Assignment 2

Batuhan Faik Derinbay derinbay18@itu.edu.tr
150180705

Question 1

i. (a) **Linearity**

$$y_1(t) = 5x_1^2(t) + 2x_1(t) + 4 \quad \wedge \quad y_2(t) = 5x_2^2(t) + 2x_2(t) + 4$$
$$y'(t) = ay_1(t) + by_2(t) = 5ax_1^2(t) + 2ax_1(t) + 4a + 5bx_2^2(t) + 2bx_2(t) + 4b$$
$$x(t) = (ax_1(t) + bx_2(t)) \to y''(t) = 5(ax_1(t) + bx_2(t)) + 2(ax_1(t) + bx_2(t)) + 4$$
$$y'(t) \neq y''(t) \Rightarrow \textit{Not Linear}$$

(b) **Time Variance**

$$x(t - t_0) \mapsto y(t - t_0) \Rightarrow y(t - t_0) = 5x^2(t - t_0) + 2x(t - t_0) + 4$$
$$t = t - t_0 \to y'(t) \Rightarrow y'(t - t_0) = 5x^2(t - t_0) + 2x(t - t_0) + 4$$
$$y(t - t_0) = y'(t - t_0) \Rightarrow \textit{Time Invariant}$$

(c) **Causality**

$$t = t_0 \to y(t) \Rightarrow y(t_0) = 5x^2(t_0) + 2x(t_0) + 4$$

For the present time $t = t_0$, $y(t_0)$ only depends on the values of the input signal $x(\tau)$ where $\tau \leq t_0$. Therefore the system is *causal*.

(d) **Stability**

$$x(t) = u(t) \mapsto y(t) = 5u^2(t) + 2u(t) + 4 = 5u(t) + 2u(t) + 4 = 7u(t) + 4$$

Given a bounded input $x(t) = u(t)$, the output $y(t)$ is also bounded. Therefore the system is *stable*.

ii. (a) **Linearity**

$$y_1(t) = \begin{cases} 0, & t < 0 \\ x_1(t) - x_1(t-4), & t \geq 0 \end{cases} \quad \wedge \quad y_2(t) = \begin{cases} 0, & t < 0 \\ x_2(t) - x_2(t-4), & t \geq 0 \end{cases}$$

$$y'(t) = ay_1(t) + by_2(t) = \begin{cases} 0, & t < 0 \\ ax_1(t) - ax_1(t-4) + bx_2(t) - bx_2(t-4), & t \geq 0 \end{cases}$$

$$(ax_1(t) + bx_2(t)) \mapsto y''(t) = \begin{cases} 0, & t < 0 \\ ax_1(t) + bx_2(t) - ax_1(t-4) - bx_2(t-4), & t \geq 0 \end{cases}$$

$$y'(t) = y''(t) \Rightarrow \textit{Linear}$$

(b) **Time Variance**

A split system with conditions dependent on the time variable t is a *time variant* system.

(c) **Causality**

Since this is an linear time-invariant (LTI) system, we can obtain impulse response $h(t)$ by simply supplying the function $\delta(t)$ as the input to the system. We can then easily determine causality of the system as follows:

$$h(t) = \begin{cases} 0, & t < 0 \\ \delta(t) - \delta(t-4), & t \geq 0 \end{cases}$$

$$\forall \tau < 0, \quad h(\tau) = 0 \Rightarrow \textit{Causal}$$

(d) **Stability**

$$\int_{-\infty}^{\infty} |h(\tau)| \mathrm{d}\tau = \int_{-\infty}^{0} |h(\tau)| \mathrm{d}\tau + \int_{0}^{\infty} |h(\tau)| \mathrm{d}\tau = 0 + \int_{0}^{\infty} \delta(\tau) - \delta(\tau - 4) \mathrm{d}\tau$$

$$= u(t) - u(t-4) < \infty \Rightarrow \textit{Stable}$$

iii. (a) **Linearity**

$$y_1[n] = \sum_{n+2}^{m=n-2} x_1[m] - 2|x_1[n]| \quad \wedge \quad y_2[n] = \sum_{n+2}^{m=n-2} x_2[m] - 2|x_2[n]|$$

$$y'[n] = ay_1[n] + by_2[n] = \sum_{n+2}^{m=n-2} ax_1[m] - 2a|x_1[n]| + \sum_{n+2}^{m=n-2} bx_2[m] - 2b|x_2[n]|$$

$$x[n] = (ax_1[n] + bx_2[n]) \rightarrow y''[n] = \sum_{n+2}^{m=n-2} (ax_1[m] + bx_2[m]) - 2|(ax_1[n] + bx_2[n])|$$

$$y'[n] \neq y''[n] \text{due to absolute value} \Rightarrow \textit{Not Linear}$$

(b) **Time Variance**

$$x[n - n_0] \mapsto y[n - n_0] \Rightarrow y[n - n_0] = \sum_{n-n_0+2}^{m=n-n_0-2} x[m] - 2|x[n - n_0]|$$

$$n = n - n_0 \rightarrow y'[n] \Rightarrow y'[n - n_0] = \sum_{n-n_0+2}^{m=n-n_0-2} x[m] - 2|x[n - n_0]|$$

$$y[n - n_0] = y'[n - n_0] \Rightarrow \textit{Time Invariant}$$

(c) **Causality**

$$y[n_0] = x[n_0 - 2] + x[n_0 - 1] + x[n_0] + x[n_0 + 1] + x[n_0 + 2] - 2|x[n_0]|$$

For the present time $n = n_0$, $y[n_0]$ doesn't only depend on the values of the input signal $x(\tau)$ where $\tau \leq t_0$. Therefore the system is *not causal.*

(d) **Stability**

Given a finite (bounded) input $x[n]$ = finite, the output yields $y[n] = -2 *$ finite $+ \sum$ finite = finite. Since response of the system of a bounded input is also bounded, the system is *stable.*

iv. (a) **Linearity**

$$y_1[n] + 5y_1[n-1] + 9y_1[n-2] + 5y_1[n-3] + y_1[n-4] = 2x_1[n] + 4x_1[n-1] + 2x_1[n-2]$$
$$\wedge$$
$$y_2[n] + 5y_2[n-1] + 9y_2[n-2] + 5y_2[n-3] + y_2[n-4] = 2x_2[n] + 4x_2[n-1] + 2x_2[n-2]$$
$$\begin{aligned} y'[n] &= ay_1[n] + by_2[n] \\ &= 2ax_1[n] + 4ax_1[n-1] + 2ax_1[n-2] - 5ay_1[n-1] - 9ay_1[n-2] \\ &- 5ay_1[n-3] - ay_1[n-4] + 2bx_2[n] + 4bx_2[n-1] + 4bx_2[n-2] \\ &- 5by_2[n-1] - 9by_2[n-2] - 5by_2[n-3] - by_2[n-4] \end{aligned}$$

(b) **Time Variance**

$$x[n-n_0] \mapsto y[n-n_0] \Rightarrow y[n-n_0]$$
$$\begin{aligned} y[n-n_0] = 2x[n-n_0] + 4x[n-n_0-1] + 2x[n-n_0-2] - 5y[n-n_0-1] \\ - 9y[n-n_0-2] - 5y[n-n_0-3] - y[n-n_0-4] \end{aligned}$$

$$n = n - n_0 \to y'[n] \Rightarrow y'[n-n_0]$$
$$\begin{aligned} y'[n-n_0] = 2x[n-n_0] + 4x[n-n_0-1] + 2x[n-n_0-2] - 5y[n-n_0-1] \\ - 9y[n-n_0-2] - 5y[n-n_0-3] - y[n-n_0-4] \end{aligned}$$

$$y[n-n_0] = y'[n-n_0] \Rightarrow \textit{Time Invariant}$$

(c) **Causality**

$$y[n_0] = 2x[n_0] + 4x[n_0-1] + 2x[n_0-2] - 5y[n_0-1] - 9y[n_0-2] - 5y[n_0-3] - y[n_0-4]$$

For the present time $n = n_0$, $y[n_0]$ only depends on the values of the input signal $x(\tau)$ where $\tau \leq t_0$. Therefore the system is *causal.*

(d) **Stability**

To check the stability of the system, let's use the z-transform.

$$y[n] + 5y[n-1] + 9y[n-2] + 5y[n-3] + y[n-4] = 2x[n] + 4x[n-1] + 2x[n-2]$$
$$h[n] + 5h[n-1] + 9h[n-2] + 5h[n-3] + h[n-4] = 2\delta[n] + 4\delta[n-1] + 2\delta[n-2]$$
$$H(z) + 5z^-1H(z) + 9z^-2H(z) + 5z^-3H(z) + z^-4H(z) = 2 + 4z^-1 + 2z^-2$$
$$H(z) = \frac{2 + 4z^-1 + 2z^-2}{1 + 5z^-1 + 9z^-2 + 5z^-3 + z^-4} = \frac{2z^4 + 4z^3 + 2z^2}{z^4 + 5z^3 + 9z^2 + 5z + 1}$$
$$\text{Poles: } z^4 + 5z^3 + 9z^2 + 5z + 1 = 0 \Rightarrow \exists i = 1, 2, 3, 4, \quad ||z_i|| > 1$$

Since some poles z_i for some (two to be exact) values of i have their norms greater than unit circle, this causal system is *not stable.*

Question 2

(a) i Given signals $x(t)$ and $z(t)$ can be written as follows.

$$x(t) = \begin{cases} 1, & 0 \leq t < 1 \\ -1, & 1 \leq t < 2 \\ 0, & else \end{cases} \quad \wedge \quad z(t) = \begin{cases} t, & -1 \leq t < 1 \\ 0, & else \end{cases}$$

In order to find $y_1(t) = x(t) * z(t)$, convolution operation is applied to signals $x(t)$ and $z(t)$.

$$y_1(t) = x(t) * z(t) = \int_{-\infty}^{\infty} x(t-\tau) z(\tau) \mathrm{d}\tau$$

First $x(t-\tau)$ is obtained by first applying time reversal (P1), then time shifting (P2), represented in Figure 6.

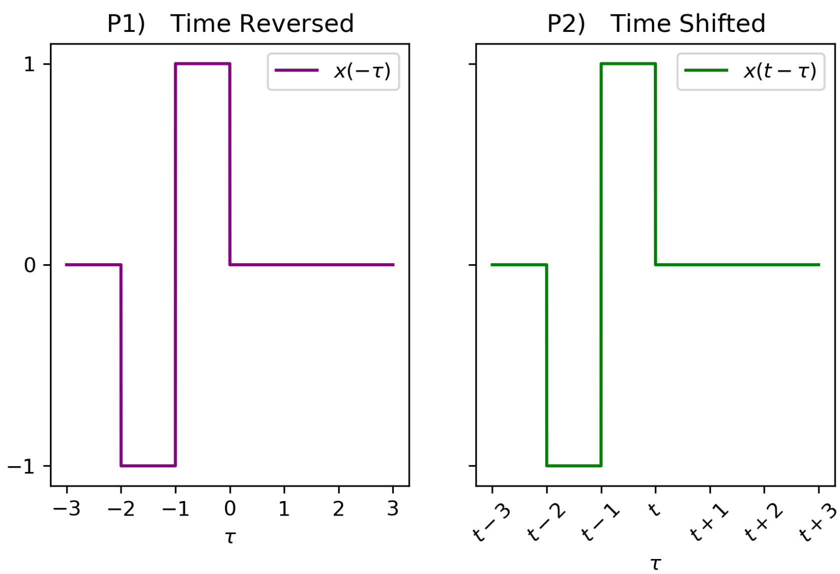


Figure 1: Time Reversal and Time Shift Operations on Signal x

Afterwards, using the graphical method, operation $\int_{-\infty}^{\infty} x(t-\tau)z(\tau)\mathrm{d}\tau$ is carried out for all valid values of t in six cases (Figure 2).

- Case 1 (S1): $-\infty < t < -1$
  Signals $x(t-\tau)$ and $z(\tau)$ don't overlap. Therefore, $\int_{-\infty}^{-1} x(t-\tau)z(\tau)\mathrm{d}\tau = 0$
- Case 2 (S2): $-1 \leq t < 0$

$$\int_{-1}^{t} x(t-\tau)z(\tau)\mathrm{d}\tau = \int_{-1}^{t} \tau \mathrm{d}\tau = \left.\frac{\tau^2}{2}\right|_{-1}^{t} = \frac{t^2-1}{2}$$

- Case 3 (S3): $0 \leq t < 1$

$$\int_{-1}^{t-1} x(t-\tau)z(\tau)\mathrm{d}\tau + \int_{t-1}^{t} x(t-\tau)z(\tau)\mathrm{d}\tau = \int_{-1}^{t-1} -\tau\mathrm{d}\tau + \int_{t-1}^{t} \tau\mathrm{d}\tau$$
$$= -\frac{\tau^2}{2}\bigg|_{-1}^{t-1} + \frac{\tau^2}{2}\bigg|_{t-1}^{t} = \frac{-t^2+4t-1}{2}$$

- Case 4 (S4): $1 \leq t < 2$

$$\int_{t-2}^{t-1} x(t-\tau)z(\tau)\mathrm{d}\tau + \int_{t-1}^{1} x(t-\tau)z(\tau)\mathrm{d}\tau = \int_{t-2}^{t-1} -\tau\mathrm{d}\tau + \int_{t-1}^{1} \tau\mathrm{d}\tau$$
$$= -\frac{\tau^2}{2}\bigg|_{t-2}^{t-1} + \frac{\tau^2}{2}\bigg|_{t-1}^{1} = \frac{-t^2+3}{2}$$

- Case 5 (S5): $2 \leq t < 3$

$$\int_{t-1}^{1} x(t-\tau)z(\tau)\mathrm{d}\tau = \int_{t-2}^{1} -\tau\mathrm{d}\tau = -\frac{\tau^2}{2}\bigg|_{t-2}^{1} = \frac{t^2-4t+3}{2}$$

- Case 6 (S6): $3 < t < \infty$
  Signals $x(t-\tau)$ and $z(\tau)$ don't overlap. Therefore, $\int_3^\infty x(t-\tau)z(\tau)\mathrm{d}\tau = 0$

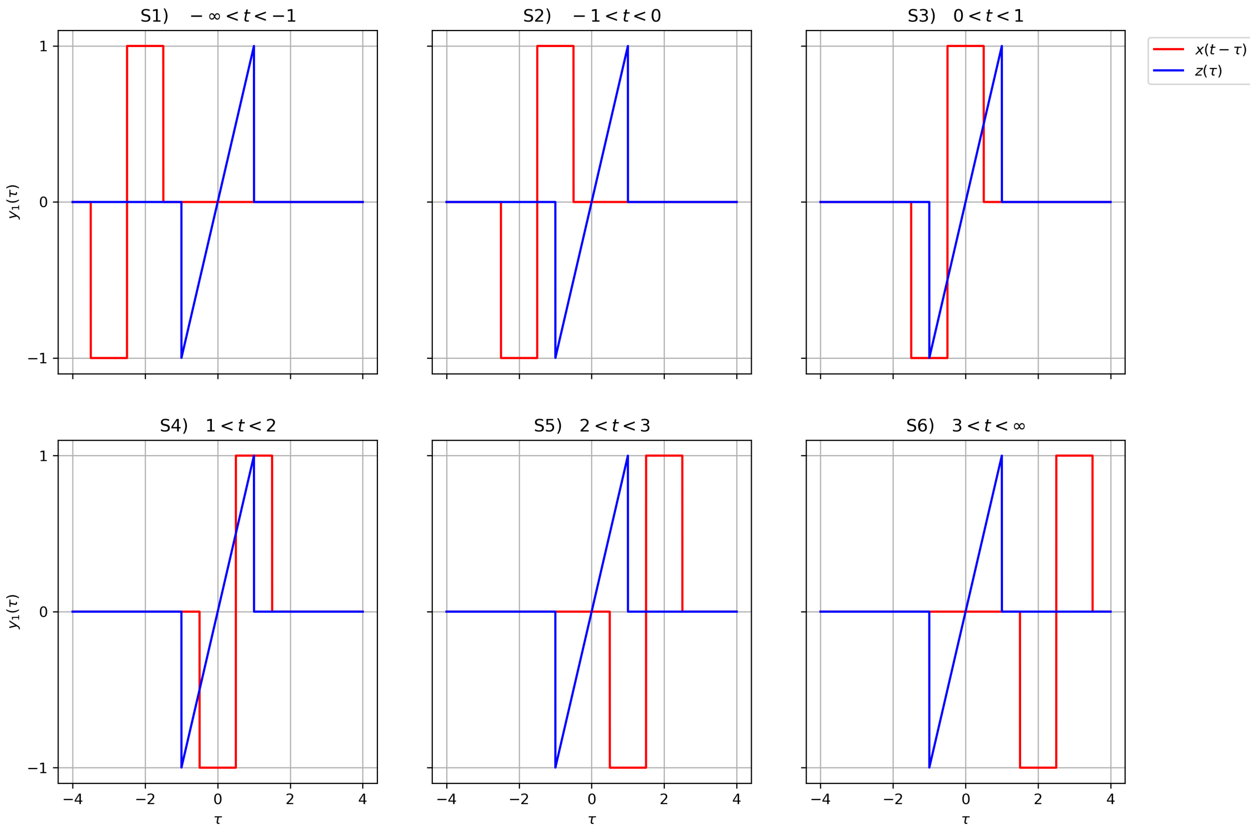
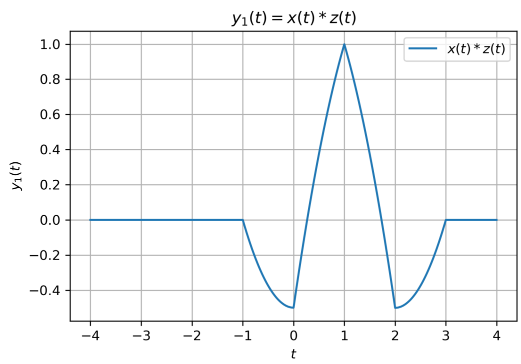


Figure 2: Convolution Steps of $x(t)$ and $z(t)$ Using Graphical Method

After finding the t dependent expressions of convolution for $-\infty < t < \infty$, $y_1(t) = x(t) * z(t)$ is obtained (Figure 3).



$$y_1(t) = \begin{cases} \frac{t^2-1}{2}, & -1 \le t < 0 \\ \frac{-t^2+4t-1}{2}, & 0 \le t < 1 \\ \frac{-t^2+3}{2}, & 1 \le t < 2 \\ \frac{t^2-4t+3}{2}, & 2 \le t < 3 \\ 0, & else \end{cases}$$

Figure 3: $y_1(t)$: Convolution of $x(t)$ and $z(t)$

ii Given signals $x(t)$ and $v(t)$ can be written as follows.

$$x(t) = \begin{cases} 1, & 0 \leq t < 1 \\ -1, & 1 \leq t < 2 \\ 0, & else \end{cases} \quad \wedge \quad v(t) = \begin{cases} e^t, & -1 \leq t < 0 \\ e^{-2t}, & 0 \leq t < 1 \\ 0, & else \end{cases}$$

In order to find $y_2(t) = x(t) * v(t)$, convolution operation is applied to signals $x(t)$ and $v(t)$.

$$y_2(t) = x(t) * v(t) = \int_{-\infty}^{\infty} x(t-\tau)v(\tau)\mathrm{d}\tau$$

First $x(t-\tau)$ is obtained by first applying time reversal, then time shifting. Represented previously in Figure 6.
Afterwards, using the graphical method, operation $\int_{-\infty}^{\infty} x(t-\tau)v(\tau)\mathrm{d}\tau$ is carried out for all valid values of t in six cases (Figure 4).

- Case 1 (S1): $-\infty < t < -1$
  Signals $x(t-\tau)$ and $v(\tau)$ don't overlap. Therefore, $\int_{-\infty}^{-1} x(t-\tau)v(\tau)\mathrm{d}\tau = 0$
- Case 2 (S2): $-1 \leq t < 0$

$$\int_{-1}^{t} x(t-\tau)v(\tau)\mathrm{d}\tau = \int_{-1}^{t} e^{2\tau}\mathrm{d}\tau = \frac{1}{2}e^{2\tau}\Big|_{-1}^{t} = \frac{1}{2}(e^{2t} - e^{-2})$$

- Case 3 (S3): $0 \leq t < 1$

$$\begin{aligned} &\int_{-1}^{t-1} x(t-\tau)v(\tau)\mathrm{d}\tau + \int_{t-1}^{0} x(t-\tau)v(\tau)\mathrm{d}\tau + \int_{0}^{t} x(t-\tau)v(\tau)\mathrm{d}\tau \\ &= \int_{-1}^{t-1} -e^{2\tau}\mathrm{d}\tau + \int_{t-1}^{0} e^{2\tau}\mathrm{d}\tau + \int_{0}^{t} e^{2\tau}\mathrm{d}\tau \\ &= -\frac{1}{2}(e^{2t-2} - e^{-2}) + \frac{1}{2}(1 - e^{2t-2}) - \frac{1}{2}(e^{-2t} - 1) \\ &= \frac{1}{2}(-e^{-2t} - 2e^{2t-2} + 2 + e^{-2}) \end{aligned}$$

- Case 4 (S4): $1 \leq t < 2$

$$\int_{t-2}^{0} x(t-\tau)v(\tau)\mathrm{d}\tau + \int_{0}^{t-1} x(t-\tau)v(\tau)\mathrm{d}\tau + \int_{t-1}^{1} x(t-\tau)v(\tau)\mathrm{d}\tau$$
$$= \int_{t-2}^{0} -e^{2\tau}\mathrm{d}\tau - \int_{0}^{t-1} e^{-2\tau}\mathrm{d}\tau + \int_{t-1}^{1} e^{-2\tau}\mathrm{d}\tau$$
$$= -\frac{1}{2}(1-e^{2t-4}) + \frac{1}{2}(e^{-2t+2}-1) - \frac{1}{2}(e^{-2}-e^{-2t+2})$$
$$= \frac{2e^{4-2t}+e^{2t-2}-2e^2-1}{2e^2}$$

- Case 5 (S5): $2 \leq t < 3$

$$\int_{t-2}^{1} x(t-\tau)v(\tau)\mathrm{d}\tau = \int_{t-2}^{1} -e^{-2\tau}\mathrm{d}\tau = \frac{1}{2}(e^{-2}-e^{-2t+4})$$

- Case 6 (S6): $3 < t < \infty$
  Signals $x(t-\tau)$ and $v(\tau)$ don't overlap. Therefore, $\int_3^\infty x(t-\tau)v(\tau)\mathrm{d}\tau = 0$

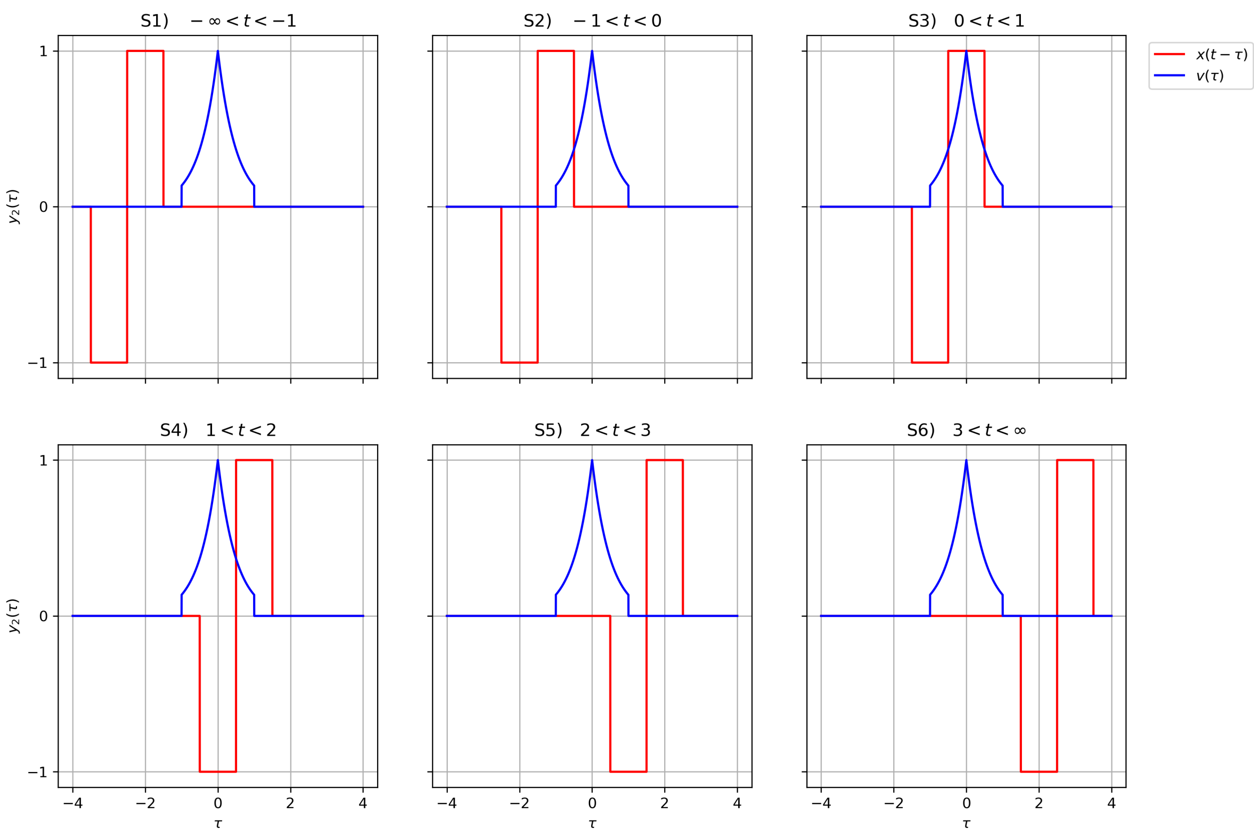
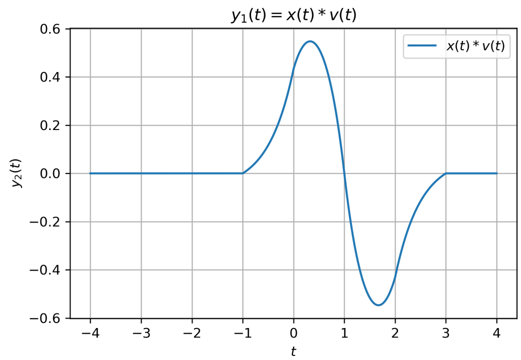


Figure 4: Convolution Steps of $x(t)$ and $v(t)$ Using Graphical Method

After finding the t dependent expressions of convolution for $-\infty < t < \infty$, $y_2(t) = x(t) * v(t)$ is obtained (Figure 5).



$$y_2(t) = \begin{cases} \frac{1}{2}(e^{2t} - e^{-2}), & -1 \leq t < 0 \\ \frac{-e^{-2t} - 2e^{2t-2} + 2 + e^{-2}}{2}, & 0 \leq t < 1 \\ \frac{2e^{4-2t} + e^{2t-2} - 2e^{2} - 1}{2e^{2}}, & 1 \leq t < 2 \\ \frac{1}{2}(e^{-2} - e^{-2t+4}), & 2 \leq t < 3 \\ 0, & else \end{cases}$$

Figure 5: $y_2(t)$: Convolution of $x(t)$ and $v(t)$

iii Given signal $w(t)$ can be written as follows.

$$w(t) = \begin{cases} 1 + t, & -1 \leq t < 0 \\ 1 - t, & 0 \leq t < 1 \\ 0, & else \end{cases}$$

In order to find $y_3(t) = w(t) * w(t)$, convolution operation is applied to signals $w(t)$ and $w(t)$.

$$y_3(t) = w(t) * w(t) = \int_{-\infty}^{\infty} w(t - \tau) w(\tau) \mathrm{d}\tau$$

First $w(t - \tau)$ is obtained by first applying time reversal (P1), then time shifting (P2), represented in Figure **??**.

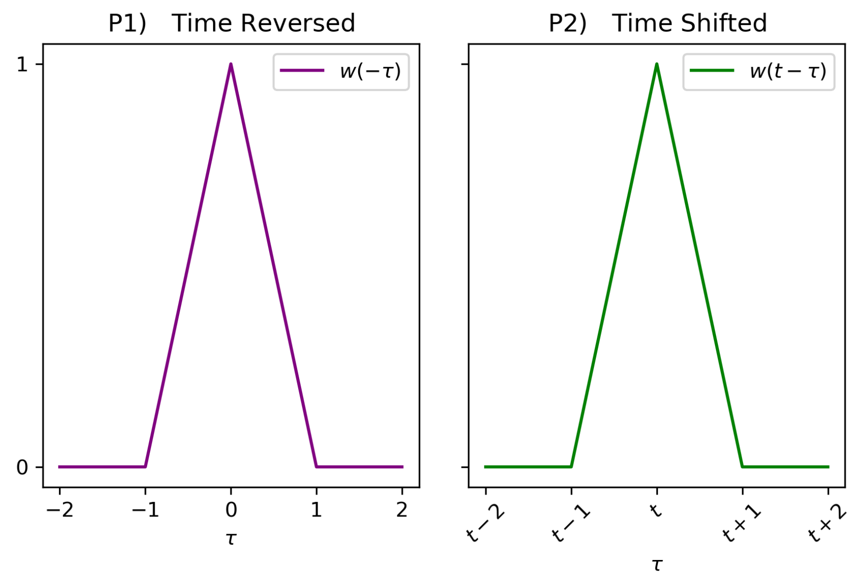


Figure 6: Time Reversal and Time Shift Operations on Signal w

Afterwards, using the graphical method, operation $\int_{-\infty}^{\infty} w(t-\tau)w(\tau)\mathrm{d}\tau$ is carried out for all valid values of t in six cases (Figure 7).

- Case 1 (S1): $-\infty < t < -2$
  Signals $w(t-\tau)$ and $w(\tau)$ don't overlap. Therefore, $\int_{-\infty}^{-2} x(t-\tau)z(\tau)\mathrm{d}\tau = 0$
- Case 2 (S2): $-2 \leq t < -1$

$$\int_{-1}^{t+1} w(t-\tau)w(\tau)\mathrm{d}\tau = \int_{-1}^{t+1} (1+\tau)(1-\tau+t)\mathrm{d}\tau = \frac{(t+2)^3}{6}$$

- Case 3 (S3): $-1 \leq t < 0$

$$\begin{aligned} &\int_{-1}^{t} w(t-\tau)w(\tau)\mathrm{d}\tau + \int_{t}^{0} w(t-\tau)w(\tau)\mathrm{d}\tau + \int_{0}^{t+1} w(t-\tau)w(\tau)\mathrm{d}\tau \\ &= \int_{-1}^{t} (1+\tau)(1+\tau-t)\mathrm{d}\tau + \int_{t}^{0} (1+\tau)(1-\tau+t)\mathrm{d}\tau + \int_{0}^{t+1} (1-\tau)(1-\tau+t)\mathrm{d}\tau \\ &= \frac{-t^3+3t+2}{3} - \frac{t^3+6t^2+6t}{6} = \frac{-3t^3-6t^2+4}{6} \end{aligned}$$

- Case 4 (S4): $0 \leq t < 1$

$$\int_{t-1}^{0} w(t-\tau)w(\tau)\mathrm{d}\tau + \int_{0}^{t} w(t-\tau)w(\tau)\mathrm{d}\tau + \int_{t}^{1} w(t-\tau)w(\tau)\mathrm{d}\tau$$
$$= \int_{t-1}^{0} (1+\tau)(1+\tau-t)\mathrm{d}\tau + \int_{0}^{t} (1-\tau)(1+\tau-t)\mathrm{d}\tau + \int_{t}^{1} (1-\tau)(1-\tau+t)\mathrm{d}\tau$$
$$= \frac{t^3-3t+2}{3} + \frac{t^3-6t^2+6t}{6} = \frac{3t^3-6t^2+4}{6}$$

- Case 5 (S5): $1 \leq t < 2$

$$\int_{t-1}^{1} w(t-\tau)w(\tau)\mathrm{d}\tau = \int_{t-1}^{1} (1-\tau)(1+\tau-t)\mathrm{d}\tau = \frac{-(t-2)^3}{6}$$

- Case 6 (S6): $2 < t < \infty$
  Signals $w(t-\tau)$ and $w(\tau)$ don't overlap. Therefore, $\int_2^\infty w(t-\tau)w(\tau)\mathrm{d}\tau = 0$

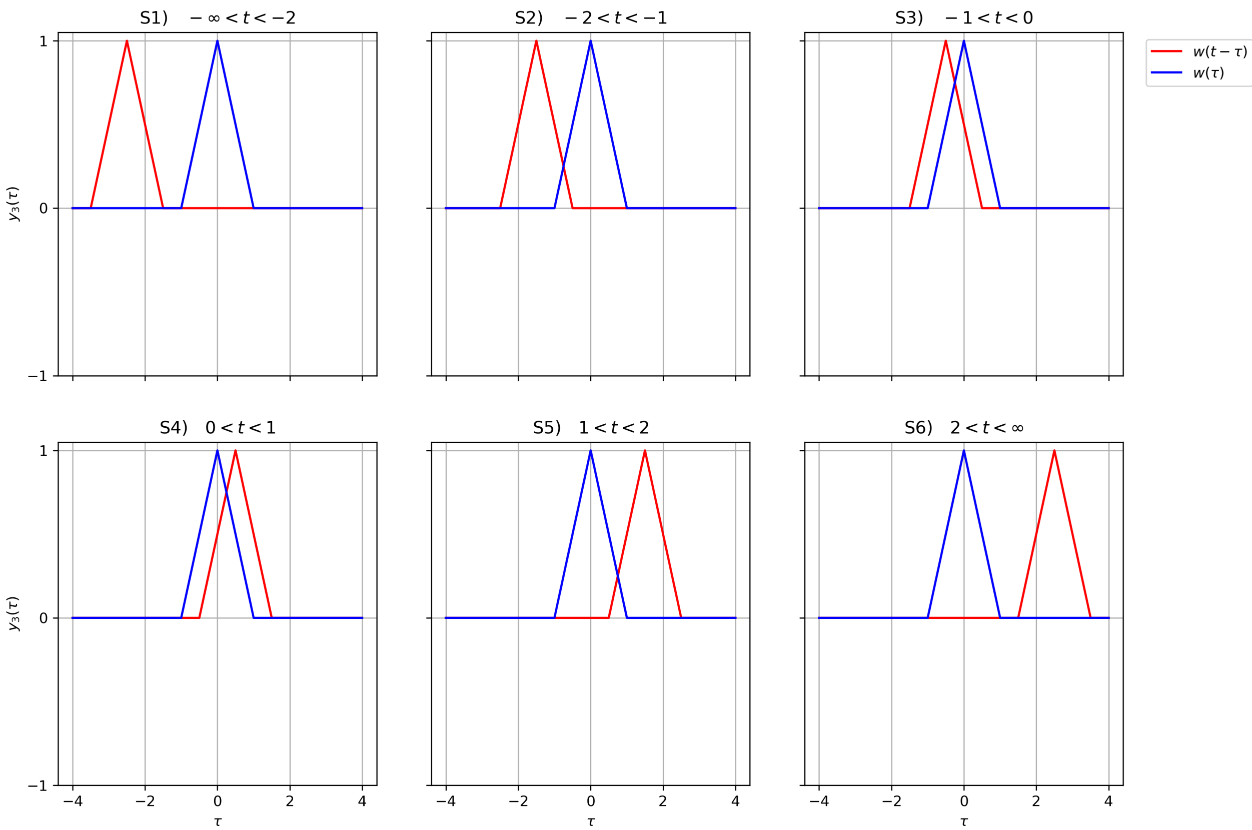
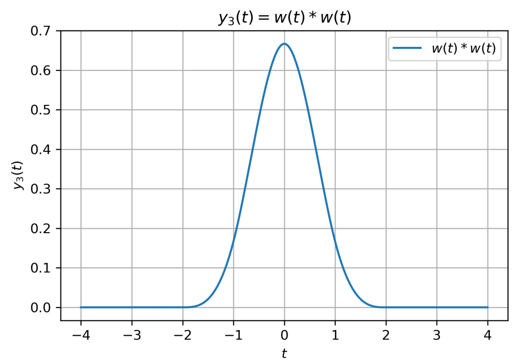


Figure 7: Convolution Steps of $w(t)$ and $w(t)$ Using Graphical Method

After finding the t dependent expressions of convolution for $-\infty < t < \infty$, $y_3(t) = w(t) * w(t)$ is obtained (Figure 8).



$$y_3(t) = \begin{cases} \frac{(t+2)^3}{6}, & -2 \leq t < -1 \\ \frac{-3t^3-6t^2+4}{6}, & -1 \leq t < 0 \\ \frac{3t^3-6t^2+4}{6}, & 0 \leq t < 1 \\ \frac{-(t-2)^3}{6}, & 1 \leq t < 2 \\ 0, & else \end{cases}$$

Figure 8: $y_3(t)$: Convolution of $w(t)$ and $w(t)$

(b) i In order to find $y_1[n] = x[n] * z[n]$, convolution operation is applied to signals $x[n]$ and $z[n]$.

$$y_1[n] = x[n] * z[n] = \sum_{k=-\infty}^{\infty} x[n-k]z[k]$$

First $x[n-k]$ is obtained by first applying time reversal ($x[-k]$), then time shifting ($x[n-k]$), represented in Figure 9.

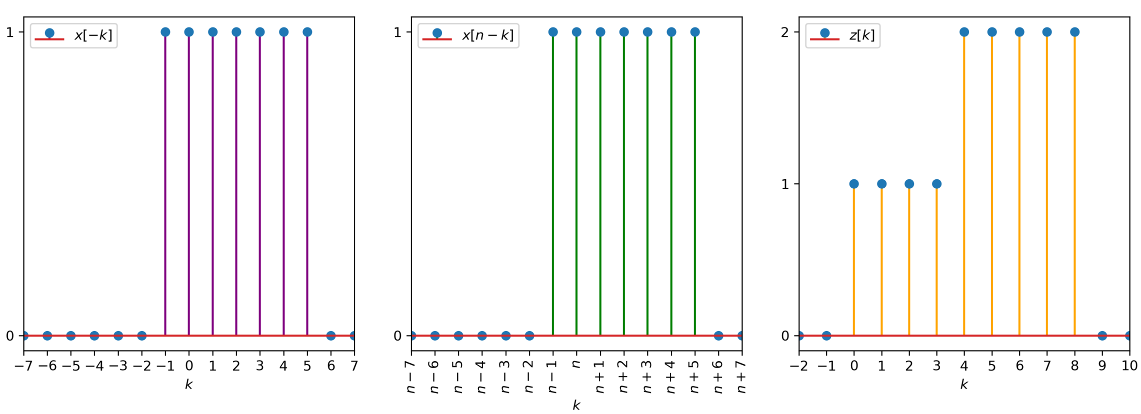


Figure 9: Time Reversal and Time Shift Operations on Signal x and the z Signal

Afterwards, using the graphical method, operation $\sum_{k=-\infty}^{\infty} x[n-k]z[k]$ is carried out for all valid values of n. In this case the result of the convolution operation is always 0 except for $-5 \leq n < 10$. So the convolution operation is performed between this range and $y_1[n]$ is found (Figure 10).

$$
\begin{aligned}
y_1[n] = {} & \delta[n+5] + 2\delta[n+4] + 3\delta[n+3] + 4\delta[n+2] + 6\delta[n+1] + 8\delta[n] \\
& + 10\delta[n-1] + 11\delta[n-2] + 12\delta[n-3] + 11\delta[n-4] + 10\delta[n-5] \\
& + 8\delta[n-6] + 6\delta[n-7] + 4\delta[n-8] + 2\delta[n-9]
\end{aligned}
$$

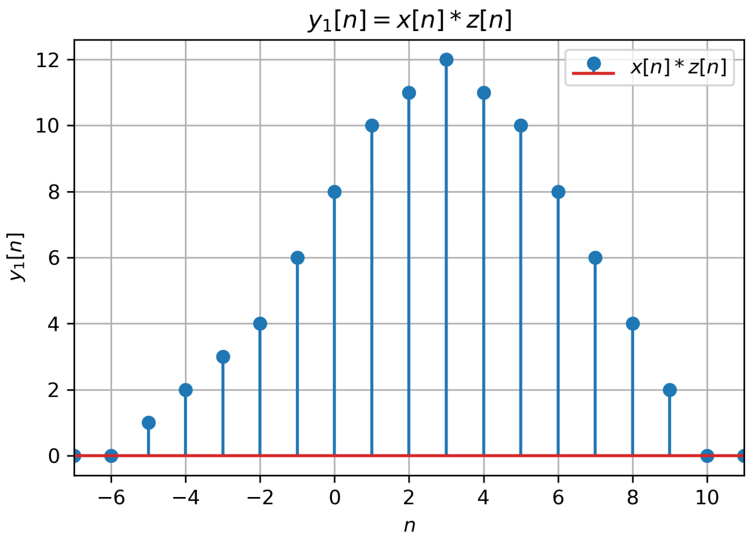


Figure 10: $y_1[n]$: Convolution of $x[n]$ and $z[n]$

ii In order to find $y_2[n] = w[n] * x[n]$, convolution operation is applied to signals $w[n]$ and $x[n]$.

$$
y_2[n] = w[n] * x[n] = \sum_{k=-\infty}^{\infty} x[n-k]w[k]
$$

First $x[n-k]$ is obtained by first applying time reversal ($x[-k]$), then time shifting ($x[n-k]$), represented in Figure 11.

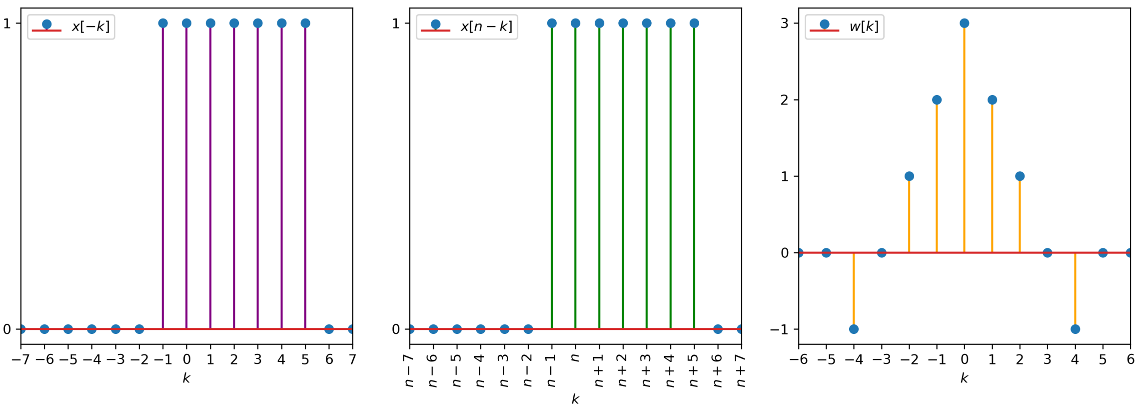


Figure 11: Time Reversal and Time Shift Operations on Signal x and the w Signal

Afterwards, using the graphical method, operation $\sum_{k=-\infty}^{\infty} x[n-k]w[k]$ is carried out for all valid values of n. In this case the result of the convolution operation is always 0 except for $-9 \leq n < 6$. So the convolution operation is performed between this range and $y_2[n]$ is found (Figure 12).

$$
\begin{aligned}
y_2[n] = &-\delta[n+9] - \delta[n+8] + 2\delta[n+6] + 5\delta[n+5] \\
&+ 7\delta[n+4] + 8\delta[n+3] + 9\delta[n+2] + 8\delta[n+1] + 7\delta[n] \\
&+ 5\delta[n-1] + 2\delta[n-2] - \delta[n-4] - \delta[n-5]
\end{aligned}
$$

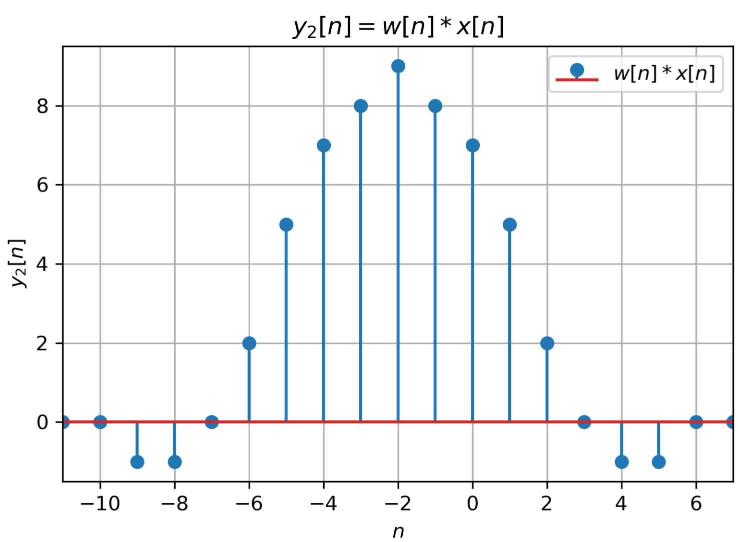


Figure 12: $y_2[n]$: Convolution of $w[n]$ and $x[n]$

iii In order to find $y_3[n] = w[n] * z[n]$, convolution operation is applied to signals $w[n]$ and $z[n]$.

$$y_3[n] = w[n] * z[n] = \sum_{k=-\infty}^{\infty} z[n-k]w[k]$$

First $z[n-k]$ is obtained by first applying time reversal ($z[-k]$), then time shifting ($z[n-k]$), represented in Figure 13.

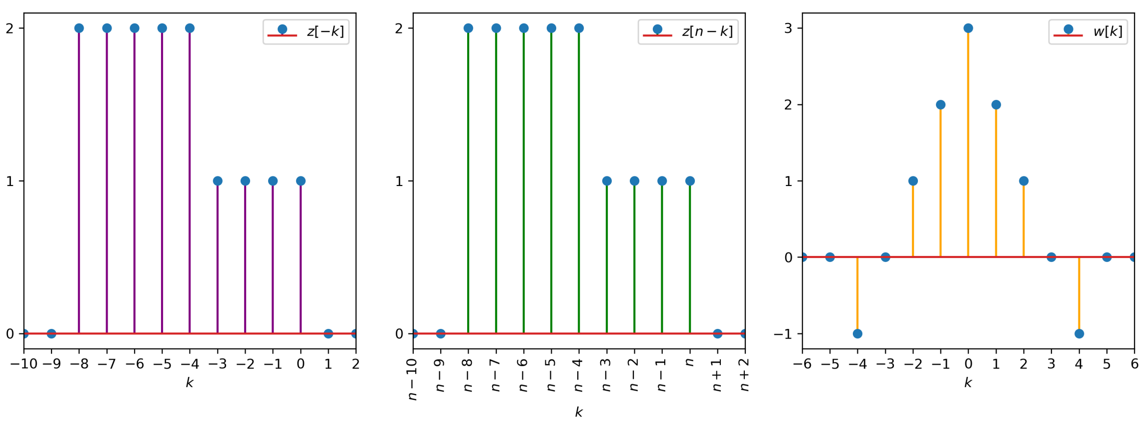


Figure 13: Time Reversal and Time Shift Operations on Signal z and the w Signal

Afterwards, using the graphical method, operation $\sum_{k=-\infty}^{\infty} z[n-k]w[k]$ is carried out for all valid values of n. In this case the result of the convolution operation is always 0 except for $-4 \leq n < 13$. So the convolution operation is performed between this range and $y_3[n]$ is found (Figure 14).

$$\begin{aligned} y_3[n] = &-\delta[n+4] - \delta[n+3] + 2\delta[n+1] + 4\delta[n] + 6\delta[n-1] \\ &+ 8\delta[n-2] + 10\delta[n-3] + 12\delta[n-4] + 16\delta[n-5] + 17\delta[n-6] \\ &+ 15\delta[n-7] + 10\delta[n-8] + 4\delta[n-9] - 4\delta[n-11] - 4\delta[n-12] \end{aligned}$$

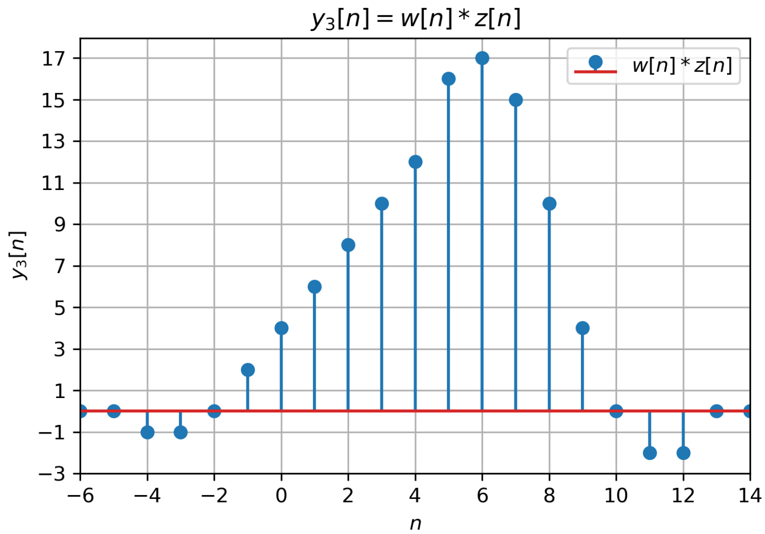


Figure 14: $y_3[n]$: Convolution of $w[n]$ and $z[n]$

Question 3

i

$$
\begin{aligned}
y_1[n] &= (u[n+10] - 2u[n] + u[n-4]) * u[n-2] \\
&= (u[n+10] - u[n] - (u[n] - u[n-4]) * u[n-2]
\end{aligned}
$$

$$
u[n-a] = \sum_{k=a}^{\infty} \delta[n-m] \Rightarrow u[n-a] - u[n-b] = \sum_{k=a}^{b+1} \delta[n-k]
$$

$$
\begin{aligned}
y_1[n] &= \left(\sum_{k=-10}^{-1} \delta[n-k] - \sum_{k=0}^{3} \delta[n-k] \right) * u[n-2] \\
&= \sum_{k=-10}^{-1} \delta[n-k] * u[n-2] - \sum_{k=0}^{3} \delta[n-k] * u[n-2] \\
&= \sum_{k=-10}^{-1} \delta[n-k-2] - \sum_{k=0}^{3} \delta[n-k-2]
\end{aligned}
$$

ii

$$y_2[n] = \sum_{k=-\infty}^{\infty} cos(\frac{\pi}{2}k)\left(\left(\frac{1}{2}\right)^{n-k} u[n-k-2]\right) = \sum_{k=-\infty}^{n-2} cos(\frac{\pi}{2}k)\left(\frac{1}{2}\right)^{n-k}$$

replacing k with $-p$

$$y_2[n] = \sum_{p=-n+2}^{\infty} cos(\frac{\pi}{2}p)\left(\frac{1}{2}\right)^{n+p}$$

$$cos(\frac{\pi}{2}p) = \begin{cases} 1, & p \bmod 4 = 1 \\ -1, & p \bmod 4 = 3 \\ 0, & p \bmod 2 = 0 \end{cases} \Rightarrow y_2[n] = \begin{cases} \sum_{p=-n+2}^{\infty}(\frac{1}{2})^{n+p}, & p \bmod 4 = 1 \\ -\sum_{p=-n+2}^{\infty}(\frac{1}{2})^{n+p}, & p \bmod 4 = 3 \\ 0, & p \bmod 2 = 0 \end{cases}$$

using the properties of sum of a geometric sequence, it is known that

$$\sum_{k=0}^{\infty} ar^k = \frac{a}{1-r}$$

therefore, it can be concluded that

$$y_2[n] = \begin{cases} -\frac{1}{5}, & n \bmod 4 = 0 \\ -\frac{1}{10}, & n \bmod 4 = 1 \\ \frac{1}{5}, & n \bmod 4 = 2 \\ \frac{1}{10}, & n \bmod 4 = 3 \end{cases}$$

iii

$$\begin{aligned} y_3(t) &= (2\delta(t+1) + \delta(t-5)) * u(t-1) \\ &= 2\delta(t+1) * u(t-1) + \delta(t-5) * u(t-1) \\ &= 2u(t) + u(t-6) \\ &= \begin{cases} 2, & 0 \le t < 6 \\ 3, & t \ge 6 \\ 0, & else \end{cases} \end{aligned}$$

iv

$$
\begin{aligned}
y_4(t) = u(t) * h(t) &= \int_{-\infty}^{\infty} u(t-\tau)h(\tau)\mathrm{d}\tau \neq 0, \forall t-\tau > 0 \\
&= \int_{-\infty}^{t} h(\tau)\mathrm{d}\tau + \int_{t}^{\infty} 0 \times h(\tau)\mathrm{d}\tau \\
&= \begin{cases} \int_{-\infty}^{t} e^{2}\tau \mathrm{d}\tau, & t < 0 \\ \int_{-\infty}^{0} e^{2\tau}\mathrm{d}\tau + \int_{0}^{\infty} e^{-3\tau}\mathrm{d}\tau, & t \geq 0 \end{cases} \\
&= \begin{cases} \left.\frac{e^{2}\tau^{2}}{2}\right|_{-\infty}^{t}, & t < 0 \\ \left.\frac{e^{2\tau}}{2}\right|_{-\infty}^{0} - \left.\frac{e^{-3\tau}}{3}\right|_{0}^{t}, & t \geq 0 \end{cases} \\
&= \begin{cases} \frac{e^{2}t}{2}, & t < 0 \\ \frac{5}{6} - \frac{e^{-3t}}{3}, & t \geq 0 \end{cases}
\end{aligned}
$$

Question 4

(a) The difference equation for discrete signal $y[n]$ is defined as follows.

$$
\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \wedge a_0 = 1 \Rightarrow y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]
$$

$$
\begin{aligned}
y[n] &= x[n] * h[n] = x[n] * (\delta[n] - \delta[n+2] - 7\delta[n-3]) \\
&= x[n] * \delta[n] - x[n] * \delta[n+2] - x[n] * 7\delta[n-3] \\
&= \delta[n] - \delta[n-5] + 4\delta[n+3] + \delta[n-3] - \delta[n+2] - 4\delta[n+5] - 7\delta[n-3] + 7\delta[n+8] - 28\delta[n] \\
&= -4\delta[n+5] + 4\delta[n+3] - \delta[n+2] - 27\delta[n] - 6\delta[n-3] - \delta[n-5] + 7\delta[n-8]
\end{aligned}
$$

(b) Output signal $y[n]$ is calculated and signals $x[n]$, $h[n]$, $y[n]$ are plotted in Python using the following piece of code.

Python Snippet That Calculates and Plots the Given Signals

```
def d_t(t):    # Impulse Function
    result = np.zeros(len(t))
    for i, val in enumerate(t):
        if val == 0:
            result[i] = 1
        else:
            result[i] = 0
    return result

n = np.arange(-10, 10, 1)
x_n = d_t(n) - d_t(n-5) + 4*d_t(n+3)
h_n = d_t(n) - d_t(n+2) - 7*d_t(n-3)
y_n = np.convolve(x_n, h_n)

fig, axs = plt.subplots(1,3)
fig.tight_layout(rect=[0, 2, 2, 3])

axs[0].set_xlim(-5,7)
axs[0].set_xticks(np.arange(-5, 8, 1))
axs[0].stem(n, x_n, label="$x[n]$", linefmt="purple", use_line_collection=True)
axs[0].legend(["$x[n]$"])
axs[0].grid()

axs[1].set_xlim(-3,5)
axs[1].stem(n, h_n, label="$h[n]$", linefmt="green", use_line_collection=True)
axs[1].legend(["$h[n]$"])
axs[1].grid()

axs[2].set_xlim(-7,10)
axs[2].set_xticks(np.arange(-7, 11, 2))
axs[2].set_yticks(np.arange(-28, 9, 4))
axs[2].stem(convolution_range(x_n, h_n), y_n, label="$y[n]$", use_line_collection=True)
axs[2].legend(['$y[n]=x[n]*h[n]$'], loc="lower right")
axs[2].grid()

for ax in axs.flat:
    ax.set(xlabel="$n$")

plt.savefig("figure_saves/q4b-ans.png", dpi=dpi, bbox_inches='tight')
```

Output of the given code, the plots, are represented in Figure 15.

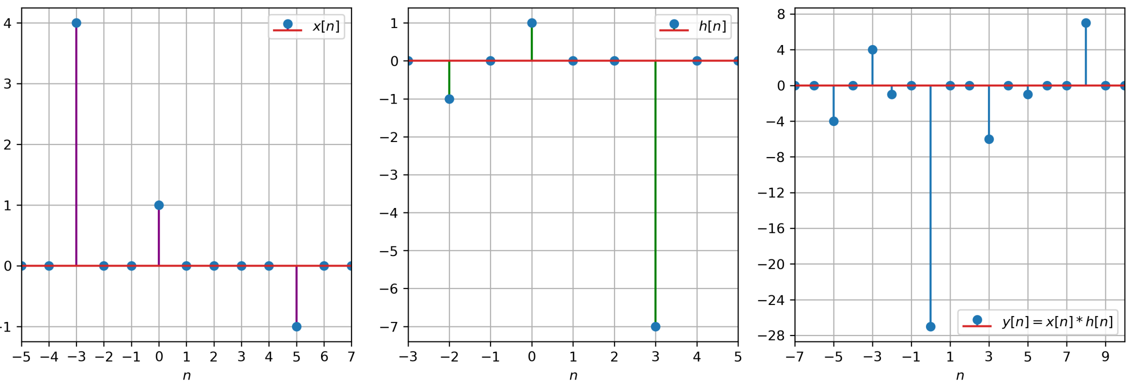


Figure 15: Plots of Signals $x[n]$, $h[n]$ and $y[n]$

Question 5

(a) Running average filter and median filters are first written in Python. Then, the given audio file is filtered using these filters. Implementations of the filters and the denoising process is given below with their respective pieces of codes.

Definition of the Running Average Filter

```python
def raf(signal, n_point, output="full"):

    if output == "same":
        # Return the signal with the same shape
        conv_length = len(signal)
        filtered = np.zeros(conv_length)
        for idx in range(n_point, conv_length):
            filtered[idx-n_point] = np.sum(signal[idx-n_point:idx])/n_point
    else:
        conv_length = len(signal)+n_point-1
        filtered = np.zeros(conv_length)
        signal = np.pad(signal, (n_point-1,0), "constant")
        for idx in range(n_point, conv_length):
            filtered[idx-n_point] = np.sum(signal[idx-n_point:idx])/n_point

    return filtered
```

Definition of the Median Filter

```python
def medf(signal, n_point, output="full"):

    if output == "same":
        conv_length = len(signal)
        filtered = np.zeros(conv_length)
```

```
            for idx in range(n_point, conv_length):
                filtered[idx-n_point] = np.median(signal[idx-n_point:idx])
        else:
            conv_length = len(signal)+n_point-1
            filtered = np.zeros(conv_length)
            signal = np.pad(signal, (n_point-1,0), "constant")
            for idx in range(n_point, conv_length):
                filtered[idx-n_point] = np.median(signal[idx-n_point:idx])

        return filtered
```

Reading the Audio Files and Applying Filters

```
import librosa
from scipy.io import wavfile

sampling_freq, noisy_data = wavfile.read("suphi_dirty.wav")
sampling_freq, clean_data = wavfile.read("suphi_clean.wav")
noisy_L = noisy_data[:,0]; noisy_R = noisy_data[:,1]
clean_L = clean_data[:,0]; clean_R = clean_data[:,1]

n_average_points = 3
suphi_raf_L = raf(noisy_L, n_average_points, "same")
suphi_raf_R = raf(noisy_R, n_average_points, "same")
suphi_raf = np.column_stack((suphi_raf_L, suphi_raf_R))

n_median_points = 3
suphi_medf_L = medf(noisy_L, n_median_points, "same")
suphi_medf_R = medf(noisy_R, n_median_points, "same")
suphi_medf = np.column_stack((suphi_medf_L, suphi_medf_R))
```

(b) A function to calculate the mean squared error (MSE) between the original clean audio file and the denoised audio files is given below.

Definition of the Mean Squared Error and its Application to Audio Files

```
def mse(s1, s2):
    error = (np.square(s1 - s2)).mean(axis=None)
    return error

raf_mse = mse(suphi_raf, clean_data)
medf_mse = mse(suphi_medf, clean_data)
raw_mse = mse(noisy_data, clean_data)
print("Running Average Filter MSE: {}\nMedian Filter MSE: {}\nMSE of Noisy Data: {}".format(raf_mse, medf_mse, raw_mse))
```

Using grid search technique, the MSE is minimized for both of the filters. When the number of averaging points is 3, MSE is minimized with a value of 65881. Moreover, when the number of median points is 3, MSE is minimized with a value of 72053. However it should be noted that filters of size 1 produce the noisy signal itself and filters of size 2 produce the same MSE of 38705. So, neither are not accepted as viable noise reduction filters.

Judging by the MSE values of both signals, running average filter seems to be better. On the other hand, it has to be noted that this MSE is calculated in the time domain. If the audio files were to be visualized in frequency domain, there would be a considerable amount of difference. Therefore it is advised to always visualize the data both in time and frequency domains (Figure 16-23). It should also be noted that denoised audio files are lack in sound quality and clarity due to the aforementioned time domain application of the filters.

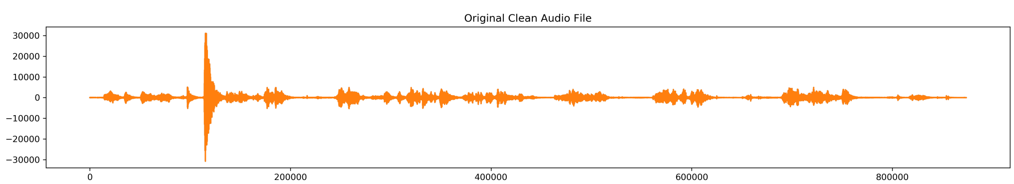


Figure 16: Clean Audio File (Time Domain)

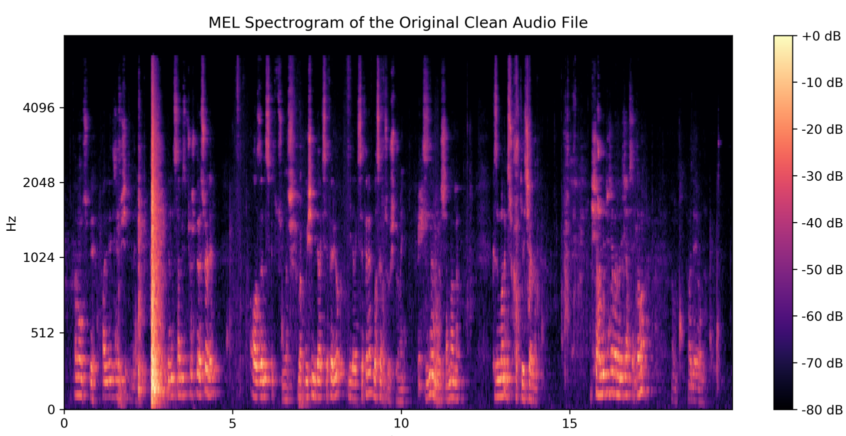


Figure 17: Mel Spectrogram of the Clean Audio File (Frequency Domain)

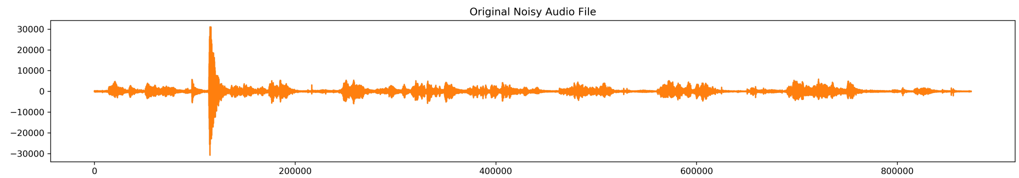


Figure 18: Noisy Audio File (Time Domain)

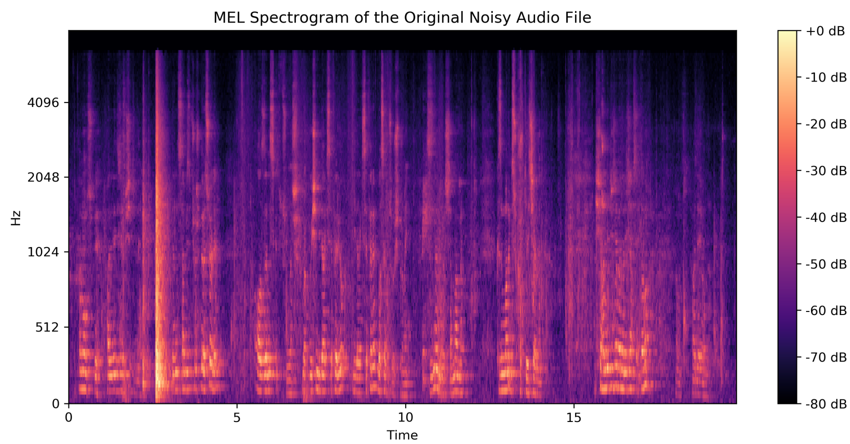


Figure 19: Mel Spectrogram of the Noisy Audio File (Frequency Domain)

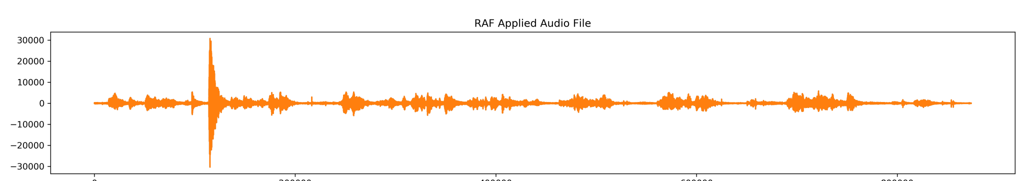


Figure 20: RAF Applied Audio File (Time Domain)

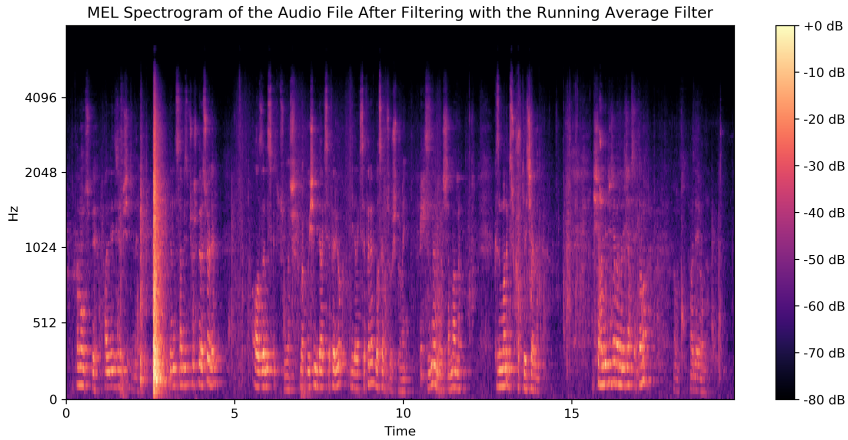


Figure 21: Mel Spectrogram of the RAF Applied Audio File (Frequency Domain)

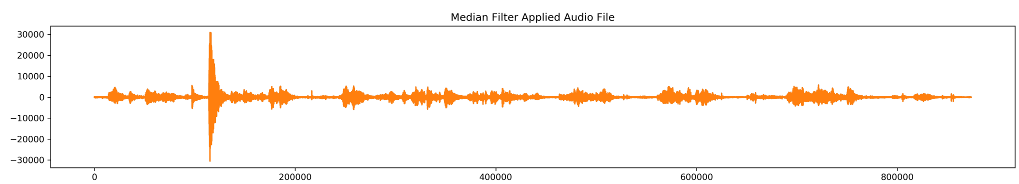


Figure 22: Median Filter Applied Audio File (Time Domain)

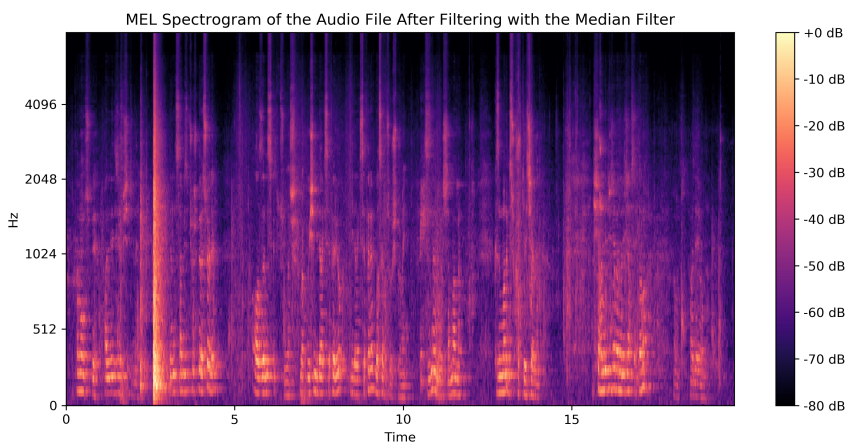


Figure 23: Mel Spectrogram of the Median Filter Applied Audio File (Frequency Domain)

(c) A linear filter F_l, is a filter that holds the condition $F_l(x_1 + kx_2) = F_l(x_1) + kF_l(x_2)$ true for two signals x_1 and x_2. But the given equation for F_l doesn't hold true for nonlinear filters.

Running average filter is a linear filter because it applies only linear operations to the signal, namely summation and multiplication. Whereas median filter is a nonlinear filter because selection of the median value is not a linear operation. For example, let's consider a signal $S_1 = \{1, 3, 4, 1, 1\}$ and both filters as $F = \{1, 1, 1\}$.

Assuming the output has the same length with the input signal S_1, the first step of the running average filter is applied as $S_o[0] = (1 \times 1 + 1 \times 3 + 1 \times 4)/3 = 8/3$. As seen by the example all of the operations are linear. On the other hand, if the median filter is applied as $S_o[0] = med(1, 3, 4) = 3$. With the given examples, it's clear that median filter is not a linear filter.

Question 6

The system consists of four blocks, three parallel ($h_2[n], h_3[n], h_4[n]$) and one in series ($h_1[n]$). Using the properties of convolution, three parallel blocks can first be summed then convoluted with the block in series. If the properties of convolution were not to be used, the block in series $h_1[n]$ has to be convoluted with all three parallel blocks $h_2[n], h_3[n], h_4[n]$ and then their results have to be summed. Let's follow this intuition and show the mathematical expressions for both ways.

$$\begin{aligned}
h_{overall,2} &= h_1[n] * (h_2[n] + h_3[n] + h_4[n]) = \alpha^n u[n] * (u[n] + u[n+3] - u[n] + \delta[n+1]) \\
&= \alpha^n u[n] * (u[n+3] + \delta[n+1]) = \alpha^n u[n] * u[n+3] + \alpha^n u[n] * \delta[n+1] \\
&= \alpha^{n+3} r[n+3] + \alpha^{n+1} u[n+1] = \alpha^{n+1}(\alpha^2 r[n+3] + u[n+1])
\end{aligned}$$

where, ramp signal $r[n]$ is $r[n] = u[n] * u[n]$.

$$\begin{aligned}
h_{overall,2} &= h_1[n] * h_2[n] + h_1[n] * h_3[n] + h_1[n] * h_4[n]) \\
&= \alpha^n u[n] * u[n] + \alpha^n u[n] * u[n+3] - \alpha^n u[n] * u[n] + \alpha^n u[n] * \delta[n+1]) \\
&= \alpha^n r[n] + \alpha^{n+3} r[n+3] - \alpha^n r[n] + \alpha^{n+1} u[n+1] \\
&= \alpha^{n+3} r[n+3] + \alpha^{n+1} u[n+1] = \alpha^{n+1}(\alpha^2 r[n+3] + u[n+1])
\end{aligned}$$

Finally, since $h_{overall,1} = h_{overall,2}$, it is proven that same output for the $h_{overall}$ can be achieved either by using the properties of convolution or not.