EEE-424 Analytical Assignment 1 Spring 2024-25 Solutions

March 2025

Answer 1

$$
\begin{aligned}
H(\omega) &= \frac{1 - a^4 \exp(-4j\omega)}{1 - a^4 \exp(-j\omega)} \\
&= (1 - a^4 \exp(-4j\omega)) \sum_{n=0}^{\infty} (a^4 \exp(-j\omega))^n \\
&= \sum_{n=0}^{\infty} a^{4n} \exp(-nj\omega) - \sum_{n=0}^{\infty} a^{4n+4} \exp(-(n+4)j\omega) \\
&= \sum_{n=0}^{3} a^{4n} \exp(-nj\omega) + \sum_{n=4}^{\infty} [a^{4n} - a^{4n-12}] \exp(-nj\omega) \\
h[n] &= \begin{cases} 0, & n < 0, \\ a^{4n}, & n = 0, 1, 2, 3, \\ a^{4n} - a^{4n-12}, & n \geq 4. \end{cases}
\end{aligned}
$$

Since the impulse response extends to ∞, this is an IIR filter.At first glance it may seem like the filter is stable only if $|a| < 1$. In fact, the filter is stable when $|a| \leq 1$, and is unstable only for $|a| > 1$. However, when $|a| = 1$, from the geometric series formula $1 + r + r^2 + r^3 = \frac{1-r^4}{1-r}$, we get that

$$
H(\omega) = \frac{1 - \exp(-4j\omega)}{1 - \exp(-j\omega)} = 1 + \exp(-j\omega) + \exp(-2j\omega) + \exp(-3j\omega)
$$

is the transfer function of a (stable) **FIR** filter.

Answer 2

Applying the orthogonality conditions we have:

$$
\begin{aligned}
a + b + c + d &= 0 \qquad (1) \\
a + b - c - d &= 0 \qquad (2) \\
a - b + c - d &= 0 \qquad (3) \\
a^2 + b^2 + c^2 + d^2 &= 1 \qquad (4)
\end{aligned}
$$

From (2) $a + b = c + d$ and then use (1) $2(c + d) = 0 => c + d = 0, a + b = 0$
From (3) $a + c = b + d$ and again use (1) $2(b + d) = 0 => b + d = 0, a + c = 0$
If $b + d = 0$ and $c + d = 0 => b = c => d = -c, a = -c$

Using (4) we substitute terms

$$
\begin{aligned}
a^2 + b^2 + c^2 + d^2 &= 1 \\
(-c)^2 + c^2 + c^2 + (-c)^2 &= 1 \\
4c^2 &= 1 \\
c &= \pm 1/2
\end{aligned}
$$

which gives $c = 1/2$ or $c = -1/2$ and all the other variables are depend on c. So we have 2 solutions for the system of equations.

Answer 3

Parts a) and b)

$$
zY(z) - \frac{5}{2}Y(z) + z^{-1}Y(z) = zX(z) + z^{-1}X(z)
$$

$$
\Rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{z + z^{-1}}{z - \frac{5}{2} + z^{-1}}
$$

$$
\begin{aligned}
H(z) &= z^{-1}\frac{z+z^{-1}}{z-\frac{5}{2}z^{-1}+z^{-2}} \\
&= \frac{1+z^{-2}}{(z^{-2}-z^{-1}\frac{5}{2}+1)} \\
&= (1+z^{-2})\left[\frac{2/3}{z^{-1}-2}+\frac{-2/3}{z^{-1}-\frac{1}{2}}\right] \\
&= (1+z^{-2})\left[\frac{-1/3}{1-\frac{1}{2}z^{-1}}+\frac{4/3}{1-2z^{-1}}\right] \\
&\Rightarrow \text{poles: } z=\frac{1}{2}, z=2
\end{aligned}
$$

Stable $\Rightarrow$ ROC must include the unit circle.

$$
\Rightarrow -\frac{1}{3}\frac{z^{-1}}{1-\frac{1}{2}z^{-1}} \quad \xrightarrow{Z^{-1}} \quad -\frac{11}{3}\left(\frac{1}{2}\right)^n u[n]
$$

(right-sided)

$$
\frac{4}{3}\frac{1}{1-2z^{-1}} \quad \xrightarrow{Z^{-1}} \quad = -\frac{4}{3}(2)^n u[-n-1]
$$

(left-sided)

$$
h[n] = \underbrace{-\frac{1}{3}\left(\frac{1}{2}\right)^n u[n] - \frac{4}{3}2^n u[-n-1]}_{h_1[n]} + \left[-\frac{1}{3}\left(\frac{1}{2}\right)^{n-2} u[n-2] + \frac{4}{3}2^{n-2}u[-n+1]\right]
$$

Since $1+z^{-2}$ term at the numerator makes

$$
h[n] = h_1[n] + h_1[n-2]
$$

$$
H(e^{j\omega}) = H(z)\Big|_{z=e^{j\omega}} = \frac{e^{j\omega}+e^{-j\omega}}{e^{j\omega}-\frac{5}{2}+e^{-j\omega}} = \frac{2\cos\omega}{-\frac{5}{2}+2\cos\omega}
$$

Part c)

$$
x[n] = 10\cos\left[\frac{\pi}{3}n-\frac{\pi}{2}\right] = \frac{10}{2}e^{-j\frac{1}{2}}e^{j\frac{\pi}{3}n} + \frac{10}{2}e^{j\frac{1}{2}}e^{-j\frac{\pi}{3}n}
$$

$$
y[n] = \frac{10}{2}e^{-j\frac{1}{2}}H(e^{j\frac{\pi}{3}})e^{j\frac{\pi}{3}n} + \frac{10}{2}e^{j\frac{1}{2}}H(e^{-j\frac{\pi}{3}})e^{-j\frac{\pi}{3}n}
$$

Remember that complex exponentials are Eigenfunctions of LTI systems:

$$
H(e^{j\frac{\pi}{3}}) = \frac{2\cos\frac{\pi}{3}}{-\frac{5}{2}+2\cos\frac{\pi}{3}} = \frac{1}{-\frac{3}{2}} = -\frac{2}{3} = H(e^{-j\frac{\pi}{3}})
$$

$$
y[n] = -\frac{10}{3}e^{j\frac{\pi}{3}n}e^{j(-\frac{1}{2})} - \frac{10}{3}e^{-j\frac{\pi}{3}n}e^{j(\frac{1}{2})}
$$

Hence,

$$
y[n] = -\frac{20}{3}\cos\left[\frac{\pi}{3}n-\frac{1}{2}\right]
$$

Answer 4

The gear rotates at 30rpm, which means it completes one revolution in 2seconds. Since the gear has 8 identical teeth, the sensor's on/off pattern repeats eight times per revolution. In other words, the gear's basic rotational frequency at 30rpm is 0.5Hz (one revolution per two seconds), but the sensor's "tooth-passing frequency" is 8 times higher, namely 4Hz (because each revolution brings 8 identical pulses). This repeating structure allows aliasing to occur if the sampling interval T is large enough that multiple teeth pass between samples, possibly causing the measured data to "jump" backward or appear stationary instead of showing forward motion.

When $T = 0.4$ seconds, the gear rotates through 0.4 seconds of motion between each sample. In 0.4 seconds at 30rpm, the gear completes $\frac{0.4}{2} = 0.2$ of a revolution. Since one revolution corresponds to 8 teeth, 0.2 revolutions correspond to $0.2 \times 8 = 1.6$ teeth. Thus, between one sample and the next, approximately 1.6 teeth pass the sensor. When we discretize 1.6 teeth, we must consider that if the gear's position "wraps around," an apparent +1.6 could also be seen as -6.4 if we think in terms of stepping from one identical tooth pattern to another. However, because 1.6 is less than 4 (half of the gear's 8-tooth cycle), the natural interpretation is that the gear is moving forward by nearly 1–2 teeth each time the microcontroller samples. Converting 1.6 teeth per 0.4 seconds to an effective rotation rate yields 4 teeth per second, which agrees with the real tooth-passing frequency of 4Hz. This means 4 teeth per second is exactly half a revolution per second (because 8 teeth = 1 revolution), or 30 rpm. So with $T = 0.4$ seconds, there is no misleading aliasing: the data would correctly show about 30 rpm forward.

When $T = 1.0$ seconds, the gear rotates for a full second between samples. In that time, it completes $\frac{1.0}{2} = 0.5$ of a revolution at 30 rpm, which corresponds to 4 teeth passing in front of the sensor. But 4 teeth out of an 8-tooth cycle brings the sensor back to an identical position: the gear looks exactly the same after half a turn because tooth 1 lines up where tooth 5 was, and all teeth are identical. In discrete sampling, moving +4 teeth can be interpreted as moving -4 teeth just as validly, because the sensor sees the same tooth profile. A +4-tooth jump is a half revolution forward, while -4 is a half revolution backward, and both end up at an identical tooth pattern. Indeed, if each sample sees the same "tooth position," the system might conclude the gear is not moving at all (aliasing to 0rpm), or it could alias to backward movement depending on minute timing offsets. In practical terms, the 1.0 second sampling interval fails to capture the gear's true forward speed, because each sample lands on positions that look the same to the sensor, leading to a severe alias.

To guarantee that the sampled data never falsely indicates backward or stationary motion, one must ensure that each sample observes a gear rotation significantly smaller than half of the 8-tooth pattern. The largest step you can allow between samples is when the gear advances just under 4 teeth in each sampling period, so that it can never be "re-labeled" as a smaller backward jump. The gear's tooth-passing frequency at 30rpm is 4Hz. According to Nyquist-style logic, you need to sample at strictly more than twice that frequency (i.e., over 8Hz) to capture the direction unambiguously. A 4Hz signal would be aliased if you only sample at 4Hz or less. Since a sampling frequency of 8Hz corresponds to a sampling period $T = 1/8$ seconds = 0.125 seconds, you must sample faster than that to avoid backward aliasing. Alternatively, one can directly argue that each 0.125 seconds of real time means the gear advances $0.125/2 = 1/16$ of a revolution, which is only 0.5 teeth; a step of less than 4 teeth between samples can never be misinterpreted as a negative jump. Hence, the maximum allowable T that always shows forward rotation at 30rpm is just under 0.125 seconds.

Finally, if the gear speed can vary up to 60 rpm, then the tooth-passing frequency can be as high as 8Hz (because 60rpm is 1 revolution per second, multiplied by 8 teeth per revolution = 8 teeth per second). You now need to sample at a frequency strictly greater than twice 8Hz, namely more than 16Hz, which corresponds to a sampling period below 0.0625 seconds. If you choose $T < 0.0625$ seconds, then even at the gear's highest speed of 60 rpm, the measured data will never incorrectly show backward or stationary motion, since every snapshot captures a small enough fraction of a revolution that there is no opportunity for a confusing wrap-around.

Answer 5

Frequency domain basis vectors are:

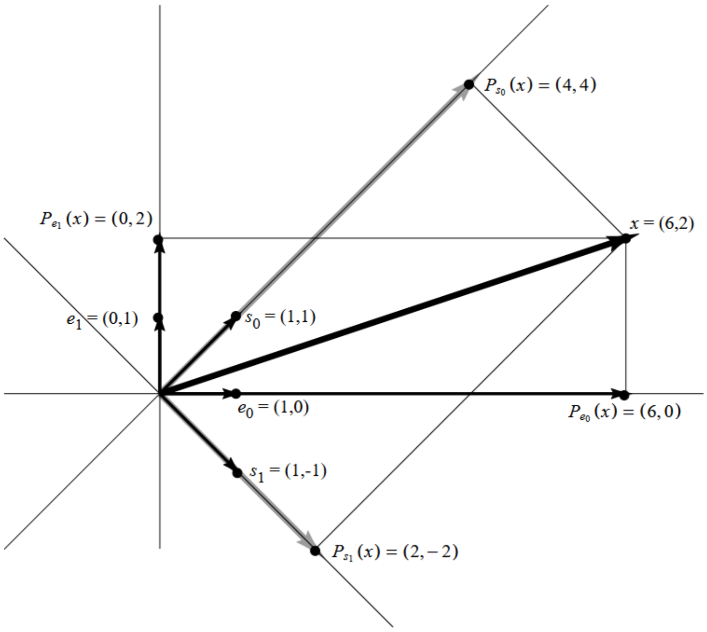
$$\underline{s_0} = (1, 1)$$
$$\underline{s_1} = (1, -1)$$

and for time domain use $e_0 = (1, 0)$ and $e_1 = (0, 1)$ so that $\underline{x} = 6e_0 + 2e_1$. Analytically, we compute the DFT to be

$$X(\omega_0) \stackrel{\text{def}}{=} \mathcal{P}_{s_0}(x) \stackrel{\text{def}}{=} \frac{\langle x, s_0\rangle}{\langle s_0, s_0\rangle} s_0 = \frac{6 \cdot 1 + 2 \cdot 1}{1^2 + 1^2} s_0 = 4s_0 = (4, 4)$$

$$X(\omega_1) \stackrel{\text{def}}{=} \mathcal{P}_{s_1}(x) \stackrel{\text{def}}{=} \frac{\langle x, s_1\rangle}{\langle s_1, s_1\rangle} s_1 = \frac{6 \cdot 1 + 2 \cdot (-1)}{1^2 + (-1)^2} s_1 = 2s_1 = (2, -2)$$

All vectors are shown in the figure below:



Note the lines of orthogonal projection illustrated in the figure. The "time domain" basis consists of the vectors e_0, e_1, and the orthogo- nal projections onto them are simply the coordinate projections $(6, 0)$ and $(0, 2)$. The "frequency domain" basis vectors are $\{s_0, s_1\}$, and they provide an orthogonal basis set which is *rotated 45 degrees* relative to the time-domain basis vectors. Projecting orthogonally onto them gives $X(\omega_0) = (4, 4)$ and $X(\omega_1) = (2, -2)$, respectively. The original signal x can be expressed as the vector sum of its coordinate projections (a time- domain representation), or as the vector sum of its projections onto the DFT sinusoids (a frequency-domain representation). Computing the co-efficients of projection is essentially "taking the DFT" and constructing $\underline{x}$ as the vector sum of its projections onto the DFT sinusoids amounts to "taking the inverse DFT."

Answer 6

To solve the problem, we begin with the definition

$$X[k] = \sum_{n=0}^{N-1} n\, e^{-j\frac{2\pi}{N}kn}.$$

It is convenient to introduce the variable

$$r = e^{-j\frac{2\pi}{N}k},$$

so that the sum becomes

$$X[k] = \sum_{n=0}^{N-1} n\, r^n$$

which should be a more recognizable sum.

Next, consider the finite geometric series

$$S(r) = \sum_{n=0}^{N-1} r^n = \frac{1-r^N}{1-r}, \quad \text{for } r \neq 1.$$

Since the derivative of r^n with respect to r is $n\,r^{n-1}$, differentiating $S(r)$ term-by-term yields

$$\frac{dS(r)}{dr} = \sum_{n=0}^{N-1} n\,r^{n-1}.$$

Multiplying by r gives

$$r\,\frac{dS(r)}{dr} = \sum_{n=0}^{N-1} n\,r^n = X[k].$$

It now remains to differentiate the closed-form expression for $S(r)$. We have

$$S(r) = \frac{1-r^N}{1-r}.$$

Differentiating with respect to r by using the quotient rule yields

$$\frac{dS(r)}{dr} = \frac{(1-r)\Big(-N\,r^{N-1}\Big) - (1-r^N)(-1)}{(1-r)^2} = \frac{-N\,r^{N-1}(1-r) + 1 - r^N}{(1-r)^2}.$$

Multiplying by r produces

$$X[k] = r\,\frac{dS(r)}{dr} = \frac{-N\,r^N(1-r) + r(1-r^N)}{(1-r)^2}.$$

Thus, for $r \neq 1$ (i.e. for $k \not\equiv 0 \pmod{N}$) the DFT is given in closed form by

$$X[k] = \frac{r(1-r^N) - N\,r^N(1-r)}{(1-r)^2} \quad \text{with } r = e^{-j\frac{2\pi}{N}k}.$$

In the special case when $k = 0$ so that $r = 1$, one may compute the sum directly as

$$X[0] = \sum_{n=0}^{N-1} n = \frac{N(N-1)}{2}.$$

It is possible to further simplify the expression for $X[k]$ for $k \neq 0$ by writing the differences in the numerator in terms of sine functions (using Euler's formula) and expressing the result in an alternative form. But that is not necessary for this question. The following final expression is sufficient:

$$X[k] = \frac{-N\,e^{-j\frac{2\pi}{N}kN}\Big(1 - e^{-j\frac{2\pi}{N}k}\Big) + e^{-j\frac{2\pi}{N}k}\Big(1 - e^{-j\frac{2\pi}{N}kN}\Big)}{\Big(1 - e^{-j\frac{2\pi}{N}k}\Big)^2},$$

Answer 7

$$\mathbf{p} = \begin{bmatrix} \langle x(t), p_1(t)\rangle \\ \langle x(t), p_2(t)\rangle \\ \langle x(t), p_3(t)\rangle \end{bmatrix} = \begin{bmatrix} \int_0^1 \cos\left(\frac{\pi t}{2}\right) dt \\ \int_0^1 \cos\left(\frac{\pi t}{2}\right) t dt \\ \int_0^1 \cos\left(\frac{\pi t}{2}\right) t^2 dt \end{bmatrix} = \begin{bmatrix} \frac{2}{\pi} \\ \frac{2}{\pi} - \frac{4}{\pi^2} \\ \frac{2}{\pi} - \frac{16}{\pi^3} \end{bmatrix}.$$

Next, we need to compute the entries of the Grammian:

$$\langle p_1(t), p_1(t)\rangle = \int_0^1 1 dt = 1$$

$$\langle p_1(t), p_2(t)\rangle = \int_0^1 t dt = \frac{1}{2}$$

$$\langle p_1(t), p_3(t)\rangle = \int_0^1 t^2 dt = \frac{1}{3}$$

$$\langle p_2(t), p_2(t)\rangle = \int_0^1 t^2 dt = \frac{1}{3}$$
$$\langle p_2(t), p_3(t)\rangle = \int_0^1 t^3 dt = \frac{1}{4}$$
$$\langle p_3(t), p_3(t)\rangle = \int_0^1 t^4 dt = \frac{1}{5}.$$

Thus, by symmetry, we have:

$$R = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}.$$

The set of optimal approximating coefficients is then:

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = R^{-1}\mathbf{p} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}^{-1} \begin{bmatrix} \frac{2}{\pi} \\ \frac{2}{\pi} - \frac{4}{\pi^2} \\ \frac{2}{\pi} - \frac{16}{\pi^3} \end{bmatrix} = \begin{bmatrix} 1.0194 \\ -0.2091 \\ -0.8346 \end{bmatrix}.$$

Hence $\cos\left(\frac{\pi t}{2}\right) \approx 1.0194 - 0.2091t - 0.8346t^2$ over the unit interval.

Answer 8

a) Compute the Circular Convolution.

For the first method any valid method including multiplication with the correct Toeplitz Circulant matrices is acceptable. The DFT method is as follows: Let

$$W_4 = e^{-j\frac{2\pi}{4}} = e^{-j\frac{\pi}{2}},$$

with

$$W_4^0 = 1, \quad W_4^1 = -j, \quad W_4^2 = -1, \quad W_4^3 = j.$$

DFT of $x(n)$:

$$X(k) = \sum_{n=0}^{3} x(n) W_4^{kn}.$$

$$\begin{aligned} X(0) &= 1 + 2 + 3 + 0 = 6, \\ X(1) &= 1 + 2(-j) + 3(-1) + 0 = -2 - 2j, \\ X(2) &= 1 + 2(-1) + 3(1) + 0 = 2, \\ X(3) &= 1 + 2(j) + 3(-1) + 0 = -2 + 2j. \end{aligned}$$

Thus, $X = [6,\ -2-2j,\ 2,\ -2+2j]$.

DFT of $h(n)$:

$$H(k) = \sum_{n=0}^{3} h(n) W_4^{kn}.$$

$$\begin{aligned} H(0) &= 4 + 5 + 6 + 7 = 22, \\ H(1) &= 4 + 5(-j) + 6(-1) + 7(j) = -2 + 2j, \\ H(2) &= 4 + 5(-1) + 6(1) + 7(-1) = -2, \\ H(3) &= 4 + 5(j) + 6(-1) + 7(-j) = -2 - 2j. \end{aligned}$$

Thus, $H = [22,\ -2+2j,\ -2,\ -2-2j]$.

$$Y(k) = X(k)\,H(k).$$

$$\begin{aligned} Y(0) &= 6 \cdot 22 = 132, \\ Y(1) &= (-2-2j)(-2+2j) = (-2)^2 + (2)^2 = 4 + 4 = 8, \\ Y(2) &= 2 \cdot (-2) = -4, \\ Y(3) &= (-2+2j)(-2-2j) = 8. \end{aligned}$$

Thus, $Y = [132,\ 8,\ -4,\ 8]$.

IDFT of $Y(k)$:

$$y(n) = \frac{1}{4}\sum_{k=0}^{3} Y(k)W_4^{kn}, \quad n = 0, 1, 2, 3.$$

<u>For $n = 0$:</u>

$$y(0) = \frac{1}{4}[132 + 8 + (-4) + 8] = \frac{144}{4} = 36.$$

<u>For $n = 1$:</u>

$$y(1) = \frac{1}{4}\Big[132 \cdot 1 + 8(-j) + (-4)(-1) + 8(j)\Big] = \frac{1}{4}[132 + 4] = 34.$$

<u>For $n = 2$:</u> (Using $W_4^2 = -1$)

$$y(2) = \frac{1}{4}\Big[132 \cdot 1 + 8(-1) + (-4) \cdot 1 + 8(-1)\Big] = \frac{1}{4}[132 - 8 - 4 - 8] = \frac{112}{4} = 28.$$

<u>For $n = 3$:</u>

$$y(3) = \frac{1}{4}\Big[132 \cdot 1 + 8(j) + (-4)(-1) + 8(-j)\Big] = \frac{1}{4}[132 + 4] = 34.$$

Thus, the 4-point circular convolution result is

$$y_{\text{circ}}(n) = [36,\ 34,\ 28,\ 34].$$

b) Compute the Linear Convolution.
Since $\text{length}(x) = 3$ and $\text{length}(h) = 4$, the linear convolution has $3 + 4 - 1 = 6$ points.

$$y_{\text{lin}}(n) = \sum_m x(m)\,h(n - m).$$

$$\begin{aligned}
y_{\text{lin}}(0) &= x(0)h(0) = 1 \cdot 4 = 4,\\
y_{\text{lin}}(1) &= x(0)h(1) + x(1)h(0) = 1 \cdot 5 + 2 \cdot 4 = 5 + 8 = 13,\\
y_{\text{lin}}(2) &= x(0)h(2) + x(1)h(1) + x(2)h(0) = 1 \cdot 6 + 2 \cdot 5 + 3 \cdot 4 = 6 + 10 + 12 = 28,\\
y_{\text{lin}}(3) &= x(0)h(3) + x(1)h(2) + x(2)h(1) = 1 \cdot 7 + 2 \cdot 6 + 3 \cdot 5 = 7 + 12 + 15 = 34,\\
y_{\text{lin}}(4) &= x(1)h(3) + x(2)h(2) = 2 \cdot 7 + 3 \cdot 6 = 14 + 18 = 32,\\
y_{\text{lin}}(5) &= x(2)h(3) = 3 \cdot 7 = 21.
\end{aligned}$$

Thus,

$$y_{\text{lin}}(n) = [4,\ 13,\ 28,\ 34,\ 32,\ 21].$$

c) Relate Circular and Linear Convolution.
In 4-point circular convolution the output indices are computed modulo 4. Thus, the linear convolution outputs at $n = 4$ and $n = 5$ wrap around as:

$$y_{\text{circ}}(0) = y_{\text{lin}}(0) + y_{\text{lin}}(4) = 4 + 32 = 36,$$

$$y_{\text{circ}}(1) = y_{\text{lin}}(1) + y_{\text{lin}}(5) = 13 + 21 = 34.$$

The values $y_{\text{circ}}(2) = 28$ and $y_{\text{circ}}(3) = 34$ match $y_{\text{lin}}(2)$ and $y_{\text{lin}}(3)$, respectively.

To recover the full linear convolution using the DFT, choose a DFT length $N \geq 6$. Zero-pad $x(n)$ and $h(n)$ to length 6, so that circular convolution equals linear convolution.

Answer 9

The correct answer is **32kHz**. If you choose 48kHz then the 23kHz wave will NOT be aliased but the 25kHz wave will get folded to an alias frequency of 23kHz and as such the receiver will not be able to distinguish between the two. Sampling at 16kHz will also produce aliased frequncies but this time in both the sine waves and cause more severe distortion/overlap than any of the other cases rendering the received signal completely useless.
Only 32kHz works because in this case the the 23kHz signal aliases to 9kHz, and the 25kHz signal aliases to 7kHz. Since 9kHz and 7kHz are distinct, the receiver can reliably distinguish which original signal is being transmitted.

Answer 10

a)

$$x[2n] = \{x[0], x[2], x[4], x[6]\} = \{1, -1, -1, 1\},$$
$$x[2n+1] = \{x[1], x[3], x[5], x[7]\} = \{1, -1, 1, -1\}.$$
$$X[k] = \sum_{n=0}^{N/2-1} x_{2n} W_{N/2}^{kn} + W_N^k \sum_{n=0}^{N/2-1} x_{2n} W_{N/2}^{kn}$$

Computing $W_{N/2}^{kn}$ and W_N^k for different values of k and n, and following accordingly, we get:

$$X_0[k] = \sum_{n=0}^{N/2-1} x_{2n} W_{N/2}^{kn} = \begin{bmatrix} W_{N/2}^{kn} \end{bmatrix} \begin{bmatrix} x_{2n} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2+2j \\ 0 \\ 2-2j \end{bmatrix}.$$

$$X_1[k] = \sum_{n=0}^{N/2-1} x_{2n+1} W_{N/2}^{kn} == \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2+2j \\ 0 \\ 2-2j \end{bmatrix}.$$

$$X[k] = X_0[k] + W_N^k X_1[k].$$

Computing the values of W_N^k for $N = 8$,

$$X[0] = X_0[0] + W_8^0 X_1[0] = 0.$$
$$X[1] = X_0[1] + W_8^1 X_1[1] = 2 + 2j.$$
$$X[2] = X_0[2] + W_8^2 X_1[2] = -4j.$$
$$X[3] = X_0[3] + W_8^3 X_1[3] = 2 - 2j.$$
$$X[4] = X_0[0] - W_8^4 X_1[0] = 0.$$

The rest of the terms can be computed in the same way, or alternatively, using an inherent property of the signal at hand. Notice that the signal $x[n]$ is real. Using the property that the DFT of a real signal is Hermitian symmetric, we can find the rest of the terms.

$$X[7] = X[1]^*.$$
$$X[6] = X[2]^*.$$
$$X[5] = X[3]^*.$$

b)

$$y[2n] = \{y[0], y[2], y[4], y[6]\} = \{1, 1, 1, 2\},$$
$$y[2n+1] = \{y[1], y[3], y[5], y[7]\} = \{-2, -1, -1, -1\}.$$
$$Y[k] = \sum_{n=0}^{N/2-1} y_{2n} W_{N/2}^{kn} + W_N^k \sum_{n=0}^{N/2-1} y_{2n} W_{N/2}^{kn}$$

$$Y_0[k] = \sum_{n=0}^{N/2-1} y_{2n} W_{N/2}^{kn} = \begin{bmatrix} W_{N/2}^{kn} \end{bmatrix} \begin{bmatrix} y_{2n} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ j \\ -1 \\ -j \end{bmatrix}.$$

$$Y_1[k] = \sum_{n=0}^{N/2-1} y_{2n+1} W_{N/2}^{kn} = \begin{bmatrix} W_{N/2}^{kn} \end{bmatrix} \begin{bmatrix} y_{2n} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -2 \\ -1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -5 \\ -1 \\ -1 \\ -1 \end{bmatrix}.$$

$$Y[k] = Y_0[k] + W_N^k Y_1[k].$$

Computing the values of W_N^k for $N = 8$,

$$Y[0] = Y_0[0] + W_8^0 Y_1[0] = 0.$$

$$Y[1] = Y_0[1] + W_8^1 Y_1[1] = -\frac{1}{\sqrt{2}} + j\frac{1}{\sqrt{2}}.$$

$$Y[2] = Y_0[2] + W_8^2 Y_1[2] = -1 + j.$$

$$Y[3] = Y_0[3] + W_8^3 Y_1[3] = (1 - \frac{1}{\sqrt{2}}) + \frac{1}{\sqrt{2}}j.$$

$$Y[4] = Y_0[0] - W_8^4 Y_1[0] = 10.$$

The rest of the terms can be computed in the same way, or alternatively, using the property that the DFT of a real signal is Hermitian symmetric around its center.

$$Y[7] = Y[1]^*.$$

$$Y[6] = Y[2]^*.$$

$$Y[5] = Y[3]^*.$$

c) Both signals can be computed more efficiently compared to arbitrary signals, by leveraging DFT properties. Since both signals are real, their DFTs are Hermitian symmetric around their centers. As a consequence, their FFTs can be computed even faster, by only computing the first $N/2 + 1$ terms.
d)

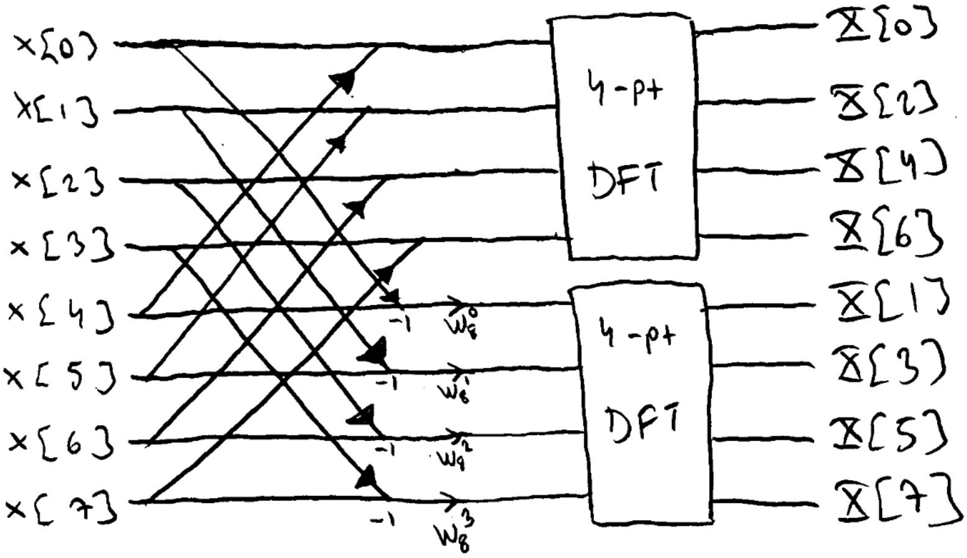


Figure 1: Flow graph for the 8-pt decimation-in-frequency FFT algorithm.