HABIB UNIVERSITY

. Digital Signal Processing — Lab Instructor: Dr. Shafayat Abrar
. Lab 03: PSD of Digital Line Codes — Lab Demonstrator: Ahmad Bilal

1 Continuous-Time Fourier Series (CTFS): Exponential Form

Synthesis equation:

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}$$

Analysis equation:

$$c_k = \frac{1}{T_0} \int_{t_0}^{t_0+T_0} \tilde{x}(t) e^{-jk\omega_0 t} dt$$

2 Continuous-Time Fourier Transform (CTFT)

Synthesis equation:

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

Analysis equation:

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

3 Discrete-Time Fourier Series (DTFS)

Synthesis equation:

$$\tilde{x}[n] = \sum_{k=0}^{N-1} \tilde{c}_k e^{j(2\pi/N)kn}, \qquad \text{all } n$$

Analysis equation:

$$\tilde{c}_k = \frac{1}{N} \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j(2\pi/N)kn}$$

4 Discrete-Time Fourier Transform (DTFT)

Synthesis equation:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\Omega) e^{j\Omega n} d\Omega$$

Analysis equation:

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n}$$

5 Parseval's theorem

For a periodic power signal $\tilde{x}[n]$ with period N and DTFS coefficients $\{\tilde{c}_k, k = 0, \ldots, N-1\}$ it can be shown that

$$\frac{1}{N}\sum_{n=0}^{N-1}|\tilde{x}[n]|^2 = \sum_{k=0}^{N-1}|\tilde{c}_k|^2$$

For a non-periodic energy signal $x[n]$ with DTFT $X(\Omega)$, the following holds true:

$$\sum_{n=-\infty}^{\infty}|\tilde{x}[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi}|X(\Omega)|^2 d\Omega$$

6 Discrete Fourier Transform (DFT)

Synthesis equation:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j(2\pi/N)kn}, \quad k = 0, \ldots, N-1$$

Analysis equation:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j(2\pi/N)kn}, \quad n = 0, \ldots, N-1$$

7 Relationship of the DFT to the DTFT

The DFT of a length- N signal is equal to its DTFT evaluated at a set of N angular frequencies equally spaced in the interval $[0, 2\pi)$. Let an indexed set of angular frequencies be defined as

$$\Omega_k = \frac{2\pi k}{N}, \quad k = 0, \ldots, N-1$$

The DFT of the signal is written as

$$X[k] = X(\Omega_k) = \sum_{n=0}^{N-1} x[n]e^{-j(2\pi k/N)n}$$

8 Digital Line Codes

In digital data transmission, a **line code** is a pattern of voltage, current, or photons used to represent data transmitted down a communication channel or written to a storage medium. This repertoire of signals is usually called a constrained code in data storage systems. Some traditional line code. are summarized in Fig. 1.

Note NRZ and RZ stand for Non-return-to-zero and Return-to-zero, respectively; an NRZ pulse stays for the whole duration of the bit period whereas an RZ pulse returns to zero level in the mid of the bit period. `Unipolar` and `Polar` differ from each other in the aspect of how bit-0 is assigned a voltage level; unipolar and polar assign zero and negative pulse to bit-0, respectively. Also, note that

`Unipolar` and `Polar` are uncorrelated schemes where voltage levels are assigned to bit-1 and bit-0 in an absolute manner (irrespective of what was transmitted earlier). On the other hand, `Bipolar` is a correlated scheme where bit-1's polarity toggles between a positive and negative level (i.e., in an alternate fashion) while bit-0 is always assigned a zero voltage.

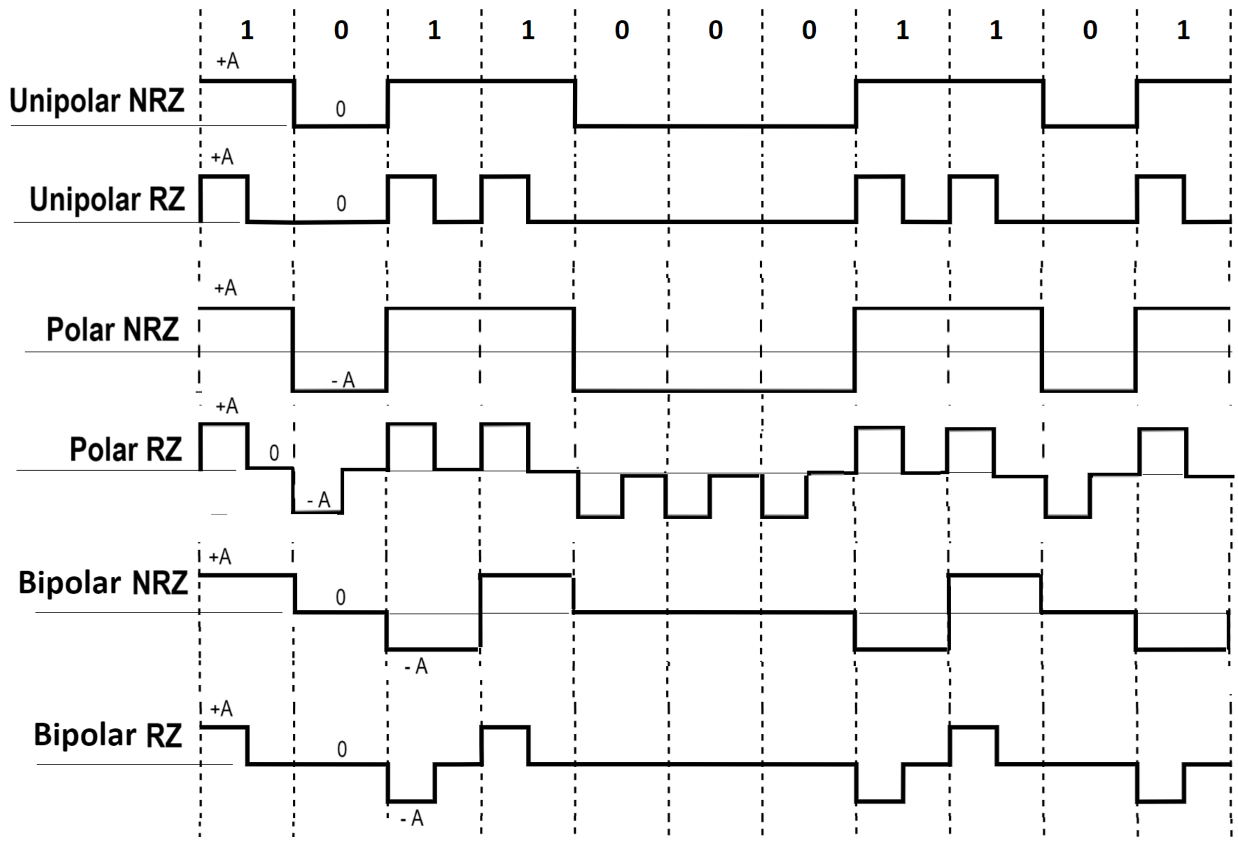

Figure 1: Digital line codes.

9 Power Spectral Density (PSD)

In signal processing, the **power spectrum** $\Psi_{xx}(f)$ of a continuous-time signal $x(t)$ describes the distribution of power into frequency components f composing that signal. [1] According to Fourier analysis, any physical signal can be decomposed into several discrete frequencies or a spectrum of frequencies over a continuous range. The statistical average of any sort of signal (including noise) as analyzed in terms of its frequency content, is called its **spectrum**.

10 PSD of Digital Line Codes Using CTFT

Digital line codes are essentially random. For example, in digital telephony, the digital signals are digitized voices, which are random. In digital television, the digital signals are digitized image and voice, which are also random. The bandwidth occupied by digital line codes is of the most concern to system design engineers. In this lab, we derive a general formula for the Power Spectral Density

(PSD) of digital signals. This general formula includes cases of correlated data and uncorrelated data. Therefore it can be used for a wide range of applications involving baseband signals.

Let the baseband digital signal be represented by

$$s(t) = \sum_{n=-\infty}^{\infty} a_n g(t - nT) \tag{1}$$

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 burst of carrier 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 (1), we first truncate the signal to get

$$s_J(t) = \sum_{i=-J}^{J} a_i g(t - iT) \tag{2}$$

Next, we assume it is not random and take the CTFT of both sides of (2), the spectrum of this truncated signal is found as follows, where $s_J(t) \longleftrightarrow S_J(f)$ and $g(t) \longleftrightarrow G(f)$

$$S_J(f) = G(f) \sum_{i=-J}^{J} a_i e^{-j\omega iT}$$

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 (1) is obtained by taking the statistical average and time limit of $|S_J(f)|^2$ as follows:

$$\begin{aligned}
\Psi_s(f) &= \lim_{J\to\infty} \frac{1}{(2J+1)T} E\left\{|S_J(f)|^2\right\} = \lim_{J\to\infty} \frac{|G(f)|^2}{(2J+1)T} E\left\{\left|\sum_{i=-J}^{J} a_i e^{-j\omega iT}\right|^2\right\} \\
&= \lim_{J\to\infty} \frac{|G(f)|^2}{(2J+1)T} E\left\{\left(\sum_{i=-J}^{J} a_i e^{-j\omega iT}\right)\left(\sum_{m=-J}^{J} a_m e^{-j\omega mT}\right)^*\right\} \\
&= |G(f)|^2 \lim_{J\to\infty} \frac{1}{(2J+1)T} \sum_{i=-J}^{J} \sum_{m=-J}^{J} E\{a_i a_m\} e^{j(m-i)\omega T} \\
&= |G(f)|^2 \lim_{J\to\infty} \frac{1}{(2J+1)T} \sum_{i=-J}^{J} \sum_{\ell=i+J}^{i-J} R(\ell) e^{-j\ell\omega T}
\end{aligned}$$

where $\ell := i - m$ and $R(\ell) := E\{a_i a_m\} = E\{a_i a_{i-\ell}\}$ 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 i when $J \to \infty$, we obtain

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

Equivalently this is

$$\Psi_s(f) = \frac{|G(f)|^2}{T} \sum_{\ell=-\infty}^{\infty} R(\ell) e^{-j\ell\omega T} \tag{3}$$

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

10.1 Case 1: 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 $E\{a_n\} = m_a$ and a variance of σ_a^2, then

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

Substitute this for $R(\ell)$ in (3), we have

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

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

$$\Psi_s(f) = \frac{|G(f)|^2}{T}\left(\sigma_a^2 + \frac{m_a^2}{T}\sum_{\ell=-\infty}^{\infty}\delta\left(f - \frac{\ell}{T}\right)\right) \tag{4}$$

That is, for uncorrelated data

$$\Psi_s(f) = \underbrace{\frac{\sigma_a^2|G(f)|^2}{T}}_{\text{continuous spectrum}} + \underbrace{\left(\frac{m_a}{T}\right)^2 \sum_{\ell=-\infty}^{\infty}\left|G\left(\frac{\ell}{T}\right)\right|^2 \delta\left(f - \frac{\ell}{T}\right)}_{\text{discrete spectrum}} \tag{5}$$

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 $(m_a/T)^2\,|G(k/T)|^2$.

10.1.1 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

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

Now refer to (5), note that $m_a = 0$ and $\sigma_a^2 = A^2$, the second term becomes 0 and the PSD of the signal is (assuming $A = 1$)

$$\Psi_s(f) = \frac{|G(f)|^2}{T} \tag{6}$$

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 (6). its PSD is

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

10.2 Case 2: 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 $\{a_i\}$ takes on three values with the following probabilities:

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

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 $\{a_i\}$ 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

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

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

$$R(k)\Big|_{k>1} = E\{a_i a_{i-k}\} = \frac{1}{8}(A^2) + \frac{1}{8}(-A^2) = 0$$

Thus, for $k > 1$, a_i and a_{i-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:

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

11 PSD Using DFT/FFT

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

$$s_{J,\delta}(t) = s_J(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} s_J(nT_s)\delta(t - nT_s) \tag{7}$$

The CT spectrum for the sampled signal is:

$$\text{CTFT}\{s_{J,\delta}(t)\} = S_{J,\delta}(f) = S_J(f) * \frac{1}{T_s}\sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{T_s}\right) = \frac{1}{T_s}\sum_{k=-\infty}^{\infty} S_J\left(f - \frac{k}{T_s}\right)$$

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 $\sum_{n=-\infty}^{\infty} s_J(nT_s)\delta(t - nT_s)$ separately, we obtain

$$S_{J,\delta}(f) = \sum_{n=-\infty}^{\infty} s_J(nT_s) e^{-j2\pi f n T_s}$$

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

$$S_{J,\delta}(\Omega) = \sum_{n=-\infty}^{\infty} s_J(nT_s)e^{-j\Omega n}$$

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 $s_J(nT_s)$ is obtained as follows:

$$S_{J,\delta}[k] = \sum_{n=0}^{N-1} s_J[n]e^{-j(2\pi k/N)n} = \text{FFT}\{s_J[n]\}$$

where $s_J[n] := s_J(nT_s)$, and $N = (2J+1)M$. So, the PSD may be computed numerically using the FFT function of MATLAB. Since, the signal $s_J(nT_s)$ contains random pulse amplitudes, this is necessary to realize $s_J(nT_s)$ 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, $\Psi_s[k]$, is expressed as:

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

where $s_J^{\ell}(nT_s)$ 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 $s_J(nT_s)$ is $N = (2J+1)M$.

12 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;
sNt = [];
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
```

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

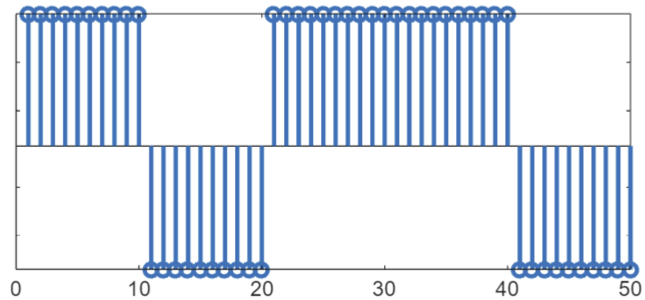

Figure 2: The waveform $s_J(t)$ in discrete-time for $K = 2$.

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
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; plot(PSD)
```

The PSD $\Psi_s[k]$ using DFT/FFT is illustrated in Fig.3(a), where the frequency discrete index k goes from 0 to $M(2J+1)-1$. If we want to obtain $\Psi_s(f)$ from $\Psi_s[k]$, 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 $\Psi_s[k]$ 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, $\Psi_s(\Omega)$, for the frequency interval $0 \leq \Omega \leq 2\pi$ may be obtained from numerically computed $\Psi_s[k]$, $0 \leq k \leq (2J+1)M-1$. 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 $-\pi \leq \Omega \leq \pi$.

4. The PSD obtained from the FFT function of MATLAB may be mapped to CTFT-based PSD, $\Psi_s(f)$, for the frequency interval $0 \leq f \leq f_s$, where $f_s = 1/T_s$. If we are interested in a two-sided

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

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

$$S_{J,\delta}(f) = \frac{1}{T_s} S_J(f) \Rightarrow S_J(f) = T_s S_{J,\delta}(f)$$

The PSD in CTFT, $\Psi_s(f)$, depends upon $S_J(f)$ (and not $S_{J,\delta}(f)$). Since, $S_J(f) = T_s S_{J,\delta}(f)$, $\Psi_s(f)$ may be obtained from $\Psi_s[k]$ 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('$\Psi_s[k]$','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('$\Psi_s(\Omega)$','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('$\Psi_s(f)$','FontSize',16,'interpreter','latex');
xlim([-fs/2 fs/2]);title('PSD vs. $Tf$ [Normalized and two-sided]',...
    'FontSize',14,'interpreter','latex');drawnow
```

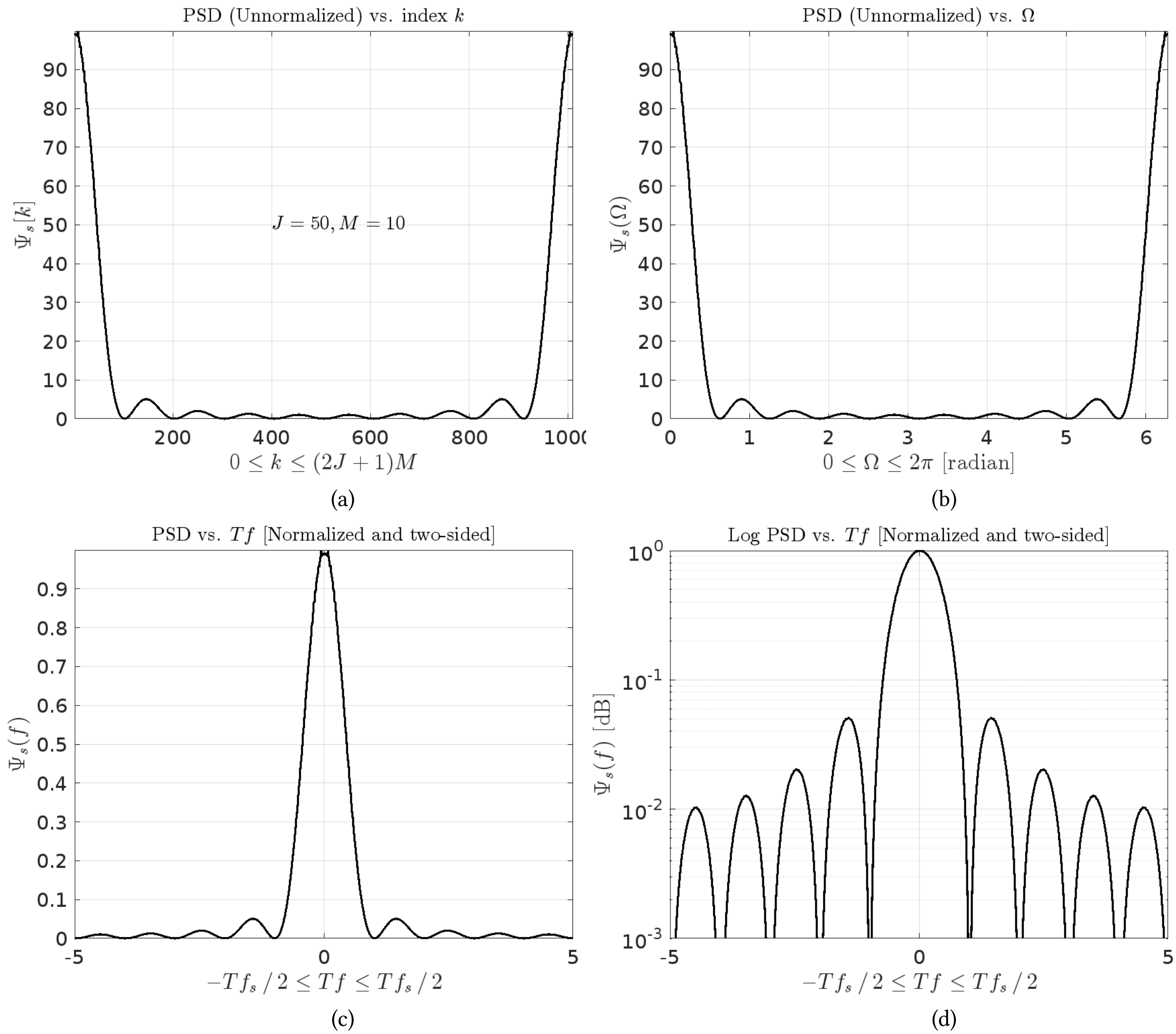

Figure 3: PSD of `Polar-NRZ`.

12.1 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. 3(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('$\Psi_s(f)$ [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
```

13 Exercises

Task 1: Obtain plots of $\Psi_s[k]$, $\Psi_s(\Omega)$, and normalized semilog $\Psi_s(f)$ of `Polar-RZ` code.

Task 2: Obtain plots of $\Psi_s[k]$, $\Psi_s(\Omega)$, and normalized semilog $\Psi_s(f)$ of `Unipolar-NRZ` code.

Task 3: Obtain plots of $\Psi_s[k]$, $\Psi_s(\Omega)$, and normalized semilog $\Psi_s(f)$ of `Unipolar-RZ` code.

Task 4: Obtain plots of $\Psi_s[k]$, $\Psi_s(\Omega)$, and normalized semilog $\Psi_s(f)$ of `Bipolar-NRZ` code.

Task 5: Obtain plots of $\Psi_s[k]$, $\Psi_s(\Omega)$, and normalized semilog $\Psi_s(f)$ of `Bipolar-RZ` code.

Task 6: Compare plots of normalized one-sided semilog $\Psi_s(f)$ of `Bipolar-NRZ` code, `Polar-NRZ` code, and `Unipolar-NRZ` code versus $0 \leq Tf \leq Tf_s/2$.

Task 7: Compare plots of normalized one-sided semilog $\Psi_s(f)$ of `Bipolar-RZ` code, `Polar-RZ` code, and `Unipolar-RZ` code versus $0 \leq Tf \leq Tf_s/2$.

Task 8: Provide a comparison of theoretical and simulated PSD using MATLAB, $\Psi_s(f)$, for `Polar-NRZ`. Repeat this task for `Polar-RZ`.

14 VIVA

1. The unipolar codes contain DC voltage in their PSD. What are the possible drawbacks of DC voltages in the context of data transmission?
2. What benefits does RZ offer over NRZ?
3. What demerits does RZ offer over NRZ?
4. Compared to polar codes, the bipolar codes have spectral null at $f = 0$; is this advantageous? or disadvantageous? briefly explain you opinion.
5. Prove that
$$\sum_{\ell=-\infty}^{\infty} e^{-j\ell\omega T} = \frac{1}{T}\sum_{\ell=-\infty}^{\infty} \delta\left(f - \frac{\ell}{T}\right)$$
6. Explain why it is possible to obtain $\Psi_s(f)$ numerically from $\Psi_s[k]$ multiplied with T_s^2.
7. It is usually believed that both Unipolar and Bipolar codes cause synchronization problems. What do you understand by that?
8. What do you know about Dicode NRZ? Briefly explain how it works.
9. What do you know about Biphase codes? Briefly explain how it works.

References

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

15 Selected Answers:

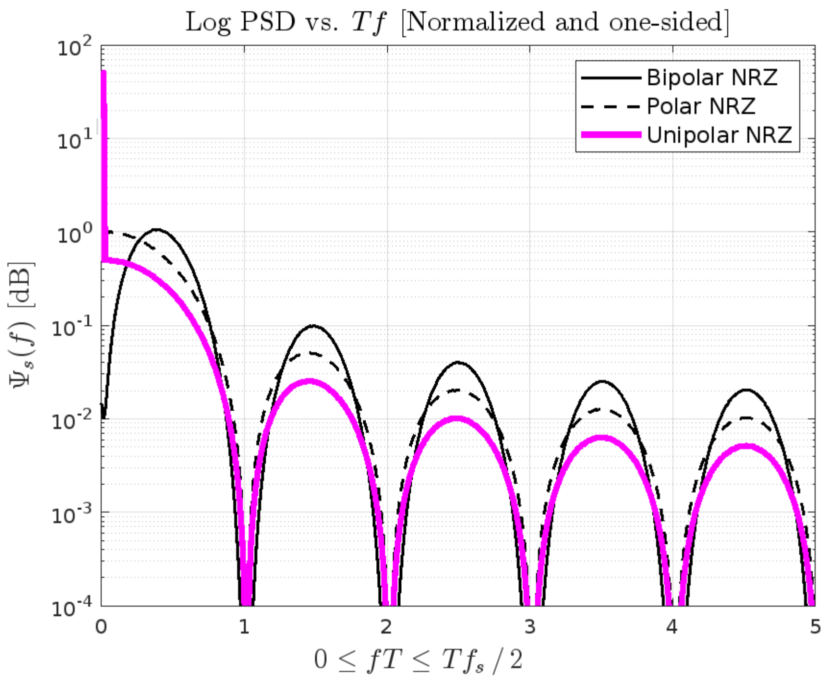

Figure 4: Task 6 plot for NRZ codes.

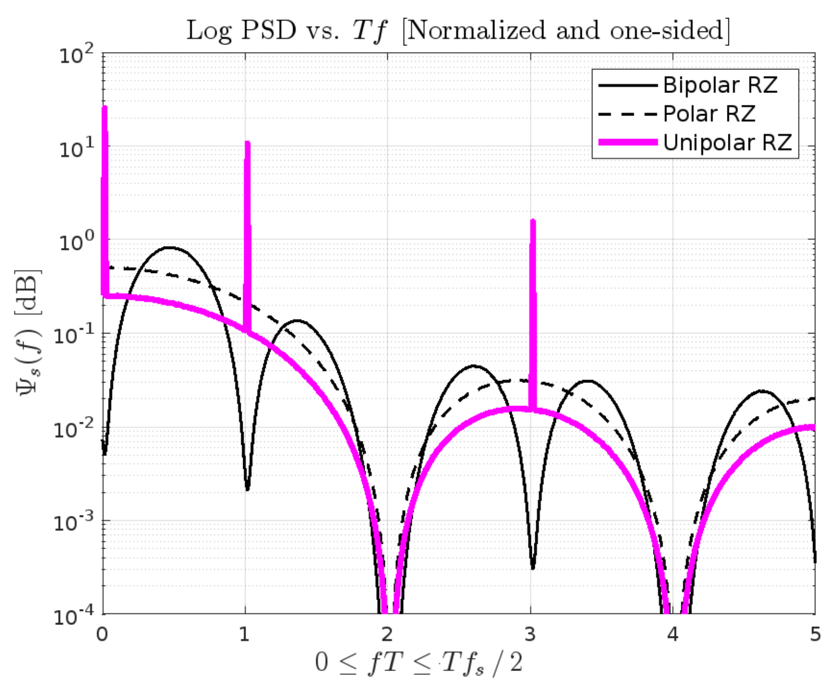

Figure 5: Task 7 plot for RZ codes.