. **Digital Signal Processing** **Lab Instructor: Dr. Shafayat Abrar**

. **Lab Demonstrator: Ahmad Bilal**

. **Lab 02: Spectral Content of Deterministic and Random Signals: Applications of FFT**

1 Continuous-Time Fourier Transform (CTFT)

Let us start from the beginning, from an analog world description. The continuous-time Fourier transforms (CTFT), direct (synthesis) and inverse (analysis), are defined as follows:

Synthesis equation: $x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df$

Analysis equation: $X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$

In computer simulations, however, we do not have continuous-time, so we sample the time-domain signal $x(t)$ at a regular interval T_s to get a sampled signal $x_s(t)$ as follows:

$$x_s(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s) \tag{1}$$

The (theoretical) continuous-time Fourier transform of the sampled signal is:

$$\begin{aligned} X_s(f) &:= \text{CTFT}\left\{x_s(t)\right\} \\ &= \text{CTFT}\left\{\sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s)\right\} \\ &= X(f) * \frac{1}{T_s}\sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{T_s}\right) \\ &= \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X(f - kf_s)\,, \text{ for } -\infty < f < \infty, \text{ where } f_s := 1/T_s \end{aligned} \tag{2}$$

which makes use of the following properties of the Fourier transform:

$$\begin{aligned} &x(t)y(t) \longleftrightarrow X(f) * Y(f) \\ &X(f) * \delta(f - f_0) = X(f - f_0) \\ &\sum_{n=-\infty}^{\infty} \delta\left(t - nT_s\right) \longleftrightarrow \frac{1}{T_s}\sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{T_s}\right) \end{aligned} \tag{3}$$

An interesting observation we can make from (2) is that (refer to Fig. 1) if $X(f)$ is band-limited, then

$$X_s(f) = \frac{1}{T_s}X(f), \text{ for } -\frac{f_s}{2} \le f \le \frac{f_s}{2}, \text{ where } f_s := \frac{1}{T_s} \tag{4}$$

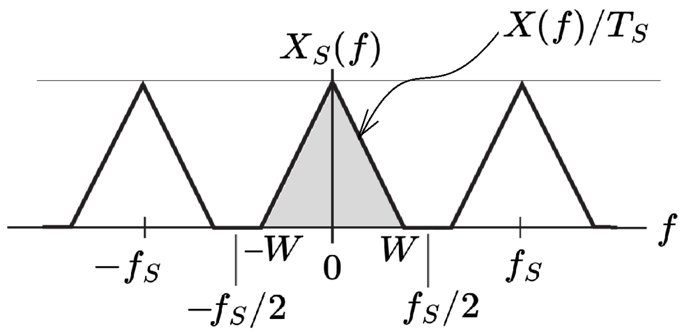


Figure 1: Caption

What do we learn from (4)?

We learn we may obtain the CTFT $X(f)$ from the DFT of the sampled signal!!!

This is explained as follows:

(1) First, we sample the continuous-time signal $x(t)$ at the rate $f_s = 1/T_s$ to obtain $x_s(t)$.

(2) Secondly, we obtain the discrete Fourier transform of $x_s(t)$ using the MATLAB function `fft`. This function, by design, provides the spectrum for the continuous-time frequencies: $0 \le f \le f_s$ [Hertz], or, equivalently, versus the discrete-time frequencies: $0 \le \Omega \le 2\pi$ [rad/sec].

(3) Thirdly, we may use the MATLAB function `fftshift` to shift the spectrum around $f = 0$, and as a result, the spectrum is now specified for the frequencies: $-0.5f_s \le f \le 0.5f_s$ [Hertz]; this gives $X_s(f)$. Or, we may refer to the spectrum as a discrete-time Fourier spectrum $X(\Omega)$ of the signal $x[n] = x(nT_s)$ specified for the frequencies: $-\pi \le \Omega \le \pi$ [rad/sec].

(4) Finally, we may obtain the CTFT of $x(t)$, i.e., $X(f)$, by simply multiplying the spectrum $X_s(f)$ with sampling time T_s as follows:

$$X(f) = T_s X_s(f), \text{ for } -\frac{f_s}{2} \le f \le \frac{f_s}{2}, \text{ where } f_s := 1/T_s$$

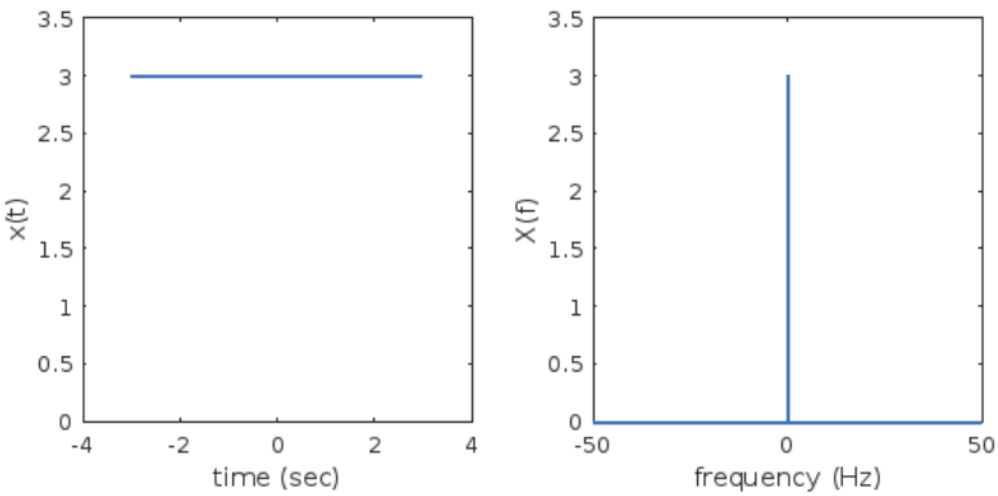
(5) **Important:** For **periodic signals** (which includes DC signal as well), if the periodic signal is defined over the time $t_{\text{min}} \le t \le t_{\text{max}}$, then the following is true:

$$X(f) = \frac{T_s}{t_{\text{max}} - t_{\text{min}}} X_s(f), \text{ for } -\frac{f_s}{2} \le f \le \frac{f_s}{2}, \text{ where } f_s := 1/T_s$$

Example 1: Compute the CTFT of the DC signal using the MATLAB function `fft`. We know that the Fourier transform of the DC signal $x(t) = A \; \forall t$ is $X(f) = A\delta(f)$.

To obtain this result in MATLAB, we first need to define the time variable, where we must mention the time step-size which may be termed as the sampling time; call it T_s. Consider the following code:

```
clc; clear; close all
A = 3; % Amplitude of the DC signal
tmax = 3; % (sec) Maximum time
tmin = -3; % (sec) Minimum time
Ts = 0.01; % (sec) Sampling time
fs = 1/Ts; % (Hertz) Sampling frequency
time = tmin:Ts:tmax; % (sec) Time axis
N = length(time); % Number of samples
xt = A*ones(1,N); % The DC signal
Xs = fft(xt,N); % Fourier Transform of the sampled signal
X = Ts*Xs/(tmax-tmin); % Fourier Transform of the continuous signal
X = fftshift(X); % Transform is centered around f = 0
freq = linspace(-fs/2, fs/2, N); % Frequency axis
figure; subplot 121; plot(time, xt, 'linewidth', 2);ylim([0 5])
xlabel('time (sec)'); ylabel('x(t)');
subplot 122; plot(freq, abs(X), 'linewidth', 2);ylim([0 3.5])
xlabel('frequency (Hz)'); ylabel('X(f)');
```



Activity 1: Spectral Contents of Periodic/Non-Periodic Deterministic Signals:

(a) Obtain the spectrum of the **periodic** sinusoidal signal, $x_1(t) = A\sin(2\pi f_0 t)$.

Note that due to Euler's identity

$$\sin(\theta) = \frac{1}{2\jmath}\exp(\jmath\theta) - \frac{1}{2\jmath}\exp(-\jmath\theta)$$

The two-sided spectrum of $A\sin(2\pi f_0 t)$ shows two delta functions at $f = \pm f_0$ with amplitude $A/2$ each.

(b) Obtain the spectrum of the **non-periodic** square-pulse signal, $x_2(t) = \Pi\left(\frac{t}{\tau}\right)$, where

$$A\Pi\left(\frac{t}{\tau}\right) = \begin{cases} A, & |t| < \frac{\tau}{2} \\ 0, & |t| > \frac{\tau}{2} \end{cases}$$

Note that the Fourier transform of $A\Pi\left(\frac{t}{\tau}\right)$ is $A\tau\text{sinc}(\tau f)$.

(c) Obtain the spectrum of the radio-frequency pulse signal, $x_3 = x_1(t)x_2(t) = A\sin(2\pi f_0 t)\Pi\left(\frac{t}{\tau}\right)$.

(d) Obtain the spectrum of the triangular pulse signal,

$$A\Lambda\left(\frac{t}{\tau}\right) = \begin{cases} \frac{A}{\tau}(\tau + t), & -\tau \le t < 0 \\ \\ \frac{A}{\tau}(\tau - t), & 0 \le t < \tau \\ 0, & \text{otherwise} \end{cases}$$

Note that the Fourier transform of $A\Lambda\left(\frac{t}{\tau}\right)$ is $A\tau\text{sinc}^2(\tau f)$.

2 Digital Line Codes

In digital data transmission, a **line code** refers to a specific pattern of voltage, current, or photons used to represent data as it is transmitted through a communication channel or stored on a medium. In data storage systems, this collection of signaling patterns is often referred to as a constrained code. Some traditional line coding schemes are summarized in Fig. 2-3.

The terms NRZ (Non-Return-to-Zero) and RZ (Return-to-Zero) describe two common pulse formats. An NRZ pulse remains constant throughout the entire duration of the bit period, while an RZ pulse returns to the zero voltage level midway through the bit period. These pulse formats differ in how they encode timing information and signal transitions.

The distinction between `Unipolar` and `Polar` line codes lies in how they assign voltage levels to represent binary data, specifically bit-0. In `Unipolar` coding, bit-0 is always assigned a zero voltage level, whereas in `Polar` coding, bit-0 is represented by a negative voltage level. Both `Unipolar` and `Polar` schemes are uncorrelated, meaning that the voltage levels assigned to bit-1 and bit-0 are independent of the previously transmitted bits.

In contrast, `Bipolar` line codes introduce correlation by alternating the polarity of bit-1 between positive and negative voltage levels for consecutive 1's. This means the polarity toggles with each occurrence of a bit-1, while bit-0 is consistently represented by a zero voltage level. This alternating behavior in `Bipolar` coding reduces the DC component in the signal, which can be beneficial for certain communication channels, but also distinguishes it as a correlated coding scheme.

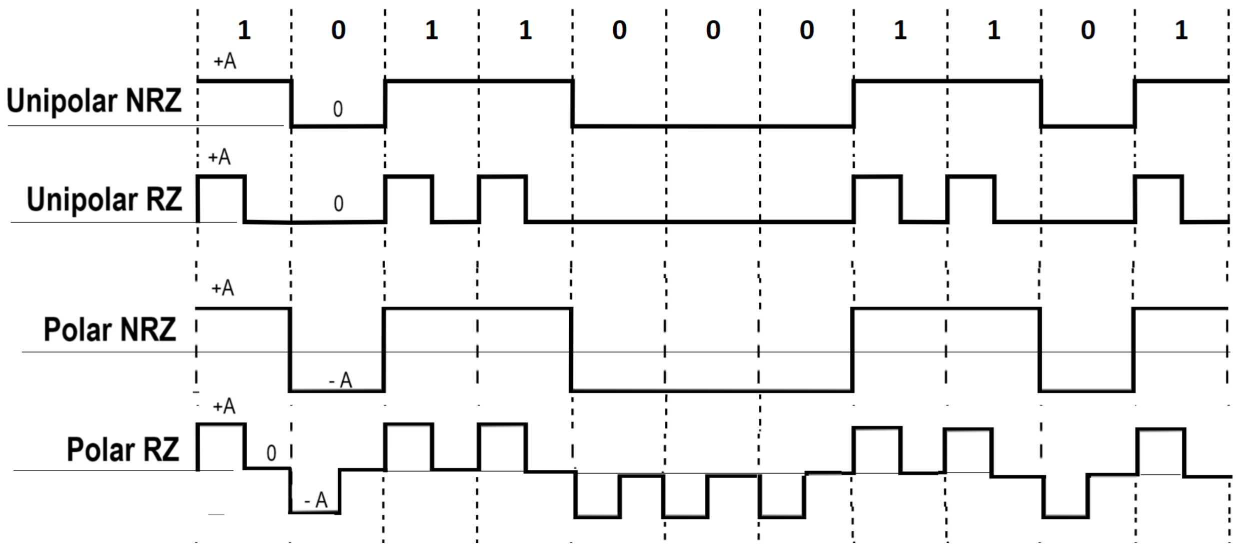


Figure 2: Uncorrelated digital line codes.

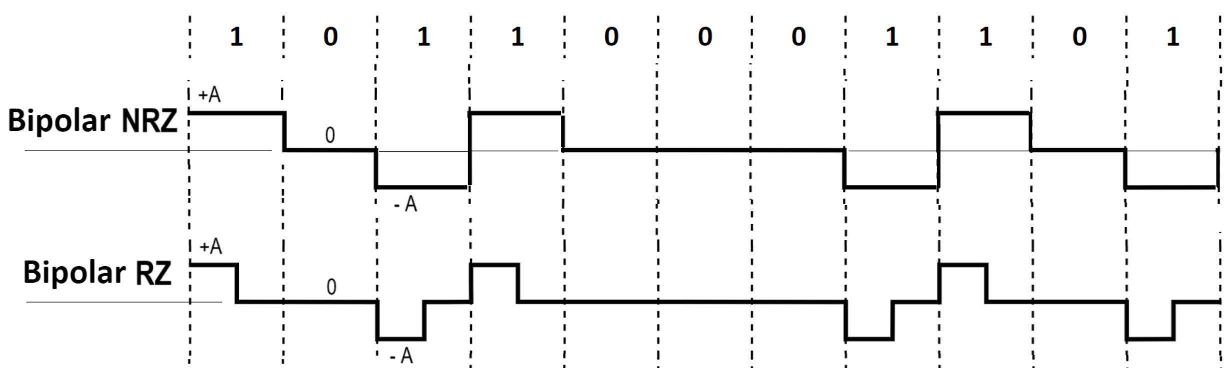


Figure 3: Correlated digital line codes.

3 Power Spectral Density (PSD)

In signal processing, the **power spectrum** $\Psi_x(f)$ of a continuous-time signal $x(t)$ represents how the signal's power is distributed among its frequency components f. Essentially, it provides a detailed view of the energy content of the signal across different frequencies.

Fourier analysis, a fundamental tool in signal processing, states that any physical signal can be expressed as a sum of sinusoidal components, each with a specific frequency, amplitude, and phase. These components may be discrete frequencies or span a continuous range, depending on the nature of the signal.

The term **spectrum** refers to the frequency-based representation of a signal, which includes the statistical average of its power or energy across different frequency bands. This concept applies to all types of signals, including deterministic signals and random noise. The power spectrum, therefore, serves as a key tool in understanding the frequency content of a signal and analyzing its behavior in various applications, such as communication, audio processing, and control systems.

4 PSD of Digital Line Codes Using CTFT

Digital line codes are inherently random. For instance, in digital telephony, the digital signals represent digitized voices, which exhibit random behavior. Similarly, in digital television, the digital signals consist of digitized images and voices, both of which are also random in nature.

One of the primary concerns for system design engineers is the bandwidth occupied by these digital line codes. In this lab, we derive a general formula for the Power Spectral Density (PSD) of digital signals. This formula accounts for both correlated and uncorrelated data, making it versatile and applicable to a wide range of scenarios involving baseband signals. By encompassing various cases, this formula provides a robust framework for analyzing the frequency characteristics of digital signals in diverse applications.

Let the baseband digital signal be represented by

$$x(t) = \sum_{n=-\infty}^{\infty} a_n g(t - nT) \tag{5}$$

where a_n are discrete random data symbols, $g(t)$ is a signal of duration T (i.e., nonzero only in $[0, T]$). Let us name $g(t)$ as the *symbol function* or *pulse shape*. It could be any signal with a Fourier transform. For example, it could be a baseband symbol-shaping pulse or a carrier burst at passband. The random sequence $\{a_n\}$ could be binary or non-binary.

Now to find out the power spectral density of the signal in (5), we first consider the transmission of $2J + 1$ *symbols*

$$x(t) = \sum_{n=-J}^{J} a_n g(t - nT) \tag{6}$$

Next, take the CTFT of both sides of (6), the spectrum of this truncated signal is found as follows, where

$x(t) \longleftrightarrow X(f)$ and $g(t) \longleftrightarrow G(f)$

$$X(f) = G(f) \sum_{n=-J}^{J} a_n e^{-j\omega nT}$$

which exploits the fact that $g(t-t_0) \longleftrightarrow G(f)e^{-j\omega t_0}$, and $\omega = 2\pi f$. The power spectral density of the original signal in (5) is obtained by taking the statistical average and time limit of $|X(f)|^2$ as follows:

$$\begin{aligned}
\Psi_x(f) = \lim_{J\to\infty} \frac{1}{(2J+1)T} E\left\{|X(f)|^2\right\} &= \lim_{J\to\infty} \frac{|G(f)|^2}{(2J+1)T} E\left\{\left|\sum_{n=-J}^{J} a_n e^{-j\omega nT}\right|^2\right\} \\
&= \lim_{J\to\infty} \frac{|G(f)|^2}{(2J+1)T} E\left\{\left(\sum_{n=-J}^{J} a_n e^{-j\omega nT}\right)\left(\sum_{m=-J}^{J} a_m e^{-j\omega mT}\right)^*\right\} \\
&= |G(f)|^2 \lim_{J\to\infty} \frac{1}{(2J+1)T} \sum_{n=-J}^{J}\sum_{m=-J}^{J} E\left\{a_n a_m\right\} e^{j(m-n)\omega T} \\
&= |G(f)|^2 \lim_{J\to\infty} \frac{1}{(2J+1)T} \sum_{n=-J}^{J}\sum_{\ell=n+J}^{n-J} R(\ell) e^{-j\ell\omega T}
\end{aligned}$$

where $\ell := n - m$ and $R(\ell) := E\left\{a_n a_m\right\} = E\left\{a_n a_{n-\ell}\right\}$ is the auto-correlation function of the data bits. Note that $R(-\ell) = R(\ell)$. Realizing the inner summations become the same regardless of the value of index n when $J \to \infty$, we obtain

$$\Psi_x(f) = \frac{|G(f)|^2}{T} \lim_{J\to\infty}\left(\frac{2J+1}{2J+1}\sum_{\ell=n+J}^{n-J} R(\ell)e^{-j\ell\omega T}\right) = \frac{|G(f)|^2}{T}\sum_{\ell=\infty}^{-\infty} R(\ell)e^{-j\ell\omega T}$$

Equivalently this is

$$\Psi_x(f) = \frac{|G(f)|^2}{T}\sum_{\ell=-\infty}^{\infty} R(\ell)e^{-j\ell\omega T} \tag{7}$$

Now, there are **two** possible cases of $R(\ell)$; one is **uncorrelated**, and the other **correlated**.

4.1 When Data Symbols Are Uncorrelated

When two random variables X and Y are uncorrelated, then $E\{XY\} = E\{X\}E\{Y\}$. Assume a_n has a mean of $E\left\{a_n\right\} = m_a$ and a variance of σ_a^2, then

$$\begin{aligned}
R(\ell) &= \begin{cases} E\left\{a_n^2\right\}, & \ell = 0 \\ E\left\{a_n\right\}E\left\{a_{n-\ell}\right\}, & \ell \neq 0 \end{cases} \\
&= \begin{cases} \sigma_a^2 + m_a^2, & \ell = 0 \\ m_a^2, & \ell \neq 0 \end{cases}
\end{aligned}$$

Substitute this for $R(\ell)$ in (7), we have

$$\Psi_x(f) = \frac{|G(f)|^2}{T}\left(\sigma_a^2 + m_a^2 \sum_{\ell=-\infty}^{\infty} e^{-j\ell\omega T}\right)$$

By revoking the Poisson sum formula:

$$\sum_{\ell=-\infty}^{\infty} e^{-j\ell\omega T} = \frac{1}{T}\sum_{\ell=-\infty}^{\infty} \delta\left(f - \frac{\ell}{T}\right)$$

where $\delta(f)$ is the impulse function, we have

$$\Psi_x(f) = \frac{|G(f)|^2}{T}\left(\sigma_a^2 + \frac{m_a^2}{T}\sum_{\ell=-\infty}^{\infty} \delta\left(f - \frac{\ell}{T}\right)\right) \tag{8}$$

That is, for uncorrelated data

$$\Psi_x(f) = \underbrace{\frac{\sigma_a^2|G(f)|^2}{T}}_{\text{continuous spectrum}} + \underbrace{\left(\frac{m_a}{T}\right)^2 \sum_{\ell=-\infty}^{\infty} \left|G\left(\frac{\ell}{T}\right)\right|^2 \delta\left(f - \frac{\ell}{T}\right)}_{\text{discrete spectrum}} \tag{9}$$

Above we exploit the sampling property of the delta function, that is,

$$G(f)\delta(f - f_0) = G(f_0)\delta(f - f_0)$$

The first term is the continuous part of the spectrum which is a scaled version of the PSD of the symbol-shaping pulse. The second term is the discrete part of the spectrum which has spectral lines at frequencies k/T (i.e., multiples of the data rate). The spectral lines have an envelope of the shape of the PSD of the symbol-shaping pulse. Each spectral line has a strength of $(m_a/T)^2\,|G(k/T)|^2$.

4.1.1 NRZ Polar, Uncorrelated Digital Signal Transmission

Consider NRZ Polar code, where present pulse voltage a_n depends only on the present value of data bit; if bit is zero, pulse a_n is negative; if bit is one, then pulse a_n is positive. That is $s(t) = \sum_{n=-\infty}^{\infty} a_n g(t - nT)$, where $a_n = +A$ or $-A$ with equal probabilities for bit one and zero, respectively, and $E\{a_n a_m\} = E\{a_n\}E\{a_m\}$ for $n \neq m$. It is easy to see that its mean is zero:

$$m_a = E\{a_n\} = 0.5(A) + 0.5(-A) = 0$$

and its variance is

$$\sigma_a^2 = E\left\{a_n^2\right\} + m_a^2 = E\left\{a_n^2\right\} = 0.5(+A)^2 + 0.5(-A)^2 = A^2$$

Further, since a_n are uncorrelated and stationary, then

$$R(\ell) = E\left\{a_n a_{n-\ell}\right\} = \begin{cases} E\left\{a_n^2\right\} = \sigma_a^2 = A^2, & \ell = 0 \\ E\left\{a_n a_{n-\ell}\right\} = E\left\{a_n\right\} E\left\{a_{n-\ell}\right\} = 0 \cdot 0 = 0, & \ell \neq 0 \end{cases}$$

Now refer to (9), note that $m_a = 0$ and $\sigma_a^2 = A^2$, the second term becomes 0 and the PSD of the signal is (assuming $A = 1$)

$$\Psi_x(f) = \frac{|G(f)|^2}{T} \tag{10}$$

It shows that the PSD of a binary (± 1), equiprobable, stationary, and uncorrelated data sequence is just equal to the energy spectral density $|G(f)|^2$ of the symbol-shaping pulse $g(t)$ divided by the symbol duration. Data bits are binary (± 1), equiprobable, stationary, and uncorrelated. The symbol-shaping pulse is the rectangular pulse: $g(t) = 1$ for $(-T/2, T/2)$, and 0 elsewhere. Or, we denote $g(t) = \Pi(t)$. Then from a Fourier transform table,

$$G(f) = T\left(\frac{\sin \pi f T}{\pi f T}\right) = T \operatorname{sinc}(fT)$$

From (10). its PSD is

$$\Psi_x(f) = \frac{|G(f)|^2}{T} = T\left(\frac{\sin \pi f T}{\pi f T}\right)^2 = T \operatorname{sinc}^2(fT)$$

4.2 When Data Symbols are Correlated

Consider a `Bipolar` line code. This group of line codes uses three voltage levels $\pm A$ and 0. This group is also referred to as `AMI` (alternative mark inversion) codes or `PT` (pseudo-ternary) codes. In `Bipolar-NRZ` format, a `1` (bit one) is represented by an NRZ pulse with **alternative polarities** if `1's` are consecutive; this means voltage levels for bit `1` are **correlated**. A `0` (bit zero) is represented by the zero level. In `Bipolar-RZ`, the coding rule is the same as `Bipolar-NRZ` except that the symbol pulse has a half-length of T. They have no DC component but like `Bipolar-NRZ` their lack of transitions in a string of `0's` may cause a synchronization problem.

For `Bipolar-NRZ` codes the data sequence $\{a_n\}$ takes on three values with the following probabilities:

$$a_n = \begin{cases} A, & \text{for binary } 1, \quad p_A = 1/4 \\ -A, & \text{for binary } 1, \quad p_{-A} = 1/4 \\ 0, & \text{for binary } 0, \quad p_0 = 1/2 \end{cases}$$

We can find $R(\ell)$ for $\ell = 0$ as follows:

$$R(0) = E\left\{a_i^2\right\} = \frac{1}{4}(A)^2 + \frac{1}{4}(-A)^2 + \frac{1}{2}(0)^2 = \frac{1}{2}A^2$$

Adjacent bits in $\{a_n\}$ are correlated due to the alternate mark inversion. The adjacent bit pattern in the original binary sequence must be one of these: $(1,1),(1,0),(0,1)$, and $(0,0)$. The possible $a_k a_{k+1}$ products are $-A^2, 0, 0, 0$. Each of them has a probability of $1/4$. Thus

$$R(1) = E\{a_n a_{n-1}\} = \frac{1}{4}(-A^2) + \frac{1}{4}\cdot 0 + \frac{1}{4}\cdot 0 + \frac{1}{4}\cdot 0 = -\frac{1}{4}A^2$$

For $k > 1$, the possible $a_i a_{i-k}$ products are $\pm A^2, 0, 0, 0$. Each case occurs with a probability of $1/4$; the probabilities of $-A^2$ and $+A^2$ are the same and equal to $1/8$. Thus

$$R(k)\bigg|_{k>1} = E\{a_n a_{n-k}\} = \frac{1}{8}(A^2) + \frac{1}{8}(-A^2) = 0$$

Thus, for $k > 1$, a_n and a_{n-k} are **uncorrelated**. Summarizing the above results, we have

$$R(\ell) = \begin{cases} \frac{1}{2}, & \ell = 0 \\ -\frac{1}{4}, & |\ell| = 1 \\ 0, & |\ell| > 1 \end{cases}$$

Substitute this $R(\ell)$, we have the PSD of `Bipolar-NRZ` code:

$$\begin{aligned}\Psi_x(f) = \frac{1}{T}|G(f)|^2\left(\frac{1}{2} - \frac{1}{4}e^{j\omega T} - \frac{1}{4}e^{-j\omega T}\right) &= \frac{1}{T}|G(f)|^2\left(\frac{1}{2} - \frac{1}{2}\cos(\omega T)\right) \\ &= \frac{A^2 T}{4}\left(\frac{\sin \pi f T/2}{\pi f T/2}\right)^2 \sin^2 \pi f T.\end{aligned}$$

5 PSD Using DFT/FFT

In computer simulations, we do not have continuous time, so we sample the signal $x(t)$ at a regular interval T_s to get

$$x_s(t) = x(t)\sum_{n=-\infty}^{\infty}\delta(t - nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s) \tag{11}$$

The CT spectrum for the sampled signal is:

$$\mathrm{CTFT}\{x_s(t)\} = X_s(f) = X(f) * \frac{1}{T_s}\sum_{k=-\infty}^{\infty}\delta\left(f - \frac{k}{T_s}\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} X\left(f - \frac{k}{T_s}\right)$$

which makes use of the following properties:

$$\begin{aligned} x(t)y(t) &\longleftrightarrow X(f) * Y(f) \\ X(f) * \delta(f - f_0) &= X(f - f_0) \end{aligned}$$

If we take CTFT of each term of $\sum_{n=-\infty}^{\infty} x(nT_s)\delta(t-nT_s)$ separately, we obtain

$$X_s(f) = \sum_{n=-\infty}^{\infty} x(nT_s)e^{-j2\pi f nT_s}$$

If we take DTFT of sampled signal $x(nT_s)$, we obtain

$$X_s(\Omega) = \sum_{n=-\infty}^{\infty} x(nT_s)e^{-j\Omega n}$$

Note, that both CTFT and DTFT provide analytical insight into the resulting spectrum; however, for numerical estimation of the spectrum, we need to rely on DFT/FFT (FFT is an efficient method to compute DFT). The DFT of signal $x(nT_s)$ is obtained as follows:

$$X_s[k] = \sum_{n=0}^{N-1} x[n]e^{-j(2\pi k/N)n} = \mathrm{FFT}\left\{x[n]\right\}$$

where $x[n] := x(nT_s)$, and $N = (2J+1)M$. So, the PSD may be computed numerically using the FFT function of MATLAB. Since, the signal $x(nT_s)$ contains random pulse amplitudes, this is necessary to realize $x(nT_s)$ L many times, compute the FFT of each realization, add all FFT values together, and divide by L to get an average behavior of FFT. The DFT/FFT-based PSD, $\Psi_x[k]$, is expressed as:

$$\Psi_x[k] = \frac{E\left\{\left|\mathrm{FFT}\{x(nT_s)\}\right|^2\right\}}{(2J+1)\,T} \approx \frac{\sum_{\ell=1}^{L}\left|\mathrm{FFT}\{x^{\ell}(nT_s)\}\right|^2}{(2J+1)\,T\,L}, \quad 0 \leq k \leq N-1$$

where $x^{\ell}(nT_s)$ is the ℓth realization of square wave pulse-shaped stream of $2J+1$ random binary data pulses, where each pulse is sampled M time; note that the total number of samples in $x(nT_s)$ is $N = (2J+1)M$.

6 MATLAB Simulations

We discuss computer simulation of digital line code and the corresponding power spectral density. We consider `NRZ-Polar` code. The pulse shape is a square wave for the bit period T. In MATLAB, however, we rely on discrete-time simulation, where time is divided into small intervals known as sampling time T_s. We need to assume an integer number of T_s in a T; assume $T = MT_s$, where M is a suitable integer, say $M = 10$. Voltage levels are $+A$ and $-A$ for bit-1 and bit-0, respectively. For convenience, we may assume that $A = 1$.

First, we generate $2J+1$ bits (a random sequence of `ones` and `zeros`) and assign the corresponding square wave to it. This can be done as follows for `NRZ-Polar` code:

```
A = 1;
J = 50;
M = 10;
Bits = (sign(randn(1,2*J+1))+1)/2;
xNt = [];
for ii=1:2*J+1
    if Bits(ii)==1
       xNt=[A*ones(1,M) xNt];
    else
       xNt=[-A*ones(1,M) xNt];
    end
end
```

For example, for $J = 2$, we may get `Bits = [0 1 1 0 1]`.[1] The resulting waveform $x(t)$ is obtained as follows for NRZ-`Polar` code:

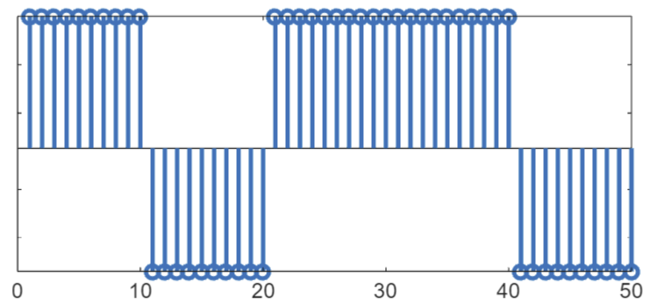


Figure 4: The waveform $x(t)$ in discrete-time for $K = 2$.

The MATLAB for NRZ-`Polar` code is obtained as follows:

```
clc; clear; close all
A = 1; J = 50; M = 10; L = 20000;
PSD = 0; T = 1;
for jj=1:L
xNt = [];
Bits = (sign(randn(1,2*J+1))+1)/2;
    for ii=1:2*J+1
        if Bits(ii)==1
            xNt=[A*ones(1,M) xNt];
        else
            xNt=[-A*ones(1,M) xNt];
        end
    end
    PSD = PSD + abs(fft(xNt)).^2;
end
PSD = PSD/L/(2*J+1)/T; plot(PSD)
```

[1] For the sake of PSD simulation, in the sequel, we will assume a relatively large value for J, say $J = 50$.

The PSD $\Psi_x[k]$ using DFT/FFT is illustrated in Fig.5(a), where the frequency discrete index k goes from 0 to $M(2J+1)-1$. If we want to obtain $\Psi_x(f)$ from $\Psi_x[k]$, then there are some normalization and shifting steps involved as mentioned below:

1. First, the x-axis is normalized as Tf so that the resulting PSD is not dependent on pulse duration. It is a standard practice to assume $T=1$.
2. This is a standard practice to normalize $\Psi_x[k]$ by dividing it by its area. The purpose is to make sure that all line code schemes under discussion consume the same amount of power.
3. A numerical DTFT-based PSD, $\Psi_x(\Omega)$, for the frequency interval $0 \le \Omega \le 2\pi$ may be obtained from numerically computed $\Psi_x[k]$, $0 \le k \le (2J+1)M-1$. Moreover, if we are interested in a two-sided spectrum, then the FFT values must be re-arranged to make them correspond to the interval $-\pi \le \Omega \le \pi$.
4. The PSD obtained from the FFT function of MATLAB may be mapped to CTFT-based PSD, $\Psi_x(f)$, for the frequency interval $0 \le f \le f_s$, where $f_s = 1/T_s$. If we are interested in a two-sided spectrum, then the FFT values must be re-arranged to map them to interval $-f_s/2 \le f \le +f_s/2$. Secondly, for $-f_s/2 \le f \le +f_s/2$, we notice that

$$X_s(f) = \frac{1}{T_s} X(f) \Rightarrow X(f) = T_s X_s(f)$$

   The PSD in CTFT, $\Psi_x(f)$, depends upon $X(f)$ (and not $X_s(f)$). Since, $X(f) = T_s X_s(f)$, $\Psi_x(f)$ may be obtained from $\Psi_x[k]$ by multiplying it with T_s^2.

```
clc;clear;close all
A = 1; J = 50; M = 10; L = 20000; PSD = 0;
T = 1; Ts=T/M; fs=1/Ts;
for jj=1:L
sNt = [];
Bits = (sign(randn(1,2*J+1))+1)/2;
    for ii=1:2*J+1
        if Bits(ii)==1
            sNt=[A*ones(1,M) sNt];
        else
            sNt=[-A*ones(1,M) sNt];
        end
    end
    PSD = PSD + abs(fft(sNt)).^2;
end
PSD = PSD/L/(2*J+1)/T; LPSD = length(PSD);
figure(1); plot(PSD,'k-','linewidth',1.5)
set(gca,'FontSize',14);axis tight;grid on;
xlabel('$0\le k\le (2J+1)M$','FontSize',16,'interpreter','latex')
```

```
title('PSD (Unnormalized) vs. index $k$','FontSize',14,'interpreter','latex')
ylabel('$\Psi_x[k]$','FontSize',16,'interpreter','latex');
h=text(400,50,['$J = ' num2str(J) ', M = ' num2str(M) '$']);
set(h,'fontsize',14,'interpreter','latex');drawnow
DTFT_xaxis = linspace(0,2*pi,LPSD);
figure(2);plot(DTFT_xaxis,PSD,'k-','linewidth',1.5)
set(gca,'FontSize',14);axis tight; grid on;
xlabel('$0 \le \Omega\le 2\pi$ [radian]','FontSize',16,'interpreter','latex')
title('PSD (Unnormalized) vs. $\Omega$','FontSize',14,'interpreter','latex')
ylabel('$\Psi_x(\Omega)$','FontSize',16,'interpreter','latex');drawnow
fT = linspace(-T*fs/2,T*fs/2,LPSD); % normalized freq axis
Area = sum(PSD)*(fT(2)-fT(1)); CTFT_PSD = PSD/Area;
CTFT_PSD = [fliplr(CTFT_PSD(end:-1:floor(LPSD/2)))...
    CTFT_PSD(1:floor(LPSD/2)-1)];
figure(3);plot(fT,CTFT_PSD,'k-','linewidth',1.5)
set(gca,'FontSize',14);axis tight; grid on;
xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$','FontSize',16,'interpreter','latex')
ylabel('$\Psi_x(f)$','FontSize',16,'interpreter','latex');
xlim([-fs/2 fs/2]);title('PSD vs. $Tf$ [Normalized and two-sided]',...
    'FontSize',14,'interpreter','latex');drawnow
```

6.1 Semilog Plot

This is a standard practice to illustrate PSD on semilog scale; this helps improve the clarity of sidelobes of the spectrum. The `Polar-NRZ`'s log PSD is shown in Fig. 5(d).

```
figure(4);semilogy(fT,CTFT_PSD,'k-','linewidth',1.5);set(gca,'FontSize',14);
xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$','FontSize',16,'interpreter','latex')
ylabel('$\Psi_x(f)$ [dB]','FontSize',16,'interpreter','latex');ylim([1e-3 1]);
xlim(T*[-fs/2 fs/2]);title('Log PSD vs. $Tf$ [Normalized and two-sided]',...
    'FontSize',14,'interpreter','latex');grid on;axis tight;drawnow
```

Activity 2: Power Spectral Density of Random Signals:

(a) Obtain plots of $\Psi_x[k]$, $\Psi_x(\Omega)$, and normalized semilog $\Psi_x(f)$ of `Polar-RZ` code.

(b) Obtain plots of $\Psi_x[k]$, $\Psi_x(\Omega)$, and normalized semilog $\Psi_x(f)$ of `Unipolar-NRZ` code.

(c) Obtain plots of $\Psi_x[k]$, $\Psi_x(\Omega)$, and normalized semilog $\Psi_x(f)$ of `Unipolar-RZ` code.

(d) Obtain plots of $\Psi_x[k]$, $\Psi_x(\Omega)$, and normalized semilog $\Psi_x(f)$ of `Bipolar-NRZ` code.

(e) Obtain plots of $\Psi_x[k]$, $\Psi_x(\Omega)$, and normalized semilog $\Psi_x(f)$ of `Bipolar-RZ` code.

(f) Compare plots of normalized one-sided semilog $\Psi_x(f)$ of `Bipolar-NRZ` code, `Polar-NRZ` code, and `Unipolar-NRZ` code versus $0 \le Tf \le Tf_s/2$.

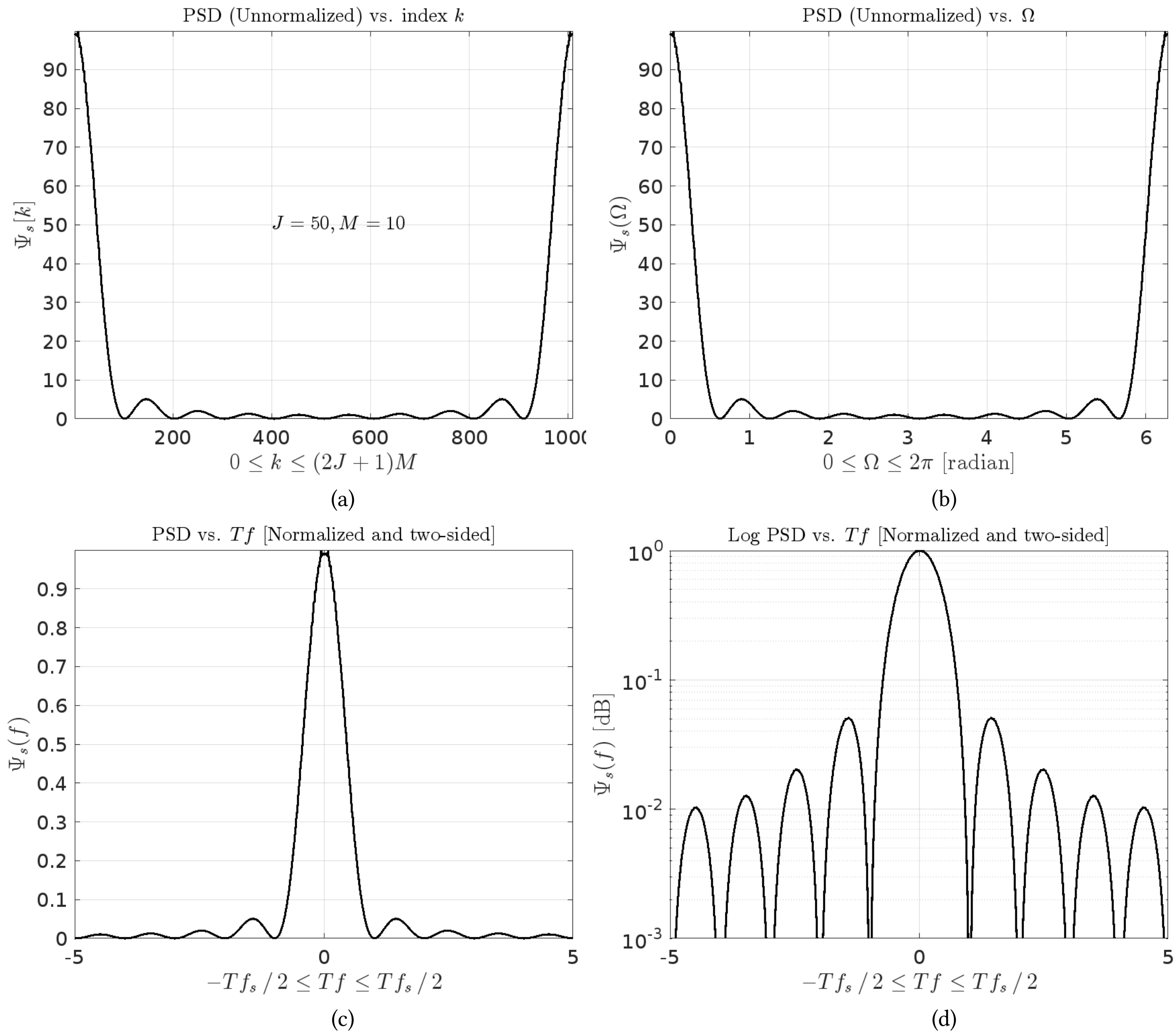


Figure 5: PSD of `Polar`-NRZ.

(g) Compare plots of normalized one-sided semilog $\Psi_x(f)$ of `Bipolar`-RZ code, `Polar`-RZ code, and `Unipolar`-RZ code versus $0 \leq Tf \leq Tf_s/2$.

(h) Provide a comparison of theoretical and simulated PSD using MATLAB, $\Psi_x(f)$, for `Polar`-NRZ. Repeat this task for `Polar`-RZ.

Activity 3: Answer the following questions:

1. Unipolar codes include DC voltage in their power spectral density (PSD). What are the potential drawbacks of DC voltage in data transmission?
2. What advantages does RZ have compared to NRZ? What disadvantages does RZ have compared to NRZ?
3. Bipolar codes, unlike polar codes, exhibit a spectral null at $f = 0$. Is this an advantage or a disadvantage? Briefly explain your opinion.
4. Prove that

$$\sum_{\ell=-\infty}^{\infty} e^{-j\ell\omega T} = \frac{1}{T} \sum_{\ell=-\infty}^{\infty} \delta\left(f - \frac{\ell}{T}\right)$$

5. Explain why it is possible to obtain $\Psi_x(f)$ numerically from $\Psi_x[k]$ multiplied with T_s^2.
6. Unipolar and Bipolar codes are commonly thought to cause synchronization issues. What is your understanding of this?
7. What is your understanding of Dicode NRZ? Briefly describe how it operates.

References

1. Xiong, Fuqin. *Digital Modulation Techniques.* Artech House Inc., London, 2006.
Refer to Chapter 2 and Appendix A.

7 Selected Answers:

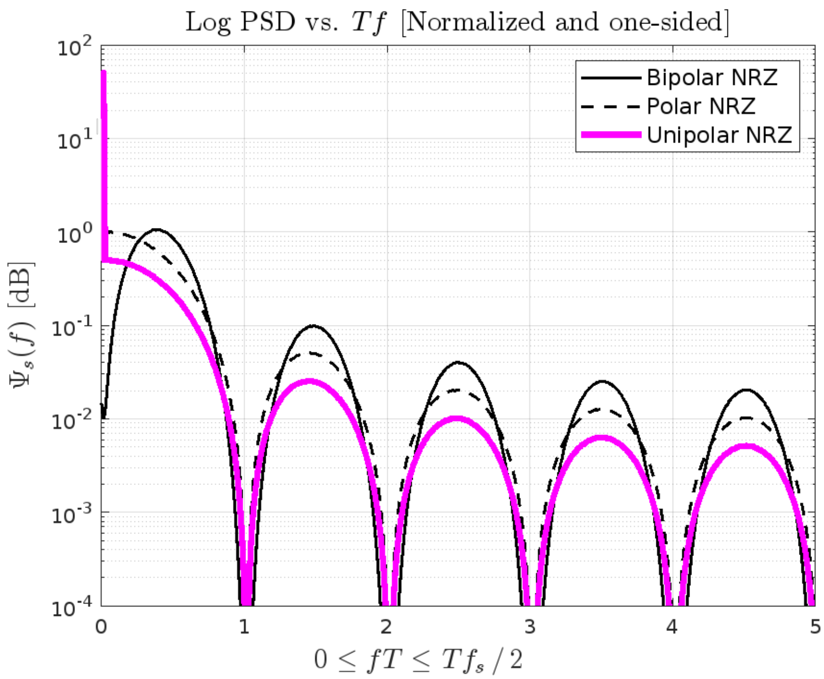


Figure 6: Task 6 plot for NRZ codes.

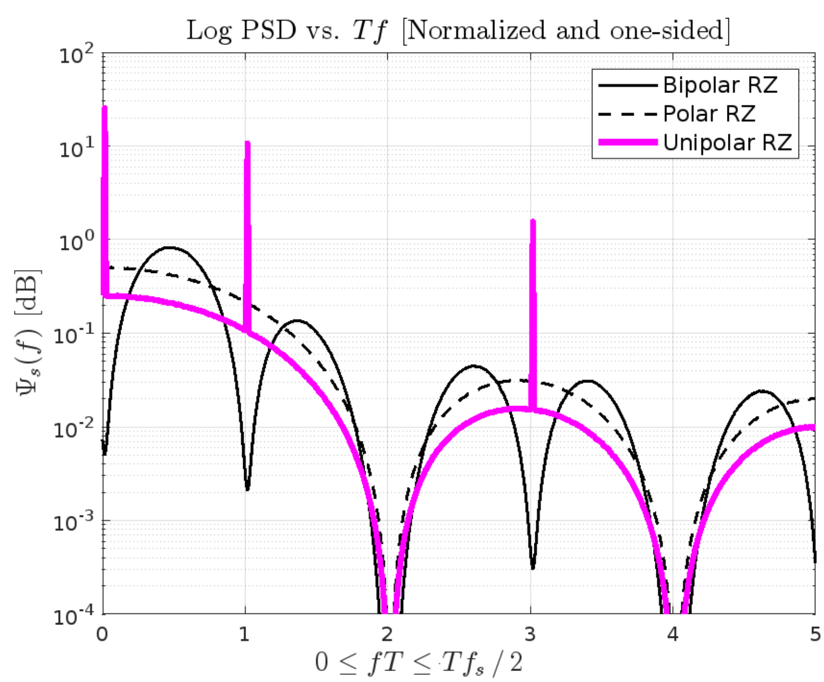


Figure 7: Task 7 plot for RZ codes.