        data1 = [h(n,N) for n in range(s0,s0+len(data0))]
        convolution(data0,s0,data1,s0)
```

(a) As we can see in the Figure 1, convolving the input signal with $\delta[n-5]$ output the shifted version of the input signal. In general, we can conclude that convolving a signal with $\delta[n-k]$ delays the signal by k steps.

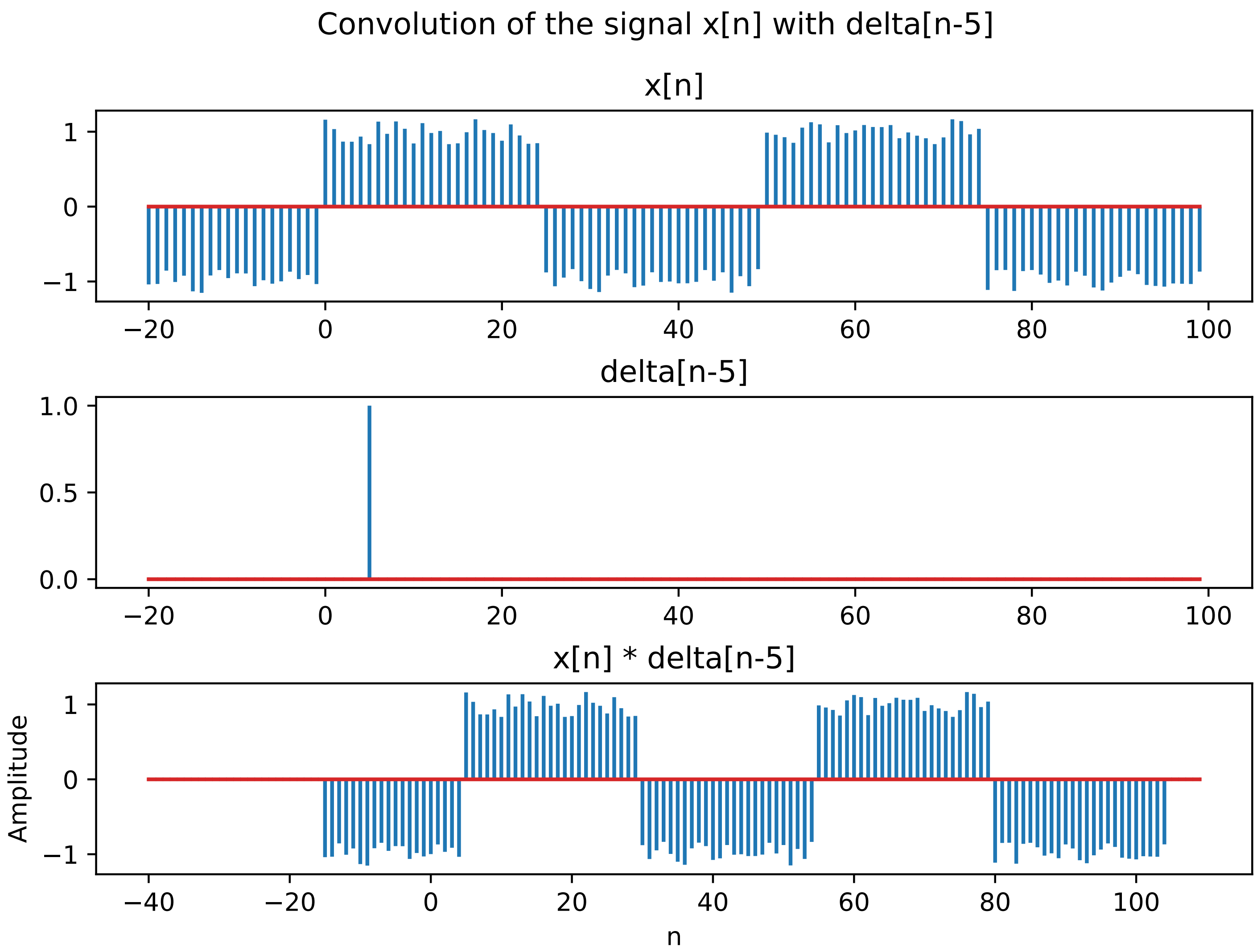

Figure 1: Convolution of the signal with $\delta[n-5]$

(b) As we can see in the Figure 2,3,4, the N-Point Moving Average filter keeps the n values of our data and takes the average of them and equates them to our current position. In this way, it makes the signal smoother. So x[n] = (x[n] + x[n - 1] + ... + x[n - N + 1]) /N

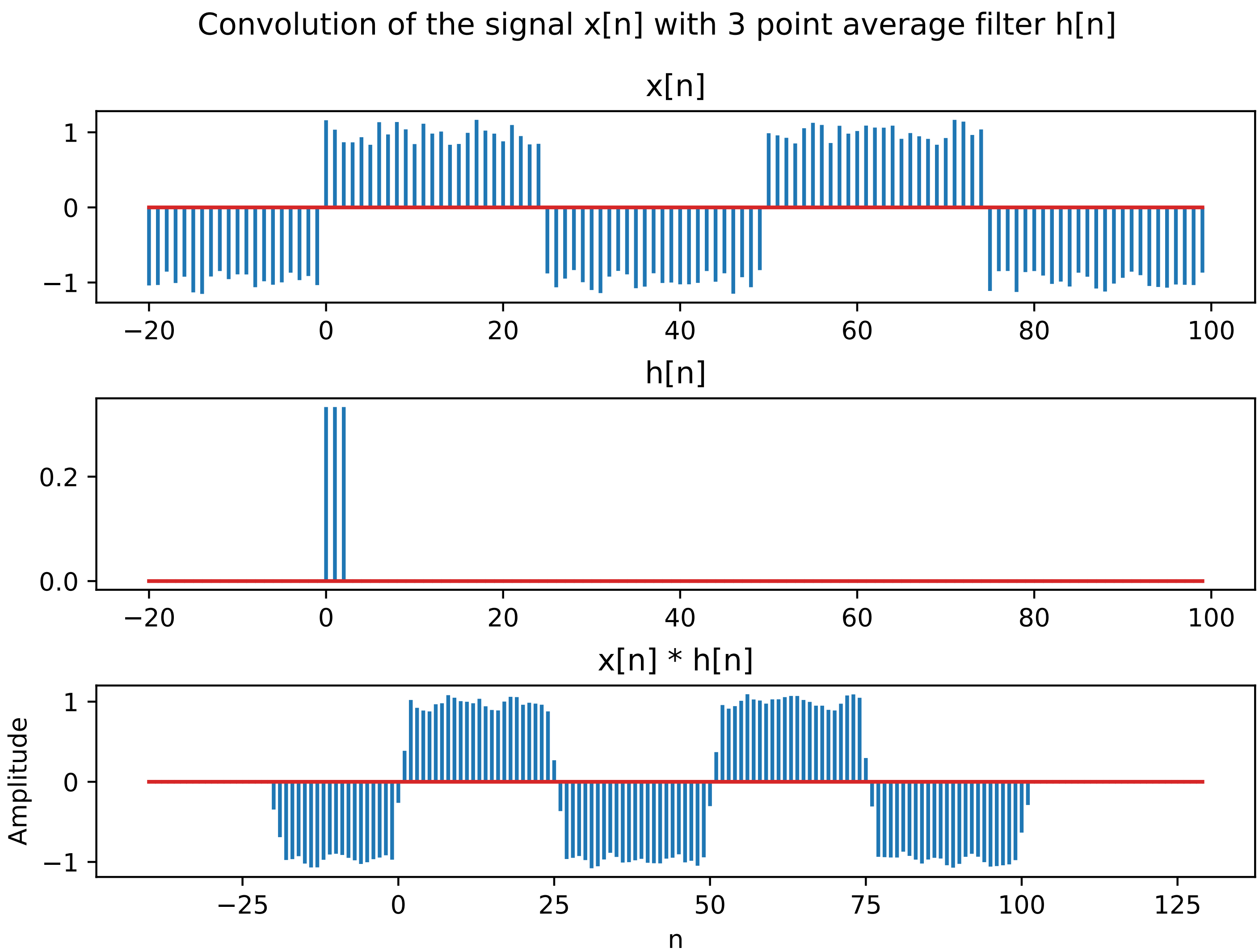

Figure 2: Convolution of the signal with 3-Point Moving Average Filter $h[n]$

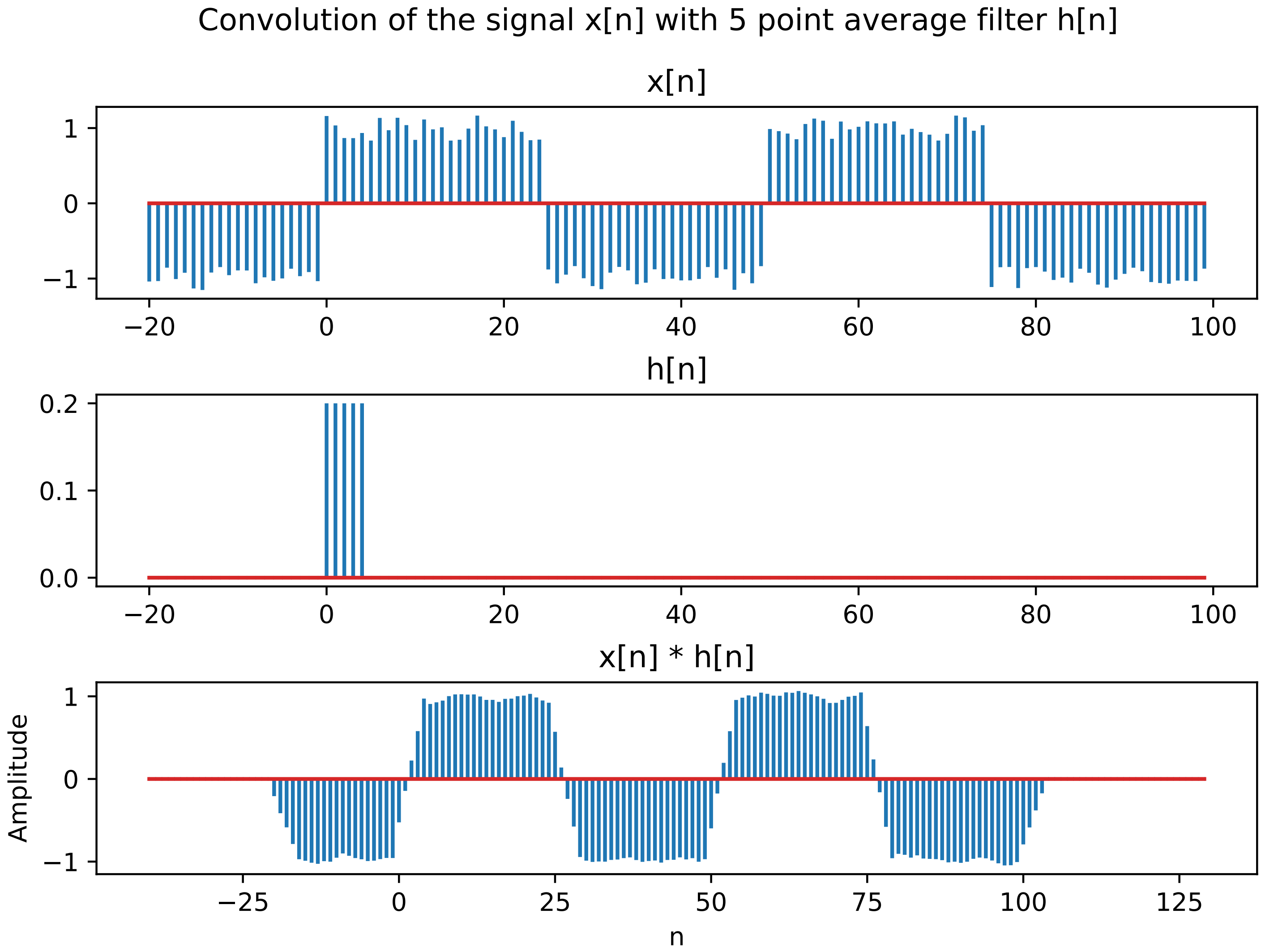

Figure 3: Convolution of the signal with 5-Point Moving Average Filter $h[n]$

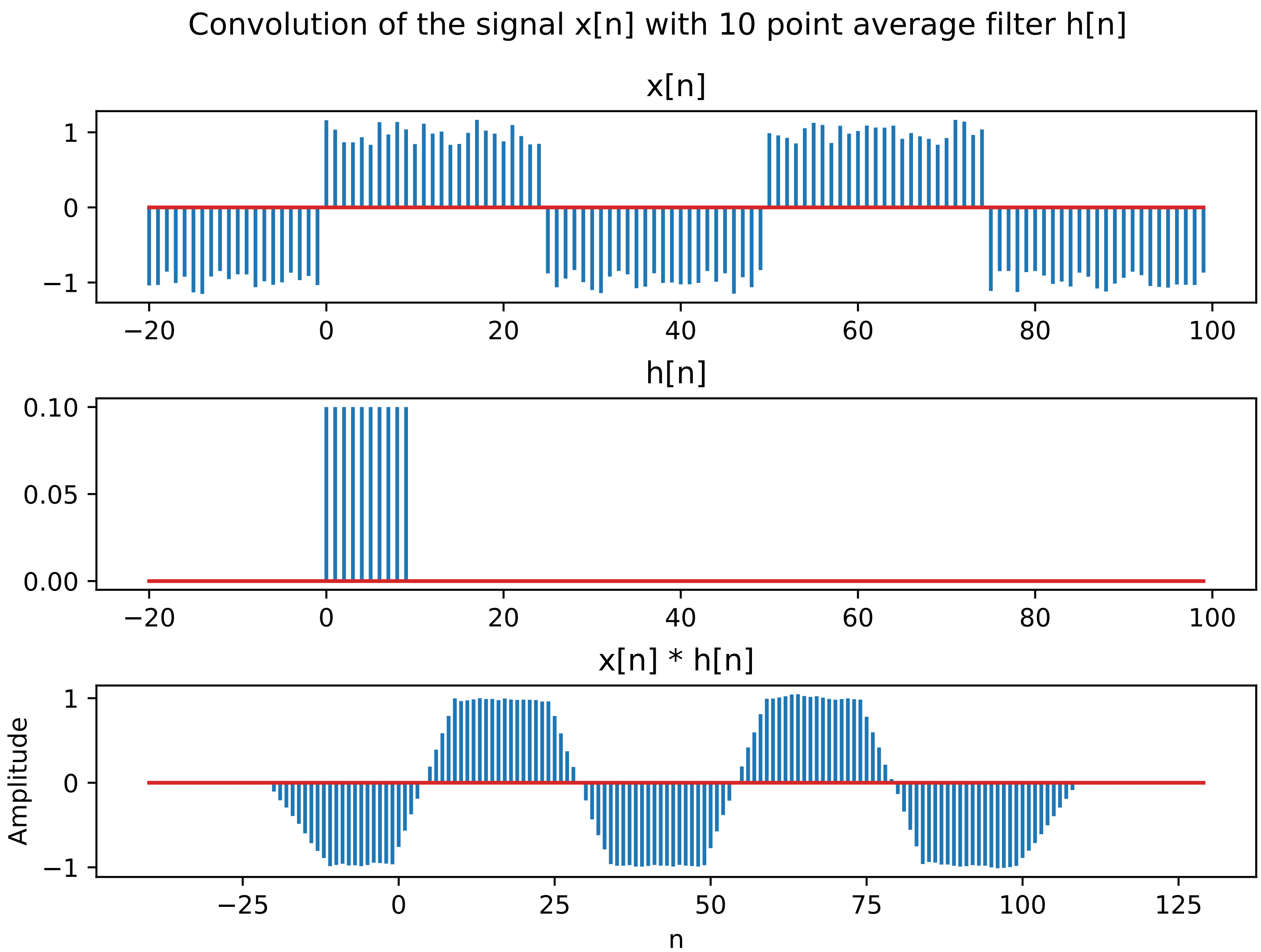

Figure 4: Convolution of the signal with 10-Point Moving Average Filter $h[n]$

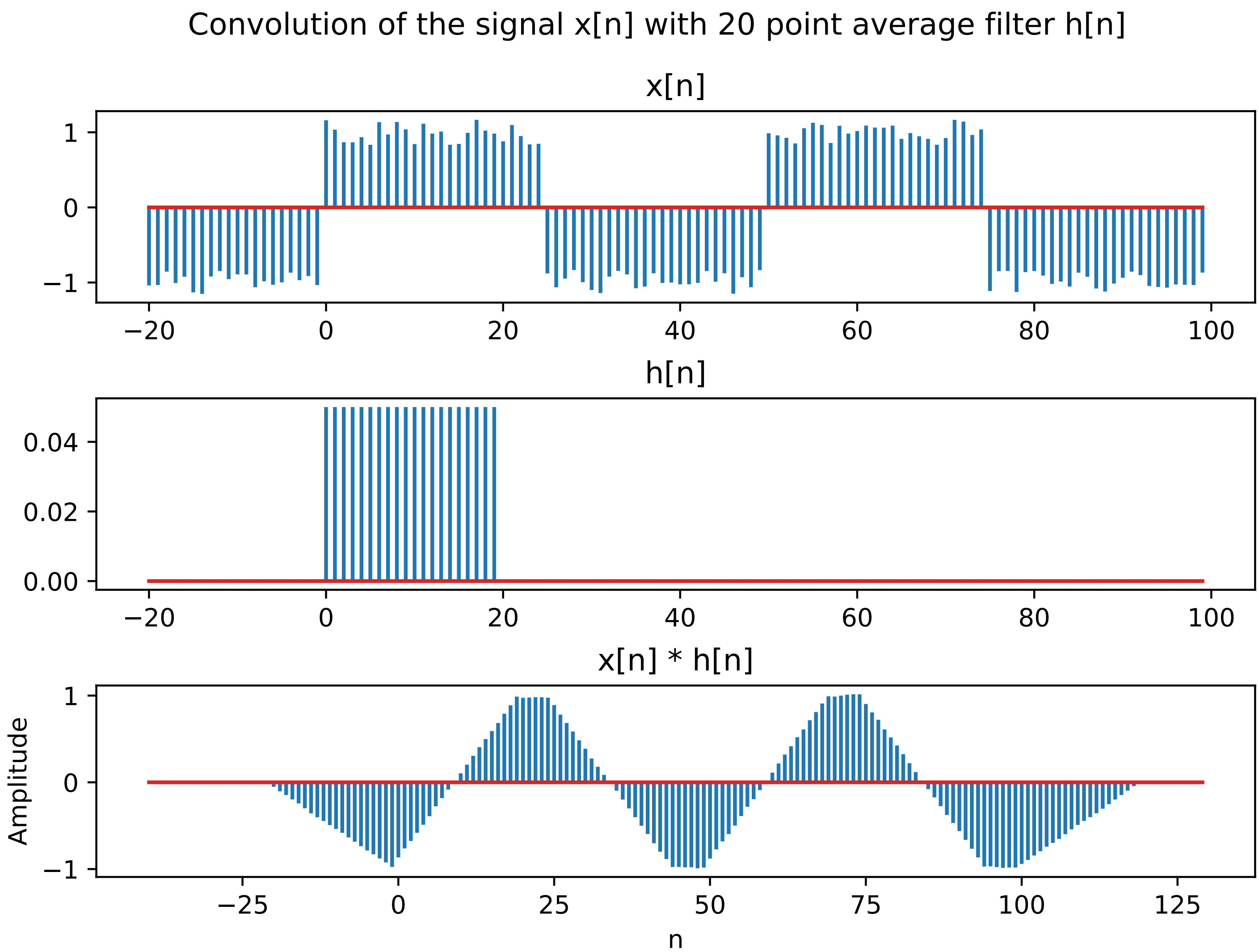

Figure 5: Convolution of the signal with 20-Point Moving Average Filter $h[n]$