Appendix A: Tutorials

**Tutorial 1**

Q: Plot the unit step function $u(t)$. Also plot the functions $u(-t)$, $u(t - 3)$, $u(-t - 3)$, $u(t + 5)$, $u(-t + 5)$, and $y(t) = u(t) - u(t - 1)$.

Solution:

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}, \quad \therefore \quad u(-t) = \begin{cases} 1, & -t \geq 0 \\ 0, & -t < 0 \end{cases} = \begin{cases} 1, & t \leq 0 \\ 0, & t > 0 \end{cases}$$

Generally we have:

$$u(t - a) = \begin{cases} 1, & t - a \geq 0 \\ 0, & t - a < 0 \end{cases} = \begin{cases} 1, & t \geq a \\ 0, & t < a \end{cases},$$

$$u(-t - a) = \begin{cases} 1, & -t - a \geq 0 \\ 0, & -t - a < 0 \end{cases} = \begin{cases} 1, & t \leq -a \\ 0, & t > -a \end{cases}$$

Shift rules:

If $g(t)$ is a function and $a > 0$, then the direction of horizontal shift from original location is found as follows:

<table>
<thead>
<tr>
<th>Function</th>
<th>Shift direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g(t + a)$</td>
<td>– (Left)</td>
</tr>
<tr>
<td>$g(t - a)$</td>
<td>+ (Right)</td>
</tr>
<tr>
<td>$g(-t + a)$</td>
<td>+ (Right)</td>
</tr>
<tr>
<td>$g(-t - a)$</td>
<td>– (Left)</td>
</tr>
</tbody>
</table>
Tutorial 2

Q1: Find the I/O (input–output) relation of the system shown below.

Solution: First, we should define the important internal points as shown below:

\[
y(t) = r(t - T_o) \\
r(t) = x(t) - 2y(t)
\]

\[
\therefore y(t) = x(t - T_o) - 2y(t - T_o).
\]
This is a feedback system since the I/O equation includes a delayed version of the output itself.

Q2: Find the I/O (input–output) relation of the system shown below.

Ans. \( y(t) = x(t) + x(t - T_o) - y(t - T_o) \).

**Tutorial 3**

Q: Determine whether the analog time-delay \( T_o \) (\( T_o \) is constant) is:

Solution:
1. Since \( y(t) = x(t - T_o) \), the output equals the input at a past time instant, \( t - T_o \), hence it is a memory system.
2. The output \( y(t) \) is not a function of \( x(t + t_0) \), \( t_0 > 0 \), hence it is causal (does not depend on future values of the input).
3. Let \( T \) represents the system transformation.
   For the input \( x(t) = ap(t) + br(t) \) we have:
   \[
   y(t) = T\{x(t)\} = x(t - T_o) = a \cdot p(t - T_o) + b \cdot r(t - T_o) \\
   = a \cdot T\{p(t)\} + b \cdot T\{r(t)\}
   \]
   Hence, the system is linear.
4. \( T\{x(t - t_0)\} = x(t - t_0 - T_o) \) (1)
   We have: \( y(t) = x(t - T_o) \), hence,
   \[
   y(t - t_0) = x(t - t_0 - T_o) \] (2)
   From Eqs. 1 and 2 we get:
   \[
   y(t - t_0) = T\{x(t - t_0)\}
   \]
   Hence, \( T \) is time-invariant.
5. If $|x(t)| \leq c \forall t$ (bounded), then we have $|x(t - T_0)| \leq c \forall t$.
Hence, $|y(t)| \leq c \forall t$ and the system is BIBO-stable.

**Tutorial 4**

**Q:** Find the output of the following linear time-invariant system for a unit-step input (i.e., find the step response). Verify that the impulse response is the time derivative of the step response.

$$x(t) = u(t) \quad \rightarrow \quad h(t) = e^{-at}u(t) \quad (a > 0) \quad \rightarrow \quad y(t)$$

**Solution:** We have:

$$y(t) = u(t) * h(t) = \int_{-\infty}^{\infty} u(\lambda)h(t - \lambda)d\lambda.$$

**Step 1:** Formulas of functions using $t$:

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad \text{and} \quad h(t) = e^{-at}u(t) = \begin{cases} e^{-at}, & t \geq 0 \\ 0, & t < 0 \end{cases}.$$

**Step 2:** Formulas of functions using $\lambda$:

$$u(\lambda) = \begin{cases} 1, & \lambda \geq 0 \\ 0, & \lambda < 0 \end{cases} \quad \text{and} \quad h(t - \lambda) = \begin{cases} e^{-a(t-\lambda)}, & t - \lambda \geq 0 \rightarrow \lambda \leq t \\ 0, & \lambda > t \end{cases}.$$

**Step 3:** Integrate to find the convolution. The range of integration can be found graphically as the zone of non-zero overlap between the two functions. This can be done by moving $h(t - \lambda)$ from left to right (i.e., by varying $t$ from $-\infty$ to $\infty$) while keeping $u(\lambda)$ fixed (see the plot below).

**Case 1:** $t > 0$

$$y(t) = \int_{0}^{t} \exp[-a(t - \lambda)]d\lambda.$$

$$= \frac{1}{a}[\exp(-at)]_{0}^{t} = \frac{1}{a}[1 - \exp(-at)].$$

**Case 2:** $t < 0$

No overlap $\Rightarrow u(\lambda)h(t - \lambda) \equiv 0 \ \forall \lambda \Rightarrow y(t) \equiv 0.$
Tutorial 5

Q: Evaluate \( y(t) = x(t) * h(t) \), where \( x(t) \) and \( h(t) \) are as shown.

Solution: The convolution is given by \( y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\lambda)h(t - \lambda) d\lambda \).
Step 1: Equations of \( x(t) \) and \( h(t) \):

\[
x(t) = \begin{cases} 
1, & 0 \leq t \leq 3 \\
0, & \text{elsewhere}
\end{cases} \quad \text{and} \quad h(t) = \begin{cases} 
1, & 0 \leq t \leq 2 \\
0, & \text{elsewhere}
\end{cases}
\]

Step 2: Equations in terms of \( \lambda \):

\[
x(\lambda) = \begin{cases} 
1, & 0 \leq \lambda \leq 3 \\
0, & \text{elsewhere}
\end{cases} \quad \text{and} \quad h(t - \lambda) = \begin{cases} 
1, & 0 \leq t - \lambda \leq 2 \\
0, & \text{elsewhere}
\end{cases} = \begin{cases} 
1, & \left[ \begin{array}{c}
\text{start} \\
\end{array} \right] \leq \lambda \leq \left[ \begin{array}{c}
\text{end}
\end{array} \right] \\
0, & \text{elsewhere}
\end{cases}
\]

Step 3: Integrate to find \( y(t) \).

Case 1: \( t < 0 \) or \( t > 5 \) \( \implies y(t) \equiv 0 \) (no overlap).

Case 2: \( 0 < t \leq 2 \) \( \implies y(t) = \int_0^t dt = t \).

Case 3: \( 2 < t \leq 3 \) \( \implies y(t) = \int_{t - 2}^t dt = 2 \).

Case 4: \( 3 < t \leq 5 \) \( \implies y(t) = \int_{t - 2}^3 dt = 5 - t \).
**Tutorial 6**

**Q:** Find the convolution of the functions \(x(t)\) and \(h(t)\) as shown below.

\[
\begin{array}{c|c}
 & r(t) \\
-5 & 0 1 2 3 4 5 6 \\
\end{array}
\quad t
\quad
\begin{array}{c|c}
 & s(t) \\
-5 & 0 1 2 3 4 5 6 \\
\end{array}
\]

**Solution:** The convolution of \(r(t)\) and \(s(t)\) is given by:

\[
y(t) = r(t) * s(t) = \int_{-\infty}^{\infty} r(\lambda) s(t - \lambda) d\lambda.
\]

**Step 1:** Write the equations of \(r(t)\) and \(s(t)\) as follows:

\[
r(t) = \begin{cases} 
2, & 0 \leq t \leq 2 \\
0, & \text{elsewhere}
\end{cases}
\]

To find the equation of \(s(t)\), we consider 2 points (as it is a straight line):

\((t_1, s_1) = (-2, 0) \text{ and } (t_2, s_2) = (2, 2)\).

\[
\therefore \frac{s - s_1}{t - t_1} = \frac{s_2 - s_1}{t_2 - t_1} \Rightarrow s = \frac{t}{2} + 1 \Rightarrow s(t) = \begin{cases} 
t/2 + 1, & -2 \leq t \leq 2 \\
0, & \text{elsewhere}
\end{cases}
\]

**Step 2:** Write the above equations in terms of \(\lambda\), then sketch \(r(\lambda)\) and \(s(t - \lambda)\):

\[
r(\lambda) = \begin{cases} 
2, & 0 \leq \lambda \leq 2 \\
0, & \text{elsewhere}
\end{cases}
\]

and

\[
s(t - \lambda) = \begin{cases} 
(t - \lambda)/2 + 1, & -2 \leq t - \lambda \leq 2 \\
0, & \text{elsewhere}
\end{cases}
\]

\[
\begin{align*}
\text{Start} & \quad \text{End} \\
(t - \lambda)/2 + 1, & \leq \lambda \leq t + 2 \quad \text{and} \\
0, & \text{elsewhere}
\end{align*}
\]

**Note:**

\[2 \leq t - \lambda \leq 2 \Rightarrow -2 - t - \lambda \leq 2 - t \Rightarrow t - 2 \leq \lambda \leq t + 2\].
Step 3: Integrate to find \( y(t) \). Consider the intervals of overlap between \( r(\lambda) \) and \( s(t - \lambda) \). These intervals can be defined using the endpoints of \( r(\lambda) \) (which is fixed here) and \( s(t - \lambda) \) (which is moving according to the shift \( t \), noting that positive \( t \) gives positive shift).

**Case 1:** End of \( s(t - \lambda) \) \([\text{start} - 2]\) moves right (when \( t \) increases), \( s(t - \lambda) \) moves right as well. The first overlap with \( r(\lambda) \) occurs when the end of \( s(t - \lambda) \) \([\text{start} - 2]\) exceeds the start of \( r(\lambda) \) \([\text{start} - 1]\). Now consider that the end of \( s(t - \lambda) \) is inside \( r(\lambda) \), hence, \( 0 \leq t + 2 \leq 3 \), or equivalently \(-2 \leq t \leq 1\). In this case we have:

\[
y = \int_{0}^{t+2} \frac{[1]}{r(\lambda)} \left[ \frac{(t - \lambda)/2 + 1}{s(t - \lambda)} \right] d\lambda = t^2/4 + t + 1
\]

(where \(-2 \leq t \leq 1\), as shown above).
Case 3: As the end of $s(t - \lambda)$ [i.e., $\lambda = t + 2$] exceeds the end of $r(\lambda)$ [i.e., $\lambda = 3$], $s(t - \lambda)$ will fully overlap $r(\lambda)$ until the start of $s(t - \lambda)$ [i.e., $\lambda = t - 2$] reaches the start of $r(\lambda)$ [i.e., $\lambda = 0$]. Hence, we consider two conditions: $t + 2 > 3$ and $t - 2 < 0$, which give $t > 1$ and $t < 2$, or equivalently $1 < t < 2$. In this case we have:

$$y = \int_{0}^{3} \left[ \frac{1}{r(\lambda)} \left( \frac{(t - \lambda)}{2} + 1 \right) \right] d\lambda = (3/2)t + 3/4$$

(where $1 < t < 2$, as shown above).

Case 4: As the start of $s(t - \lambda)$ [i.e., $\lambda = t - 2$] moves beyond the start of $r(\lambda)$ [i.e., $\lambda = 0$], overlap takes place only between $\lambda = t - 2$ [start of $s(t - \lambda)$] and $\lambda = 3$ [end of $r(\lambda)$]. Hence, we consider the condition $0 \leq t - 2 \leq 3$, or equivalently, $2 \leq t \leq 5$. Here we have:

$$y = \int_{t-2}^{3} \left[ \frac{1}{r(\lambda)} \left( \frac{(t - \lambda)}{2} + 1 \right) \right] d\lambda = 15/4 + t/2 - t^2/2$$

where $2 \leq t \leq 5$, as shown above).
Case 4: $2 \leq t \leq 5$.

Case 5: If the start of $s(t - \lambda)$ [i.e., $\lambda = t - 2$] moves past the end of $r(\lambda)$ [i.e., $\lambda = 3$], no overlap occurs. The condition here is written as $t - 2 \geq 3$, or equivalently $t > 5$. Here $y(t) = 0$.

Case 5: $t \geq 5$, no overlap.

Summary of Results

\[
y(t) = \begin{cases} 
  t^2/4 + t + 1, & -2 \leq t \leq 1 \\
  (3/2)t + 3/4, & 1 < t < 2 \\
  (15/4) + t/2 - t^2/4, & 2 \leq t \leq 5 \\
  0 & \text{elsewhere}
\end{cases}
\]

Q: Solve the above problem as $y(t) = s(t) \ast r(t) = \int_{-\infty}^{\infty} s(\lambda) r(t - \lambda) d\lambda$. You should get the same answer, since convolution is commutative.
Tutorial 7

Q: Show that \( \delta(t) = \lim_{a \to 0} \frac{1}{2a} \Pi_{2a}(t) \) \( [a > 0] \). What is the importance of this result?

Solution: We should prove that the function \( g(t) = \frac{1}{2a} \Pi_{2a}(t) \) (shown below) satisfies the definition and properties of the delta function as \( a \to 0 \).

The delta function is defined as \( \int_{-\infty}^{\infty} x(t) \delta(t - t_o) dt = x(t_o) \) for any continuous function \( x(t) \) [see Tables]. Applying the same integral to \( g(t) \) we get:

\[
\int_{-\infty}^{\infty} x(t)g(t - t_o) dt = \frac{1}{2a} \int_{t_o-a}^{t_o+a} x(t) dt \tag{1}
\]

The **Mean Value Theorem for integrals** states that for any continuous function \( s(t) \) we have:

\[
\int_{c}^{d} s(t) dt = (d - c)s(t_m),
\]

where \( t_m \) is a number between \( c \) and \( d \) \( (c < t_m < d) \). Applying this theorem to Eq. 1 above we get:

\[
\int_{-\infty}^{\infty} x(t)g(t - t_o) dt = \frac{1}{2a} \int_{t_o-a}^{t_o+a} x(t) dt = \frac{1}{2a} [(t_o + a) - (t_o - a)] s(t_m) = x(t_m),
\]

where \( t_o - a < t_m < t_o + a \). As \( a \to 0 \), we have \( t_m \to t_o \); and hence

\[
\int_{-\infty}^{\infty} x(t) \lim_{a \to 0} \{g(t - t_o)\} dt = \lim_{a \to 0} \int_{-\infty}^{\infty} x(t)g(t - t_o) dt = \lim_{a \to 0} \{x(t_m)\} = x(t_o).
\]
The distinguishing Properties of the delta function are evenness, unit area, and spiky shape (Tables). These properties are satisfied by the above function:

**Evenness:** we have \( g(-t) = g(t) \).

**Unit area:** \( \int_{-\infty}^{\infty} g(t) \, dt = (1/2a)(2a) = 1 \).

**Spiky shape:** \( \lim_{a \to 0} g(t) = \lim_{a \to 0} \left\{ \begin{array}{ll} \frac{1}{2a} & |t| < a \\ 0 & \text{elsewhere} \end{array} \right\} = \left\{ \begin{array}{ll} \infty & t = 0 \\ 0 & \text{elsewhere} \end{array} \right\} \).

This result justifies the use of narrow pulse to approximate the delta function in practical applications, noting that it is impossible to generate an exact delta function.

**Tutorial 8**

**Q:** Show that the unit-step response \( \rho(t) \) of any LTI system is related to its impulse response \( h(t) \) by: \( h(t) = \frac{d\rho(t)}{dt} \).

**Solution:** The I/O relation for LTI systems gives: \( \rho(t) = h(t) * u(t) = \int_{-\infty}^{\infty} h(\lambda) \, u(t - \lambda) \, d\lambda \).

The unit-step function is given by:

\[
\begin{align*}
u(\lambda) &= \left\{ \egin{array}{ll} 1, & \lambda > 0 \\ 0, & \text{elsewhere} \end{array} \right\} \\
\end{align*}
\]

Hence, \( u(t - \lambda) = \left\{ \egin{array}{ll} 1, & t - \lambda > 0 \Rightarrow \lambda < \frac{t}{\text{end}} \\ 0, & \text{elsewhere} \end{array} \right\} \).

Now we have \( \rho(t) = \int_{-\infty}^{t} h(\lambda) \, d\lambda \).

Therefore, \( \rho'(t) = h(t) \) [see Tables].

![Diagram](image)
**Tutorial 9**

**Q:** For the periodic square wave $x(t)$ shown below, find:
(A) The complex Fourier series, (B) the trigonometric Fourier series.

![Periodic Square Wave](image)

**Solution:**
(A) Since $x(t)$ is periodic, it has Fourier series expansion [Tables]:

$$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{j2\pi kf_0 t}$$

From the above figure, the signal fundamental frequency is $f_0 = 1/T_o$ Hz. The Fourier coefficients (for $k \neq 0$) are given by [see Tables]:

$$X_k = \frac{1}{T_o} \int_{0}^{T_o} x(t)e^{-j2\pi kf_0 t} dt = \frac{1}{T_o} \int_{0}^{T_o} e^{-j2\pi kf_0 t} dt$$

$$= \left( \frac{1}{T_o} \right) \left( \frac{1}{-j2\pi kf_0} \right) \left[ e^{-j2\pi kf_0 t} \right]_{j0}^{T_o} = \frac{1 - e^{-j2\pi k}}{j2\pi k} = \frac{1 - (-1)^k}{j2\pi k}$$

[Note that we used $f_o T_o = 1$ and $e^{-j\pi} = \cos(\pi) - j\sin(\pi) = -1$.]

For $k = 0$ we have: $X_0 = \frac{1}{T_o} \int_{0}^{T_o} x(t) dt = \frac{1}{2}$.

Now:

1. If $k$ is even, i.e., $k = 2n$, then $X_k = X_{2n} = 0$ (if $n \neq 0$)
2. If $k$ is odd, i.e., $k = 2n + 1$, then $X_k = X_{2n+1} = \frac{1}{j\pi(2n+1)}$.

$$\therefore \quad x(t) = \frac{1}{2} + \sum_{n=-\infty}^{\infty} \frac{1}{2n+1} e^{j(2n+1)2\pi f_0 t}$$

(B) From above we have:

$$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{j2\pi kf_0 t} = X_0 + \sum_{k=1}^{\infty} \left[ X_k e^{j2\pi kf_0 t} + X_{-k} e^{-j2\pi kf_0 t} \right]$$

$$= a_0 + \sum_{k=1}^{\infty} \left\{ a_k \cos(2\pi kf_0 t) + b_k \sin(2\pi kf_0 t) \right\}$$
where we used Euler’s formula to get:

\[ a_0 = X_0, \quad a_k = X_k + X_{-k}, \quad b_k = j(X_k - X_{-k}). \]

Hence, \( a_0 = X_0 = 1/2. \)

If \( k \) is even \((k \neq 0)\), then \( a_k = b_k = 0 \) (see above).

If \( k \) is odd, then:

\[ a_k = a_{2n+1} = X_{(2n+1)} + X_{-(2n+1)} = 0 \quad \text{and} \quad b_k = b_{2n+1} = \frac{2}{(2n + 1)\pi}. \]

Therefore, we can write the signal \( x(t) \) as follows:

\[
x(t) = \frac{1}{2} + \sum_{n=0}^{\infty} b_{2n+1} \sin(2\pi(2n + 1)f_0 t) \\
= \frac{1}{2} + \frac{2}{\pi} \left[ \sin(\omega_0 t) + \frac{1}{3} \sin(3\omega_0 t) + \frac{1}{5} \sin(5\omega_0 t) + \cdots \right].
\]

Q: If \( x(t) \) above is passed through the system shown below, find the output \( y(t) \) and the average power in \( x(t) \) and \( y(t) \).

Ans. \( P_x = 0.5; \ P_y = 0.47. \)

\[
\begin{array}{c}
x(t) \\
\downarrow \\
\text{LPF, } H(f) \\
\downarrow \\
y(t)
\end{array}
\]

**Tutorial 10**

Q1: Show that \( x(t)\delta(t - t_o) = x(t_o)\delta(t - t_o). \)

**Solution:** Let \( g(t) \) be any function that is continuous at \( t = t_o \), and let \( \phi(t) = g(t)x(t) \). Then using Tables we get:

\[
\int_{-\infty}^{\infty} g(t)[x(t)\delta(t - t_o)]dt = \int_{-\infty}^{\infty} [g(t)x(t)]\delta(t - t_o)dt = \int_{-\infty}^{\infty} \phi(t)\delta(t - t_o)dt = \phi(t_o).
\]
Now: \( \phi(t_0) = g(t_0)x(t_0) = x(t) \int_{-\infty}^{\infty} g(t) \delta(t-t_o) dt = \int_{-\infty}^{\infty} g(t) [x(t_0) \delta(t-t_0)] dt. \)

Hence we have: \( \int_{-\infty}^{\infty} g(t) [x(t) \delta(t-t_o)] dt = \int_{-\infty}^{\infty} g(t) [x(t_0) \delta(t-t_0)] dt \)

Since \( g(t) \) is arbitrary, the above equation implies that:

\[
 s(t) = r(t), \quad \text{or} \quad x(t) \delta(t-t_0) = x(t_0) \delta(t-t_0). 
\]

Q2: Consider the cascaded and parallel systems shown in Figs. a and b. Find the equivalent impulse responses for the equivalent system shown in Fig. c.

**Solution:**

(a) Let \( y_1(t) = x(t) * h_1(t) \).

\[
 y(t) = y_1(t) * h_2(t) = [x(t) * h_1(t)] * h_2(t) = x(t) * [h_1(t) * h_2(t)] \Rightarrow h(t)
\]

\[
 = h_1(t) * h_2(t)
\]

where we used the *associative* property of convolution.

(b) Let \( y_1(t) = x(t) * h_1(t), y_2(t) = x(t) * h_2(t) \).

\[
 y(t) = y_1(t) + y_2(t) = x(t) * h_1(t) + x(t) * h_2(t) = x(t) * [h_1(t) + h_2(t)] \Rightarrow h(t)
\]

\[
 = h_1(t) + h_2(t)
\]

where we used the *distributive* property of convolution.
**Tutorial 11**

**Q:** For the periodic pulse train \( p(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT) \) find:
(A) Complex Fourier series (FS), (B) trigonometric FS, (C) Fourier Transform (FT).

**Solution:**

(A) \( p(t) = \sum_{k=-\infty}^{\infty} P_k e^{j2\pi kf_0 t} \), where \( \omega_o = 2\pi f_o = \frac{2\pi}{T_o} \) is the fundamental frequency, \( T_o \) being the fundamental period, and the Fourier coefficients are given by [see Tables]:

\[
P_o = \frac{1}{T_o} \int_{-T_o/2}^{T_o/2} p(t) dt = \frac{1}{T_o} \int_{-T_o/2}^{T_o/2} \delta(t) dt = \frac{1}{T_o}
\]
\[
P_k = \frac{1}{T_o} \int_{-T_o/2}^{T_o/2} p(t) e^{-j2\pi kf_0 t} dt = \frac{1}{T_o} \int_{-T_o/2}^{T_o/2} \delta(t) e^{-j2\pi kf_0 t} dt = \frac{1}{T_o}
\]

\[p(t) = \frac{1}{T_o} \sum_{k=-\infty}^{\infty} e^{j2\pi kf_0 t}\]

(B) \( p(t) = a_o + \sum_{k=1}^{\infty} [a_k \cos(k\omega_o t) + b_k \sin(k\omega_o t)] \), where [see Tables]:

\[a_o = P_o = \frac{1}{T_o}, \quad a_k = P_k + P_{-k} = \frac{2}{T_o}, \quad b_k = j(P_k - P_{-k}) = 0.
\]

\[p(t) = \frac{1}{T_o} + \frac{2}{T_o} \sum_{k=1}^{\infty} \cos(k\omega_o t).
\]

(C) Using part (A) we have [using Tables]:

\[
\mathcal{F}\{p(t)\} = \mathcal{F}\left\{ \frac{1}{T_o} \sum_{k=-\infty}^{\infty} e^{j2\pi kf_0 t} \right\} = \frac{1}{T_o} \sum_{k=-\infty}^{\infty} \mathcal{F}\{e^{j2\pi kf_0 t}\}
\]
\[
= \frac{1}{T_o} \sum_{k=-\infty}^{\infty} \delta(f - kf_0) = f_o \sum_{k=-\infty}^{\infty} \delta(f - kf_0)
\]

Hence, a pulse train in the time domain is Fourier-transformed to a pulse train in the frequency domain.
**Tutorial 12**

**Q1:** Show that $x(t)\cos\omega_0 t \xrightarrow{F} \frac{1}{2}X(f-f_0) + \frac{1}{2}X(f+f_0)$.

**Solution:** From Tables we have:

$$\cos\omega_0 t \xrightarrow{F} \frac{1}{2}\delta(f-f_0) + \frac{1}{2}\delta(f+f_0).$$

Let $c(t) = \cos\omega_0 t$, $C(f) = \mathcal{F}\{\cos\omega_0 t\}$. Using Tables we have:

$$F\{x(t) \cdot c(t)\} = X(f) \ast C(f) = X(f) \ast \left[\frac{1}{2}\delta(f-f_0) + \frac{1}{2}\delta(f+f_0)\right] = \frac{1}{2}X(f-f_0) + \frac{1}{2}X(f+f_0)$$

where we used the relation $X(f) \ast \delta(f-f_0) = X(f-f_0)$ from Tables.

**Q2:** (A) Find $\mathcal{F}\{x(t) = e^{-|t|}, a > 0\}$. (B) Using part (A), find $\mathcal{F}\left\{\frac{2}{1+r^2}\right\}$.

**Solution:**

(A)

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt = \int_{-\infty}^{\infty} e^{-a|t|} e^{-j2\pi ft} dt = \int_{-\infty}^{0} e^{at} e^{-j2\pi ft} dt + \int_{0}^{\infty} e^{-at} e^{-j2\pi ft} dt = \frac{1}{a-j2\pi f} + \frac{1}{a+j2\pi f} = \frac{2a}{a^2 + (2\pi f)^2}.$$

(B)

$$x(t) \xrightarrow{F} X(f) \Rightarrow X(t) \xrightarrow{F} x(-f) \quad [\text{duality (Tables)}]$$

$$\Rightarrow X(t) \xrightarrow{F} x(f) \quad [\text{if } x(t) \text{ is even}]$$
Using duality of FT and part (A) above we get: \[ \mathcal{F}\left\{ \frac{2}{1+(2\pi t)^2} \right\} = e^{-|f|}. \]

From Tables (scaling property) we have: \( s(kt) \leftrightarrow \frac{1}{|k|} S\left(\frac{f}{k}\right) \).

Applying this to the above result we get:

\[ \mathcal{F}\left\{ \frac{2}{1+(t)^2} \right\} = 2\pi e^{-2\pi|f|} \]

**Tutorial 13**

**Q:** Determine the impulse response of the ideal bandstop filter whose frequency response is shown below.

![Frequency response graph](image)

**Solution:** It is better that we arrange \( H(f) \) in terms of well-known functions. In this question it can be written as follows:

\[ H(f) = 1 - G(f), \]

where \( G(f) \) is shown below.

To find an explicit formula for \( G(f) \), let \( f_0 = \frac{f_1+f_2}{2}, B = f_2 - f_1 \).

\[ G(f) = \prod_{B} (f - f_0) + \prod_{B} (f + f_0) = X(f - f_0) + X(f + f_0), \]

where \( X(f) = \prod_{B}(f) \).

Now from Tables we have:

1. \( \text{sinc}(Lt) \xrightarrow{\mathcal{F}} \frac{1}{L} \prod_{L}(f) \Rightarrow L \cdot \text{sinc}(Lt) \xrightarrow{\mathcal{F}} \prod_{L}(f) \) [multiply both sides by \( L \)]

   Hence, for this question we have: \( B \text{sinc}(Bt) \xrightarrow{\mathcal{F}} \prod_{B}(f). \)

2. \( x(t)\cos(2\pi f_0 t) \xrightarrow{\mathcal{F}} \frac{1}{2} X(f - f_0) + \frac{1}{2} X(f + f_0), \)

\[ \therefore \quad h(t) = F^{-1}\{1 - G(f)\} = F^{-1}\{1\} - F^{-1}\{G(f)\} = \delta(t) - g(t) \]

\[ = \delta(t) - 2B \text{sinc}(Bt) \cos(2\pi f_0 t). \]
**Tutorial 14**

**Q:** (A) Determine the impulse response of the ideal band-stop filter whose frequency response is shown in Fig. 1 below.

(B) Determine the impulse response of the ideal band-stop filter whose frequency response is shown in Fig. 2 below.

**Solution:**

(A) The frequency response $H(f)$ can be written as:

$$H(f) = 1 - G(f),$$

where $G(f) = \frac{Kc^2}{f}$ [see the figure below].

From Tables we have: $\Lambda_{T}(t) \leftrightarrow \frac{2}{\pi} \text{sinc}^2(\frac{f}{2}t)$, and:

$$X(t) \leftrightarrow X(f) \Rightarrow X(t) \leftrightarrow x(-f) \quad \text{[duality (Tables)]}$$

$$\Rightarrow X(t) \leftrightarrow x(f)[\text{If } x(t) \text{ is even}]$$

$$\Rightarrow \frac{B}{2} \text{sinc}^2(\frac{B}{2}t) \leftrightarrow \Lambda_{B}(f) \Rightarrow \frac{2f_c}{2} \text{sinc}^2(\frac{2f_c}{2}t) \leftrightarrow \Lambda_{2f_c}(f)$$

$$\Rightarrow h(t) = \mathcal{F}^{-1}\{H(f)\} = \mathcal{F}^{-1}\{1 - G(f)\} = \delta(t) - f_c \text{sinc}^2(f_c t).$$
In this part, the frequency response reminds us of the modulation property. It can be written as:

\[ H(f) = G(f - f_c) + G(f + f_c). \]

Hence, using Tables or Tutorial 13 we get:

\[ h(t) = 2g(t)\cos(2\pi f_c t) = 2f_c \text{sinc}^2(f_c t) \cdot \cos(2\pi f_c t). \]

**Tutorial 15**

**Q:** Show that:

(A) \( x(t) \cdot y(t) \xrightarrow{\mathcal{F}} X(f) \ast Y(f) \)

(B) \( x(t) \ast y(t) \xrightarrow{\mathcal{F}} X(f) \cdot Y(f) \)

**Solution:**

(A)

\[
\mathcal{F} \{x(t)y(t)\} = \int_{-\infty}^{\infty} x(t)y(t)e^{-j2\pi ft} dt
\]

\[= \int_{-\infty}^{\infty} \mathcal{F}^{-1}\{X(f)\}y(t)e^{-j2\pi ft} dt
\]

\[= \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} X(\lambda)e^{j2\pi \lambda t} d\lambda \right] y(t)e^{-j2\pi ft} dt
\]

\[= \int_{-\infty}^{\infty} X(\lambda) \left[ \int_{-\infty}^{\infty} y(t)e^{-j2\pi(f-\lambda)t} dt \right] d\lambda
\]

\[= \int_{-\infty}^{\infty} X(\lambda)Y(f - \lambda)d\lambda = X(f) \ast Y(f).
\]
(B)

\[
\mathcal{F}\{x(t) * y(t)\} = \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} x(\lambda) y(t - \lambda) d\lambda \right] e^{-j2\pi ft} dt \\
= \int_{-\infty}^{\infty} x(\lambda) \left[ \int_{-\infty}^{\infty} y(t - \lambda) e^{-j2\pi ft} d\lambda \right] d\lambda \\
= \int_{-\infty}^{\infty} x(\lambda) \left[ \int_{-\infty}^{\infty} y(\nu) e^{-j2\pi f(\nu + \lambda)} d\nu \right] d\lambda \\
= \left( \int_{-\infty}^{\infty} x(\lambda) e^{-j2\pi f\lambda} d\lambda \right) \left( \int_{-\infty}^{\infty} y(\nu) e^{-j2\pi f\nu} d\nu \right) \\
[\text{since } \lambda \text{ and } \nu \text{ are independent}] \\
= \mathcal{F}\{x(t)\} \cdot \mathcal{F}\{y(t)\} = X(f) \cdot Y(f)
\]

**Tutorial 16**

**Q:** Find the s-domain voltage transfer function and the voltage impulse response for the analog systems shown below. Assume zero initial conditions (IC’s).

![Electrical Circuit Diagrams](image)

**Solution:**

\[
v_L(t) = L\frac{di(t)}{dt} \quad \text{Tables} \quad V_L(s) = L[sI(s) - i(0^-)] = Ls \cdot I(s) \quad [\text{zero IC’s}].
\]

Hence, inductor impedance is represented in the complex-frequency domain as:

\[
X_L(s) = \frac{V_L(s)}{I(s)} = sL \quad [\text{zero IC’s}].
\]
Now \( v_R(t) = R \cdot i(t) \overset{\mathcal{L}}{\longrightarrow} V_R(s) = RI(s) \Rightarrow X_R(s) = V_R(s)/I(s) = R. \)

\[ i_C(t) = C \frac{dv_C(t)}{dt} \overset{\mathcal{L}}{\longrightarrow} I_C(s) = C \cdot [sV_C(s) - v_C(0^-)] = Cs \cdot V_C(s) \quad \text{[zero IC’s]} \]

\[ \Rightarrow X_C(s) = V_C(s)/I_C(s) = 1/sC. \]

To summarize impedance transformations to the s-domain (zero IC’s):

\( R \Rightarrow R; \ L \Rightarrow sL; \ C \Rightarrow 1/(sC). \)

Now we can transform the above circuits to the s-domain using \( \mathcal{L} \)-Tables:

(a) \( H(s) = \frac{V_o(s)}{V_i(s)} = 2/[4s + 2] \Rightarrow h(t) = \mathcal{L}^{-1}\{\{1/2\}[1/(s + 0.5)]\} = 0.5e^{-0.5t} \)

(b) \( H(s) = \frac{2}{4s^2 + 1} = 1 - \frac{1}{s^2 + 1} \Rightarrow h(t) = \delta(t) - (1/2)e^{-\frac{t}{2}}. \)

(c) \( H(s) = \frac{1/2s^2}{(1/2s)^2 + 1} = \frac{1}{4s + (1/4)} \Rightarrow h(t) = \frac{1}{4}e^{-\frac{t}{4}}. \)

**Tutorial 17**

**Q:** Find the current and voltage across the capacitor below after the switch is closed at \( t = 0. \)
Solution: Let the capacitor initial voltage be \( V_o \) (volts). After the switch is closed, a d.c. voltage of \( V \) volts will suddenly be applied, hence, \( v_i(t) = Vu(t) \) volts.

\[
i(t) = i_C(t) = C \frac{dv_C(t)}{dt} \rightarrow I(s) = C \cdot [sV_C(s) - v_C(0^-)] = C[sV_C(s) - V_o],
\]

or simply:

\[
I = C \cdot [sV_C - V_o] \tag{1}
\]

\[
v_c = v_i - iR \rightarrow V_C = V_i - IR = V/s - IR \tag{2}
\]

From (1), (2) we get:

\[
I = \frac{1}{R} \frac{V - V_o}{s + 1/RC} = I_o \frac{1}{s + 1/RC} \tag{3}
\]

[where \( I_o = (V - V_o)/R \).]

From Tables: \( i(t) = I_o e^{-t/RC} u(t) \) amperes.

Substituting (3) in (2) we get: \( V_C = V \left[ \frac{1}{s} - \frac{1}{s+1/RC} \right] + \frac{V_o}{s+1/RC} \)

From Tables: \( v_c(t) = [V(1 - e^{-t/RC}) + V_o e^{-t/RC}] u(t) \) volts.

**Tutorial 18**

**Q:** The switch S in Fig. 1 below has been closed for a long time. Find the current across the capacitor after the switch is opened at \( t = 0 \).

\[
\begin{align*}
V \text{ (d.c.)} & \quad + \quad R_1 \quad - \\
& \quad + \quad R_2 \quad - \\
& \quad i(t) \\
& \quad + \quad v_c \\
& \quad - \quad v_L \\
& \quad - \quad v_i \\
\end{align*}
\]

**Solution:** As the switch was closed for a long time, both \( v_C(0^-) \) and \( i_L(0^-) \) are zero. Even if there was an initial voltage \( V_o \) on \( C \) and an initial current \( I_o \) in \( L \) when \( S \) was closed, the final current and voltages in the right loop will all be zero. This is due to the consumption of energy by \( R_2 \). To prove this, consider the right loop where the switch has been closed as shown in Fig. 2.
Let $R = R_2$. Now $V_R = IR$.

$$i(t) = C \frac{dv_C}{dt} \to i(s) = C[sV_C(s) - v_C(0^-)] \to I = C[sV_C - V_0] \to V_C = \frac{I}{Cs} + \frac{V_0}{s}.$$  

$$v_L(t) = L \frac{di(t)}{dt} \to v_L(s) = L[sI(s) - i(0^-)] \to V_L = L[sI - Io].$$

$$V_C + V_L + V_R = 0 \to I = \frac{LCIo s - CV_o}{LCs^2 + RCS + 1}.$$

$$I = Io \frac{s - \frac{V_o}{Io}}{s^2 + R s + \frac{1}{LC}} = Io \frac{s - \frac{V_o}{Io} + \left(\frac{R}{2L} - \frac{R}{2L}\right)}{(s + \frac{R}{2L})^2 + \left(\frac{1}{LC}\right)} = Io \left[\frac{s + a}{F(s)} - \frac{\omega_o}{F(s)}\right].$$

where

$$a = \frac{R}{2L}, \quad \omega_o^2 = \frac{1}{LC} - \frac{R^2}{4L^2}, \quad F(s) = (s + a)^2 + \omega_o^2, \quad \xi = \frac{\left(V_o/LIo\right) + \frac{R}{2L}}{\omega_o}.$$

$$i(t) = Io e^{-at} [\cos(\omega_o t) - \xi \sin(\omega_o t)] u(t).$$

If $I_0$, then $i(t) = -(V_o/L\omega_o) e^{-at} \sin(\omega_o t) u(t)$.

If $D = \frac{R^2}{L^2} - \frac{4}{LC} > 0$, we use partial fractions to get:

$$i(t) = [k_1 e^{-bt_1} + k_2 e^{-bt_2}] u(t), \quad b_1, b_2 = \frac{R}{2L} \pm \frac{1}{2} \sqrt{\frac{R^2}{L^2} - \frac{4}{LC} > 0},$$

while $k_1 = -\lambda l$.

$$k_1 = Io (k - b_1)/b \quad (\text{where } k = V_o/(Li_o), b = b_2 - b_1)$$

$k_2 = Io (k - b_2)/b$ for $Io \neq 0$.

$b$ (where $\lambda = V_o/L$) and $k_2 = \lambda b$ for $Io = 0$.

Unless $R = 0$, the expressions for $i(t)$ tend to zero after a long time, so are the expressions for $v_L(t) = Ldi(t)/dt$ and $v_R = i(t)R$. Hence, $v_C = -(v_L + v_R) \to 0$ as well.
Now back to the main question. Let \( R = R_1 + R_2 \). We have \( v_i(t) = Vu(t) \rightarrow V_i(s) = 1/s \). Using \( \mathcal{L} \) rules with zero IC’s we get:

\[
V_C = I/(Cs), \quad V_L = LsI.
\]

\[
v_i(t) = v_R + v_C + v_L \rightarrow 1/s = IR + I/(Cs) + LsI
\]

\[
I = \frac{(1/L)}{s^2 + (R/L)s + 1/(LC)}
\]

If \( D < 0 \), we get: \( I = (1/\omega_o L)[\omega_o / F(s)] \rightarrow i(t) = (1/\omega_o L)e^{-\omega_o t} \sin(\omega_o t)u(t) \).
If \( D > 0 \), we get \( I = (1/L)[a_1/(s + b_1) + a_2/(s + b_2)] \) where \( a_1 = 1/b, \quad a_2 = -1/b \), hence

\[
i(t) = (1/bL)[e^{-b_1t} - e^{-b_2t}]u(t).
\]

**Tutorial 19**

**Q1**: Find the transfer function of the feedback system shown below. All signals are voltages.

![Diagram of Feedback System](image)

**Solution**: First, we define important auxiliary points [i.e., \( e(t) \) and \( r(t) \)] as shown below.

![Diagram of Feedback System with Auxiliary Points](image)

Now we write the system equations as follows:

\[
e(t) = x(t) + r(t)
\]  

(1)

This equation is in the time-domain. It is easier to transform to the s-domain using \( \mathcal{L} \), since the convolution in the t-domain would be transformed into a simple multiplication in the s-domain. Taking the \( \mathcal{L} \) of both sides of (1) we get:

\[
E(s) = X(s) + R(s)
\]  

(2)
Similarly,

\[ R(s) = G(s)Y(s) \quad (3) \]

\[ Y(s) = D(s)E(s) \quad (4) \]

Using (2), (3), and (4) we get: 
\[ Y(s)[1 - D(s)G(s)] = D(s)X(s) \]

\[ \therefore H(s) = \frac{Y(s)}{X(s)} = \frac{D(s)}{1 - D(s)G(s)}. \]

Q2: Find the transfer function of the following feedback system.

\[ \begin{align*}
X(s) & \quad + \quad R(s) \quad + \quad D(s) \\
\text{x(t)} & \quad + \quad G(s) \\
Y(s) & \quad = \quad \text{y(t)}
\end{align*} \]

Tutorial 20

Q: The relationship between \( \mathcal{F}\{x(t)\} \) and \( \mathcal{L}\{x(t)\} \) is given by:

\[ F\{x(t)\} = L\{x(t)\}\big|_{s=j\omega} \]

or \( X(\omega) = X(s)\big|_{s=j\omega} \)

where \( \omega = 2\pi f \). This holds on the condition that:

\[ \int_{-\infty}^{\infty} |x(t)|dt < \infty \quad \text{[i.e., } x(t) \text{ is absolutely integrable].} \]

Now decide whether \( X(\omega) \) can be found from \( X(s) \) for the functions:

1. \( \delta(t) \), 2. \( u(t) \), 3. \( t^n u(t) \), \( n \) is integer \( \geq 0 \).

Solution:

1. \( \int_{-\infty}^{\infty} |x(t)|dt = \int_{-\infty}^{\infty} \delta(t)dt = 1 < \infty \). Hence, \( X(\omega) = X(s)\big|_{s=j\omega} \).

From Tables: \( \Delta(s) = L\{\delta(t)\} = 1. \therefore \Delta(\omega) = \Delta(s)\big|_{s=j\omega} = 1. \)

2. \( \int_{-\infty}^{\infty} |u(t)|dt = \int_{0}^{\infty} u(t)dt = \int_{0}^{\infty} 1.dt \rightarrow \infty. \)

Hence, we cannot use \( X(\omega) = X(s)\big|_{s=j\omega} \).

Note that the Laplace transform of \( u(t) \) is: \( U(s) = \frac{1}{s} \) (Tables), and the Fourier transform of \( u(t) \) is: \( U(f) = \frac{1}{2} \delta(f) + \frac{1}{j2\pi f} \) (Tables).
Therefore, \( U(\omega) = \pi \delta(\omega) + \frac{1}{j\omega} \) [Using the relation \( \delta(2\pi f) = \frac{1}{2\pi} \delta(f) \) from Tables.]

It is apparent that \( U(\omega) \neq U(s)|_{s=j\omega} \).

3. The function \( t^nu(t) \) has no Fourier transform, but it has Laplace transform given by \( \frac{n!}{s^{n+1}} \).

**Tutorial 21**

**Q:** Expand the following expression using partial fraction expansion, then find \( y(t): Y(s) = \frac{2s^2 + 13s + 12}{(s+1)(s^2+5s+6)} \).

**Solution:** To find the inverse Laplace transform \( (L^{-1}) \) for an expression of the form:

\[
Y(s) = \frac{N(s)}{D(s)} = \frac{a_0 + a_1s + \cdots + a_ms^m}{(s - p_1)(s - p_2) \cdots (s - p_n)},
\]

we apply partial fraction expansion rules as follows:

\[
Y(s) = \frac{c_1}{s - p_1} + \frac{c_2}{s - p_2} + \cdots + \frac{c_n}{s - p_n}.
\]

**Notes:**

1. If \( m = \text{degree} \ [N(s)] \geq n = \text{degree} \ [D(s)] \), perform long division first.
2. For complex conjugate poles \( p_{k+1} = p_k^* \) we have \( c_{k+1} = c_k^* \).
3. For a pole of multiplicity \( r \) at \( s = p_k \) there will be \( r \) terms: \( c_{k1}(s - p_k) + c_{k2}(s - p_k)^2 + \cdots + c_{kr}(s - p_k)^r \) in the expansion.

Now back to the main question.

\[
Y(s) = \frac{2s^2 + 13s + 12}{(s+1)(s^2+5s+6)} \Rightarrow s^2 + 5s + 6 = 0 \Rightarrow p_{1,2} = \frac{-5 \pm \sqrt{5^2 - 4 \times 6}}{2} = -2, -3.
\]

Hence, \( Y(s) = \frac{2s^2 + 13s + 12}{(s+1)(s^2+2)(s+3)} \). Let \( Y(s) = \frac{a}{s+1} + \frac{b}{s+2} + \frac{c}{s+3} \).

\[
\therefore \frac{2s^2 + 13s + 12}{(s+1)(s+2)(s+3)} = \frac{a}{s+1} + \frac{b}{s+2} + \frac{c}{s+3}
\]

To find \( a \) do the following:

1. Multiply by \((s+1)\) \( \text{\small \frac{2s^2 + 13s + 12}{(s+1)(s+2)(s+3)}} = a + \frac{b(s+1)}{s+2} + \frac{c(s+1)}{s+3} \).
2. Put \( s = -1: \frac{2-13+12}{(-1+2)(-1+3)} = a + 0 + 0 \Rightarrow a = \frac{1}{2} \).
Similarly we get: $b = 6$ and $c = -\frac{9}{2}$.

From Tables we find $\mathcal{L}^{-1}$ of $Y(s)$ as follows:

$$y(t) = ae^{-t} + be^{-2t} + ce^{-3t} = \frac{1}{2}e^{-t} + 6e^{-2t} - \frac{9}{2}e^{-3t}.$$

**MATLAB:** The above question can be solved on MATLAB® as follows:

$$A = \begin{bmatrix} 2 & 13 & 12 \\ \end{bmatrix}; \quad B = \begin{bmatrix} 1 & 6 & 11 & 6 \\ \end{bmatrix}; \quad [R, P, K] = \text{residue}(A, B)$$

where $R$ gives the coefficients $c_i$, $P$ gives the poles $p_i$, and $K$ is empty if degree($A$) < degree($B$). Note that $B$ can be obtained using $B = \text{conv}([1 1], [1 5 6])$ (multiplication of two polynomials).

**Tutorial 22**

**Q:** A linear electrical system is described by the second-order differential equation:

$$y''(t) + 5y'(t) + 6y(t) = x(t) \quad (1)$$

with initial conditions:

$$y(0) = 2; \quad y'(0) = 1.$$

If the input voltage is given by $x(t) = e^{-t} u(t)$, find the output $y(t)$.

**Solution:** From $\mathcal{L}$ Tables we have:

$$y'(t) \xrightarrow{\mathcal{L}} sY(s) - y(0^-) = sY(s) - 2$$

$$y''(t) \xrightarrow{\mathcal{L}} s[sY(s) - y(0^-)] - y'(0^-) = s^2 Y(s) - 2s - 1$$

$$x(t) = e^{-t} u(t) \xrightarrow{\mathcal{L}} X(s) = \frac{1}{s + 1}$$

Taking $\mathcal{L}$ of both sides of Eq. 1, we get:

$$[s^2 Y(s) - 2s - 1] + 5[sY(s) - 2] + 6Y(s) = \frac{1}{s + 1}$$

$$Y(s)[s^2 + 5s + 6] = 2s + 11 + \frac{1}{s + 1} = \frac{2s^2 + 13s + 12}{s + 1}$$

$$\therefore Y(s) = \frac{2s^2 + 13s + 12}{(s + 1)(s^2 + 5s + 6)}$$

$$\therefore y(t) = \frac{1}{2}e^{-t} + 6e^{-2t} - \frac{9}{2}e^{-3t} \quad \text{(From Tutorial 21)}$$
**Tutorial 23**

**Q:** Find the autocorrelation function of \( x(t) = \prod_1(t - 0.5) \).

Note that \( x(t) \) is a **deterministic** signal.

**Solution:** From Tables: \( R(\tau) = \int_{-\infty}^{\infty} x(\lambda)x(\tau + \lambda)d\lambda \). Note that \( R(\tau) = x(\tau) * x(-\tau) \).

We follow the same steps that are necessary to find the convolution of two signals.

From Tables: \( \prod_T(t) = \begin{cases} 1, & -T/2 \leq t \leq T/2 \\ 0, & \text{elsewhere} \end{cases} \)

\[ \therefore x(\lambda) = \prod_1(\lambda - 0.5) = \begin{cases} 1, & -0.5 \leq \lambda - 0.5 \leq 0.5 \\ 0, & \text{elsewhere} \end{cases} = \begin{cases} 1, & 0 \leq \lambda \leq 1 \\ 0, & \text{elsewhere} \end{cases} \]

\[ \therefore x(\tau + \lambda) = \begin{cases} 1, & 0 \leq \tau + \lambda \leq 1 \\ 0, & \text{elsewhere} \end{cases} = \begin{cases} 1, & \frac{-\tau}{\text{start}} \leq \lambda \leq \frac{1 - \tau}{\text{end}} \\ 0, & \text{elsewhere} \end{cases} \]

Note that since we have \( x(\tau + \lambda) \) inside the integral, a positive \( \lambda \)-shift \( \tau \) will give negative shift to the function w.r.t the y-axis (contrary to the case of convolution).

Now we move the function \( x(\tau + \lambda) \) from left to right while \( x(\lambda) \) is fixed. We get the following cases:

**Case 1:** \( 1 - \tau < 0 \) (i.e., \( \tau > 1 \)) \( \Rightarrow R(\tau) = 0 \) [no overlap].

**Case 2:** \( 0 \leq 1 - \tau < 1 \) (i.e., \( 0 < \tau \leq 1 \)) \( \Rightarrow R(\tau) = \int_0^{1-\tau} d\lambda = 1 - \tau \).

**Case 3:** \( 0 \leq -\tau \leq 1 \) (i.e., \( -1 \leq \tau \leq 0 \)) \( \Rightarrow R(\tau) = \int_{-\tau}^{1} d\lambda = 1 + \tau \).

**Case 4:** \( -\tau > 1 \) (i.e., \( \tau < -1 \)) \( \Rightarrow R(\tau) = 0 \) [no overlap].
Notice that $R(\tau)$ is **Case even**, with an **Case absolute maximum at** $\tau = 0$.

**MATLAB:** The above question can be solved on MATLAB® as follows:

\[
x = \text{stepfun}(t, 0) - \text{stepfun}(t, 1); \quad y = \text{xcorr}(x) \ast Ts;
\]

**Tutorial 24**

**Q:** A white noise signal $n(t)$ is passed through a $RC$ circuit as shown. Find the autocorrelation function of the output noise.

![RC Circuit Diagram]

**Solution:**

\[
H(f) = \frac{X_C}{R + X_C} = \frac{\frac{1}{j\omega C}}{R + \frac{1}{j\omega C}} = \frac{1}{1 + j2\pi fCR},
\]

\[
G_n(f) = \eta/2 \xrightarrow{\text{WKT}} R_n(\tau) = \frac{\eta}{2} \delta(\tau),
\]

\[
G_{n_0}(f) = |H(f)|^2 G_n(f) = \frac{\eta/2}{1 + 4\pi^2 f^2 R^2 C^2} = \frac{\eta}{2} \frac{1}{1 + \frac{4\pi^2 f^2}{R^2 C^2}}
\]

\[
= \frac{\eta}{4RC} \frac{2 \frac{1}{R^2}}{(\frac{1}{R^2})^2 + (2\pi f)^2}.
\]

\[\therefore \quad R_{n_0}(\tau) = F^{-1}\{G_{n_0}(f)\} = \frac{\eta}{4RC} e^{-\frac{|\tau|}{4RC}} [\text{Tables; Wiener–Kinchin Theorem}].\]

Note that $R_\eta(\tau) = \mathcal{E}\{n(t)n(t + \tau)\}$, where $\mathcal{E}$ is the statistical expectation functional [Compare with the deterministic signal in Tutorial 22].

![Autocorrelation Function Graph]
**Tutorial 25**

**Q1:** A random telegraph signal \( x(t) \) with autocorrelation function \( R_x(\tau) = e^{-a|\tau|} \), \( a \) being a constant, is applied to an RC circuit as shown. Find the power spectral density (PSD) of the output random signal \( y(t) \).

![Circuit Diagram]

**Solution:** We have: \( H(f) = \frac{1}{1 + j(2\pi f RC)} \)

\[
\therefore \quad G_x(f) = F\{R_x(\tau)\} = F\{e^{-a|\tau|}\} = \frac{2a}{a^2 + (2\pi f)^2} \quad \text{[Tables; using WKT].}
\]

\[
\therefore \quad G_y(f) = |H(f)|^2 G_x(f) = \left\{ \frac{1}{1 + (2\pi f)^2} \right\} \left\{ \frac{2a}{a^2 + (2\pi f)^2} \right\}.
\]

**Q2:** A random noise current signal \( x(t) \) has a **Case double-sided** PSD normalized to unit resistance given by:

\[
G_x(f) = e^{-0.01|f|} \quad \text{W/Hz/} \Omega \quad (\text{Amp}^2/\text{Hz}).
\]

The signal is passed through an ideal BPF with lower cutoff frequency at 1 kHz and upper cutoff frequency at 5 kHz as shown. Determine the noise power at the output if the load resistance is 100 \( \Omega \).

![BPF Diagram]

**Solution:** The normalized output power is:

\[
P_y = \int_{-\infty}^{\infty} G_y(f) df = \int_{-\infty}^{\infty} G_x(f)|H(f)|^2 df = 2 \int_{1k}^{5k} e^{-0.01f} df = 0.008 \text{ W/} \Omega.
\]

The total output noise power is 100 \( P_y = 0.8 \text{ W.} \)
**Tutorial 26**

**Q:** The autocorrelation function of a signal \(x(t)\) is defined as follows:

\[
R_x(\tau) = \begin{cases} 
\int_{-\infty}^{\infty} x(\lambda)x(\tau + \lambda)d\lambda, & \text{If } x \text{ is an energy signal} \\
\frac{T}{T_0} \int_{-T_0/2}^{T_0/2} x(\lambda)x(\tau + \lambda)d\lambda, & \text{If } x \text{ is a periodic power signal} \\
\mathcal{E}\{x(t)x(t+\tau)\}, & \text{If } x \text{ is WSS random signal} \\
\lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(\lambda)x(\tau + \lambda)d\lambda & \text{when } x \text{ is ergodic}
\end{cases}
\]

with the following properties:

1. \(R_x(-\tau) = R_x(\tau)\) [i.e., it is an even function].

2. \(R_x(0) = \frac{E(\text{signal energy})}{C_0} = P(\text{average signal power})\) if \(x\) is an energy signal, \(P(\text{average signal power}) = \mathcal{E}\{x^2(t)\}\) if \(x\) is a periodic power signal, and \(\mathcal{E}\{x(t)x(t+\tau)\}\) if \(x\) is WSS random signal.

3. \(R_x(\tau)\) has an absolute maximum at \(\tau = 0\), i.e., \(|R_x(\tau)| \leq R_x(0)\).

**Wiener–Kinchin Relations:**

\[
R_x(\tau) \leftarrow \mathcal{F} \begin{cases} 
\frac{|X(f)|^2}{\text{ESD}} & \text{If } x \text{ is an energy signal} \\
\frac{|X(f)|^2}{\text{PSD}} & \text{If } x \text{ is a periodic power signal} \\
G_x(f) & \text{If } x \text{ is WSS random signal}
\end{cases}
\]

**Prove properties 1, 2, and 3 for energy signals.**

**Solution:**

1. \(R_x(-\tau) = \int_{-\infty}^{\infty} x(\lambda)x(-\tau + \lambda)d\lambda \xrightarrow{\gamma=-\tau+\lambda} \int_{-\infty}^{\infty} x(v)x(v)dv = R_x(\tau)\).

2. \(R_x(0) = \int_{-\infty}^{\infty} x^2(\lambda)d\lambda = E(\text{always positive})\).

3. \(|R_x(\tau)|^2 \leq \int_{-\infty}^{\infty} |x(\lambda)|^2d\lambda \int_{-\infty}^{\infty} |x(\tau + \lambda)|^2d\lambda = \left[\int_{-\infty}^{\infty} |x(\lambda)|^2d\lambda\right]^2\) \[using Schwarz’s inequality, Tables.\]

For real signals: \(|x(t)|^2 = x^2(\lambda) \rightarrow R_x^2(\tau) \leq R_x^2(0) \rightarrow |R_x(\tau)| \leq R_x(0)\).


Tutorial 27

Q: A random current signal \( x(t) \) has a two-sided, normalized (to unit resistance) power spectral density (PSD) given by:

\[
G_x(f) = \begin{cases} 
0, & 0 < |f| < 1 \text{ Hz} \\
\frac{0.01}{|f|}, & 1 < |f| < \infty \text{ Hz} 
\end{cases} \text{ Amp}^2/\text{Hz}
\]

The signal is passed through a filter whose transfer function is \( H(s) = \frac{1}{s} \).

1. Determine the magnitude transfer function of the filter.
2. Determine the power transfer function of the filter.
3. Determine the average power (normalized to 1 \( \Omega \)) of the output signal \( y(t) \).

Solution:

1. \( |H(\omega)| = |H(s)|_{s=j\omega} = \frac{1}{|j\omega|} = \frac{1}{\omega} \) or \( |H(f)| = \frac{1}{2\pi f} \).
2. \( |H(f)|^2 = \frac{1}{4\pi^2 f^2} \).
3. \( P = \int_{-\infty}^{\infty} G_x(f)|H(f)|^2 df = 2 \int_{1}^{\infty} \frac{0.01}{4\pi^2 f^2} df = 0.000253 \text{ W}/\Omega \).

Tutorial 28

Q: A signal \( x(t) = 4\cos(\omega_0 t) \) is transmitted through two additive white Gaussian noise (AWGN) channels and received as \( s_1(t) = x(t) + n_1(t) \) and \( s_2(t) = x(t) + n_2(t) \), where SNR\(_1\) = 10 dB and SNR\(_2\) = 0 dB. Find:

1. The expected values of \( s_1(t) \) and \( s_2(t) \).
2. Noise power in both cases.
3. Roughly plot the pdf of \( n_1(t) \) and \( n_2(t) \). Which one has more spread around the mean?

Solution:

1. Since we have AWGN, then \( \mathcal{E}\{n_1(t)\} = \mathcal{E}\{n_2(t)\} = 0 \) [at any time \( t \)].
   Note that \( \mathcal{E}\{n(t)\} = \text{statistical mean} \) of \( n(t) = m = \int_{-\infty}^{\infty} np(n)dn \), while the time mean is \( m_t = \frac{1}{T} \int_{0}^{T} n(t)dt \), \( T \) being the total time. If \( t \) is discrete, then \( m_t = \frac{1}{N} \sum_{k=0}^{N-1} n(k) \). If \( n(t) \) is \textit{ergodic}, then \( m = m_t \).
\[
\mathcal{E}\{s_1(t)\} = \mathcal{E}\{x(t) + n_1(t)\} = \mathcal{E}\{x(t)\} + \mathcal{E}\{n_1(t)\} = x(t).
\]
Similarly: \(\mathcal{E}\{s_2(t)\} = x(t)\).

2. \(\text{SNR}_1 \text{ dB} = 10 \log_{10}(\text{SNR}_1) \rightarrow \text{SNR}_1 = 10^{\text{SNR}_1/10} = 10\).
\[
\text{SNR}_1 = P_x/P_{n_1} \Rightarrow P_{n_1} = P_x/\text{SNR}_1 = P_x/10.
\]
\[P_x = 4^2/2 = 8 \Rightarrow P_{n_1} = 8/10 = 0.8 \equiv -0.96 \text{ dB}.
\]
Similarly: \(P_{n_2} = 8 \equiv 9 \text{ dB} \).

3. Gaussian pdf is given by: \(P(n) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(n-m)^2/2\sigma^2}\) where \(m = \text{mean}, \sigma^2 = \text{variance}\).
Both \(n_1\) and \(n_2\) have zero means. Their variances are:
\[
\sigma_1^2 = P_{n_1} = 0.8 \rightarrow \sigma_1 \approx 0.9 \quad \text{and} \quad \sigma_2^2 = P_{n_2} = 8 \rightarrow \sigma_2 \approx 2.8.
\]
Since \(n_2\) has more power, i.e., more variance, we expect that it has wider spread as shown.

\[\text{Tutorial 29}\]

\[\textbf{Q: Range Estimation by Radar}: \text{A narrow pulse } s(t) \text{ is transmitted by the radar station towards the airplane. The pulse will hit the plane and reflect back to the radar, where a matched filter (correlator) is utilized to estimate the distance between the plane and the radar. Explain how the correlator is used for range estimation.}\]
Solution: The received signal \( r(t) \) is a delayed version of the transmitted signal \( s(t) \), which is also corrupted by noise. It is given by: \( r(t) = s(t - t_0) + n(t) \).

The **correlator is a matched filter** with impulse response given by:

\[
h(t) = s(-t) \quad \text{[A reflected version of } s(t) \text{ around vertical axis].}
\]

\[
y(t) = r(t) * h(t) = \int_{-\infty}^{\infty} r(\lambda) h(t - \lambda) d\lambda = \int_{-\infty}^{\infty} r(\lambda) s(\lambda - t) d\lambda
\]

\[
= \int_{-\infty}^{\infty} [s(\lambda - t_0) + n(\lambda)] s(\lambda - t) d\lambda
\]

\[
y = \lambda - t \quad \rightarrow \quad \int_{-\infty}^{\infty} s(v + t - t_0) s(v) dv + \int_{-\infty}^{\infty} n(v + t) s(v) dv = R_s(t - t_0) + R_{ns}(t).
\]

Note: since \( h(t) = s(-t) \), the system convolution is essentially a **correlation**.

Since the correlation is a measure of similarity, \( R_{ns}(t) \approx 0 \).

From Tutorial 21, the autocorrelation \( R_s(t - t_0) \) of a square pulse \( s(t) \) is a triangular function with a maximum at \( t - t_0 = 0 \), or \( t = t_0 \). Hence, if the pulse width \( a \) is very small, we have just a pulse at the time instant \( t = t_0 \) at the receiver, from which we can decide the time delay \( t_0 \).

Now \( 2d = ct_0 \) (where \( c = \) velocity of signal propagation \( \approx 10^8 \text{ m/s} \)).

Hence, the distance to the airplane is \( d = ct_0/2 \).

**Tutorial 30**

**Q:** In **binary baseband communication**, binary data (a sequence of 0’s and 1’s) are transmitted through a channel using two **orthogonal signals**, \( s_0(t) \) [to represent
logic 0] and $s_1(t)$ [to represent logic 1]. If the data rate is $R$ bps, then the time duration of $s_0$ and $s_1$ is $T = 1/R$ (s). A bank of two matched filters and a comparator are used at the receiver for optimal detection of signals in noise. In this question assume noise-free transmission, with $s_0$ and $s_1$ as shown, and $R = 1$ kbps.

1. Find the energy of $s_0$ and $s_1$ over $T$ s.
2. Show that $s_0$ and $s_1$ are orthogonal.
3. Find the outputs of the two matched filters at the receiver as functions of time over $(0, T)$, knowing logic 0 was transmitted and received as $r(t)$.
4. What is the decision of the receiver in step (3) above at $t = T$?

Solution:
1. $E_o = \int_0^T s_0^2(t)dt = T = 0.001$ J; $E_1 = \int_0^T s_1^2(t)dt = 0.001$ J.
2. $\int_0^T s_0(t)s_1(t)dt = \int_0^{T/2} dt - \int_{T/2}^T dt = 0 \rightarrow s_o$ and $s_1$ are orthogonal.

$$r_0 = \text{Output of the 1st correlator} = \int_0^t r(t)s_0(t)dt = \int_0^t s_o^2(t)dt = t$$

$$r_1 = \text{Output of the 2nd correlator} = \int_0^t r(t)s_1(t)dt = \int_0^t s_o(t)s_1(t)dt$$

$$= \begin{cases} 
\int_0^t dt = t, & 0 < t \leq T/2 \\
\int_0^{T/2} dt - \int_{T/2}^t dt = T - t, & T/2 < t \leq T
\end{cases}$$

4. At $t = T$, the receiver makes a decision based on the output of the comparator. Since $r_o(T) = T > r_1(T) = 0$, the decision is that logic 0 was transmitted. No error is expected since we considered a noise-free condition.
**Tutorial 31**

**Q:** A random signal frequently encountered in application is a sinusoid with random phase as follows: $s(t) = A \cos(\omega_o \cdot t + \theta)$ where $\theta$ is a random variable uniformly distributed over $(0, 2\pi)$. Find the autocorrelation function of $s(t)$.

**Solution:** Uniform distribution implies that: $2\pi b = 1$; hence, $b = 1/(2\pi)$. The pdf of $\theta$ is given by:

$$p(\theta) = \begin{cases} 
1/(2\pi), & 0 \leq \theta < 2\pi \\
0, & \text{elsewhere}
\end{cases}.$$  

Since $s(t)$ is a random signal, its autocorrelation is given by:

$$R_s(\tau) = \mathcal{E}\{s(t)s(t + \tau)\} = \mathcal{E}\left\{A^2 \cos(\omega_o \cdot \theta) \cos(\omega_o(t + \tau) + \theta)\right\}$$

$$= \mathcal{E}\left\{\frac{A^2}{2} \left[\cos(2\omega_o t + \omega_o \cdot \tau + 2\theta) + \cos(\omega_o \cdot \tau)\right]\right\}$$ (using Tables)

$$= \frac{A^2}{2} \mathcal{E}\{\cos(2\omega_o t + \omega_o \cdot \tau + 2\theta)\} + \frac{A^2}{2} \mathcal{E}\{\cos(\omega_o \cdot \tau)\}$$

$$= \frac{A^2}{2} \int_0^{2\pi} \cos(2\omega_o t + \omega_o \cdot \tau + 2\theta) \frac{1}{2\pi} d\theta + \frac{A^2}{2} \cos(\omega_o \cdot \tau)$$

$$= \frac{A^2}{2} \cos(\omega_o \cdot \tau)$$

**Note:** If we sample $s(t)$ with a sampling rate of $f_s = 1/T_s = 1/(T_o/4) = 4f_o$, the resulting samples are uncorrelated since $R_s(T_s) = R_s(T_o/4) = 0.$
**Tutorial 32**

**Q:** The pulse $x(t)$ shown in Fig. a below is sent along a communication line. At the receiver, a matched filter $h(t) = x(-t)$ is used for optimal detection.

1. Assume that the noise at the receiver has a flat spectrum, but is bandlimited to 10 kHz. The variance of noise is $\sigma^2$. What is the maximum signal power to noise power ratio (SNR) at the input of the matched filter (i.e., what is $\text{SNR}_i$ at the time of optimal detection)?

2. What is the maximum SNR at the output of the matched filter, $\text{SNR}_o$?

3. What is the ratio $\left(\frac{\text{SNR}_o}{\text{max}}\right) / \left(\frac{\text{SNR}_i}{\text{max}}\right)$?

4. Repeat the above steps for $s(t)$ in Fig. b.

---

**Solution:**

Since $h(t) = x(t_o - t)$ will maximize $\text{SNR}_o$ at $t = t_o$, we expect $\text{SNR}_o|_{\text{max}}$ at $t = 0$ here, as $t_o = 0$. Note that $t_o$ is the time of optimal reception. We can reach this result from $x(t) \ast h(t)$ as follows:

$$y(t) = x(t) \ast h(t) = \int_{-\infty}^{\infty} x(\lambda) h(t-\lambda) \, d\lambda = \int_{-\infty}^{\infty} x(v+t)x(v) \, dv = R_x(t)$$

Following Tutorial 23, we find $y(t)$ as shown below.
1. \[ \text{SNR}_i(t) = \frac{|x(t)|^2}{\sigma^2} = \frac{1}{\sigma^2} \quad \forall t \in (0, T) \] (1)

2. \[ \text{SNR}_o(t) \big|_{\text{max}} = \frac{|y(0)|^2}{\sigma_o^2} = \frac{1}{\sigma_o^2} \quad \text{(using the figure above)} \] (2)

Now we find \( \sigma_o^2 \) as follows:

\[
P_n = \sigma^2 = \int_{-\infty}^{\infty} G_n(f)df = \int_{-B}^{B} \frac{\eta}{2}df = \eta B
\] (3)

\[
P_{n_0} = \sigma_o^2 = \int_{-\infty}^{\infty} G_{n_0}(f)df = \int_{-\infty}^{\infty} G_n(f)|H(f)|^2df = \frac{\eta}{2} \int_{-B}^{B} |H(f)|^2df
\] (4)

Since \( h(t) = x(-t) \), we have

\[ H(f) = X(-f) \quad [\text{Tables}] \] (5)

Since \( x(t) \) is real, we have

\[ X(-f) = X^*(f) \] (6)

\[ |H(f)|^2 = H(f)H^*(f) = X^*(f)X(f) = |X(f)|^2 \] (7)

From (4) and (7) we have:

\[
\sigma_o^2 = \frac{\eta}{2} \int_{-B}^{B} |X(f)|^2df
\] (8)

Since \(|X(f)| \) is a sinc function [Tables], then \(|X(f)|^2 \) decays rapidly and we have:

\[
\sigma_o^2 \approx \frac{\eta}{2} \int_{-\infty}^{\infty} |X(f)|^2df
\] (9)

Using Parseval’s theorem we have:

\[
\sigma_o^2 \approx \frac{\eta}{2} \int_{-\infty}^{\infty} |X(f)|^2df = \frac{\eta}{2} \int_{-\infty}^{\infty} |x(t)|^2dt = \frac{\eta}{2} \int_{0}^{1} 1 \cdot dt = \frac{\eta}{2}
\] (10)

From (2) and (10) we have:

\[ \text{SNR}_o(t) \big|_{\text{max}} = \frac{1}{\sigma_o^2} = \frac{2}{\eta} \] (11)
3. From (1), (2), (3) and (11) we have:

\[
\frac{\text{SNR}_o(t)|_{\text{max}}}{\text{SNR}_i(t)|_{\text{max}}} = \frac{1/\sigma_0^2}{1/\sigma_i^2} = \frac{\sigma_i^2}{\sigma_0^2} = \frac{\eta \cdot B}{\eta/2} = 2B = 20000
\]

**Tutorial 33**

**Q1:** Find the magnitude and phase response of the constant time-delay system.

![Time-delay, T](image)

**Solution:** The time-delay I/O equation is \( y(t) = x(t - T) \).

Taking the Fourier transform we get: \( Y(f) = X(f)e^{-j2\pi fT} \) [Tables].

\[ H(f) = Y(f)/X(f) = e^{-j2\pi fT} \]

\[ h(t) = \delta(t - T), \text{using Tables} \]

Hence, the magnitude response is \( |H(f)| = 1 \) (constant), i.e., the time-delay is an all pass filter. The phase response is \( \phi(f) = \angle H(f) = -2\pi \cdot fT \). Note that the phase is frequency dependent in a linear relationship (hence the name linear phase).

**Q2:** Find the magnitude and phase response of the Hilbert transformer defined by:

\[ H(f) = \begin{cases} -j, & f \geq 0 \\ j, & f < 0 \end{cases} \]

![Hilbert Transformer, H(f)](image)

**Solution:** \( H(f) = -j \text{sgn}(f) \rightarrow h(t) = 1/(\pi \cdot) \) [Tables, Duality of \( \mathcal{F} \)]

\[ |H(f)| = |-j \text{sgn}(f)| = 1 \) (constant), i.e., Hilbert transformer (HT) is an all-pass filter.
\( \phi(f) = \angle H(f) = \angle [0 - j \text{sgn}(f)] = \begin{cases} \tan^{-1}(\frac{-1}{0}) = \tan^{-1}(-\infty) = -\frac{\pi}{2}, & f \geq 0 \\ \tan^{-1}(\frac{1}{0}) = \tan^{-1}(+\infty) = \frac{\pi}{2}, & f < 0 \end{cases} \).

Hence \( \phi(f) \) is \textit{constant} for physical frequencies \( f \geq 0 \).

**Comparison:** A constant time-delay \( T \) gives a constant (frequency-independent) time-delay and a frequency-dependent phase shift \( \phi(f) \); HT gives frequency-independent (constant) phase shift \( 90^\circ \) and frequency-dependent time-delay \( [t_d = (\pi/2)/\omega] \).

\[
\begin{array}{c|c|c}
|H(f)| & f & \phi(f) \\
\hline
1 & 0 & +\pi/2 \\
0 & f & -\pi/2
\end{array}
\]

**Tutorial 34**

**Q1:** Design a BP Chebychev-I 1 dB ripple filter with center frequency \( f_0 = 10 \) kHz, bandwidth \( BW = 1 \) kHz, maximum gain \( G_m = 1 \), and stopband gain \( \leq -10 \) dB for \( f \leq f_1 = 7 \) kHz and \( f \geq f_2 = 13 \) kHz. True load resistance is \( R_L = 10 \Omega \).

**Solution:**

\[
\omega_u = \omega_0 + \omega_b/2 = 2\pi(10k) + 2\pi(1k)/2 = 21\pi k \text{ rad/s}
\]

where \( \omega_b = BW = \omega_u - \omega_l \).

\[
\omega_l = \omega_0 - \omega_b/2 = 2\pi(10k) - 2\pi(1k)/2 = 19\pi k \text{ rad/s}.
\]

\[
\omega_g = \sqrt{\omega_l \omega_u} = \sqrt{(21\pi k)(19\pi k)} = 19.97\pi k \text{ rad/s}.
\]

Using LP \( \rightarrow \) BP transformation (Tables): \( \omega_N = \frac{\omega^2 - \omega_0^2}{\omega_0 \omega_b} \), we find the normalized LP frequencies that correspond to \( f_1 \) and \( f_2 \) as follows:

\( \omega_{1N} = -7.25 \) and \( \omega_{2N} = 5.32 \).

We check the order \( n \) that gives gain \( \leq -10 \) dB for \( |\omega_{2N}| \leq \omega_N \leq |\omega_{1N}| \); or \( 5.32 \leq \omega_N \leq 7.25 \). From Chebychev-I curves (Tables), \( n = 1 \) gives gain \( \leq -10 \) dB for \( \omega_N = 7.25 \), but not for \( \omega_N = 5.32 \). Hence, \( n = 2 \) is the suitable choice. From Tables we obtain the \textit{normalized} transfer function as follows:

\[
H(s_N) = \frac{a_0}{1.1 + 1.09s_N + s_N^2} \Rightarrow G_{dc} = \frac{G_M}{\sqrt{1 + \varepsilon^2}} = \frac{1}{\sqrt{1 + 0.25}} = \frac{a_0}{1.1} \Rightarrow a_0 = 0.98.
\]
Using the transformation LPN $\rightarrow$ BP (Tables), we obtain the final transfer function:

$$H(s) = \frac{0.98}{1.1 + 1.09\left(\frac{s^2 + \omega_0^2}{\omega_0k}\right) + \left(\frac{s^2 + \omega_0^2}{\omega_0k}\right)^2}$$

which is a 4th-order filter. Note that since $s_N \rightarrow \frac{s^2 + \omega_0^2}{\omega_0k}$ is quadratic, number of poles is doubled by using this transformation.

**Circuit Design:** Using Tables, the scaling factors $Z = 10$ and $F = \omega_p = 2\pi k$, and the relation $\omega_b^2 = 1/(LC)$, we design the BPF circuit as shown below.

![Circuit Diagram](image)

**Tutorial 35**

**Q:** Find the values of $k$ for which the analog system shown below is stable ($k$ is real).

![System Diagram](image)

**Solution:** Using same approach as in Tutorial 19, we find the transfer function of the system as follows:

$$H(s) = \frac{1}{s^2 + s + k} = \frac{1}{(s - p_1)(s - p_2)}$$

$$p_{1,2} = \frac{-1 \pm \sqrt{1 - 4k}}{2} = -\frac{1}{2} \pm \frac{1}{2} \sqrt{1 - 4k} = -\frac{1}{2} \pm \sqrt{1 - 4k}$$

For the system to be stable, its poles $p_1$ and $p_2$ should be in the left half of the $s$-plane.
Since \( k \) is real, \( \sqrt{\frac{1}{4} - k} \) is either real (if \( \frac{1}{4} \geq k \)) or pure imaginary (but not complex) (if \( \frac{1}{4} < k \)). If \( \sqrt{\frac{1}{4} - k} \) is imaginary, then both \( p_1 \) and \( p_2 \) will be in the left half of the \( s \)-plane and the system is always stable.

Now assume \( \sqrt{\frac{1}{4} - k} \) is real, i.e., \( k \leq \frac{1}{4} \). For stability we need the following condition:

\[
-\frac{1}{2} + \sqrt{\frac{1}{4} - k} < 0 \quad (\text{to keep } p_1 < 0).
\]

\[
\therefore \quad \frac{1}{4} - k < \frac{1}{4} \rightarrow k > 0.
\]

Combining \( k \leq \frac{1}{4} \) and \( k > 0 \) gives \( 0 < k \leq \frac{1}{4} \) as the overall condition in this case.

Now we summarize the real and imaginary cases above:

1. Case 1 (the root is imaginary): \( k > \frac{1}{4} \)
2. Case 2 (the root is real): \( 0 < k \leq \frac{1}{4} \)

The combination of the two cases gives the final condition \( k > 0 \).

**Tutorial 36**

**Q:** Design a passive bandpass Butterworth filter with the following specifications:

1. 4th-order
2. Center frequency = 10 kHz
3. Bandwidth = 3 kHz
4. True load impedance = 600 \( \Omega \)
5. Maximum gain = 1.

**Solution:**

\[
\omega_l = \omega_0 - \frac{\omega_b}{2} = 17k\pi; \quad \omega_g = \sqrt{\omega_u \omega_l} = 19.77k\pi
\]

The scaling factors are: ISF = \( Z = 600 \) and FSF = \( F = \omega_b = 6k\pi \).

Since LP \( \rightarrow \) BP transformation is quadratic (Tables), we need a 2nd-order LPF as the LP-prototype. Its transfer function is found from Tables as follows:

\[
H(s_N) = \frac{a_o}{b_o + b_1 s_N + s_N^2}
\]

where \( b_o = 1 \) and \( b_1 = 1.41 \).

\[
G_{d,c} = G_m = 1 = \frac{a_o}{b_o} \rightarrow a_o = b_o = 1.
\]
Using the LP $\rightarrow$ BP transformation $s_N = \frac{s^2 + \omega_b^2}{s\omega_b}$ (from Tables), we get the transfer function of the BPF as follows:

$$H(s) = \frac{1}{1 + 1.41 \frac{s^2 + \omega_b^2}{s\omega_b} + \left(\frac{s^2 + \omega_b^2}{s\omega_b}\right)^2}$$

**Circuit Design:** Using Tables, the scaling factors $Z = 600$ and $F = \omega_b = 6\pi k$, and the relation $\omega_g^2 = 1/(LC)$, we design the BPF circuit as shown below.

**Tutorial 37**

**Q:** Find the transfer function of the following active filter and explain its function.

**Solution:**

$$\frac{Y}{V^+} = \left(1 + \frac{R_2}{R_1}\right),$$

$$V^+ = X \frac{X_c}{R + X_c} = X \frac{1/(j\omega C)}{R + 1/(j\omega C)} = X \frac{1}{1 + j\omega RC}$$

$$\therefore \frac{Y}{X} = \frac{Y}{V^+} \frac{V^+}{X} = \left(1 + \frac{R_2}{R_1}\right) \frac{1}{1 + j\omega RC} = \frac{G_m}{1 + j\omega RC}$$

where $G_m = 1 + R_2/R_1$ is the gain of the filter.
The magnitude response is:

\[ |H(\omega)| = \frac{G_m}{\sqrt{1 + (\frac{\omega}{\omega_o})^2}} \]

where \( \omega_o = \frac{1}{\sqrt{RC}} \).

Now we have:

\[ H(0) = G_m; \]
\[ H(\infty) = 0; \]
\[ |H(\omega_o)| = \frac{G_m}{\sqrt{2}}. \]

Hence, the system is a LPF with cutoff frequency \( \omega_o = \frac{1}{\sqrt{RC}} \).

**Tutorial 38**

**Q:** Determine and plot the spectrum at all points shown in the signal processing system below. Consider only the frequency range 0–40 Hz.

**Solution:** The signal frequency is \( 2\pi f_o = 10\pi \rightarrow f_o = 5 \) Hz.

\[ X(f) = \frac{1}{2j} \delta(f - f_o) - \frac{1}{2j} \delta(f + f_o) \text{ (Tables)} \Rightarrow |X(f)| = \frac{1}{2} \delta(f - f_o) + \frac{1}{2} \delta(f + f_o). \]

For \( |R(f)| \) we have: \( A = 0.5|H_a(j2\pi f_o)| = 0.5 \left| \frac{100}{10\pi + 100} \right| = 0.47 \) (see the figure below).

For \( |V(f)| \) we have: \( B_1 = 18.8|H(e^{j2\pi(f - f_o)T_s})| = 18.8|1 - e^{-j2\pi5/40}| = 14.4, \) and \( B_2 = 18.8|H(e^{j2\pi(f + f_o)T_s})| = 18.8|H(e^{-j2\pi f_o T_s})| = 18.8|H(e^{-j2\pi f_o T_s})| = B_1. \)

For \( |Y(f)| \) we have:

\[ C = B_1|H_r(f_o)| = 14.4 \left| \frac{1}{40} \text{sinc} \left( \frac{5}{40} \right) \right| = 0.35 \] and \( D = B_2|H_r(35)| = 0.05. \)

*Note: no need here to divide \((1 - z^{-1})\) by \(T_s\) for accurate differentiation, since the sampler performs this function.*
Tutorial 39

Q: If the interest rate $r$ is fixed and there are no account fees, then a savings account in a bank represents an LTI system. Let $x(n)$ denote the amount of money deposited (or withdrawn) in the $n$th day, and $y(n)$ be the total amount of money in the account at the end of the $n$th day. Assume $r$ is compounded daily.

1. Find the impulse response $h(n)$ of the system using a time-domain approach.
2. Find the system output [using $h(n)$] at the $n$th day.
3. Find the system difference equation, the transfer function $H(z)$, and the impulse response $h(n)$. Can we use the fft to find the output $y(n)$?

Solution:

1. We need an input which is a delta function $\delta(n)$. Put $\$1$ in the opening day ($n = 0$) and $\$0$ afterwards. Now we have:

   $y(0) = x(0) = \delta(0) = 1; \quad y(1) = 1 + r; \quad y(2) = y(1) + ry(1) = y(1)(1 + r) = (1 + r)^2. \quad \text{Similarly: } y(n) = (1 + r)^n.$

   This is the impulse response of the system, $\{h(n)\}$.

2. $y(n) = x(n) * h(n) = \sum_{k=0}^{n} x(k)h(n-k) = \sum_{k=0}^{n} x(k)[1 + r]^{n-k}.$  
   For example, if $r = 0.01\% = 0.0001$ and we put $\$100$ in the opening day, $\$50$ in the next day, $\$50$ in the 3rd day, and nothing afterwards, then at the end of the year we have: $y(365) = x(0)(1 + r)^{365} + x(1)(1 + r)^{364} + x(1)(1 + r)^{363} = \$207.41$.

3. New balance = old + $r \cdot$ old + new deposit
\[ y(n) = y(n-1) + ry(n-1) + x(n) = (1 + r)y(n-1) + x(n) \]

\[ Y(z) = (1 + r)z^{-1}Y(z) + X(z) \Rightarrow Y(z)[1 - (1 + r)z^{-1}] = X(z) \]

\[ H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - (1 + r)z^{-1}} = \frac{z}{z - (1 + r)} \Rightarrow h(n) = (1 + r)^n \text{ (Tables).} \]

This is an IIR filter, with a pole at \( z = 1 + r \) and a zero at \( z = 0 \). It is unstable. We can use the FFT if \( \{x(n)\} \) has finite length and we are interested in \( y(n) \) for a finite range of \( n \), i.e., if \( 0 \leq n \leq M \), where \( M \) is a finite number.

**Tutorial 40**

**Q:** An engineer wants to make a decision about the trend of a fluctuating stock price. Based on a study of the stock market, he found that a period of 3 days is sufficient for a temporary decision, where the importance (weight) of the price at any day is twice that at the previous day. How can this engineer use the fft algorithm to estimate the weighted average of the stock price over the last week, knowing that the last seven prices were \{12, 9, 10, 13, 8, 6, 9\} dollars?

**Solution:** We need a 3-tap moving average FIR filter \( (N = 3) \). We should find its impulse response first.

Assume that the percentage weight of today’s price is \( a \Rightarrow h(0) = a \).

The weight of yesterday’s price will be \( a/2 \Rightarrow h(1) = a/2 \).

The weight of the price 2 days ago will be \( (a/2)/2 = a/4 \Rightarrow h(2) = a/4 \).

Now we have \( 1 = a + \frac{a}{2} + \frac{a}{4} \) from which \( a = 0.57 \). Hence, the tap weights are: \( h_0 = 0.57, h_1 = 0.28, \) and \( h_2 = 0.14 \). The average price suitable for a decision will be the output of the FIR filter:

\[ y(n) = h(n) \ast x(n) = \sum_{k=0}^{N-1} h(k)x(n-k); \quad \text{where} \quad h = [h_0 \ h_1 \ h_2] \quad \text{and} \]

\[ x = \begin{bmatrix} 12 & 9 & 10 & 13 & 8 & 6 & 9 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix} \]

The length of \( \{y(n)\} \) sequence is \( L = N_h + N_x - 1 = 3 + 7 - 1 = 9 \).

For efficient computation (especially in real applications, where \( N_h \) and \( N_x \) can be very large), we should use the fft as follows:
1. Zero-pad $h$ to get $h_z = [h_0, h_1, h_2, 0, 0, 0, 0, 0]$ and $x$ to get $x_z = [12, 9, 10, 13, 8, 6, 9, 0, 0]$.

2. Compute: $H_z = \text{fft}(h_z)$, $X_z = \text{fft}(x_z)$, and $Y_z = X_z H_z$.

3. Find $y_z = \text{ifft}(Y_z)$ to get $y_z = \begin{bmatrix} 6.8 & 8.5 & 10 & 11.5 & 9.7 & 7.5 & 8 & 3.4 & 1.2 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{bmatrix}$.

Consider only the central 5 values of $\{y(n)\}$, i.e., items no. 2, 3, ..., 6.

*Note*: the coefficients $\{h(n)\}$ can be adaptive based on change of market variables.

---

**Tutorial 41**

**Q:** Find $x(n)$ if

(A) $X(z) = \frac{3}{z-2}$,

(B) $X(z) = \frac{1}{z-3z^2+2}$.

**Solution:**

(A) $X(z) = \frac{3}{z-2} = 3z^{-1} \frac{1}{z-2} = 3z^{-1}R(z)$.

From Tables we find: $r(n) = 2^n u(n)$ and $z^{-1}R(z) \xrightarrow{zT} r(n-1) = 2^{n-1}u(n-1)$.

Hence we have $x(n) = 3r(n-1) = 3 \cdot 2^{n-1} u(n-1)$.

(B) $z^2 - 3z + 2 = 0 \Rightarrow p_{1,2} = \frac{-(-3) \pm \sqrt{(-3)^2 - 4(1)(2)}}{2(1)} = 1, 2$.

$X(z) = \frac{1}{(z-p_1)(z-p_2)} = \frac{1}{(z-1)(z-2)} = \frac{a}{z-1} + \frac{b}{z-2}$ (where we used partial fraction expansion).

To find $a$:

1. Multiply by $z - 1$: $\frac{z^{-1}}{(z-1)(z-2)} = a + b \frac{1}{z-2}$

2. Put $z = 1$: $\frac{1}{(1-2)} = a + 0 \Rightarrow a = -1$.

Similarly we find $b = 1$.

$\therefore \quad X(z) = \frac{-1}{z-1} + \frac{1}{z-2} = (-1)z^{-1} \frac{z}{z-1} + z^{-1} \frac{z}{z-2}$.

From Tables and (A) above we find:

$x(n) = -u(n-1) + 2^{n-1}u(n-1) = [2^{n-1} - 1]u(n-1)$.

*Note*: Normally in such questions, it is better to expand $X(z)/z$ rather than $X(z)$. Applying this to Q2 above, we get $x(n) = \frac{1}{2} \delta(n) + [2^{n-1} - 1]u(n)$. Show that the two answers are equivalent.
**Tutorial 42**

**Q:** Find the values of $b$ for which the system shown below is stable, knowing that $b$ is real.

![System Diagram](image)

**Solution:** We first define intermediate signals $r(n)$, $g(n)$ as shown below.

![Intermediate Signals](image)

We write the system equations as follows:

$$y(n) = r(n) + g(n) \quad (1)$$
$$g(n) = y(n - 1) \quad (2)$$
$$r(n) = x(n) - by(n - 2) \quad (3)$$

From (1), (2), and (3) we get:

$$y(n) = x(n) + y(n - 1) - by(n - 2) \quad (4)$$

Taking the ZT of both sides of (4) and re-arranging terms we get:

$$Y(z)[1 - z^{-1} + bz^{-2}] = X(z) \Rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - z^{-1} + bz^{-2}} = \frac{z^2}{z^2 - z + b}$$

The system has two poles at $p_{1,2} = \frac{1 \pm \sqrt{1 - 4b}}{2}$.

Since $b$ is real, then $\sqrt{1 - 4b}$ is either real or pure imaginary, but not complex.

**Case 1:** If $1 > 4b$ (i.e., $\sqrt{1 - 4b}$ is real positive), then $|z_p| < 1 \Rightarrow -3 < \pm \sqrt{1 - 4b} < 1$. 

$$\begin{align*}
\sqrt{1 - 4b} < 1 & \Rightarrow 0 < b \\
-3 < -\sqrt{1 - 4b} & \Rightarrow -2 < b \\
\end{align*}$$

(Intersection of $b > -2$ and $b > 0$.)
Case 2: If $1 \leq 4b$ (i.e., $\sqrt{1 - 4b}$ is pure imaginary), then

$$|z_p| < 1 \Rightarrow |1 \pm j\sqrt{4b - 1}| < 2 \Rightarrow b < 1.$$ 

Hence, combining Case 1 and Case 2 we get $0 < b < 1$.

**MATLAB:** try $b = 0.8; r = \text{roots([1 -1 b])}; a = \text{abs}(r)$

**Tutorial 43**

**Q:** Find the values of $b$ for which the system shown below is stable, knowing that $b$ is real.

**Solution:** It is better to define some auxiliary points on such diagrams. In this question we define $r(n)$, hence the other side of $z^{-1}$ will be $r(n - 1)$.

$$y(n) = r(n) + (b/3)r(n - 1) \quad (1)$$

$$r(n) = x(n) + (b/2)r(n - 1) \quad (2)$$

It is not easy to find the I/O relationship from the time domain equations. Hence, take the z-transform of (1) and (2) as follows:

$$Y(z) = R(z) + \frac{b}{3}z^{-1}R(z) = \left[1 + \frac{b}{3}z^{-1}\right]R(z) \quad (3)$$

$$R(z) = X(z) + \frac{b}{2}z^{-1}R(z) \Rightarrow X(z) = \left[1 - \frac{b}{2}z^{-1}\right]R(z) \quad (4)$$
\[ H(z) = \frac{Y(z)}{X(z)} = \frac{1 + (b/3)z^{-1}}{1 - (b/2)z^{-1}} = \frac{z + (b/3)}{z - (b/2)} \]  

The system has a zero at \( z = -b/3 \) and a pole at \( z = b/2 \).

For stability we should have: \( |b/2| < 1 \rightarrow |b| < 2 \).

Q: From Eq. 5 above, find the impulse response \( h(n) \) and the difference equation of the above system.

**Tutorial 44**

Q: Using the impulse invariance method, design a digital Chebychev LPF with the following specifications:

1. \( T_s = 0.01 \) s (hence, \( f_s = 100 \) Hz),
2. \( f_c = 10 \) Hz (\( \omega_c = 20\pi \) rad/s),
3. \( G_m = 1 \), gain < 0.1 (i.e., \( -20 \) dB) for \( 30 \leq f \leq f_s/2 = 50 \) Hz),
4. \( 3 \) dB ripple is allowed in the passband.

Solution: We need an analog Chebychev filter with the above specifications, with gain less than \( -20 \) dB for normalized frequency \( f_n \geq 30/10 = 3 \) (normalized w.r.t \( f_c \)). From Tables we find the filter order \( n = 2 \), with normalized transfer function:

\[ H_N(s) = \frac{a_o}{0.7079 + 0.6449s_N + s_N^2} \]

Since \( n \) is even, we have \( G_{dc} = G_m/\sqrt{1 + e^2} = 1/\sqrt{1.9953} = 0.7079 \).

Hence, \( a_o/0.7079 = 0.7079 \Rightarrow a_o = 0.5 \). The denormalized analog transfer function is obtained by the substitution \( s_N = s/\omega_c = s/(20\pi) \) to get:

\[ H_a(s) = \frac{1974}{2795 + 40.5s + s^2} \]

Using Partial Fraction Expansion (Tutorial 19), we can write \( H_a(s) \) as follows:

\[ H_a(s) = \frac{c_1}{s - p_1} + \frac{c_2}{s - p_2}, \]

where \( p_1 = -20.25 + 48.18i, \quad p_2 = \overline{p_1} = -20.25 - 48.18i, \quad c_1 = -20.2i, \quad c_2 = \overline{c_1} = 20.2i \).

The \( z \)-domain poles are:

\[ z_1 = \exp(p_1T_s) = 0.68 + 0.45i, \quad z_2 = \exp(p_2T_s) = \overline{z_1} = 0.68 - 0.45i. \]
Hence, the transfer function of the digital filter would be:

\[
H(z) = T_s \left[ c_1 \frac{z}{z - z_1} + c_2 \frac{z}{z - z_2} \right] = T_s z \left[ c_1 \frac{1}{z - z_1} + \bar{c}_1 \frac{1}{z - \bar{z}_1} \right] = T_s z \left[ \frac{c_1 (z - \bar{z}_1) + \bar{c}_1 (z - z_1)}{(z - z_1) (z - \bar{z}_1)} \right] = T_s z \left( \frac{(c_1 + \bar{c}_1) z - (c_1 \bar{z}_1 + \bar{c}_1 z_1)}{z^2 - (z_1 + \bar{z}_1) z + |z_1|^2} \right)
\]

\[
= \frac{0.15 z}{z^2 - 1.36 z + 0.667}.
\]

**Tutorial 45**

**Q:** Using the impulse invariance method, design a digital filter with sampling frequency 100 Hz and impulse response that matches the response of the following 3rd-order analog Butterworth filter:

\[
H(s) = \frac{1}{(s + 1)(s - p)(s - \bar{p})},
\]

where \( p = -0.5 + 0.866i \)

**Solution:** Using Partial Fraction Expansion, we write the transfer function as follows:

\[
H_a(s) = \frac{1}{s + 1} + \frac{c}{s - p} + \frac{\bar{c}}{s - \bar{p}},
\]

where \( c = -0.5 - 0.28i \).

The z-domain poles are:

\[
z_1 = \exp (p T_s) = \exp (-T_s) = 0.99,
\]

\[
z_2 = \exp (p T_s) = 0.995 + 0.0086i,
\]

\[
z_3 = \bar{z}_2 = 0.995 - 0.0086i.
\]

\[
\therefore \quad H(z) = T_s \left[ \frac{z}{z - 0.99} + c \frac{z}{z - z_2} + \bar{c} \frac{z}{z - \bar{z}_2} \right]
\]
**Tutorial 46**

**Q:** Using the bilinear transform, design a 4th-order BP Butterworth digital filter with center frequency $\Omega_o = 1.5$, maximum gain $G_m = 1$, and bandwidth $\Omega_b = 0.4$.

**Solution:** Since the transformation $LP \rightarrow BP$ is quadratic (Tables), we need a 2nd-order prototype analog LPF. The transfer function of this filter is given by (Tables):

$$H(s) = \frac{a_o}{b_o + b_1 s + s^2}$$

where $b_o = 1$ and $b_1 = 1.41$.

$$G_{d,c} = G_m = 1 = \frac{a_o}{b_o} \Rightarrow a_o = b_o = 1. \Rightarrow H(s) = \frac{1}{1 + 1.41 s + s^2}.$$ 

$$\omega_u = \tan(\Omega_u/2) = \tan\left(\frac{\Omega_u + \Omega_b}{2}\right)/2 = \tan(1.7/2) = 1.14$$

$$\omega_l = \tan(\Omega_l/2) = \tan\left(\frac{\Omega_o - \Omega_b}{2}\right)/2 = \tan(1.3/2) = 0.76$$

$$\omega_b = \omega_u - \omega_l = 0.38; \omega_g = \sqrt{\omega_l \omega_u} = 0.93$$

Using the LP $\rightarrow$ BP transformation $s = \frac{s^2 + \alpha_o^2}{s \omega_b}$ (from Tables), we get the transfer function of the analog BPF as follows:

$$H(s) = \frac{1}{1 + 1.41 \frac{s^2 + \alpha_o^2}{s \omega_b} + \left(\frac{s^2 + \alpha_o^2}{s \omega_b}\right)^2}.$$ 

Using the bilinear transformation $s = (z - 1)/(z + 1)$, we obtain the transfer function of the required BP digital filter as follows:

$$H(z) = \frac{1}{1 + 1.41 \frac{[(z-1)/(z+1)]^2 + \alpha_o^2}{[(z-1)/(z+1)] \omega_b} + \left(\frac{[(z-1)/(z+1)]^2 + \alpha_o^2}{[(z-1)/(z+1)] \omega_b}\right)^2}.$$

**Tutorial 47**

**Q:** A radar station received the two signals at two different time intervals. A correlator is used to analyze these signals using their autocorrelations. The correlator outputs are as shown below. Comment on the structure of these signals.
Solution: Figure a shows approximately a delta function with some disturbance, hence the first signal is just a broadband noise. The pdf, mean, and variance (power) of this noise can be found from the input signal itself. Figure b shows a symmetric shape with disturbance everywhere and a maximum at the origin, along with a narrow spike at the origin, hence the second signal is a piece of information corrupted by broadband noise.

**Tutorial 48**

Q:
(A) Find the Hilbert transform of \( x(t) = \sin(\omega_o t) \).
(B) Find the Hilbert transform of \( x(t) = \sin(\omega_o t) + \delta(t) \).
(C) Find the analytic signal \( z(t) \) associated with the real signal \( x(t) = \cos(\omega_o t) \).
(D) Compare the spectra of these signals and comment.

Solution:
(A) Let \( \mathcal{H}[\sin(\omega_o t)] = y(t) \).

The transfer function of \( \mathcal{H} \) is \( H(f) = \begin{cases} e^{-j\pi/2} = -j, & f > 0 \\ e^{+j\pi/2} = j, & f < 0 \end{cases} \) = \(-j\text{sgn}(f)\).

From Tables: \( X(f) = \frac{1}{2j} \delta(f - f_o) - \frac{1}{2j} \delta(f + f_o) \). Hence we have:

\[
Y(f) = H(f) \cdot X(f) = -j\text{sgn}(f) \left[ \frac{1}{2j} \delta(f - f_o) - \frac{1}{2j} \delta(f + f_o) \right] \\
= -\frac{1}{2} \delta(f - f_o) - \frac{1}{2} \delta(f + f_o) \Rightarrow y(t) = -\cos(\omega_o t) \text{ (Tables)}
\]

(B) Let \( \mathcal{H}[\delta(t)] = d(t) \), \( \mathcal{H}[\sin(\omega_o t)] = y(t) \), and \( \mathcal{H}[x(t)] = z(t) \).

From Tables, \( \Delta(f) = FT\{\delta(t)\} = 1 \)

\[
\therefore \quad D(f) = -j\text{sgn}(f) \cdot \Delta(f) = -j\text{sgn}(f)
\]
From Tables we have: \( d(t) = 1/(\pi \cdot t) \); an from (A) we have: \( y(t) = -\cos(\omega_o t) \).

Since \( \mathcal{H} \) is a linear system, we have: \( z(t) = y(t) + d(t) = -\cos(\omega_o t) + 1/(\pi \cdot t) \).

(C) \( z(t) = x(t) + j\mathcal{H}\{x(t)\} \).

Using an approach similar to that in (A) above, we have
\[
\mathcal{H}\{\cos(\omega_o t)\} = \sin(\omega_o t).
\]
Hence, \( z(t) = \cos(\omega_o t) + j\sin(\omega_o t) = e^{j\omega_o t} \).

Now from Tables we have:
\[
X(f) = \frac{1}{2} \delta(f - f_o) + \frac{1}{2} \delta(f + f_o); \quad Z(f) = \delta(f - f_o).
\]

(D) Using the analytic signal, the negative part of spectrum is removed, while the positive part is scaled by 2. This indicates that the use of the analytic signal will lead to spectrum economy.

**Tutorial 49**

**Q:** Consider a first-order sinusoidal DPLL (SDPLL) under noise-free conditions, center frequency \( f_o = 1 \) Hz, and input signal \( x(t) = \sin(6t + \pi/4) \).

1. Find the system equation in terms of the digital filter gain \( G_1 \).
2. Find the steady-state phase error \( \phi_{ss} \) in terms of \( G_1 \).
3. Find the range of \( G_1 \) that ensures locking on the above incoming frequency.
4. Choose a value for \( G_1 \) inside the locking range and find the corresponding \( \phi_{ss} \).
5. Assuming \( t(0) = 0 \), plot \( x(t) \) with the first three DCO pulses for the above value of \( G_1 \).

**Solution:**

1. From Tables we find the system equation as follows:
\[
\phi(k + 1) = \phi(k) - \omega G_1 \sin[\phi(k)] + (\omega - \omega_o)T_o \tag{1}
\]
We have: \( \omega = 6 \) rad/s, \( \omega_o = 2\pi \) rad/s, and \( T_o = 1 \) s.
Hence, the system equation will be given by:
\[
\phi(k + 1) = \phi(k) - 6G_1 \sin[\phi(k)] - 0.28 \tag{2}
\]

2. From (2) we have:
\[
\phi_{ss} = \phi_{ss} - 6G_1 \sin[\phi_{ss}] - 0.28 \Rightarrow \phi_{ss} = \sin^{-1}(-0.047/G_1) \tag{3}
\]

3. Locking conditions are as follows:

   **Condition 1:** From (3) we have:
   \[
   | -0.047/G_1 | < 1 \Rightarrow G_1 > 0.047 \tag{4}
   \]
**Condition 2:** From (2) we have:

\[ g(w) = \frac{w}{C_0} - 6G_1 \sin(w) - 0.28 \]  

\[ \therefore g'(w) = 1 - 6G_1 \cos(w). \]

\[ \therefore \text{If a solution } \phi_{ss} \text{ of (5) exists, then } g'(\phi_{ss}) = 1 - 6G_1 \cos(\phi_{ss}). \]

*Fixed point analysis* says that (Tables):

\[ \phi_{ss} \text{ exists if } |g'(\phi_{ss})| < 1 \Rightarrow |1 - 6G_1 \cos(\phi_{ss})| < 1 \]  

From Tables: \( \cos(\phi_{ss}) = \pm \sqrt{1 - \sin^2(\phi_{ss})}. \)

From (3) we have

\[ |\phi_{ss}| < \pi/2 \Rightarrow \cos(\phi_{ss}) > 0 \]  

Hence, using (3) and (7) we have:

\[ \cos(\phi_{ss}) = +\sqrt{1 - \sin^2(\phi_{ss})} = \sqrt{1 - (0.047/G_1)^2}. \]  

Now substituting (8) in (6) we get:

\[ |1 - 6G_1 \sqrt{1 - 0.0022/G_1^2}| < 1 \Rightarrow |1 - \sqrt{36G_1^2 - 0.08}| < 1 \]

\[ \Rightarrow -1 < 1 - \sqrt{36G_1^2 - 0.08} < 1 \]

\[ \Rightarrow -2 < - \sqrt{36G_1^2 - 0.08} < 0 \]  

\[ \Rightarrow 0 < \sqrt{36G_1^2 - 0.08} < 2 \]

\[ \Rightarrow 0 < 36G_1^2 - 0.08 < 4 \]

\[ \Rightarrow 0.047 < G_1 < 0.33 \]

4. We have \( \phi(0) = \pi/4, x(0) = \sin(\pi/4) = 0.7. \)

Using (9), let \( G_1 = 0.1. \) Hence:

\[ y(0) = G_1x(0) = 0.1(0.7) = 0.07, \]

\[ T(1) = T_o - y(0) = 1 - 0.07 = 0.93 \text{ s}, \]

\[ t(1) = T(1) = 0.93 \text{ s}, \]

\[ \phi(1) = \omega t(1) + \pi/4 \equiv 0.08 \text{ rad}, \]

\[ x(1) = \sin(\phi(1)) = 0.08, \]

\[ y(1) = G_1x(1) = 0.1(0.08) = 0.008, \]

\[ T(2) = T_o - y(1) = 1 - 0.008 = 0.99, \]

\[ t(2) = T(1) + T(2) = 1.92, \]

\[ \phi(2) = \omega t(2) + \pi/4 \equiv -0.26 \text{ rad}. \]
Tutorial 50

Q: The phase equation of a first-order sinusoidal DPLL is shown below. Find approximately the steady-state phase error \( \phi_{ss} \).

Solution: First, we plot the curve \( y = x \). If there is a steady-state, this curve will have two intersections with phase equation curve; one of them is the steady-state point, where its projection on the y-axis gives \( \phi_{ss} \). To determine which one is the steady-state point, we choose randomly some value for the initial phase \( \phi(0) \), and then plot the phase plane diagram by successive projections on the equation and \( y = x \). For the above figure we choose \( \phi(0) = -1.5 \) and we find \( \phi_{ss} \approx -0.5 \).
**Tutorial 51**

**Q:** Design a digital controlled oscillator (DCO) with a clock rate of 1 MHz and a center frequency of 10 kHz. Draw the block diagram and the output signal for an input sequence of 0, 30, −30, 70,…. Indicate sampling times.

**Solution:** We have \( f_c = 1 \text{ MHz} \), hence, \( T_c = 1/f_c = 1\mu s \).

The center frequency is \( f_o = f_c/M_o = 10 \text{ kHz} \).

Hence, the free-running input number is \( M_o = f_c/f_o = 100 \).

For the given sequence, the values of the counter initial number are \( R_o = 100 - 0 = 100 \); \( 100 - 30 = 70 \); \( 100 - (-30) = 130 \); and \( 100 - 70 = 30 \); respectively.

Hence, the sampling times are

0 (initial), 100 \( T_c = 100\mu s \), 170 \( \mu s \), 300 \( \mu s \), and 330 \( \mu s \).

An illustrative (not to scale) diagram is shown below.
**Tutorial 52**

**Q:** A block diagram of a delta modulation system is shown below, along with the input signal. The step of the integrator is arranged to be 0.2 V, while the sampling frequency is $f_s = 1$ kHz. Plot on the same figure the estimated signal and the output waveform. Assume that the integrator output was initially zero and the quantizer output is 1. Suggest a demodulation circuit.

![Block diagram of a delta modulation system](image)

**Solution:** The sampling period is $T_s = 1/f_s = 1/1000 = 1$ ms. The plots are as shown below.

![Estimated signal and output waveform](image)
Tutorial 53

Q: Use MATLAB to design a fourth -order elliptic lowpass filter with the following specifications:

Pass-band peak-to-peak ripple, $R_p = 0.6 \text{ dB}$
Minimum stop-band attenuation $R_s = 18 \text{ dB}$
The normalized pass-band-edge frequency $w_p = 0.63$ Then, implement this filter
by changing the filter properties to fixed-point representation.

Solution: We may use the dfilt object for this purpose.
The first step is to design the filter using the Matlab default double-precision format:

$$M = 4; \ Rp = .6; \ Rs = 18; \ wp = .63; \ [\text{Den}, \text{Num}] = \text{ellip}(M, \text{Rp}, \text{Rs}, \text{wp});$$

Digital filter implementation using direct form—I:

$$H = \text{dfilt.dfl}(\text{Den}, \text{Num})$$

Second step is to convert the filter coefficients to fixed point format:

H.Arithmetic = 'fixed'
get(H)
H.FilterInternals = 'SpecifyPrecision'

Note: “SpecifyPrecision” enables you to set your own wordlength and fraction length for the output and accumulator-related properties.
Now set the hardware specifications:

set(H,'InputWordlength', 16,...
'InputFracLength', 13,...
'ProductWordLength', 24,...
'AccumulatorWordLength', 28,...
'OutputWordLength', 16);
H

Verify that the object dfilt has made an appropriate auto-scaling (Hint: check the range of the filter coefficients).

Tutorial 54

Q: Verify the equivalence of the identities shown in the figure below. What is the potential of these identities?
Solution: In both cases $H(z) = z^{-1}$, that is a time delay. In Fig. a, the right-hand side can be expressed as

$$y(n+1) = x(Mn)$$
$$y(n) = x(M(n-1)) = x(Mn - M)$$

while the left-hand side is

$$v(n) = x(n-M)$$
$$y(n) = v(Mn)) = x(Mn - M)$$

that is, both sides are of Fig. a are equivalent. The up-sampler shown in Fig. b can be verified similarly. In both cases the right-hand side implementation is more computationally efficient since after the down-sampler or before the up-sampler the filter $H(z)$ will operate at lower sampling rate. On the other hand, the left-hand side implementation is wasteful as it processes a lot of zero terms.
Appendix B: Miscellaneous Exercises

Miscellaneous DSP Exercises—A

Q1: Explain the meaning of a “signal”. Give five examples of real-life signals.
Q2: State five classes of signals with brief explanations.
Q3: Draw a block diagram for an analog/digital signal processing system.
Q4: Show whether the Hilbert transform [that gives constant 90° phase shift for all sinusoidal signals of the form \( x(t) = \sin(\omega t + c) \)] is:
   (a) Memoryless, (b) causal, (c) linear, (d) time-invariant, (e) BIBO stable.
Q5: Define the Dirac delta function. How can we approximate it in applications?
Q6: How can we represent an analog system and its I/O relationship?
Q7: Both Fourier and Laplace Transforms are used to represent analog systems. Which one is the more general?
Q8: State the conditions in the time-domain that an analog system is BIBO stable. What are the equivalent conditions in the frequency domain?
Q9: Explain what is the physical meaning of the cross-correlation integral. State an application for this integral.
Q10: What kind of signals can Fourier series represent? Is the Fourier series a frequency transform? Can it reveal the frequency content of the signal?
Q11: How does the trigonometric Fourier series of an odd periodic signal look like?
Q12: Can we use Fourier series to represent energy signals?
Q13: Can we use Fourier transform to represent periodic signals? How?
Q14: From the basic definition of Fourier transform, find and plot the amplitude and phase spectra of the signal \( x(t) = \exp(-5t)u(t) \). Is this an energy or power signal? Why?
Q15: What is meant by the duality of the Fourier transform?
Q16: Using Tables, find the Fourier transform of the following signals and plot their magnitude spectra:
   1. \( \sin(2t + 1) \), 2. \( \cos(5t) \), 3. 1, (4) \( \delta(t) \), 5. \( u(t) \), 6. \( \text{sgn}(5t) \), 7. \( \Pi_1(t) \), 8. \( \text{sinc}(t - 5) \), 9. \( \text{sinc}(t) \cos(20t) \).
Q17: State Parseval’s theorem for periodic and non-periodic analog signals and define the power and energy spectra and spectral densities.

Q18: Explain why the single-sided Laplace transform is sufficient for engineering applications.

Q19: From the basic definition, find the Laplace transform and its ROC for the signal \( x(t) = \exp(t)u(t) \).

Q20: Plot the pole-zero diagram of the system \( H(s) = s(s + 1)/[(s^2 + 5s + 6)(s + 5)] \). Is the system stable? Justify your answer.

Q21: What is the sample space of the die tossing experiment? How can we represent outcomes of this experiment using a random variable? Is this random variable continuous or discrete? Plot the pdf of this random variable.

Q22: Two dice are tossed. What is the probability that the first is 1 and the second is 6?

Q23: Explain the physical meaning of the statistical mean and variance of a random variable.

Q24: Plot and find the mean and the variance of the following Gaussian pdf
\[
p(x) = \frac{1}{\sqrt{\pi}} e^{-x^2}.
\]

Q25: Two Gaussian noise processes \( n_1(t) \) and \( n_2(t) \), where the first has more power than the second. Approximately plot the pdf’s of the two noise random variables. Are \( n_1 \) and \( n_2 \) energy or power signals?

Q26: Find the mean and variance of the random signal \( x(t) = \sin(\omega t) + n(t) \), where \( n(t) \) is Gaussian noise with variance = 0.1.

Q27: State the Wiener–Kinchin theorem for WSS random signals.

Q28: Define the autocorrelation function of a random variable \( X(t) \).

Q29: Explain the meaning of white noise. What is the effect of an ideal LPF on the correlation between white noise samples?

Q30: Outline the principle of a matched filter. State an application for matched filters in binary communications (Plot the block diagram).

Q31: Show that the matched filter is essentially a correlator.

Q32: Show that the ideal analog LPF is a non-causal system. How can we modify this theoretical system to be practically realizable?

Q33: What are the advantages of active filters over passive filters?

Q34: What is the importance of impedance matching between successive stages of an electrical system? How can we attain this aim when designing analog passive filters?

Q35: What does the 3-dB point of a filter mean?

Q36: If a sixth-order analog BPF is required, what is the order of the prototype normalized LPF? Why?

Q37: Find the transfer function and draw the circuit diagram of a second-order Butterworth LPF with cutoff frequency \( f_c = 50 \text{ Hz} \), maximum gain \( G_m = 1 \), and \( RL = 10 \text{ Ω} \).

Q38: What is the function of the following circuit? Find the transfer function and plot approximately its magnitude frequency response (Fig. B.1).

Q39: What does uniform sampling mean?
**Q40:** The signal \( x(t) = \text{sinc}(t - 5) \) is sampled with \( f_s = 2 \text{ Hz} \). Plot the magnitude spectrum of the digitized signal \( x(n) \). Is there aliasing? How can we reconstruct \( x(t) \) from \( x(n) \)?

**Q41:** Explain what frequency aliasing means.

**Q42:** Find the convolution of the two signals: \( x(n) = \{2, 3, 4\} \) and \( y(n) = \{1, -5\} \).

**Q43:** Explain the meaning of the DTFT and the DFT. How can we represent digital systems and their I/O relationships in the time and the frequency domains using these transforms? Which transform is more useful in practice?

**Q44:** Which is more general: the DFT or the ZT? How can we represent digital systems and their I/O relationships using the ZT? How is the ZT related to DTFT?

**Q45:** State the conditions in the time-domain that a system is BIBO stable. What are the equivalent conditions in the z-domain?

**Q46:** What is the relationship between the s-plane and the z-plane?

**Q47:** Explain why the single-sided z-transform is sufficient for engineering applications.

**Q48:** Explain why the digital subsystem \( H(z) = z^{-1} \) represents a unit delay for causal systems. How can we implement this system using digital hardware?

**Q49:** What is meant by the FFT?

**Q50:** Find the difference equation, the transfer function, the impulse response, and the frequency response of the digital system shown in figure below, noting that \( f_s = 1 \text{ Hz} \). Is it FIR or IIR filter? LP or HP? What do you expect the function of this system? Does it have an exact linear phase? Plot the pole-zero diagram. Is the system stable?

**Q51:** a) What is the energy of the digital signal \( x(n) = 0.5 n T_s u(n) \), knowing that \( f_s = 100 \text{ Hz} \)? (Ans. 0.726 J) (Hint: use Tables for summation.)

**Q52:** What is the energy of the analog signal \( x(t) = 0.5 t u(t) \)? (Ans. 0.721 J)

**Q53:** How can we make the circular convolution of two finite digital sequences \( x(n) \) (length \( N_1 \)) and \( h(n) \) (length \( N_2 \)) equal to their linear convolution?

**Q54:** State the advantages of DSP over ASP.

**Q55:** Explain the function of the moving average digital filter. Is it FIR or IIR? State a real-life application for this filter.

**Q56:** Design a digital integrator. Is it FIR or IIR? LP or HP?
Q57: Implement the following digital system in cascade:

\[ H(z) = \frac{(2z - 5)(z + 1)}{(z^3 + 5z^2 + 6z)}. \]

Q58: Why do we normally choose complex poles as conjugates in digital systems?

Q59: How can we equalize the distorting effect of a communication channel?

Q60: What are the advantages of FIR over IIR filters?

Q61: Explain Gibbs phenomenon and how it can be dealt with in digital systems.

Q62: Design an ideal FIR HPF with cutoff frequency \( f_c = 20 \) Hz, knowing that the sampling frequency is \( f_s = 100 \) Hz. (Hint: use Tables to find \( h(n) \) for LPF, then convert to HPF).

Q63: Plot the pole diagram of the sinusoidal digital oscillator. How can we deduce the frequency of oscillation from these poles?

Q64: Simulate the following DSP systems using MATLAB (find the output time signal for arbitrary causal input \( x(n) \) and the frequency response of the system):

(a) The system shown in Fig. B.3
(b) The system shown in Fig. B.2.

Q65: Design a digital DC blocker with \( f_s = 20 \) kHz that attenuates the frequency \( f = 20 \) Hz by \(-0.1\) dB.

Q66: A signal \( x(t) = \sin(t) \) is corrupted by AWGN \( n(t) \) with SNR = 0 dB. Simulate the noisy signal \( x(t) = x(t) + n(t) \) using MATLAB and find the spectra of all signals. Design a Butterworth analog LPF to reduce noise and find the output of this filter.

Q67: A random signal \( n(t) \) is received and saved as a data file with \( f_s = 1 \) kHz and length \( N = 1000 \) samples. Write a MATLAB code to find the pdf of this signal.

Q68: An analog signal has a time duration of 3 ms. How long would be the frequency range of this signal?

Answer to Q4 (Hilbert Transformer):

1. We have \( y(t) = x(\omega t - \pi/2) = x\left[\omega \left(t - \frac{\pi/2}{\omega}\right)\right] \). Hence, there is a time-delay of \( \pi/2/\omega \), where the new time axis is \( t_r = t - \pi/2/\omega \). Therefore, the system has memory.

2. It is causal since there is no time-advance.

3. Consider only positive frequencies, hence the phase shift is \(-\pi/2\) [see Tutorial 33]. Consider sinusoids in complex exponential form.

Let \( z(t) = ae^{j(\omega_1t + c)} + be^{j(\omega_2t + d)} \), where \( a, b, c \) and \( d \) are constants. We have:
Hence, it is linear.

4. Let $y(t) = T[x(t)] = e^{j(\omega t + c - \pi t^2)}$ (output of HT). Now

$$T[x(t - t_o)] = T\left[e^{j(\omega(t-t_o)+c)}\right] = e^{j[\omega(t-t_o)+c-\pi/2]} = y(t - t_o).$$

Hence, it is time-invariant.

5. Clearly it is BIBO stable, as it does not affect the magnitude of the signal.

**Miscellaneous DSP Exercises—B**

Q1: Explain why we cannot use the impulse invariance method to design high pass digital filters (Hint: due to aliasing).

Q2: Show that the bilinear transformation preserves stability between the analog prototype filter and the digital filter (Hint: show that the LHS of the s-plane is transformed inside the unit circle in the z-plane, as shown in the Lecture Notes).

Q3: Can we use the bilinear transformation to design a digital HPF from an analog prototype HPF? (Hint: yes, since there is no aliasing).

Q4: Explain how the analog frequency is transformed using the bilinear transformation (see Tables).

Q5: Can we use the global average to find the trend of prices in a stock market? Why? What kind of filter do you suggest for this purpose? Why? (Hint: no, since the price trend is non-stationary, i.e., time-varying. We use either an FIR moving average filter, or an alpha filter. We prefer the alpha filter for its simple structure).

Q6: Explain how we estimate a communication channel transfer function using an FIR filter.
Q7: Where do you expect the poles of a digital resonator to be located in the z-plane?

Q8: (a) Draw a block diagram for a generic FIR filter of order N.
   (b) Plot the impulse response of this filter.
   (c) What is the condition on this filter to have a linear phase transfer function?
   (d) Plot the impulse response of a linear FIR filter and draw an efficient block diagram assuming N is odd.

Q9: Two digital filters have impulse responses given by \( h_1(n) = \{1, 2, -1, -1, 2, 1\} \) (starting at \( n = 9\)) and \( h_2(n) = \{1, 2, -1, 5, -1, 2, 1\} \) (starting at \( n = 0\)).
   (a) Which one is an FIR filter? (Ans. both)
   (b) Which one have a linear phase transfer function? (Ans. both, as they have symmetric impulse response.)
   (c) Plot the above impulse responses.
   (d) Draw the efficient implementation diagrams of these filters.

Q10: (a) Plot the transfer function of an ideal digital LPF with cutoff frequency 10 Hz and sampling frequency 50 Hz. Use the frequency range \([-100, 100\) Hz].
   (b) Roughly plot the impulse response of this filter.
   (c) repeat part (a) for a high-pass filter with cutoff 10 Hz.

Q11: Explain Gibbs phenomenon. How can we alleviate the effect of this phenomenon? (Hint: when we truncate the infinite impulse response of an ideal digital filter using a time window, we get magnitude ripples in the filter transfer function, with maxima at the ends of the transfer function. This ripple is significant when we use a rectangular widow.)

Q12: Explain why we need just one matched filter at the receiver of a binary communication system with antipodal signals (symbols).

Q13: Explain the basic operation of the optimal receiver in a binary communication system with orthogonal signals. [Hint: it consists of two matched filters, one matched to the symbol that represents logic “0”, the other to “1” (explain how we choose the impulse responses). If “0” was transmitted, the output of the first matched filter (which is matched to “0”) will be higher than that of the second (which is matched to “1”). This is because the matched filter is a correlator. Hence, the comparator will decide that “0” was transmitted. Similar reasoning if “1” was transmitted, where the output of the second filter will be higher this time. Draw a block diagram.]

Q14: A signal \( x(t) = \sin(\omega t) \) was corrupted by a white Gaussian noise \( n(t) \) with variance 0.1. Find the statistical mean, time mean, and variance of the signal \( s(t) = x(t) + n(t) \). [Ans. Statistical mean = \( \text{mean}\{x(t) + n(t)\} = \text{mean}\{x(t)\} + \text{mean}\{n(t)\} = x(t) + 0 = x(t) \), the time mean = \( \text{mean}\{x(t)\} + \text{mean}\{n(t)\} = 0 + 0 = 0 \). Note that since \( n(t) \) is ergodic, its time mean = its statistical mean.]
Q15: The analog Hilbert transform is a linear system that shifts the phase of the input signal by $-90^\circ$. Explain the operation of this system in the frequency domain and in the time domain.

Q16: Define the analytic signal and explain how it can be used for spectral economy.

Q17: Explain with an example why we modulate signals before radio transmission.

Q18: The figure below shows SSBSC AM generator. Show how this structure is used for spectral economy. Suggest a demodulation circuit (Fig. B.4).

Q19: Explain the operation of the first-order sinusoidal DPLL with a sinusoidal input. State and comment on the system equations. Draw the block diagram and the sampling process.

Q20: What are the main advantages of digital PLLs over analog PLLs?

Q21: Draw the block diagrams of a generic analog PLL and a generic DPLL.

Q22: Explain the meaning of the locking range of a DPLL.

Q23: Explain how can the DPLL track the frequency of a sinusoidal signal corrupted by noise. [Hint: the peak of the PLL frequency pdf will estimate the input frequency, if the original frequency is inside the loop locking range.]

Q24: How do you expect the relation between the variance of the PLL frequency and the input SNR? [Ans. the variance decreases when SNR increases; it approaches zero (hence the frequency pdf will approach a spike over the true input frequency) if the SNR is very high.]

Q25: For a sinusoidal input signal $x(t) = A \sin(\omega_c t + \theta(t))$, the phase equation in a first-order SDPLL is given by $\phi(k) = \theta(k) - \omega_c \sum_{i=0}^{k-1} y(i)$, where $y(i)$ is the output of the digital filter at the $i$th sampling instant, and $\theta(k)$ is the information-bearing phase. Explain (with a block diagram) how to demodulate PM signals using SDPLL.

Q26: Find the system equation for the second-order SDPLL.
Q27: Explain the basic operation of an adaptive filter. State how can we choose the reference signal in communications and noise reduction. Draw a block diagram for an adaptive noise canceler.

Q28: The algorithm for adaptive Wiener filter is as follows:

Define:
\[ h(k) = [h_0(k) h_1(k) h_2(k) \ldots h_M(k)] \]
the filter coefficients at the \( k \)th instant.
\[ y(n) = [y(n) y(n-1) \ldots y(n-M)] \]
the observed signal vector at the time instant \( n \). The algorithm can be described in vector form (MATLAB-like code) as follows:

\[ h(0) = 0; \] % Initialize the filter coefficients.
For \( n = 1: N \) % \( N = \text{length}(y); \)
\[ \hat{x}(n) = h(n-1) * y^T(n); \] % Filter output (this is matrix multiplication).
\[ e(n) = d(n) - \hat{x}(n); \]
\[ h(n) = h(n-1) + \mu * e(n) \]
\[ y(n); \] % \( \mu \) is the convergence-accuracy coefficient.
end Show how to implement this algorithm using DSP hardware.

Q29: What is the basis of finding the coefficients of the Wiener filter? [Ans. minimizing the mean-squared error between the estimated signal and the reference signal.]

Q30: What is meant by the intersymbol interference (ISI)? Explain how an adaptive filter with a training sequence can be used for ISI reduction.

Q31: What are the mean and variance of uniform quantization noise?

Q32: Explain why we need non-uniform quantization for audio applications, and how to achieve this kind of quantization.

Q33: How can we improve the signal-to-quantization noise ratio? Explain [Hint: by oversampling.]

Q34: Explain how the oversampling process can assist in reducing the order of the anti-aliasing filter of the compact disc system.

Q35: With a block diagram, explain the operation of the analog delta modulation system. Suggest a demodulation circuit. What is the slope overload? What is the granular noise? How can we improve the accuracy of this system? Implement the system in the digital domain.
Q36: Why do we need an adaptive delta modulation system? An adaptive algorithm for the step size is given by: \( \Delta_n = \Delta_{n-1} K^{y(n)y(n-1)} \), where \( K > 1 \) is a constant. Implement this algorithm using DSP hardware. Draw the block diagram of the adaptive modulator and the demodulator.

Q37: A block diagram of a sigma-delta modulator is shown below. Suggest a decoder. Represent this system in Laplace domain and show how it can be used for noise shaping (Fig. B.5).

Q38: Why is the sigma-delta modulator more appropriate for audio applications than the delta modulator?

Q39: Design a circuit for DC blocking.

Q40: Design a digital SD modulator and demodulator.

**Miscellaneous DSP Exercises: C**

Q1: A fifth-order elliptic low-pass filter with the following specifications:

- Pass-band peak-to-peak ripple = 0.5 dB
- Minimum stop-band attenuation = 40 dB
- The pass-band-edge frequency = 0.3\( \pi \)  Use Matlab to implemented this filter with second-order sections.

Q2: A 12-bit A/D converter is used to sample an analog signal. The sampled signal \( x(n) \) is stored in the lower bits of DSP processor such that the corresponding maximum dynamic range will be 1/16. Then \( x(n) \) is to be passed to a 16-bit IIR filter whose transfer function is

\[
H(z) = \frac{1}{1 - 0.98z^{-1}}.
\]

Scale the output signal such that its upper limit does not exceed unity.

Q3: Consider the first-order IIR filter described by the infinite-precision difference equation

\[
y(n) = -0.625y(n-1) + x(n).
\]

Show whether this system would trap into limit cycle when implemented using 4-bit (a) rounding arithmetic, (b) truncation arithmetic. Let \( x(n) = 0 \) and \( y(0) = 3/8 \). Find the magnitude and frequency of the oscillation for each case, if any.

Q4: A second-order IIR filter described by the following infinite-precision difference equation

\[
y(n) = -a_1y(n-1) - a_2y(n-2) + x(n).
\]

Derive the dead band bound that govern the occurrence of limit cycles in this structure when implemented using rounding arithmetic.
Miscellaneous DSP Exercises: D

Q1: Prove that \( c_M(n) = \frac{1}{M} \sum_{k=0}^{M-1} e^{-j2\pi kn/M} \).

Q2: Figure B.6 shows the spectrum of a band-pass signal. Show that down-sampling this signal by \( M = 3 \) produces an alias-free spectrum.

Q3: Write a MATLAB code to change the sampling rate from 48 to 44.1 kHz as explained in Example (2), Chap. 5.

Q4: Consider the sinusoidal signal \( x(n) = \cos(0.1 \pi n) \). Determine the frequency spectrum of the 3-fold down-sampled version.

Q5: Prove the decimation and interpolation identities shown in Fig. 5.13.

Q6: Given the following specifications of the sampling rate conversion DSP system shown in Fig. 5.11: \( L = 2 \), \( M = 3 \) and the input audio signal is sampled at 6 kHz, whereas the output signal sampling rate should be 9 kHz. Determine the filter length and cutoff frequencies for the required filter (window design method could be used).

Q7: The signal \( x(n) \) was sampled at a frequency \( f_s = 10 \) kHz. Consider the following two cases: (a) Resample \( x(n) \) at a new sampling frequency \( f_{s1} = 22 \) kHz. (b) The signal signal \( x(n) \) is to be resampled to a new sampling frequency \( f_{s2} = 8 \) kHz.

Q8: Consider the transfer function of a FIR filter: \( H(z) = 0.25 + 0.5 z^{-1} + 0.75 z^{-2} + z^{-3} + 1.25 z^{-4} + 1.5 z^{-5} \). Use polyphase decomposition technique to implement a factor of \( M = 3 \) decimator.

Q9: A single stage decimator structure is used to reduce the sampling rate of a signal from 12000 to 500 Hz. The specifications of the single-stage decimator low-pass FIR filter \( H(z) \) are: pass-band edge = 225 Hz, stop-band edge = 250 Hz, pas-band ripple = 0.004, and stop-band attenuation = 0.001. Assume \( H(z) \) as an equiripple linear-phase FIR filter.
Q10: To convert the CD audio at the sampling rate of 44.1 kHz to the MP3 sampling rate of 48 kHz, a conversion factor of $L/M = 160/147$ is needed. Single-stage scheme would require an unfeasible FIR filter sizes for interpolation and decimation. How would you tackle this problem?
Appendix C: Tables and Formulas

Basic Definitions

The rectangular pulse: \( \Pi(t/T) = \Pi_T(t) = \begin{cases} 1, & |t| \leq T/2 \\ 0, & \text{elsewhere} \end{cases} \)

The triangular pulse: \( \Lambda(t/T) = \Lambda_T(t) = \begin{cases} 1 - 2|t|/T, & |t| \leq T/2 \\ 0, & \text{elsewhere} \end{cases} \)

The unit step function: \( u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \)

The sinc function: \( \text{sinc}(t) = \frac{\sin(\pi t)}{(\pi t)} \)

The signum function: \( \text{sgn}(t) = \begin{cases} 1, & t \geq 0 \\ -1, & t < 0 \end{cases} \)

Useful Formulas

1. \( e^{\pm j\phi} = \cos(\phi) \pm j\sin(\phi) \) (Euler);
   \( |e^{\pm j\phi}| = 1 \);
   \( \cos(x) = \cosh(jx) = (e^{jx} + e^{-jx})/2; \)
   \( \sin(x) = (1/j)\sinh(jx) = (e^{jx} - e^{-jx})/(2j); \)

2. \( \cos^2(\phi) + \sin^2(\phi) = 1; \)
   \( \tan(x) = \sin(x)/\cos(x); \)
   \( \cos^2(\phi) = (1/2) + (1/2)\cos(2\phi); \)
   \( \sin^2(\phi) = (1/2) - (1/2)\cos(2\phi) \)
3. \[
\cos(x \pm y) = \cos(x) \cos(y) \mp \sin(x) \sin(y);
\]
\[
\sin(x \pm y) = \sin(x) \cos(y) \pm \cos(x) \sin(y);
\]
\[
\tan(x \pm y) = \tan(x) \pm \tan(y)/[1 \mp \tan(x) \tan(y)];
\]
\[
\sin(2x) = 2 \sin(x) \cos(x); \cos(2x) = 2 \cos^2(x) - 1.
\]

4. \[
\cos(x) \cos(y) = (1/2)[\cos(x - y) + \cos(x + y)];
\]
\[
\sin(x) \sin(y) = (1/2)[\cos(x - y) - \cos(x + y)];
\]
\[
\sin(x) \cos(y) = (1/2)[\sin(x - y) + \sin(x + y)];
\]

5. \[
\tan(x/2) = \sin(x)/[1 + \cos(x)] = [1 - \cos(x)]/\sin(x);
\]
\[
\sin(x/2) = \pm \sqrt{1 - \cos(x)}/2;
\]
\[
\cos(x/2) = \pm \sqrt{1 + \cos(x)}/2.
\]

6. \[
\sinh(x) = (e^x - e^{-x})/2;
\]
\[
\cosh(x) = (e^x + e^{-x})/2;
\]
\[
\tanh(x) = \sinh(x)/\cosh(x);
\]
\[
\sinh^{-1}(x) = \ln(x + \sqrt{x^2 - 1});
\]
\[
\cosh^{-1}(x) = \ln(x + \sqrt{x^2 - 1});
\]
\[
\tanh^{-1}(x) = \frac{1}{2} \ln \frac{1 + x}{1 - x};
\]
\[
\cosh^2(x) - \sinh^2(x) = 1;
\]

7. \[
df(y)/dx = [df(y)/dy][dy/dx];
\]
\[
d \left[ \int_a^t f(u)du \right] /dt = f(t), \quad \text{where } a \text{ is constant}
\]
8. \( dx^a / dx = ax^{a-1}; \)
\[
\int x^a \, dx = x^{a+1} / (a + 1), \quad (a \neq -1);
\]
\( d \ln(x) / dx = 1/x; \)
\[
\int x^{-1} \, dx = \ln(x);
\]

9. \( df(y) / dx = [df(y) / dy][dy / dx]; \)
\( d \sin(x) / dx = \cos(x); \)
\( d \cos(x) / dx = -\sin(x); \)
\( d \tan(x) / dx = -1 / \cos^2(x); \)
\( de^x / dx = e^x; \)
\( dc^x / dx = \ln(c) e^x; \)
\( d \tan^{-1}(x) / dx = 1 / (1 + x^2); \)
\( d \tanh^{-1}(x) / dx = 1 / (1 - x^2); \)
\( d \sinh^{-1}(x) / dx = 1 / \sqrt{x^2 + 1}; \)
\( d \cosh^{-1}(x) / dx = 1 / \sqrt{x^2 - 1}; \)
\( d \sin^{-1}(x) = -d \cos^{-1}(x) = 1 / \sqrt{1 - x^2}. \)

10. \[
\sum_{n=0}^{N-1} r^n = \begin{cases} 
(1 - r^N) / (1 - r), & r \neq 1 \\
N, & r = 1
\end{cases};
\]
\[
\sum_{n=0}^{N-1} (a + nr) = \frac{N}{2} [2a + (N - 1)r] = \frac{N}{2} \text{ [first + last]}. 
\]

11. If \( \log_b(y) = x \Leftrightarrow y = b^x; \)
\( \log_b(xy) = \log_b(x) + \log_b(y); \)
\( \log_b(x/y) = \log_b(x) - \log_b(y); \)
\( \log_b(x^n) = n \log_b(x); \quad x^0 = 1; \)
\( \log_b b = 1; \)
\( \log_b(y) = \log_c(y) / \log_c(b); \)
\( \log_a(a^x) = x. \)
Natural logarithm: \( \ln(x) = \log_e(x) \), \( e \approx 2.718281828459 \cdots \)
Common logarithm: \( \log_{10}(x) \).

12. Power series expansion: \( \frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots \) (for \( |x| < 1 \));
\[
\sin(x) = x - x^3/3! + x^5/5! - \cdots;
\]
\[
\sin^{-1}(x) = x + x^3/6 + 3x^5/40 + 5x^7/112 + 35x^9/1152 + \cdots;
\]
\[
\cos(x) = 1 - x^2/2! + x^4/4! - \cdots;
\]
\[
\cos^{-1}(x) = \pi/2 - x - x^3/6 - 3x^5/40 - 5x^7/112 - \cdots \) (for \( |x| < 1 \));
\[
\tan(x) = x + x^3/3 + 2x^5/15 + 17x^7/315 + 62x^9/2835 + \cdots \) (for \( |x| < \pi/2 \);
\]
\[
\tan^{-1}(x) = x - x^3/3 + x^5/5 - \cdots \) (for \( |x| < 1 \));
\]
\[
\tan^{-1}(1+x) = \pi/4 + x/2 - x^2/4 + x^3/12 + x^5/40 + \cdots \) (for \( |x| < \infty \));
\]
\[
\ln(1+x) = x - x^2/2 + x^3/3 - x^4/4 + \cdots \) (for \( |x| < 1 \)).
\]

13. The convolution integral: \( x(t) * h(t) = \int_{-\infty}^{\infty} x(\lambda)h(t-\lambda)d\lambda \);

14. The correlation integral: \( R_{xy}(\tau) = \int_{-\infty}^{\infty} x(t)y(t+\tau)dt \);
\text{Autocorrelation: } R_x(\tau) = \int_{-\infty}^{\infty} x(t)x(t+\tau)dt.

15. Discrete convolution: \( x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \);

16. Dirac delta function, \( \delta(t) \): It is defined by the integral
\[
\int_{-\infty}^{\infty} x(t)\delta(t-t_0)dt = x(t_0),
\]
where \( x(t) \) is any continuous function.
Note that: \( \delta(at) = (1/|a|)\delta(t) \); \( \delta(-t) = \delta(t) \); \( \int_{-\infty}^{\infty} \delta(t)dt = 1. \)

17. \( x(t) * \delta(t-t_0) = x(t-t_0); x(t)\delta(t-t_0) = x(t_0)\delta(t-t_0) \)

18. Gaussian distribution: \( p(\lambda) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(\lambda-m)^2/2\sigma^2} \), \( m = \text{mean}, \sigma^2 = \text{variance} \)

19. Schwarz’s inequality: \( | \int g_1(x)g_2(x)dx |^2 \leq (\int |g_1(x)|^2dx)(\int |g_2(x)|^2dx) \)
\text{(equality holds only when } g_1(x) = kg_2(x), k \text{ being a constant.})

20. Roots of the quadratic equation: \( ax^2 + bx + c = 0 \) are given by
\[
r_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.
\]
21. Complex cubic roots of 1: Call them \( \{u_k \}_{k=0,1,2} \). Let \( z = 1 \). Then:

\[
z = 1 = 1e^{i2\pi} \Rightarrow \sqrt[3]{z} = (e^{i2\pi})^{1/3} = \left\{ 0, 1, 2, \ldots \right\} = \{1, e^{i2\pi/3}, e^{i4\pi/3}\} = \left\{1, -1 \pm j\sqrt{3}/2 \right\} = \{u_k | k = 0, 1, 2\}.
\]

Complex cubic roots of a real number \( a = h^3 \) are \( \{r_k = hu_k | k = 0, 1, 2\} \).

22. Roots of the cubic equation: \( x^3 + ax^2 + bx + c = 0 \).

Let \( p = b - \frac{a^2}{3} \);

\[
q = c + \frac{2a^3 - 9ab}{27} ;
\]

\[
h = \frac{1}{2} \left[ \frac{q}{2} \pm \sqrt{\frac{q^2}{4} + \frac{p^3}{27}} \right] (\pm : \text{take either } + \text{ or } - ; \text{ but choose carefully when } p = 0 \text{ so that } h \neq 0);
\]

\[
r_k = hu_k \{ k = 0, 1, 2 \}.
\]

Then: \( x_k = \frac{p}{3r_k} - \frac{a}{3} - r_k \)

23. \( (x + y)^n = \sum_{k=0}^{n} C(n, k)x^n y^k \) where \( C(n, k) = \frac{n!}{k! (n - k)!} \).

### Laplace transform pairs and theorems

<table>
<thead>
<tr>
<th>( x(t) )</th>
<th>( X(s) )</th>
<th>Property</th>
<th>Transform pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \delta(t) )</td>
<td>1</td>
<td>Linearity</td>
<td>( ax(t) + by(t) \quad \overset{L}{\longleftrightarrow} \quad aX(s) + bY(s) )</td>
</tr>
<tr>
<td>( u(t) )</td>
<td>( \frac{1}{s} )</td>
<td>Scaling</td>
<td>( x(at) \quad \overset{L}{\longleftrightarrow} \quad X(s/a)/a )</td>
</tr>
<tr>
<td>( \frac{1}{s} u(t) )</td>
<td>( \frac{1}{s^2} )</td>
<td>Time shift (delay)</td>
<td>( x(t - t_0) \quad \overset{L}{\longleftrightarrow} \quad X(s)e^{-st_0}, t_0 &gt; 0 )</td>
</tr>
<tr>
<td>( e^{-at} u(t) )</td>
<td>( \frac{1}{s+a} )</td>
<td>s-Shift</td>
<td>( x(t)e^{-at} \quad \overset{L}{\longleftrightarrow} \quad X(s+a) )</td>
</tr>
<tr>
<td>( \frac{1}{s} e^{-at} u(t) )</td>
<td>( \frac{1}{(s+a)^2} )</td>
<td>Convolution</td>
<td>( x(t) * y(t) \quad \overset{L}{\longleftrightarrow} \quad X(s)Y(s) )</td>
</tr>
<tr>
<td>( \sin(\omega_0 t) u(t) )</td>
<td>( \frac{\omega_0}{s^2 + \omega_0^2} )</td>
<td>Conjugation</td>
<td>( x^<em>(t) \quad \overset{L}{\longleftrightarrow} \quad X^</em>(s^*) )</td>
</tr>
<tr>
<td>( \cos(\omega_0 t) u(t) )</td>
<td>( \frac{s}{s^2 + \omega_0^2} )</td>
<td>Initial value theorem</td>
<td>( \text{lim}<em>{t \to 0} x(t) = \lim</em>{s \to \infty} sX(s) )</td>
</tr>
<tr>
<td>( t \sin(\omega_0 t) u(t) )</td>
<td>( \frac{2\omega_0}{(s^2 + \omega_0^2)^2} )</td>
<td>Final value theorem</td>
<td>( \text{lim}<em>{s \to 0} sX(s) = \lim</em>{t \to \infty} x(t) )</td>
</tr>
<tr>
<td>( t \cos(\omega_0 t) u(t) )</td>
<td>( \frac{\omega_0^2}{(s^2 + \omega_0^2)^2} )</td>
<td>Time differentiation</td>
<td>( dx(t)/dt \quad \overset{L}{\longleftrightarrow} \quad sX(s) - x(0^-) )</td>
</tr>
<tr>
<td>( e^{-at} \sin(\omega_0 t) u(t) )</td>
<td>( \frac{a\omega_0}{(s+a)^2 + \omega_0^2} )</td>
<td>Time integration</td>
<td>( \int_0^t x(u) du \quad \overset{L}{\longleftrightarrow} \quad X(s)/s )</td>
</tr>
<tr>
<td>( e^{-at} \cos(\omega_0 t) u(t) )</td>
<td>( \frac{s+a}{(s+a)^2 + \omega_0^2} )</td>
<td>Multiplication by ( t^n )</td>
<td>( t^n x(t) \quad \overset{L}{\longleftrightarrow} \quad (-1)^n \frac{d^n X(s)}{ds^n}, n \text{ integer} )</td>
</tr>
</tbody>
</table>
## Appendix C: Tables and Formulas

### Fourier transform pairs and theorems

<table>
<thead>
<tr>
<th>$x(t)$</th>
<th>$X(f)$</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Pi(t/T) = \Pi_T(t)$</td>
<td>$T \text{ sinc}(Tf)$</td>
<td>$a(x(t) + bY(t)) \xrightarrow{f} aX(f) + bY(f)$</td>
</tr>
<tr>
<td>$\Lambda(t/T) = \Lambda_T(t)$</td>
<td>$\frac{1}{2} \text{ sinc}^2 \left( \frac{tf}{2} \right)$</td>
<td>$x(at) \xrightarrow{f} \frac{X(f/a)}{</td>
</tr>
<tr>
<td>$\text{sinc}(t)$</td>
<td>$\frac{1}{2} \Pi_L(f)$</td>
<td>$x(-t) \xrightarrow{f} X(-f)$</td>
</tr>
<tr>
<td>$e^{-at} u(t), a &gt; 0$</td>
<td>$\frac{1}{s + 2\pi i}$</td>
<td>$x^<em>(t) \xrightarrow{f} X^</em>(-f), x^<em>(-t) = X^</em>(f)$</td>
</tr>
<tr>
<td>$te^{-at} u(t), a &gt; 0$</td>
<td>$\frac{1}{(s + 2\pi i)^2}$</td>
<td>$x(t) \xrightarrow{f} X(f) \xrightarrow{f} x(f), x$ even</td>
</tr>
<tr>
<td>$e^{-\alpha^2 t^2}$</td>
<td>$\sqrt{\pi} e^{-\pi \alpha^2 f^2}$</td>
<td>$x(t) e^{\alpha^2 f^2} \xrightarrow{f} X(f - \alpha)$</td>
</tr>
<tr>
<td>1</td>
<td>$\delta(f)$</td>
<td>$x(t) \xrightarrow{f} \frac{1}{2} X(f - f_0) + \frac{1}{2} X(f + f_0)$</td>
</tr>
<tr>
<td>$\text{sgn}(t)$</td>
<td>$\frac{1}{2i\pi} \delta(f) + \frac{1}{2\pi i}$</td>
<td>$d^n x(t)/dt^n \xrightarrow{f} (2\pi f)^n X(f)$</td>
</tr>
<tr>
<td>$u(t)$</td>
<td>$\frac{1}{2} \delta(f) + \frac{1}{2\pi i}$</td>
<td>$x(t) + y(t) \xrightarrow{f} X(f) Y(f)$</td>
</tr>
<tr>
<td>$\cos(2\pi f_0 t + \theta_0)$</td>
<td>$\frac{1}{2} e^{j\theta_0} \delta(f - f_0) + \frac{1}{2} e^{-j\theta_0} \delta(f + f_0)$</td>
<td>$x(t) y(t) \xrightarrow{f} X(f) * Y(f)$</td>
</tr>
<tr>
<td>$\sin(2\pi f_0 t + \theta_0)$</td>
<td>$\frac{1}{2} e^{j\theta_0} \delta(f - f_0) - \frac{1}{2} e^{-j\theta_0} \delta(f + f_0)$</td>
<td>If $x(t)$ is real, then $X(-f) = X^*(f)$</td>
</tr>
<tr>
<td>$\sum_{n=-\infty}^{\infty} \delta(t - nT_s)$</td>
<td>$f_s \sum_{n=-\infty}^{\infty} \delta(t - kf_s)$, where $f_s = \frac{1}{T_s}$</td>
<td></td>
</tr>
<tr>
<td>$e^{\pm j2\pi f_0 t}$</td>
<td>$\delta(f \pm f_0)$</td>
<td></td>
</tr>
</tbody>
</table>

### z-Transform pairs and theorems

<table>
<thead>
<tr>
<th>$x(n)$</th>
<th>$X(z)$</th>
<th>Property</th>
<th>Transform pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta(n)$</td>
<td>1</td>
<td>Linearity</td>
<td>$a(x(n) + bY(n)) \xrightarrow{z} aX(z) + bY(z)$</td>
</tr>
<tr>
<td>$u(n)$</td>
<td>$\frac{1}{z-1}$</td>
<td>z-Scaling</td>
<td>$a^n x(n) \xrightarrow{z} X(z/a)$</td>
</tr>
<tr>
<td>$a^n u(n)$</td>
<td>$\frac{1}{z-a}$</td>
<td>Convolution</td>
<td>$x(n) * y(n) \xrightarrow{z} X(z) Y(z)x(n)$</td>
</tr>
<tr>
<td>$na^n u(n)$</td>
<td>$\frac{z}{(z-a)^2}$</td>
<td>Conjugation</td>
<td>$x^<em>(n) \xrightarrow{z} X^</em>(z^*)$</td>
</tr>
<tr>
<td>$(n+1)a^n u(n)$</td>
<td>$(\frac{z}{z-a})^2$</td>
<td>z-Differentiation</td>
<td>$nx(n) \xrightarrow{z} -z\frac{dX(z)}{dz}$</td>
</tr>
<tr>
<td>$\sin(bn)u(n)$</td>
<td>$\frac{\sin(b)z}{z^2 - 2\cos(b)z + 1}$</td>
<td>Time integration</td>
<td>$\sum_{k=-\infty}^{n} x(k) \xrightarrow{z} \frac{z}{1-z} X(z)$</td>
</tr>
<tr>
<td>$\cos(bn)u(n)$</td>
<td>$\frac{z^2 - \cos(b)z}{z^2 - 2\cos(b)z + 1}$</td>
<td>Time differentiation</td>
<td>$x(n) - x(n-1) \xrightarrow{z} (1 - z^{-1}) X(z)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time shift (delay)</td>
<td>$x(n-m) \xrightarrow{z} z^{-m} X(z)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Initial value</td>
<td>If $x(n) = 0$ for $n &lt; 0$, then $x(0) = \lim_{z \to \infty} Z(x(z))$</td>
</tr>
</tbody>
</table>
### Denominator polynomial coefficients for normalized LPF’s

<table>
<thead>
<tr>
<th>n</th>
<th>( b_o )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( b_3 )</th>
<th>( b_4 )</th>
<th>( b_5 )</th>
<th>( b_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>1.4142</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>2.0</td>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
<td>2.6131</td>
<td>3.4142</td>
<td>2.6131</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.0</td>
<td>3.2361</td>
<td>5.2361</td>
<td>5.2361</td>
<td>3.2361</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.0</td>
<td>3.8637</td>
<td>7.4641</td>
<td>9.1416</td>
<td>7.4641</td>
<td>3.8637</td>
<td></td>
</tr>
</tbody>
</table>

### Butterworth

- **0.5-dB passband ripple Chebychev** \( (\varepsilon^2 = 0.1220) \)
<table>
<thead>
<tr>
<th>n</th>
<th>( b_o )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( b_3 )</th>
<th>( b_4 )</th>
<th>( b_5 )</th>
<th>( b_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.8628</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.5162</td>
<td>1.425</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.7157</td>
<td>1.5349</td>
<td>1.2529</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.3791</td>
<td>1.0255</td>
<td>1.7169</td>
<td>1.1974</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.1789</td>
<td>0.7525</td>
<td>1.3096</td>
<td>1.9374</td>
<td>1.1725</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.0948</td>
<td>0.4324</td>
<td>1.1719</td>
<td>1.5898</td>
<td>2.1718</td>
<td>1.1592</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.0447</td>
<td>0.2821</td>
<td>0.7557</td>
<td>1.6479</td>
<td>1.8694</td>
<td>2.4127</td>
<td>1.1512</td>
</tr>
</tbody>
</table>

- **1.0-dB passband ripple Chebychev** \( (\varepsilon^2 = 0.2589) \)
<table>
<thead>
<tr>
<th>n</th>
<th>( b_o )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( b_3 )</th>
<th>( b_4 )</th>
<th>( b_5 )</th>
<th>( b_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.9652</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.1025</td>
<td>1.0977</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.4913</td>
<td>1.2384</td>
<td>0.9883</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.2756</td>
<td>0.7426</td>
<td>1.4539</td>
<td>0.9528</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.1228</td>
<td>0.5805</td>
<td>0.9744</td>
<td>1.6888</td>
<td>0.9368</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.0689</td>
<td>0.3071</td>
<td>0.9393</td>
<td>1.2021</td>
<td>1.9308</td>
<td>0.9283</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.0307</td>
<td>0.2137</td>
<td>0.5486</td>
<td>1.3575</td>
<td>1.4288</td>
<td>2.1761</td>
<td>0.9231</td>
</tr>
</tbody>
</table>

- **3-dB passband ripple Chebychev** \( (\varepsilon^2 = 0.9953) \)
<table>
<thead>
<tr>
<th>n</th>
<th>( b_o )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( b_3 )</th>
<th>( b_4 )</th>
<th>( b_5 )</th>
<th>( b_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0024</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.7079</td>
<td>0.6449</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.2506</td>
<td>0.9283</td>
<td>0.5972</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.1770</td>
<td>0.4048</td>
<td>1.1691</td>
<td>0.5816</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.0626</td>
<td>0.4080</td>
<td>0.5489</td>
<td>1.4150</td>
<td>0.5745</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.0442</td>
<td>0.1634</td>
<td>0.6991</td>
<td>0.6906</td>
<td>1.6628</td>
<td>0.5707</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.0157</td>
<td>0.1462</td>
<td>0.3000</td>
<td>1.0518</td>
<td>0.8314</td>
<td>1.9116</td>
<td>0.5684</td>
</tr>
</tbody>
</table>
Normalized low-pass filter LC element values

<table>
<thead>
<tr>
<th>n</th>
<th>( R_s )</th>
<th>( C_1 )</th>
<th>( L_2 )</th>
<th>( C_3 )</th>
<th>( L_4 )</th>
<th>( C_5 )</th>
<th>( L_6 )</th>
<th>( C_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.0</td>
<td>1.4142</td>
<td>1.412</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>1.0</td>
<td>2.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
<td>0.7654</td>
<td>1.8478</td>
<td>1.8478</td>
<td>0.7654</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.0</td>
<td>0.6180</td>
<td>1.6180</td>
<td>2.0</td>
<td>1.6180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.0</td>
<td>0.5176</td>
<td>1.4142</td>
<td>1.9319</td>
<td>1.9319</td>
<td>1.4142</td>
<td></td>
<td>0.5176</td>
</tr>
<tr>
<td>7</td>
<td>1.0</td>
<td>0.4450</td>
<td>1.2470</td>
<td>1.8019</td>
<td>2.0</td>
<td>1.8019</td>
<td></td>
<td>1.2470</td>
</tr>
</tbody>
</table>

**LPF Butterworth LC element values \((R_L = 1 \Omega)\)**

<table>
<thead>
<tr>
<th>n</th>
<th>( R_s )</th>
<th>( C_1 )</th>
<th>( L_2 )</th>
<th>( C_3 )</th>
<th>( L_4 )</th>
<th>( C_5 )</th>
<th>( L_6 )</th>
<th>( C_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.9841</td>
<td>0.9827</td>
<td>1.9497</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>1.8636</td>
<td>1.2804</td>
<td>1.8636</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.9841</td>
<td>0.9202</td>
<td>2.5864</td>
<td>1.3036</td>
<td>1.8258</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.0</td>
<td>1.8068</td>
<td>1.3025</td>
<td>2.6914</td>
<td>1.3025</td>
<td>1.8068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.9841</td>
<td>0.9053</td>
<td>2.5774</td>
<td>1.3675</td>
<td>2.7133</td>
<td>1.2991</td>
<td>1.7961</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.0</td>
<td>1.7896</td>
<td>1.2961</td>
<td>2.7177</td>
<td>1.3848</td>
<td>2.7177</td>
<td>1.2961</td>
<td>1.7896</td>
</tr>
</tbody>
</table>

**LPF 0.5-dB ripple Chebychev LC element values \((R_L = 1 \Omega)\)**

<table>
<thead>
<tr>
<th>n</th>
<th>( R_s )</th>
<th>( C_1 )</th>
<th>( L_2 )</th>
<th>( C_3 )</th>
<th>( L_4 )</th>
<th>( C_5 )</th>
<th>( L_6 )</th>
<th>( C_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3.0</td>
<td>0.5723</td>
<td>3.1317</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>2.2160</td>
<td>1.0883</td>
<td>2.2160</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.0</td>
<td>0.6529</td>
<td>4.4110</td>
<td>0.8140</td>
<td>2.5346</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.0</td>
<td>2.0722</td>
<td>1.1279</td>
<td>3.1025</td>
<td>1.1279</td>
<td>2.0722</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.0</td>
<td>0.6729</td>
<td>3.8725</td>
<td>0.7706</td>
<td>4.7107</td>
<td>0.9692</td>
<td>2.4060</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.0</td>
<td>2.0423</td>
<td>1.1311</td>
<td>3.1472</td>
<td>1.1942</td>
<td>3.1472</td>
<td>1.1311</td>
<td>2.0423</td>
</tr>
</tbody>
</table>

**LPF 1-dB ripple Chebychev LC element values \((R_L = 1 \Omega)\)**

![Diagram of a normalized low-pass filter with normalized frequency, \( \omega / \omega_c \), and stopband gain vs. normalized frequency for B–LPF.](image)
Stopband gain vs. norm. freq. for C−LPF ($r = 0.5$ dB)

Normalized frequency, $\omega/\omega_c$

Stopband gain vs. norm. freq. for C−LPF ($r = 1$ dB)

Normalized frequency, $\omega/\omega_c$
Useful Definitions and Relations

Complex Fourier Series (FS):

\[ x(t) = \sum_{k=-\infty}^{\infty} X_k e^{j2\pi k f_0 t}, \]
where:

\[ X_0 = \frac{1}{T_o} \int_{0}^{T_o} x(t) dt, \]

\[ X_k = \frac{1}{T_o} \int_{0}^{T_o} x(t) e^{-j2\pi k f_0 t} dt \]

Trigonometric FS:

\[ x(t) = a_o + \sum_{n=1}^{\infty} a_n \cos(2\pi n f_0 t) + b_n \sin(2\pi n f_0 t), \]
where:

\[ a_o = X_0, a_n = X_n + X_{-n}, b_n = j(X_n - X_{-n}). \]

[Note that \(X_n\)’s are the coefficients of the Complex FS.]
Fourier Transform (FT) pair:

\[ X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} \, dt, \]
\[ x(t) = \int_{-\infty}^{\infty} X(f) e^{+j2\pi ft} \, df \]

(Single-sided) Laplace Transform (LT):

\[ X(s) = \int_{0^-}^{\infty} x(t) e^{-st} \, dt, \text{ where:} \]
\[ s = \sigma + j\omega, \quad \omega = 2\pi f. \]

Discrete-Time Fourier Transform (DTFT) pair:

\[ X_s(f) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nf}, \]
\[ x(n) = \frac{1}{f_s} \int_{f_s}^{0} X_s(f) e^{+j2\pi nf} \, df \]

Some important DTFT pairs:

1. \( \frac{2f_c}{f_s} \text{sinc} \left( \frac{2f_c}{f_s} n \right) \overset{\text{DTFT}}{\longleftrightarrow} \Pi_{2f_c}(f), -\frac{f_c}{2} < f < \frac{f_c}{2} \]
   \((\text{Periodic rectangular pulse, period } = f_s)\)

2. \( \Pi_N(n) \overset{\text{DTFT}}{\longleftrightarrow} N \text{sinc}(Nf / f_s) / \text{sinc}(f / f_s) \quad [\text{a sinc-like function}] \)

The DTFT modulation property:

If \( x(n) \overset{\text{DTFT}}{\longleftrightarrow} X_s(f) \), then: \( x(n) \cos(2\pi f_0 / f_s) \overset{\text{DTFT}}{\longleftrightarrow} X_s(f - f_0) + X_s(f + f_0) \)

Discrete Fourier Transform (DFT) pair:

\[ X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \]
\[ x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{+j2\pi kn/N} \]

(Single-sided) z-Transform (ZT):

\[ X(z) = \sum_{n=0}^{\infty} x(n) z^{-n} \]

Low-pass to band-pass transformations

\[ \omega_{LN} = \left( \omega^2 - \omega_b^2 \right) / (\omega \omega_b), \quad s_{LN} = \left( s^2 + \omega_b^2 \right) / (s \omega_b), \quad \text{where:} \]
\[ \omega_b = \omega_a - \omega_l, \quad \omega_L = \sqrt{\omega_a \omega_l}. \]
Low-pass to high-pass transformations

\[ \omega_{LN} = \omega_c / \omega; \quad s_{LN} = \omega_c / s. \]

Low-pass to band-stop transformations

\[ \omega_{LN} = \omega \omega_b / \left( \omega^2 - \omega_s^2 \right), \quad s_{LN} = s \omega_b / \left( s^2 + \omega_s^2 \right), \quad \text{where:} \]
\[ \omega_b = \omega_u - \omega_l, \quad \omega_s = \sqrt{\omega_u \omega_l}. \]

Circuit denormalization

\[ R \rightarrow RZ; \quad L \rightarrow L \frac{Z}{W}; \quad C \rightarrow C \frac{Z}{W}, \quad \text{where:} \]
\[ Z = \text{Impedance scaling factor (ISF)} \]
\[ W = \text{Frequency scaling factor (FSF)} \]

Hardware analog filter transformations

\[ \frac{C}{L} \rightarrow \frac{L}{C} \]
\[ \frac{L}{C} \rightarrow \frac{L}{C} \frac{1}{\omega^2 C}, \quad \text{i.e.,} \quad L = 1 / \left( \omega^2 C \right) \]
\[ \text{and} \quad C = 1 / \left( \omega^2 L \right) \]

Hardware filter design rules (using normalized LPF standard circuits)

HPF design:
1. Transform \( LP \rightarrow HP \) circuit components
2. Denormalize \( HP \rightarrow HP \)

BPF design:
1. Denormalize \( LP \rightarrow LP \)
2. Transform \( LP \rightarrow BP \) circuit components

BSF design:
1. Transform \( LP \rightarrow HP \) circuit components
2. Denormalize \( HP \rightarrow HP \)
3. Transform \( HP \rightarrow BS \) circuit components

Magnitude response of Butterworth LPF:

\[ |H(\omega)| = G_m / \sqrt{1 + \left( \omega / \omega_c \right)^{2n}}; \]

where \( G_m = \text{maximum gain} \).
Butterworth DC gain: $G_{dc} = G_m$

Magnitude response of Chebychev-I LPF:

$$|H(\omega)| = G_m \sqrt{1 + \varepsilon^2 C_n^2 \left( \frac{\omega}{\omega_c} \right)}$$

where $C_o(x) = 1$, $C_1(x) = x$, $C_{n+1}(x) = 2x C_n(x) - C_{n-1}(x)$.

Chebychev DC gain:

$$G_{dc} = \begin{cases} 
G_m & \text{n odd} \\
G_m/\sqrt{1 + \varepsilon^2} & \text{n even}
\end{cases}$$

Matched filter: $h(t) = s(T - t)$.

Inverting and non-inverting amplifiers:

Hilbert Transformer (HT): $H(f) = -j \text{sgn}(f)$ with $h(t) = 1/(\pi \cdot t)$.

Hilbert Transform of $x(t)$ is: $\hat{x}(t) = x(t) * h(t)$ or: $\hat{X}(f) = -j \text{sgn}(f)X(f)$.

The analytic associate of $x(t)$ is: $z(t) = x(t) + j\hat{x}(t)$

Digital HT:

$$H(e^{j\Omega}) = -j \text{sgn}(\Omega), |\Omega| < \pi; h(n) = \begin{cases} 
\frac{2}{\pi} \sin^2(n\pi/2), & n \neq 0 \\
0 & \text{elsewhere}
\end{cases}$$

Bilinear Transform: $s \Rightarrow \frac{z-1}{z+1}$ with $\Omega = 2 \tan^{-1}(\omega) = 2 \tan^{-1}(2\pi f)$.

Impulse invariant transform:

$$H_a(s) = \sum_{m=1}^{M} \frac{c_m}{s - p_m} \Rightarrow H(z) = T_s \sum_{m=1}^{M} c_m \frac{z}{z - z_m} \quad \text{where} \quad z_m = e^{p_m T_s}.$$
Note that \( G_2 = 0 \) in this case.

Sinusoidal PLL: 2nd-order system equation:
\[
\phi(k+2) - 2\phi(k+1) + \phi(k) = K_2\sin[\phi(k)] - rK_2\sin[\phi(k+1)],
\]
where \( r = 1 + G_2/G_1 \).

Fixed point analysis: \( g(x) = x \) has a solution \( x^* \) only if \( |g'(x^*)| < 1 \).

Jacobian matrix:
\[
\frac{\partial(u,v)}{\partial(x,y)} = \begin{bmatrix}
\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\
\frac{\partial v}{\partial x} & \frac{\partial v}{\partial y}
\end{bmatrix}
\]
where \( u, v \) are functions of \( x, y \).

Eigenvalues \( \{ \lambda \} \) of a matrix \( X \):
\[
|\lambda I - X| = 0; \quad \text{where } I \text{ is the identity matrix.}
\]

Integrator \( H(z) = 1/(1 - z^{-1}) \); Differentiator \( H(z) = (1 - z^{-1})/T_s \).

Error functions:
\[
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-v^2} dv
\]
\[
Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-v^2/2} dv = \frac{1}{2} \text{erfc} \left( \frac{x}{\sqrt{2}} \right) = \frac{1}{2} - \frac{1}{2} \text{erf} \left( \frac{x}{\sqrt{2}} \right)
\]
For \( x > 0 \) we have:
\[
Q(x) \approx \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \sum_{k=1}^{5} b_k g^k; \quad \text{where}
\]
\[
g = 1/(1 + 0.2316419x),
\]
\[
b_1 = 0.3198, \quad b_2 = -0.3565, \quad b_3 = 1.7814,
\]
\[
b_4 = -1.8212, \quad b_5 = 1.3302
\]

Direct forms of IIR implementation:
\[
y(n) = b_0x(n) + b_1x(n-1) + \cdots + b_Mx(n-M) - a_1y(n-1) - \cdots - a_Ny(n-N)
\]
\[
H(z) = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{k=1}^N a_k z^{-k}} = \left(\sum_{i=0}^M b_i z^{-i}\right) \left(\frac{1}{1 + \sum_{k=1}^N a_k z^{-k}}\right) = H_1(z)H_2(z) = H_2(z)H_1(z).
\]
Direct Form – I

\[ x(n) \rightarrow b_0 \rightarrow b_1 \rightarrow \cdots \rightarrow b_M \rightarrow 1 \rightarrow a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_M \rightarrow y(n) \]

\[ H_1(z) \text{ [The Zeros]} \]

\[ H_2(z) \text{ [The Poles]} \]

Direct Form – II

\[ x(n) \rightarrow 1 \rightarrow a_1 \rightarrow b_0 \rightarrow b_1 \rightarrow \cdots \rightarrow b_M \rightarrow a_2 \rightarrow \cdots \rightarrow a_M \rightarrow y(n) \]

\[ \text{Poles} \]

\[ \text{Zeros} \]
Appendix D: DSP Lab Experiments

Experiment # 1: Computing the Convolution Integral

Introduction

The convolution integral of two functions \( x(t) \) and \( h(t) \) is one of the most significant topics in signal processing. This is so because the output \( y(t) \) of any linear time-invariant system is given by the convolution integral of the input signal \( x(t) \) and the system impulse response \( h(t) \) as follows:

\[ y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\lambda)h(t - \lambda)d\lambda = h(t) * x(t) = \int_{-\infty}^{\infty} h(\lambda)x(t - \lambda)d\lambda. \]

Steps of finding the convolution can be found in the tutorials. In this experiment we discuss the numerical implementation of this integral.

Due to the finite word-length and memory on the computer, we can only deal with finite length discrete-time signals. Analog signals and systems should be approximated by their discrete counterparts. This approximation should be mathematically well founded.

First, we should note that if \( x(t) \) and \( h(t) \) are finite-length functions defined over the intervals \( a_1 \leq t \leq b_1 \) and \( a_2 \leq t \leq b_2 \), respectively, then the interval of their convolution \( y(t) \) will be \( a_1 + a_2 \leq t \leq b_1 + b_2 \).

From Calculus we have:

\[ y(t) = \int_{-\infty}^{\infty} x(\lambda)h(t - \lambda)d\lambda = \lim_{\Delta t \to 0} \sum_{k=-\infty}^{\infty} x(k\Delta t)h(t - k\Delta t)\Delta \lambda. \]

Hence, we can approximate the above continuous-time convolution as follows:

\[ y(t) \approx \Delta \lambda \left[ \sum_{k=-\infty}^{\infty} x(k\Delta \lambda)h(t - k\Delta \lambda) \right]. \]
The variable \( t \) is considered as a constant in the above integration and summation. It has the characteristics of the dummy variable \( \lambda \), hence, it can be approximated as \( t \approx n\Delta\lambda \), for some integer \( n \). The above equation will be better written as follows:

\[
y(n) \approx \Delta\lambda \left[ \sum_{k=-\infty}^{\infty} x(k)h(n-k) \right],
\]

where \( \Delta\lambda \) has been removed from expressions of the form \( k\Delta\lambda \) for easier notation and suitability for implementation on the digital computer.

MATLAB\textsuperscript{®} is currently the best tool for mathematical modeling. In addition to the basic mathematical features, it has some dedicated toolboxes like the signal processing toolbox, the communications toolbox, and the control toolbox. Also, symbolic solutions are possible through the MATLAB symbolic toolbox. In this book we adopt MATLAB for DSP simulations.

Straightforward calculation of the above discrete convolution using for-next loops is time-consuming and not suitable for DSP purposes, especially for real-time processing of long signals (like speech signals). As we will see later in this course, this problem is closely related to that of calculating the Fourier transform of a signal. Scientists have discovered efficient algorithms for calculating the above summation by making use of symmetric terms in the discrete Fourier transform. This algorithm is called the Fast Fourier Transform (FFT), which is available on MATLAB as fft and ifft (for inverse Fourier Transform).

The above discrete convolution can be implemented on MATLAB as follows:

\[
y = Ts \ast \text{conv}(x,h);
\]

where \( y \) is the output vector, \( Ts \) is the time step (equivalent to \( \Delta\lambda \)), \( x \) and \( h \) are two vectors representing the two signals to be convolved. Note that * on MATLAB means scalar or matrix multiplication, while .* means vector multiplication (i.e., multiplication of corresponding elements in the two vectors).

The accuracy of computation would be dependent on \( Ts \). However, the less \( Ts \) the more computation time. Therefore, there is a compromise between speed and accuracy.

Using \( y = Ts \ast \text{conv}(x,h) \) gives a function whose length is larger than both \( x \) and \( h \). In many applications we need to fit all signals using the same word length. In this case, we can use the following fft-based algorithm:

\[
y = Ts \ast \text{ifft}(\text{fft}(x) \ast \text{fft}(h)); \quad \% \text{If we do not allow negative time},
\]

or

\[
y = Ts \ast \text{fftshift}(\text{ifft}(\text{fft}(x) \ast \text{fft}(h)));
\]

\% If we allow negative time.
MATLAB Simulation

Task 1

Write a MATLAB code to find the convolution of the two continuous-time finite-length signals \( x(t) \) and \( h(t) \) as shown in Fig. D.1.

- Open a new m-file and give it a name (e.g., DSP_Exp1.m) and save it in your directory.
- Choose a unified interval for \( x \) and \( h \) (i.e., \( a_1 = a_2 = -10, b_1 = b_2 = 10 \)).
- Take \( T_s = 0.01 \) and define the time vector \( t \).
- Express \( x \) and \( h \) first in terms of the unit step function, which is denoted in MATLAB as \( \text{stepfun}(t,t_0) \). On MATLAB command line, type \( \text{help stepfun} \) to see how this function works. Plot all step functions associated with the functions \( x \) and \( h \), then plot \( x, h \), and their convolution \( y \) on the same graph.
- Plot the theoretical convolution (see the Note below) and the numerical result on another graph.
- Define all axes using xlabel and ylabel commands.
- Limit the axes lengths using \( \text{axis([x_min x_max y_min y_max])} \) command.
- Define all functions using \( \text{text(a,b,'text')} \) command to write on figures.
- Find the convolution using the fft-based algorithm and compare with the first method.

**Note:** Using Fig. D.1 we have:

\[
y(t) = h(t) \ast x(t) = \int_{-\infty}^{\infty} h(\lambda)x(t-\lambda)d\lambda = \int_{0}^{3} x(t-\lambda)d\lambda = \int_{0}^{0} 3e^{-2(t-\lambda)}u(t-\lambda)d\lambda
\]

\[
v=t-\lambda \rightarrow = - \int_{t-3}^{t} 3e^{-2v}u(v)dv = \int_{t-3}^{t} 3e^{-2v}u(v)dv
\]

\[
= \begin{cases} 
0, & t < 0 \text{[Since } u(v) = 0 \text{ for } v < 0] \\
3 \int_{0}^{t} e^{-2v}dv = (3/2)(1 - e^{-2t}) & t \geq 0 \text{ and } 3 < 0 \rightarrow 0 \leq t < 3 \\
3 \int_{t-3}^{t} e^{-2v}dv \approx (3/2)e^{-2t+6} & t \geq 0 \text{ and } t \geq 0 \rightarrow t \geq 3.
\end{cases}
\]
Task 2

Write a code to invert the function $x(t)$ around the y-axis to get $x(-t)$. Then a code to find the shifted versions $x(t + t_o)$ and $x(-t + t_o)$. Take $t_o = 3$ and $-3$. Plot all functions on the same graph. Comment on these operations.

Task 3

The Dirac delta function $\delta(t)$ is an important tool in signal processing. Roughly speaking, it is equivalent to a pulse of infinite height and very narrow width (approaching zero). It is defined by the following integral:

$$\int_{-\infty}^{\infty} g(t)\delta(t - t_o)dt = g(t_o)$$

where $g(t)$ is a continuous function, $t_o$ is a constant. The delta function has the following properties:

P1: $\int_{-\infty}^{\infty} \delta(t)dt = 1$ (unit area),

P2: $\delta(t) = \delta(-t)$ (even),

P3: $x(t) \ast \delta(t) = x(t)$, or, generally, $x(t) \ast \delta(t - t_o) = x(t - t_o)$, where $t_o$ is a constant.

The Dirac delta function can also be defined as the limit of several even functions that can satisfy the above properties in the limit. These definitions include:

1. Limit of the weighted rectangular pulse (box), $\Pi_{2a}(t)$ (see Fig. D.2, left):

   $$\delta(t) = \lim_{a \to 0} \frac{1}{2a} \Pi_{2a}(t) = \lim_{a \to 0} \frac{1}{2a} \begin{cases} 1, & |t| \leq a \\ 0, & |t| > 0 \end{cases}$$

2. Limit of the weighted absolutely-decaying exponential:

---

**Fig. D.1** A plot of signals $h(t) = \Pi_3(t - 1.5)$ and $x(t) = 3e^{-2t} u(t)$

---
3. Limit of the weighted triangular pulse $\Lambda_{2a}(t)$ (see Fig. D.2, right):

$$\delta(t) = \lim_{a \to 0} \frac{1}{2a} e^{-|t|/a}$$

The above definitions are of practical importance since they can be used for approximating the delta function.

On MATLAB, use `stepfun` to define a weighted rectangular pulse that approximates $\delta(t)$ and call it $d$. Choose $a = 11 \times Ts$. Replace $h$ in the above code with $d$. To see whether this approximation is successful, convolve $d$ with $x$. According to P3 above, you should obtain $y \approx x$. Plot your results and comment.

Task 4

In Task 3 we chose the pulse width as $a = 11 \times Ts$. This choice will decide the accuracy of approximating the delta function. To see the effect of this choice, consider different widths for the rectangular pulse (hence, different approximations for the delta function). Let the width parameter $a = r \times Ts$, where $r$ is an integer ranging from 1 to $R$. Take $R = 100$. For each value of $r$ (hence, for each approximation of the delta function), find the mean squared error (MSE), which is defined as follows:

$$e(r) = \frac{1}{N} \sum_{n=1}^{N} [y_r(n) - x(n)]^2$$

where $N$ is the total number of samples in each signal. Use the function `mean` in MATLAB (type `help mean`) to find the above summation. If you use $y = \text{conv}(d, x)$, the interval of $y$ is larger than that of $x$, you should extend $x$ range (by zero padding) to be able to find the above summation. Plot MSE versus $a = r \times Ts$ and comment on the shape of the resulting curve.
Task 5

In the above code, use the other two approximations for the delta function and repeat the procedure in Task 4. Plot the three MSE curves on one figure and compare your results. Decide which approximation is better for practical implementation of the delta function. Take $T_S = 0.005$, repeat the same steps, and see whether the MSE is reduced.

Experiment # 2: Generation and Demodulation of AM Signals

Introduction

Information-bearing signals are normally of low-frequency nature. For example, the frequency content of the human speech signal ranges from 200 to 4 kHz, while audio signals (in general) can reach up to 20 kHz. For long-range transmission of signals, we should convert them to radio waves and transmit them through antennas. The antenna dimensions [length or diameter (for dish antenna)] should be proportional to $\lambda/4$ or $\lambda/2$, $\lambda$ being the wavelength, where $c = \lambda f = 3 \times 10^8$ m/s. Hence, for baseband radio transmission of a speech signal through, we need an antenna of length $c/(4 \times 4000) \approx 18.7$ km. Therefore, we should increase the frequency of the signal (without changing the information) before radio transmission. This is called modulation. In fact, high frequency is also more resistive to noise in space. Mobile phones operate near 900 MHz, UHF TV operates in the range 300–800 MHz, and satellite links operate in the frequency range from few hundred MHz to 40 GHz or more.

Amplitude modulation (AM) is widely used in communication systems. The standard AM modulator multiplies the message $m(t)$ by a scaling factor $\beta$ (called the modulation index) and a high-frequency sinusoid called the carrier, $c(t) = A_c \cos(\omega_c t)$, and then add the result to the carrier (for synchronization purposes) to get the standard AM signal

$$x(t) = A_c [1 + \beta m(t)] \cos(\omega_c t) \quad (1)$$

The magnitude spectrum of this signal is given by:

$$X(f) = A_c \left[ \frac{1}{2} \left\{ \delta(f - f_c) + \delta(f + f_c) \right\} + \frac{\beta}{2} \left\{ M(f - f_c) + M(f + f_c) \right\} \right] \quad (2)$$

For demodulation of the AM signal at the receiver, an envelope detector is usually used. However, if the AM signal is weak and the modulation index is small, it is better to use a square-law device followed by a low-pass filter as shown in Fig. D.3 below, where $r(t)$ is the received AM signal and $s(t)$ is the demodulated signal.
Under noise-free conditions, we have $r(t) = x(t)$. Hence, the squarer output will be as follows:

$$y(t) = x^2(t) = A_c^2[1 + \beta m(t)]^2[\cos(\omega_c t)]^2$$

$$= A_c^2[1 + 2\beta m(t) + \beta^2 m^2(t)] \left[ \frac{1}{2} + \frac{1}{2} \cos(2\omega_c t) \right].$$

The LPF will remove the high frequency components. Hence, the output signal will be given by

$$s(t) = 0.5A_c^2 + A_c^2 \beta m(t) + 0.5A_c^2 \beta^2 m^2(t),$$

which contains a d.c. term, the original message (scaled by $A_c^2 \beta$), and an error term $0.5A_c^2 \beta^2 m^2(t)$, which we cannot get rid of by filtering. The relative error is given by:

$$e = \frac{0.5A_c^2 \beta^2 m^2(t)}{A_c^2 \beta m(t)} = \frac{\beta m(t)}{2}$$ (3)

To reduce error, we should have $|\beta m(t)/2| < 1 \; \forall t$. Hence, this method is efficient only if the message is weak and the modulation index is small.

**MATLAB Simulation**

In this experiment we will simulate the generation and demodulation of standard AM signals as explained above.

**Task 1**

Write a MATLAB code to generate a standard AM signal. First, generate a sinusoid with frequency $f_o = 2$ Hz and a carrier with frequency 10 Hz. Take all amplitudes to be 1. Select a value for the modulation index and generate the modulated signal that would be transmitted. Plot all time signals and their spectra.

**Task 2**

Write a code to simulate the function of the receiver. First comes the square-law device, followed by the LPF. Use an ideal LPF with a carefully chosen cutoff.
frequency $f_1$ (Take $f_1 = 3f_0$). Filter the squared received signal, then plot the demodulated signal in the time and frequency domains. Compare with the original signal. Change the modulation index and comment on the results.

Task 3

Repeat Tasks 1 and 2 above for the rectangular pulse $\Pi_2(t-2)$, using the same carrier.

**Experiment # 3: Random Signal Analysis**

**Introduction**

A random process is a function of two variables: an event and time. A realization of the random process is called a “sample function”. All sample functions constitute an “ensemble”. At a specific time instant $t = t_o$, the values of sample functions are represented by a “random variable”. For example, noise $n(t)$ is a random process.

If we have a signal $x(t) = \cos(\omega_o t)$ and this signal is corrupted by noise $n(t)$, then the result would be the random signal $y(t) = \cos(\omega_o t) + n(t)$.

To study the statistical properties of noise and noisy signals, we may repeat the realization of the random signal $y(t)$ many times. If we have three realizations (repetitions) of the noise process $n(t)$ with a given noise power, we get the sample functions or “Realizations” as shown in Fig. D.4.

The “ensemble average” at $t = t_o$ is given by:

$$n_{av} = \frac{1}{3} \left[ n_1(t_o) + n_2(t_o) + n_3(t_o) \right]$$

If we have a large number of realizations and the process is stationary, then we may have:

$$\text{Ensemble average } (m) = \text{Time average } (m_t)$$

In this case we call the process “ergodic”. For ergodic processes we can calculate the ensemble mean using the time mean, which is much easier and does not require more than one realization. On MATLAB, this is obtained by using the instruction “mean”. Note that the ensemble mean for the above signal at the time instant $t$ is given by:

$$m = \mathcal{E}\{n(t)\} = \int_{-\infty}^{\infty} np(n)dn$$

where $p(n)$ is the probability density function (pdf) of noise, and $\mathcal{E}\{\cdot\}$ is the statistical expectation.
MATLAB Simulation

In this experiment we will study the statistical properties of a sinusoidal signal imbedded in AWGN (additive white Gaussian noise) at different SNRs (signal-to-noise-ratios).

Task 1

Write a MATLAB code to generate \( M = 10 \) realizations of a sinusoidal signal \( x(t) = a \cos(\omega_0 t) \) corrupted by AWGN process \( n(t) \) to give the noisy signal \( y(t) = a \cos(\omega_0 t) + n(t) \). Take \( f_o = 0.2 \) Hz, \( a = 1 \), and SNR = 1 dB. Show that the ensemble mean of noise is approximately zero, and the ensemble mean of the noisy signal is the deterministic signal \( s(t) \). This approximation is improved if we take a larger number of realizations \( M \) (e.g., 50, 100). Plot the time signals and their spectra for the first realization; also plot the ensemble means. Repeat the above steps for different SNR values (e.g., -5, -1, 0, 1, 3, 10).

Task 2

For each realization in Task 1, find the pdf of noise \( p_n(n) \) and take the average of all realizations. Compare this pdf with the theoretical pdf given by:

\[
p_n(n) = \frac{1}{\sqrt{2\pi\sigma}} e^{-n^2/2\sigma^2}
\]

Show practically that

\[
\int_{-\infty}^{\infty} p_n(n) dn \approx 1
\]
Compare this result with the theoretical result, which is 1 exactly. Find the mean and variance of one realization and compare with the theoretical values. As in Task 1, consider different SNRs.

Task 3

Find the autocorrelation function of the signals \( x(t) \), \( n(t) \), and \( y(t) \). Find the cross-correlation \( R_{xn}(\tau) \) between the signal \( x(t) \) and the noise process \( n(t) \).

**Experiment # 4: Filter Design with Application to Noise Reduction**

**Introduction**

MATLAB provides built-in codes for designing analog and digital filters. In this experiment we will consider Butterworth and Chebychev-I filters. You can type on the command line `>> help butter` and `>> help cheby1` to know the design parameters of these filters. As an application of filtering, we will design a filter for the purpose of reducing noise that corrupts a narrowband signal of known frequency (if the signal frequency is unknown, then this approach fails and we need an adaptive filter for noise reduction).

White Gaussian noise \( n(t) \) is a broadband signal since it is uncorrelated (i.e. its autocorrelation function is a weighted delta function, hence its power spectral density is constant for all frequencies as shown in Fig. D.5.

Normally, we are interested in a specific frequency band \((-B < f < B)\) for practical applications. For example, in speech signal processing, the important frequency band is about 4 kHz, and the whole audible spectrum is less than 20 kHz. Since noise power is given by

\[
p_n = \int_{-\infty}^{\infty} G_n(f) df,
\]

![Fig. D.5 Spectrum of a time delta function](image)
then we can reduce this power by applying a filter (LPF or BPF) to extract only the area of interest in the spectrum.

**MATLAB Simulation**

In this experiment we will design filters for the purpose of noise reduction in narrowband signals.

**Task 1**

Consider a signal $x(t)$ corrupted by additive Gaussian noise $n(t)$, where $x(t) = a \sin (\omega_0 t)$, with $a = 1, f_0 = 1 \text{ Hz}$, $0 \leq t \leq 10 \text{ s}$, and SNR $= 2 \text{ dB}$. Plot the time signals $x(t)$ and $s(t) = x(t) + n(t)$ with their spectra. Design a LPF (Butterworth and Chebychev 3 dB ripple) of minimum possible order $K$ such that the attenuation is less than $-80 \text{ dB}$ for $f > 10 \text{ Hz}$. You should carefully specify the appropriate cut-off frequency $f_c$. Plot the filter transfer function given by

$$H(f)_{\text{dB}} = 20 \log_{10} |H(f)|,$$

and vary $K$ until you reach the proper order. Then filter $s(t)$ and plot the time output signal $y(t)$ with its spectrum and compare with $s(t)$.

**Task 2**

In this task we study the effect of the sampling frequency on a recorded signal, and we conclude that if we want to process a recorded signal, we should know its sampling frequency. Listen to any test audio signal, e.g., Ohno.wav (can be downloaded from http://free-loops.com/download-free-loop-3022.html), using a media player (just double-click on the file icon). On MATLAB use $x = \text{wavread}('\text{Ohno}')$ to convert the audio file to a vector of numerical values (voltages) suitable for mathematical processing. The sampling frequency of Ohno can be read from the media player; it is $f_s = 22 \text{ kHz}$. Try changing $f_s$ to 35 kHz then re-write the signal on another file using the statement

$$\text{Ohno1} = \text{wavwrite}(x,f_s, '\text{Ohno1.wav}');$$

Now listen to the new file. Change the sampling frequency to 15 kHz and repeat the process.

**Task 3**

In this task we consider audio effects as explained in the lecture notes. Implement an FIR filter with 4 coefficients to simulate echoes, and then listen to the resulting audio signal. Change the magnitude of the coefficients and listen again.
Task 4

Now listen to a single-tone test audio signal saved as “stone” (e.g., 440 Hz sound from [http://www.mediacollege.com/audio/tone/download/](http://www.mediacollege.com/audio/tone/download/)). Find its sampling frequency $f_s$ (it is 44.1k Hz). Read the signal as $x$. Add Gaussian noise $n$ (of power $= -20$ dB) to the signal and write the result as $s = x + n$ in your directory as an audio signal using `wavwrite(s,fs,‘stonen.wav’). Listen to the corrupted signal and compare with the original one. Plot the time signals $x$ and $s$ with their spectra versus the normalized frequency ranges $f_n = f/f_s$ and $f_N = f/(f_s/2)$. Now design a digital filter (LPF Butterworth) of order 10 and normalized cut-off frequency $w_{cr}=wc/(f_s/2)$, where $w_{cr}$ is the cut-off frequency which should be chosen carefully to reduce noise. Plot the frequency response of the digital filter using `freqz`. Then filter the discrete signal $s$ to get the signal $y$. Plot $y$ and its spectrum $Y$, then compare with the original and noisy signals.

**Experiment # 5: A Sinusoidal Digital Oscillator**

**Introduction**

A digital oscillator is a system that generates an output waveform (like a sinusoid) without a need for an input signal, except for a D.C. supply and perhaps a trigger (like a delta function) at the starting time. The theory of operation is based on the fact that a digital system with poles on the circumference of the unit circle in the $z$-plane is neither stable nor divergent, but oscillatory (or, *marginally stable*).

To design a digital sinusoidal oscillator, we need a transfer function in the $z$-domain whose impulse response is a sinusoid. Using Tables we can reach at the following $z$-transform pair:

$$h(n) = \sin[(n + 1)b] \xrightarrow{z} H(z) = \frac{\sin(b)z^2}{z^2 - 2\cos(b)z + 1} \quad (1)$$

Since $n$ represents the time count, $b$ would represent the normalized radian frequency $\Omega_o = \omega_o T_s$, hence the frequency of oscillation is $f_o = \omega_o / 2\pi = (b/T_s)/2\pi = (b/2\pi)f_s$ Hz, and we should have $|b| < \pi$.

Note that the two poles of this system are the roots of $z^2 - 2\cos(b)z + 1 = 0$, which are given by:

$$p_{1,2} = \cos(b) \pm \sqrt{\cos^2(b) - 1} = \cos(b) \pm jsin(b) = e^{\pm jb} \quad (2)$$

Hence, the poles are *exactly on the circumference* of the unit circle, and the system is neither stable nor unstable (oscillatory). Figure D.6 shows the implementation.
diagram of this system. The pole-zero diagram and frequency response are shown in Fig. D.7.

**MATLAB Simulation**

In this experiment we will design a second-order IIR filter to perform as a sinusoidal oscillator.

**Task 1**

The generated sinusoid has an amplitude $A = 1$ and frequency $f_o = 1$ Hz. Choose the sampling frequency of the system as $f_s = 100$ Hz ($f_s \gg f_o$). Write the transfer function of the system as in Eq. 1 above, from which find the frequency response of the system and plot it.

**Task 2**

Now analyze the system using `zplane` function on MATLAB. Find the poles of the system using `roots`. Find the angles of these poles, from which find the frequency of the oscillator.
Task 3

Define the system true time and plot the theoretical impulse response as in Eq. 1.

Task 4

Analyze the system using tf, impulse, and freqz. Plot all results.

Task 5

Now simulate the oscillator circuit as shown in Fig. D.6. Enter a delta signal and find the output. Plot and compare with previous methods of finding the impulse response.

Task 6

Find the effect of changing the sampling frequency on the system. Reduce the sampling frequency to 10 Hz and compare.

Task 7

With $f_s = 100$ Hz and $A = 1$, try to generate 2, 10, 45, 60, and 70 Hz sinusoids. See what happen if $f_o > f_s/2$. Change the amplitude and plot the resulting output signals.

**Experiment # 6: Sampling and Reconstruction**

**Introduction**

To process analog (i.e., continuous-time) signals using digital technology, we should convert these signals into digital signals through sampling and A/D conversion. In most applications, sampling is uniform, i.e., the sampling interval $T_s$ and the sampling frequency, $f_s = 1/T_s$, are constant. Ideal sampling can be formulated as multiplication of the analog signal $x(t)$ by the a train of time impulses $p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$ as shown in Fig. D.8. The spectrum of this time impulse train is the frequency impulse train $P(f) = f_s \sum_{k=-\infty}^{\infty} \delta(f - kf_s)$. The above multiplication in the time domain would be a convolution in the frequency domain between the frequency impulse train and the spectrum of the analog signal, $X(f)$, which results in the scaling (by $f_s$) and repetition (every $f_s$) of this spectrum as shown in Fig. D.9, where we used the normalized frequency $\nu = f/2f_s$. The new spectrum is called the discrete-time Fourier transform (DTFT). Reconstruction of the sampled signal (back to the analog signal) is possible
through an analog low-pass filter on the condition that the sampling frequency is more than twice the signal bandwidth $B$, i.e.,

$$B \leq \frac{f_s}{2}.$$ 

The sampling rate $f_s = 2B$ is called the **Nyquist rate**, which is the minimum rate for safe sampling. If $B > f_s/2$, replicas of the signal spectrum $X(f)$ that constitute the DTFT will overlap, and part of the information is damaged. This overlap is called frequency aliasing. Hence, in practice, we normally bandlimit the signal before sampling to remove the unimportant frequency content that may cause aliasing. This can be achieved using a LPF (which is called anti-aliasing filter). If the sampling frequency is near the Nyquist rate, the anti-aliasing filter should have a sharp cutoff at $f = B$, otherwise aliasing may occur. If $f_s \gg 2B$, we can relax this condition on the anti-aliasing filter. This is important in hardware implementation to reduce the LPF complexity.
MATLAB Simulations

Task 1

Write a MATLAB code to simulate the sampling process of an analog signal. Since analog signals are approximated on the computer, you should distinguish between the simulation sampling period (frequency) and the actual sampling period (frequency). Take the system sampling period to be $T = 1\times10^{-4}$ s. With a time limit of 10 s, generate the global time and frequency vectors. Sinusoidal and linear FM signals are important in applications. Simulate a sinusoid of frequency 20 Hz (hence, to avoid aliasing, the sampling frequency should be more than 40 Hz). Plot the time signal and its spectrum.

Task 2

Use the MATLAB function `square` to simulate the time impulse train. Its duty cycle (the “ON”/“OFF” duration ratio) should be very small to simulate impulses. Its frequency would be the actual sampling frequency. Plot it in the time and the frequency domains and verify the theoretical formulas stated above.

Task 3

Now multiply the analog signal by the sampling signal (the impulse train). Plot the resulting spectrum for a sampling frequency of 25, 100, and 500 Hz and check for aliasing.
Task 4

Design an analog 9th-order Butterworth LPF with cutoff frequency of 50 Hz and filter the sampled signal. Plot the output signal and its spectrum and compare with the original analog signal for sampling frequency of 50, 100, and 500 Hz and check for aliasing.

Task 5

Now generate a LFM signal of initial frequency 10 Hz and modulation index of 0.7. Note that the sinusoid is a periodic function (hence, its spectrum is impulses) while the LFM is non-periodic (hence, its spectrum is a continuous function of frequency). From the spectrum we see that the maximum frequency in the LFM is about 50 Hz, hence, we expect the sampling frequency to be at least 100 Hz to avoid aliasing. Try $f_S = 50, 100, \text{ and } 500 \text{ Hz}$ then compare the results.

Task 6

In this task we study the audio effect of aliasing. Download and listen to an audio signal such as Ohno.wav which was used before in Experiment 4. On MATLAB use $[x, f_s, \text{bits}] = \text{wavread}('\text{Ohno}.' )$ to convert the audio file Ohno.wav to a vector of numerical values (and know its original sampling frequency and number of quantization bits). The sampling frequency of Ohno can be read $f_s = 22$ kHz. Try downsampling Ohno to $f_k = f_s / k$ (k integer) then re-write the signal on another file using the statement $\text{wavwrite}(x, f_s, '\text{Ohno}.wav')$. Now listen to Ohno1.wav. At what value of $k$ does aliasing start to occur? Why?

**Experiment #7: Binary Signal Transmission**

**Introduction**

In binary communication systems, binary data (which is a sequence of 0’s and 1’s) are often transmitted through a channel using two orthogonal signals, $s_0(t)$ and $s_1(t)$. One possible signal configuration is shown in Fig. D.10.

If data is large, then 0’s and 1’s are equally probable ($p(1) = p(0) = 1/2$) and statistically independent. The AWGN channel adds Gaussian noise (wideband, approximately white) with power spectral density (PSD) = $\eta/2$ (W/Hz). Hence, the received signal will be in the following form:

$$r(t) = s_i(t) + n(t), i \in \{0, 1\}, 0 \leq t \leq T.$$
Assuming that \( s_0 \) and \( s_1 \) are as shown in Fig. D.10, and that \( s_0 \) was transmitted, the outputs of the two matched filters at the time instant \( t = T \) are as follows:

\[
\begin{align*}
  r_0 &= \int_0^T r(t)s_0(t)dt = \int_0^T [s_0(t) + n(t)]s_0(t)dt \\
       &= \int_0^T s_0^2(t)dt + \int_0^T n(t)s_0(t)dt = E + n_0 \\
  r_1 &= \int_0^T r(t)s_1(t)dt = \int_0^T [s_0(t) + n(t)]s_1(t)dt \\
       &= \int_0^T s_0(t)s_1(t)dt + \int_0^T n(t)s_1(t)dt = 0 + \int_0^T n(t)s_1(t)dt = n_1
\end{align*}
\]

Both \( n_0 \) and \( n_1 \) are Gaussian and have zero means. The variances of \( n_0 \) and \( n_1 \), \( \sigma_i^2 (i \in \{1, 2\}) \), are given by:

\[
\sigma_i^2 = \frac{\eta}{2} E,
\]

where \( E \) is the energy of the signals \( s_0 \) and \( s_1 \) [Note that \( s_1^2 (t) = s_0^2 (t) \)].

Now if \( s_1 (t) \) was transmitted, then \( r_0 = n_0, \ r_1 = E + n_1 \) with same statistics as above.

**Probability of Error**

The matched filter compares \( r_1 \) and \( r_0 \). It will decide that a “0” was transmitted if \( r_0 > r_1 \), and that “1” was transmitted if \( r_0 < r_1 \). If “0” was transmitted, then error will occur only if \( r_1 > r_0 \).

It can be shown that the above probability of error can be expressed as follows:

![Fig. D.10](image-url)
where the error function \( \text{erf}(y) = \frac{2}{\sqrt{\pi}} \int_0^y e^{-u^2} du \) is a reference function in MATLAB, and SNR is the signal-to-noise ratio defined by:

\[
\text{SNR} = \frac{E}{\eta},
\]

and is normally given in dB as \( \text{SNR}_{\text{dB}} \), where \( \text{SNR}_{\text{dB}} = 10 \log_{10}(\text{SNR}) \).

The probability of error \( P_e \) is the basis for performance evaluation of communication systems. The same probability of error \( P_e \) is obtained if “1” was transmitted. Hence, the average probability of error is given by:

\[
P_{e,\text{av}} = P_e.
\]

Figure D.11 shows the general shape of \( P_e \) against SNR.

**Binary Transmission Using Antipodal Signals**

Two signals \( s_0(t) \) and \( s_1(t) \) are said to be antipodal if \( s_0(t) = -s_1(t) \). One possible configuration is to use two voltage levels, \( \pm V \). Figure D.12 shows another configuration.

Using orthogonal signals, we need a bank of two matched filters for optimum reception. However, if we use two antipodal signals, we need only one matched filter. The received signal is \( r(t) = \pm s(t) + n(t) \). Following the same analysis as for orthogonal signals, we find the following results:

- **Output of the matched filter** \( z = \pm E + n_a \) (where \( n_a = \int_0^T n(t)s(t)dt \)),
- **Variance of** \( n_a = \frac{E}{2} \) (same as that of \( n_1 \) and \( n_0 \) for orthogonal transmission).
- **Probability of error** \( P_e = \frac{1}{2} - \frac{1}{2} \text{erf}(\sqrt{\text{SNR}}) \). The decision will be as follows: if \( z > 0 \), then \( s(t) \) was transmitted (which may represent “1”), otherwise \(-s(t)\) was transmitted.

Hence, \( P_e \) using antipodal signals is less than \( P_e \) using orthogonal signals for the same SNR. Therefore, if no modulation is used later in the system, antipodal signal transmission is more efficient for baseband binary communications.

**MATLAB Simulation**

In this experiment we will consider simulation of baseband binary signal generation and transmission, as well as performance analysis using the error
function. This is important in many applications like communications between computers operating in the same neighborhood.

**Task 1**

Simulate a binary communication system with two orthogonal signals. Use the flowchart shown in Fig. D.13. Take $E = 1$ and generate $N = 1000$ bits data sequence $\{d(n)\}$. To generate a sequence of equiprobable 1’s and 0’s, use “rand” function on MATLAB, which gives you a random number uniformly distributed over the interval $(0,1)$. Use the simple rule: if rand $>0.5$, then $d(n)$ is “1”, otherwise “0”. However, you should use a smart algorithm for data generation that avoids the use of loops. Mix data with noise to simulate transmission and matched filter reception, then find the probability of error as a function of the SNR. Better to use a separate function for the probability of error calculation. Compare with the theoretical curve.

**Fig. D.11** Probability of error versus SNR in baseband binary communications

**Fig. D.12** Two antipodal signals
**Task 2**

Using the same approach (but one matched filter), simulate a binary communication system with two antipodal signals. Find the probability of error versus SNR. Compare with the theoretical result and with orthogonal signal transmission.

**Task 3**

Simulate the above system without matched filters and compare with results in Tasks 1 and 2 above.

---

**Experiment # 8: Simulation of the Sinusoidal Digital Phase-Locked Loop**

**Introduction**

The sinusoidal DPLL (SDPLL) is an important system in signal processing and communications. Like other PLLs, the SDPLL is a feedback system that arranges its local frequency to be equal to the input frequency. It can be used for signal detection, frequency tracking, and synchronization. Unless the incoming frequency $\omega$ is equal to the center frequency $\omega_0$, the first-order SDPLL (which utilizes a multiplication constant $G_1$ only in its filter) has always a non-zero steady-state phase error, $\phi_{ss}$. The 2nd-order SDPLL [which utilizes a first-order digital filter $H(z) = G_1 + G_2/(1 - z^{-1})$] always locks on zero $\phi_{ss}$. A block diagram of the SDPLL is shown in Figure D.14 below. The sampler here operates as phase error detector (PED).
The input $x(t)$ is assumed to be a sinusoidal signal as follows:

$$x(t) = A \sin(\omega t + \theta_0) + n(t).$$

The phase difference equation for the 1st-order SDPLL is given by:

$$\phi(k+1) = \phi(k) - K_2 \sin[\phi(k)] + \Lambda_o$$

where $\Lambda_o = 2\pi(\omega - \omega_0)/\omega_0$ and $K_2 = \omega G_1 A$. If we define $K_1 = \omega_0 G_1 A$, and the frequency ratio $W = \omega_0/\omega_0$, then we have $K_2 = K_1 (\omega_0/\omega_0) = K_1/W$.

The locking range is determined by the conditions:

$$K_1 > 2\pi|1 - W|$$

and

$$K_1 < \sqrt{(4 + 4\pi^2)W^2 - 8\pi^2W + 4\pi^2}.$$  

It should be noted that extreme points in this range does not ensure locking for all values of the initial phase error. The steady-state phase error is given by:

$$\phi_{ss} = \sin^{-1}(\Lambda_o/K_2).$$
MATLAB Simulations

Task 1

Write a MATLAB code to simulate the locking process of the 1st-order SDPLL. First, plot the locking range. Then, let $A = 1$, $\omega_o = 2\pi \text{ (rad/s)}$, $W = 0.9$, $K_1 = 0.8$ and take $\theta_o = \phi (0) = 0 \text{ rad}$. Hence, the incoming frequency is $f = f_o / W = 1/0.9 = 1.1 \text{ Hz}$. Make sure that the loop is inside the lock range. As we expect locking normally in less than 50 cycles, consider only 50 samples. Plot the input signal and the sampled signal. Also plot the phase $\phi(k)$ and the instantaneous frequency $[\omega(k) = 2\pi / T(k)]$ as functions of time. Check with the theoretical value of $\phi_{ss}$. Vary the initial phase $\phi (0) = \theta_o$ to take on the values $-3, -2, -1, 0, 1, 2, 3$ and see the difference in the locking process. Does the initial phase affect $\phi_{ss}$?

Task 2

Repeat Tasks 1 and 2 for various combinations of $(W,K_1)$ as follows: $(0.9,1.5)$, $(1.2,1.7)$, and $(1.4,3)$. Let $(W,K_1)$ be outside the locking range and plot the phase and frequency transients.

Task 3

Plot the phase plane diagram of the 1st-order SDPLL for Tasks 1 and 2.

Experiment # 9: Adaptive Wiener Filter for Noise Reduction and Channel Estimation

Introduction

Wiener filter is an optimum filter for estimation or prediction signals corrupted by noise or distorted by the transmission channel. Adaptive Wiener filter is a programmable filter whose coefficients [i.e., its impulse response non-zero values, $\{h(k)\}$] are changed (adapted) according to the current available samples of the observed signal $\{y(n)\}$ and a desired (reference) signal $\{d(n)\}$, to give an optimal estimate $\hat{x}(n)$ of the original signal $\{x(n)\}$ at the time instant $n$ [see Fig. D.15]. An adaptive filter utilizes a feedback algorithm to update the filter coefficients at each time instant $n$; hence, it can compensate for time-varying channel conditions.

The filter coefficients are adapted according to an algorithm, which can be implemented by hardware or simulated on a microprocessor or a computer. The adaptive FIR Wiener filter algorithm is a least mean-squared (LMS) error
algorithm, based on minimizing the MSE error $e_{mse} = \mathcal{E}\{[e(n)]^2\} = \mathcal{E}\{[d(n) - \hat{x}(n)]^2\}$ at every time instant $n$ as we have shown earlier.

Define:

$h(k) = [h_0(k) \ h_1(k) \ h_2(k) \cdots h_M(k)]$, the filter coefficients at the $k$th instant.
$y(k) = [y(k) \ y(k-1) \ y(k-2) \cdots y(k-M)]$, observed signal vector at $k$th instant.

The algorithm can be described in vector form (MATLAB-like code) as follows:

$h(0) = 0$; % Initialize the filter coefficients.
for $n = 1 : N$ \% $N =$ length(y);
$\hat{x}(n) = h(n-1)y^T(n)$; % Filter output (this is matrix multiplication).
$e(n) = d(n) - \hat{x}(n)$;
$h(n) = h(n-1) + \mu * e(n)y(n)$; \% $\mu$ is the step-size.
end

The choice of $\mu$ will affect the estimation accuracy and the convergence speed of the algorithm. Small values of $\mu$ will give better accuracy but slower convergence. Large values will do the contrary. Very small or very large values for $\mu$ will cause significant errors. Hence, a compromise would be optimal.

Larger filter length $M + 1$ gives better estimation, but more delay.

**Application 1: Noise reduction in narrowband signals:** For estimation of narrowband signals (like single-tone sinusoids) with known frequency band, we can use a normal LPF for noise reduction, but for unknown frequency band, we use adaptive Wiener filter with $d(n) = y(n)$ and input sequence \{\(y(n-1)\)}, as shown in Fig. D.16.

**Application 2: Channel estimation:** In mobile communications, a “training sequence” is sent before transmission of data. The receiver knows this signal and utilizes a copy of it as the desired signal $d(n)$. The adaptive Wiener filter can arrange its optimal coefficients during the short period of transmitting the training sequence before actual data are transmitted. Hardware or software implementation of the above algorithm is possible.
MATLAB Simulation

Task 1

On MATLAB simulate the signal \( y(k) = x(k) + n(k) \), where \( x(k) = \sin (\omega_0 k T_s) \), \( T_s = 0.01 \), the time vector is \( 0 < t < 10 \) (s), \( f_o = 2 \) Hz, \( n(t) \) is Gaussian noise, \( SNR = 2 \) dB. Assume that you know \( f_o \) and try to reduce noise using a LPF (choose the cutoff frequency carefully).

Task 2

Now assume you don’t know the signal frequency. Implement the adaptive Wiener filter for the purpose of noise reduction as shown in Fig. 2. Choose the number of taps \( M + 1 = 101 \), \( \mu = 0.001 \) and plot all signals. Vary \( M + 1 \) to 5, 10 and compare.

Task 3

With \( M + 1 = 101 \), vary \( \mu \) to take the values \( 1e^{-4}, 2e^{-4}, 5e^{-4}, 1e^{-3}, 2e^{-3}, 5e^{-3}, 1e^{-2}, 2e^{-2} \) and find the mean-squared error (MSE) as a function of \( \mu \). Find \( \mu_{\text{min}} \), the value of \( \mu \) that gives minimum MSE, and plot the corresponding signals and spectra.

Task 4

Assume that the communication channel causes ISI that spans 11 symbols, where the channel transfer function is

\[
H_c = [0.05 - 0.063088 - 0.126 - 0.259047, 0.250, 0.126, 0.038] .
\]

Assume channel noise of variance \(-40 \) dB. Generate a random sequence of length \( N = 20 \) binary symbols of 0’s and 1’s (take number of realizations \( R = 20 \) and \( R = 200 \)) and use them as the transmitted signal which will pass through the channel \( H_c \), also use them as a reference signal (the receiver knows these sequences). The channel estimated transfer function will be given by the inverse of the filter updated transfer function, \( \hat{H}_e(z) \). Plot \(|\hat{H}_e(f)|\) and \(|H_c(f)|\) and compare. Plot the mean-squared error signal (MSE) used by the filter versus the number of realizations. Compare this with the MSE in \( H_c \) estimation.
Task 5

Re-run Task 4 several times (for $R = 20$ and $200$) and check how this affects the MSE of the adaptive filter error signal. Change channel noise to $-20$ dB and check the MSE for a given $R$ and $N$.

Task 6

Take $R = 200$ and $N = 50, 500, 1000$ and compare the MSEs.

Task 7

Change the channel noise to $-20$ dB with $R = 200, N = 500$ and check the MSE. Re-run the code and check this result.

**Experiment # 10: Delta Modulation System**

**Introduction**

Delta modulation (DM) system is a single-bit differential PCM system that utilizes higher sampling rates to reduce the number of bits necessary for quantization, hence it reduces the cost of the system since increasing the sampling frequency is less expensive than increasing the word length. This makes DM suitable as ADC for audio applications, where the signal band is 20 kHz, hence the DM can use moderately high sampling frequencies.

DM system can be built in the analog domain or in the digital domain. However, it is not possible to implement an adaptive analog DM, hence we prefer the digital DM. Figure D.17 shows a digital DM system that consists of a 2-level quantizer and an integrator in the feedback loop. If the step $\Delta$ is small and the input signal has a steep slope, the DM can lose tracking and a slope overload occur. To avoid this problem, adaptive DM should be used. Figure D.18 shows an adaptive DM system.

![Fig. D.17 DM configuration](image-url)
MATLAB Simulation

In this experiment we will simulate DM and adaptive DM systems and compare their performance using sinusoidal and LFM input signals.

Task 1

On MATLAB generate the signal \( x(k) = A \sin(\omega_0 k T_s) \), with sampling period \( T_s = 0.02 \) s, amplitude \( A = 0.5 \) V, frequency \( f_0 = 1 \) Hz, and time vector \( 0 < t < 10 \) (s). With a step of \( \Delta = 0.07 \), simulate the delta modulation system in Fig. D.17. Plot the input signal, the estimated signal, and the output of the DM system along with their spectra. Can you find the effect of quantization on the output signal spectrum? The output signal is a square wave, however, its spectrum reveals a sinusoidal content.

Task 2

Implement the DM demodulator shown in Fig. D.17. Use a digital 4th-order Butterworth low-pass filter. Plot the demodulated signal and its spectrum and compare with the original signal.

Task 3

Vary the quantization step to 0.03, 0.1, and 0.2. Check the quantization noise and the slope overload.

Task 4

Repeat Tasks 1, 2, and 3 for a linear FM signal with amplitude 0.2 V, initial frequency 0.5 Hz, and modulation index of 0.5.

Task 5

Now implement the adaptive delta modulation system as shown in Fig. D.18, with an initial step of 0.03 and a step modification constant \( K = 1.3 \). For sinusoidal and LFM inputs, plot the input signal and its modulated version along with their spectra. Compare with the non-adaptive DM system.
DSP Project

Part 1: Problem Solving Using MATLAB

(A) Prime numbers

(A-1) A prime number (p) is an integer that has no factors or divisors except 1 and itself. The first 12 prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, and 37. Note that 2 is the only even prime number. Prime numbers play a significant role in coding and computation algorithms and circuits.

(A-2) Write a MATLAB code to find whether a given integer \( x \) is a prime.

(A-3) Write a MATLAB code to find the number of primes less than or equal to a given real number \( x \) (not necessarily an integer). This is called “prime counting function, \( \pi(x) \)”. For example, \( \pi(10) = \pi(10.9) = 4 \).

(A-4) Prime number theorem As \( x \) becomes large, \( \pi(x) \) approaches \( x/\ln(x) \). Prove using MATLAB simulation.

(B) Non-linear equations

Using MATLAB, find whether the equation \( \cos(x) = x^2 \) has a solution over the interval \([-3, +3]\).
(C) **3D plotting**

Plot the function \( z(x, y) = \cos(x) \exp(-|y|) \) and its first derivatives. Plot a few 2D cross-sections of \( z \).

(D) **Computation algorithms** are of fundamental importance in designing new generations of faster and more efficient computers and digital signal processors (DSP). Mersenne primes (MP’s) are important in computation algorithms and hardware as they enable binary arithmetic while using “Number Theoretic Transforms”. An integer \( m \) is a **Mersenne Prime** if it is a prime number of the form \( m = 2^p - 1 \), where \( p \) is a prime. There are **only 44** MP’s discovered so far. The first 9 MP’s are 3, 7, 31, 127, 8191, 131071, 524287, 2147483647, 2305843009213693951 (corresponding to \( p = 2, 3, 5, 7, 13, 17, 19, 31, 61 \)), while the last known MP is \( m_{44} = 232582657^2 - 1 \) (discovered in 2006 and is composed of 9808358 digits). Write a MATLAB code to find the first 5 MP’s.

**Part 2: System Implementation**

Choose Project-2a or Project-2b as follows.

**Project-2a: Hardware Design of Digital Systems**

I. Design a digital integrating or differentiating circuit. Using ADC/DAC, apply analog sinusoids (from a signal generator) with different frequencies and find the magnitude and phase spectra. Verify the function of the circuit on the oscilloscope.

II. Design a first-order digital phase-locked loop or a digital delta modulation system and verify the circuit operation using a sinusoidal signal generator and an oscilloscope.

**Project-2b: Software Analysis of the Sinusoidal Digital Phase-Locked Loop**

**A: Frequency Tracking of a Sinusoid**

**Task 1: Noise-Free Analysis**

Using MATLAB, simulate the 1st-order sinusoidal DPLL shown in Fig. D.19 Consider a sinusoidal input signal, plot the locking range of the loop, and study the loop behavior for different circuit parameters. Consider the phase plane, the sampling process and the transient phase and frequency. For a fixed frequency ratio \( W \) and initial phase (but different combinations), find the effect of the loop gain \( K_1 \) on the locking speed. Locking is practically reached when the difference
between successive phase errors is smaller than some positive number (e.g., 0.01). Also find the effect of the initial phase error on the locking speed assuming the frequency ratio and the loop gain are fixed.

Task 2: Noise Analysis

Consider a sinusoidal input signal corrupted by an AWGN noise. Find the pdf of the SDPLL output frequency for different SNRs. Discuss whether the SDPLL can estimate the input frequency in noisy environments. Take several combinations of the loop gain and the frequency ratio. Plot the variance of the loop frequency estimate as a function of SNR.

B: Demodulation of PM Signals Using SDPLL

The first-order SDPLL can demodulate PM signals as shown in Fig. D.20 below. Using MATLAB, simulate the PM demodulation circuit and study its behavior under noise-free conditions for different values of the modulation index.

C: Second-Order SDPLL

The second-order SDPLL utilizes a proportional-plus-accumulation filter and locks on zero phase. Simulate this loop and study its performance (as a frequency estimator) in Gaussian noise. Note that two initial phases should be considered.
Zahir M. Hussain took the first rank in Iraq in the General Baccalaureate Examinations 1979 with an average of 99%. He received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Baghdad in 1983 and 1989, respectively, and the PhD degree in electrical engineering from Queensland University of Technology, Brisbane, Australia, in 2002. From 1989 to 1998 he researched and lectured on electrical engineering and mathematics. In 2001 he joined the School of Electrical and Computer Engineering (SECE), RMIT University, Melbourne, Australia, as a researcher then lecturer of signal processing and communications. He was the academic leader of a 3G commercial communications project at RMIT 2001–2002. In 2002 he was promoted to Senior Lecturer. Since 2001 he has been the senior supervisor for 15 PhD candidates at RMIT (with 9 completions between 2001 and 2009); also the second supervisor for another 7 PhD candidates. Dr. Hussain has over 200 internationally refereed technical publications on signal processing, communications, and electronics. His joint work on single-bit processing with Professor Peter O’Shea has led to an Australian Research Council (ARC) Discovery Grant (2005–2007). In 2005 he was promoted to Associate Professor of Signal Processing. He got the RMIT 2005 and 2006 Publication Awards (also shared the 2004 Publication Award with Professor Richard Harris, now Chair of Telecommunications at Massey University, NZ). In 2006 he was the Head of the Communication Engineering Discipline at SECE, RMIT. In 2007 he won the RMIT Teaching Award. Dr. Hussain is a member of Engineers Australia (formerly IEAust), IET (formerly IEE), and a senior member of IEEE and the Australian Computer Society (ACS). He worked on the technical program committees of many conferences.
and served as a reviewer for leading IEEE and Elsevier journals. Prof. Hussain has recently joined Kufa University, staying as Adjunct Professor at RMIT.

Amin Z. Sadik received the B.Sc. (1983) and M.Sc. degrees (1988) in electrical engineering from the University of Baghdad, Iraq, and Baghdad University of Technology, Iraq, respectively. From 1989 to 1995, he was a lecturer in the School of Electrical Engineering, University of Technology, Baghdad, Iraq. From 1995 to 2001 he was a lecturer at the University of Salahuddin, Erbil, Iraq, and from 2001 to 2004 he was a lecturer at the University of Al-Balqa, Jordan. In 2006 he received his PhD degree in Electrical Engineering from the School of Electrical and Computer Engineering, RMIT University, Melbourne, Australia. Since then, he worked as a Research Fellow in RMIT University and as a Senior Research Fellow (Digital Signal Processing) in QUT. Dr. Sadik received the Best Paper Award in IEEE TENCON 2005 and DSPCDC 2006. He contributed to the technical committees of several international conferences and to the review process in Elsevier and other journals. In Feb. 2009 he joined the School of Engineering and Physical Sciences, Heriot-Watt University, Dubai Campus. His research interests include digital signal processing and digital communications.

Peter O'Shea received the B.E., Dip.Ed., and Ph.D. degrees, all from the University of Queensland, Australia. He has worked as an engineer at the Overseas Telecommunications Commission for 3 years, at University of Queensland for 4 years, at Royal Melbourne Institute of Technology (RMIT) for 7 years, and at Queensland University of Technology (QUT) for 9 years, where he is currently a Professor. He has received awards in Student Centered Teaching from the Faculty of Engineering and the University President at both RMIT and QUT. His interests are in signal processing for communications, power systems and biomedicine, and reconfigurable computing.