CHAPTER 3

The PSD of TH-UWB Signals

Power Spectral Density (PSD) for TH-UWB signals using PPM is derived in this chapter. The adopted approach (Di Benedetto and Vojcic, 2003) follows the analog PPM theory of the old days — practical pulse communication equipment is described in Black, Beyer, Grieser, and Polkinghorn as early as 1946 (Black et al., 1946) — and reconciles this well-known modulation method with its digital variant currently in vogue in wireless communications.

3.1 BORROWING FROM PPM

The signal of Eq. (2–4) has strong similarities with the output of a PPM modulator in its analog form. Given the modulating signal *m(t)*, the PPM wave *x(t)* in its analog form consists of a train of identically shaped and strictly non-overlapping pulses that are shifted from nominal instances of time T_s by the signal samples $m(kT_s)$. The expression of the PPM analog wave is given as:

$$x_{PPM}(t) = \sum_{j=-\infty}^{+\infty} p\left(t - jT_s - m\left(jT_s\right)\right) = p(t) * \sum_{j=-\infty}^{+\infty} \delta\left(t - jT_s - m\left(jT_s\right)\right) \quad (3\text{–}1)$$

Equation (3–1) is commonly known as the uniform sampling representation. A slightly different version of analog PPM, called natural sampling, is obtained without explicit sampling of the modulating wave *m(t)*. The two forms, however, differ very little if the maximum shift is small compared to the pulse period T_s. We should also note that the idea back in time behind natural sampling was to avoid sampling to simplify the equipment used at that time. This is the last of our worries nowadays. We can, therefore, consider Eq. (3–1) as a valid expression for the PPM analog wave.

A sufficient condition for the pulses of Eq. (3–1) to be strictly non-overlapping is as follows:

$$p(t) = 0 \quad for \quad |t| \geq \frac{T_s}{2} - |m(t)|_{\max} \Rightarrow |m(t)|_{\max} \leq \frac{T_s}{2} \tag{3–2}$$

If the condition expressed by Eq. (3–2) is verified, then the pulses are non-overlapping and the order of the pulses unaltered. For special modulating signals, for example, a sine wave at a frequency much lower than $1/T_s$, weaker conditions than Eq. (3–2) can be found. For the purpose of generality we will suppose that Eq. (3–2) is verified in all the cases we will examine.

The PSD of a PPM signal is difficult to evaluate due to the non-linear nature of PPM modulation. The complete derivation of this spectrum can be found in (Rowe, 1964), and is based on various articles by Bennett published from 1933 to 1947 (Bennett, 1933, 1944, and 1947). We shall report here only the principal results for three particular cases that are of interest for understanding the spectrum of a TH-UWB signal. The three relevant cases are: a) sinusoidal modulating signals; b) generic periodic modulating signals; c) random modulating signals.

3.1.1 Sinusoidal Modulating Signals

Consider the sinusoidal modulating signal *m(t)* at frequency f_0:

$$m(t) = A\cos\left(2\pi f_0 t\right) \tag{3–3}$$

Note that according to the sampling theorem, to be able to reconstruct *m(t)* from the modulated waveform the sampling frequency, $1/T_s$, must be at least equal to $2f_0$. Common practice sets $f_0 << 1/T_s$.

The PPM wave of Eq. (3–1) in this case becomes:

$$x_{PPM}(t) = p(t) * \sum_{j=-\infty}^{+\infty} \delta\left(t - jT_s - A\cos\left(2\pi f_0 jT_s\right)\right) \tag{3–4}$$

If one indicates by *P(f)* the Fourier transform of *p(t)*:

$$P(f) = \int_{-\infty}^{+\infty} p(t) e^{-j2\pi f t} dt \tag{3–5}$$

an expansion of x_{PPM}(t) into sinusoidal components as shown by (Rowe, 1965) can be found:

$$
\begin{aligned}
x_{PPM}(t) &= p(t) * \sum_{j=-\infty}^{+\infty} \delta\left(t - jT_s - A\cos\left(2\pi f_0 jT_s\right)\right) \\
&= \frac{1}{T_s} \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \left(-j\right)^n J_n\left(2\pi A\left(m\frac{1}{T_s} + nf_0\right)\right) \cdot \\
&\cdot P\left(m\frac{1}{T_s} + nf_0\right) e^{j2\pi(m\frac{1}{T_s}+nf_0)t}
\end{aligned} \tag{3–6}
$$

where $J_n(\cdot)$ are the Bessel functions of the first kind. Properties and curves for the Bessel functions can be found in many communication systems books since these functions appear in the computation of the spectrum of angle modulated signals (see, for example, Proakis and Salehi, 1994). The general expression and a few of the properties of the Bessel functions should be recalled here:

$$
\begin{aligned}
&J_n(x) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} e^{jx\sin\psi} e^{-jn\psi} d\psi \\
&J_{-n}(x) = (-1)^n J_n(x), \quad J_n(-x) = (-1)^n J_n(x) \\
&\text{and} \\
&J_n(x) \cong 0 \quad \mathit{for} \quad |n| > |x|
\end{aligned} \tag{3–7}
$$

From Eq. (3–6), one can derive the PSD of $x_{PPM}(t)$:

$$
\begin{aligned}
P_{x_{PPM}}(f) = \frac{1}{T_s^2} \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \left| J_n\left(2\pi A\left(m\frac{1}{T_s} + nf_0\right)\right)\right|^2 \cdot \\
\cdot \left| P\left(m\frac{1}{T_s} + nf_0\right)\right|^2 \delta\left(f - \left(m\frac{1}{T_s} + nf_0\right)\right)
\end{aligned} \tag{3–8}
$$

Equation (3–8) shows that, for sinusoidal modulating signals, the PPM signal has discrete frequency components located at the sinewave and pulse repetition frequencies and their harmonics, and at their sum and difference frequencies and harmonics. The amplitude of the pulses in frequency is governed by two terms, $P(f)$ and $J_n(x)$. We will now analyze the effect of each of these two terms separately.

Effect of the $|P(f)|^2$ Term

The values assumed by $|P(f)|^2$ in correspondence to the frequencies of the Dirac pulses contribute to determine the amplitudes of the Dirac pulses. If $P(f)$ has a limited bandwidth, the bandwidth of the PPM signal is limited as well.

Effect of the $J_n(x)$ Term

First, observe that for $n = 0$, $J_n(x) = 1$ if $x = 0$. Since $x = 2\pi Am/T_s$, one has $x = 0$ when $m = 0$. Recall that since m is the index referring to the harmonics of $1/T_s$ for all $m \neq 0$, discrete frequency components at frequencies m/T_s are present with amplitude given by $|J_n(2\pi Am/T_s)|^2$. Since m spans the infinite interval of summation, we can therefore assert that the term $J_n(x)$ does not introduce a limitation in the bandwidth of the signal of Eq. (3–8). However, it regulates the presence or absence of Dirac pulses located at multiples of f_0 and of linear combinations of f_0 and $1/T_s$.

From Eq. (3–7) we know that $J_n(x)$ approaches zero for $|n|>|x|$. Since $J_{-n}(x) = (-1)^n J_n(x)$ then $|J_{-n}(x)|^2 = |J_n(x)|^2$; therefore, we can limit the analysis to the case, $J_n(x) \geq 0$. In this case one has:

$$\begin{aligned} &J_n(x) \cong 0 \quad for \quad |n| > |x| \\ &i.e., for\, n\, positive, n > \left| 2\pi A\left(m\frac{1}{T_s} + nf_0 \right) \right| \end{aligned} \tag{3–9}$$

We now suppose that $m > 0$ since the case $m < 0$ can be obtained by symmetry. In this case, note that the condition of Eq. (3–9) becomes $n > 2\pi A(m/T_s + nf_0)$, which implies:

$$n > \left(\frac{2\pi A / T_s}{1 - 2\pi A f_0} \right) m \tag{3–10}$$

Observe that $A < T_s/2$ and put $A = \beta T_s/2$. Equation (3–10) becomes:

$$n > \left(\frac{\pi\beta}{1 - \pi\beta T_s f_0} \right) m \tag{3–11}$$

Note that since $f_0 << 1/T_s$, the quantity $(1-\pi f_0 T_s)$ approaches 1 and the condition of Eq. (3–11) becomes:

$$n > \pi\beta m \tag{3–12}$$

CHECKPOINT 3–1

In this checkpoint we will use computer simulation to analyze analog PPM in the case of a sinusoidal modulating signal.

Two MATLAB functions are introduced. The first, Function 3.1, generates a train of rectangular pulses that are analog PPM modulated by a sinusoidal wave. The second, Function 3.2, evaluates the spectrum of the signal generated by Function 3.1. We choose a rectangular waveform for the transmitted pulses to simplify the analysis in the frequency domain. The Fourier transform of a rectangular waveform has the well-known sin(x)/x shape.

Function 3.1 (see Appendix 3.A) generates a PPM-UWB signal in the case of a sinusoidal modulating signal and rectangular pulses. Within the function, the user must set the following parameters: the average transmitted power in dBm `Pow`, the sampling frequency for representing the signal `fc`, the number of pulses to be generated `np`, the time duration of each rectangular pulse `Tr`, the average pulse repetition period in seconds `Ts`, the amplitude and frequency of the sinusoidal modulating signal `A` and `f0`. Function 3.1 returns two outputs: the generated train of pulses `Stx`, and the corresponding sampling frequency `fc`. The command line for generating the signal is:

```
[Stx,fc]=cp0301_PPM_sin;
```

We will use Function 3.1 to generate two signals. The first, signal `S1`, represents the output of the transmitter in the absence of modulation. The second, signal `S2`, represents the output of the transmitter when the PPM scheme is applied in the special case of a sinusoidal modulating signal.

With reference to signal `S1`, we set the following parameters within Function 3.1: `Pow=-30; fc=1e11; np=10000; Tr=0.5e-9; Ts=2e-9; A=0; f0=0`. The output signal is composed of 10,000 equally spaced rectangular pulses. The average pulse repetition period is 2 ns, that is, four times the length of each pulse. The command line for generating signal `S1` is:

```
[S1,fc]=cp0301_PPM_sin;
```

In the case of signal `S2`, the PPM block is introduced within the transmission chain. The following parameters characterize the generated waveform: `Pow=-30; fc=1e11; np=10000; Tr=0.5e-9; Ts=2e-9; A=1e-9; f0=5e7`. Note that the amplitude of the modulating signal `A` is half the value of the average pulse repetition period `Ts`, while the frequency `f0` is ten times smaller than the average pulse repetition frequency `1/Ts`. The command line for generating signal `S2` is:

```
[S2,fc]=cp0301_PPM_sin;
```

To better understand the effect of PPM on the PSD of UWB signals, we compare `S1` and `S2` in the frequency domain. This comparison is carried out by Function 3.2.

In **Function 3.2** (see Appendix 3.A), the PSD of an input signal `x(t)` is computed by dividing the ESD of `x(t)` derived in Checkpoint 1–1 by the length `T` of the time window in which `x(t)` is represented. Function 3.1 receives in input vector `x` representing the signal in the time domain, and the value of the sampling frequency `fc`. Function 3.1 returns two outputs: vector `PSD` containing the PSD of the input signal and the value `df` of frequency separation between the samples of the PSD. This value is useful to derive the amount of power P which is associated with the input signal from the PSD,

```
P = sum(PSD.*df)
```

For a given signal `x(t)`, one can thus verify the exactness of the PSD provided by Function 3.2 by comparing the above `P` value with the one that can be evaluated in the time domain:

```
P = (1/T) * sum((x.^2).*(1/fc))
```

The command line for evaluating the PSD of signal `S1` is the following;

```
[PSD1,df] = cp0301_PSD(S1,fc);
```

The above command line stores vector `PSD1` in memory and produces the graphical output shown in Figures 3–1 and 3–2.

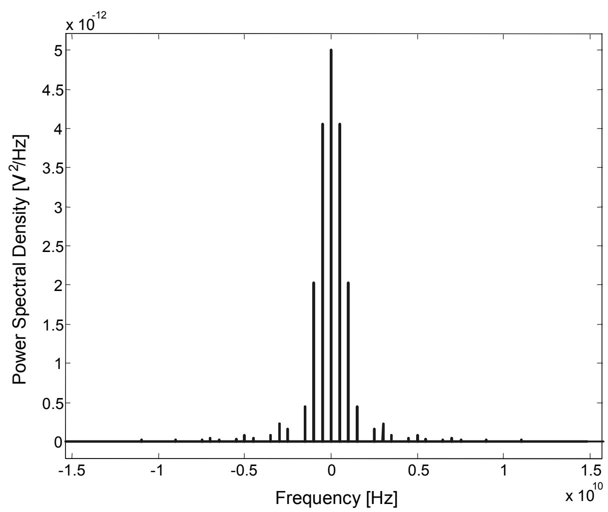

Figure 3–1 PSD of a train of equally spaced rectangular pulses (signal S1), in the absence of modulation.

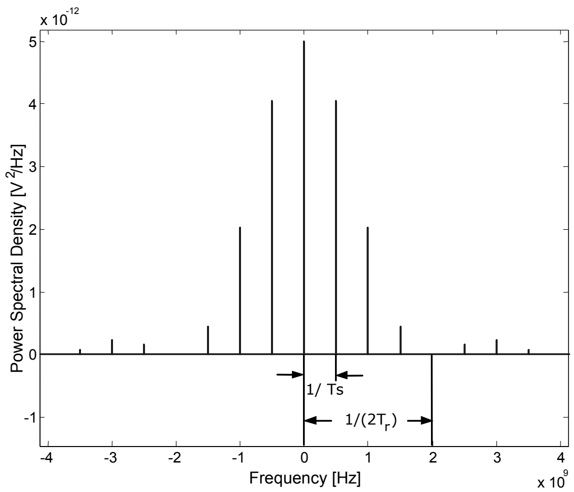

Figure 3–2 Detail of Figure 3–1: PSD of a train of equally spaced rectangular pulses (signal S1).

As expected, the PSD of signal `S1` is characterized by the presence of equally spaced spectral lines. This result is due to the absence of modulation in the time domain. The output of the transmitter is a periodic signal with its period equal to the average pulse repetition period T_s. As a consequence, the Fourier transform of `S1` is non-zero only for those frequencies that are integer multiples of the average pulse repetition frequency $f_s = 1/T_s$ = 500 MHz. The envelope of the PSD in Figures 3–1 and 3–2 follows the sin(x) / x shape of the Fourier transform of the rectangular pulse. As shown in Figure 3–2, the PSD presents in fact zero values for all frequency multiples of $1/(2T_r)$ = 2 GHz, where `Tr` is the time duration of the rectangular pulse.

We conclude the analysis of signal `S1` with the verification of the amount of transmitted power. When considering `S1` in the time domain, we obtain:

```
P_time=(fc/length(S1))*sum((S1.^2).*(1/fc))
>> P_time = 1.0000e-006
```

where the term `(fc/length(S1))` is the inverse of the time duration of the signal. In the frequency domain, one has:

```
P_freq=sum(PSD1.*df)
>> P_freq = 1.0000e-006
```

As expected, `P_time` and `P_freq` are identical. Moreover, these values confirm the input parameter `Pow` in Function 3.1, which was set equal to -30 dBm.

The spectral analysis of signal `S1` can be repeated in the case of the modulated signal `S2`. The following command must be executed:

```
[PSD2,df] = cp0301_PSD(S2,fc);
```

which stores vector `PSD2` in memory and produces the graphical output of Figures 3–3 and 3–4:

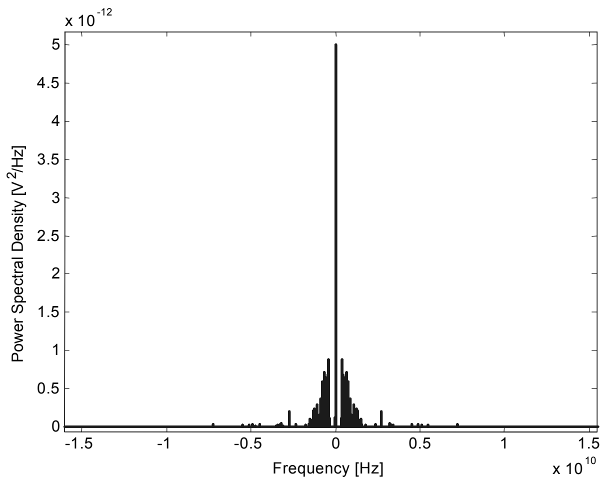

Figure 3–3 PSD of a train of rectangular pulses that is PPM-modulated by a sinusoidal modulating signal (signal S2).

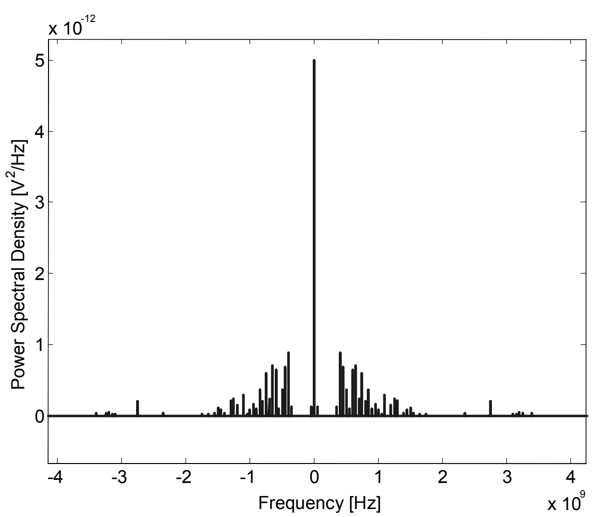

Figure 3–4 Detail of Figure 3–3: PSD of a train of PPM-modulated rectangular pulses in the case of a sinusoidal modulating signal (signal S2).

Figures 3–3 and 3–4 represent the `PSD` of signal `S2` in the same frequency range of Figures 3–1 and 3–2. We can observe that the presence of the sinusoidal modulating signal has the effect of altering the sin(x)/x shape of the signal spectrum. In addition, a higher number of spectral lines is present in the PSD than in the preceding case. According to Eq. (3–8), these new lines are located at frequencies $f(m,n) = (m/T_s+nf_0)$, that is, at the sum and difference frequencies of the modulating signal frequency f_0 and the pulse repetition frequency $1/T_s$ and their harmonics. The amplitude of the spectral line at frequency f(m,n) depends on two terms: the modulus of the Fourier transform of the rectangular pulse evaluated in f(m,n) and the modulus of the Bessel function $J_n(x)$ of order n and argument $x = (2\pi Af(m,n))$. In the case under examination, that is, $A = T_s/2$ and $f_0 = 1/(10T_s)$, the spectral lines are equally spaced with frequency separation $df = 0.1/T_s$. In other words, the spectral lines of `PSD2` are ten times closer compared to the PSD of signal `S1`. Each spectral line has an amplitude that depends on the Fourier transform of the pulse waveform and on the modulus of the following function:

$$J_n\left(2\pi\frac{T_S}{2}\left(\frac{m}{T_S}+\frac{n}{10T_S}\right)\right)=J_n\left(m\pi+\frac{n\pi}{10}\right) \quad (3\text{–}13)$$

According to Eq. (3–7), the amplitude of the Bessel function of order n and argument x tends to zero when $|n| > |x|$. As a consequence, we derive from Eq. (3–13) that the PSD of signal `S2` is composed of "clusters" of spectral lines, each cluster being located in correspondence of a multiple of the average pulse repetition frequency $f_s = 1/T_s$. This result is shown in Figure 3–5, which compares the PSD of signals `S1` and `S2`.

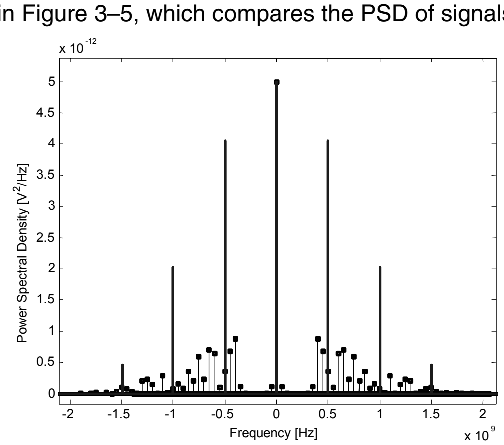

Figure 3–5 Comparison of the PSD of signals S1 (solid lines) and S2 (dashed lines).

The analysis of each cluster of spectral lines can be performed by introducing the MATLAB function `besselj(nu,z)`, which computes the value of the Bessel function of the first kind with order `nu` and argument `z`.

We start by considering the cluster of spectral lines located around frequency zero. According to Eqs. (3–8) and (3–13), this cluster consists of spectral lines at frequencies f(n) =

$n/(10T_s)$. For the single cluster, we may neglect the shaping effect of the Fourier transform of the pulse waveform. The amplitude of the spectral line at frequency f(n) is therefore proportional to:

$$A_0(n) = \left| J_n\left(\frac{n\pi}{10}\right) \right|^2 \quad (3–14)$$

Equation (3–14) can be visualized by executing the following MATLAB code:

```
n=(-20:1:20);
A0=abs(besselj(n,(pi/10).*n)).^2;
figure(1)
stem(n,A0)
```

The above code stores vector `A0` in memory and generates the plot of Figure 3–6, in which we can recognize the cluster of spectral lines located at the center of the PSD of Figure 3–5.

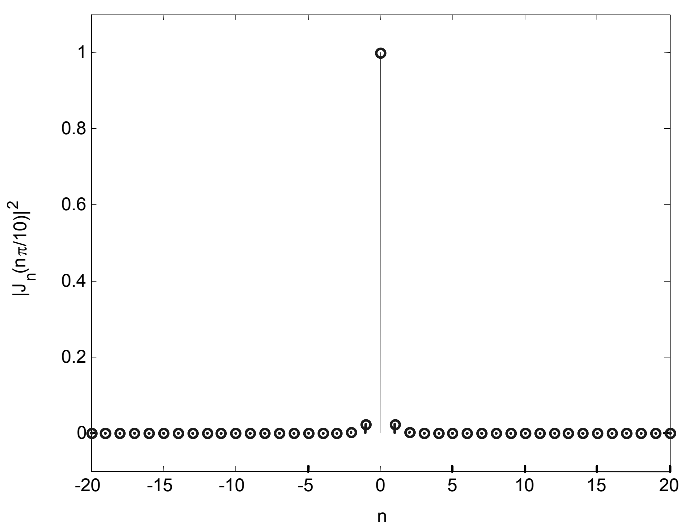

Figure 3–6 Cluster of spectral lines located around frequency zero — $|J_n(n\pi/10)|^2$ for different values of n.

The pair of clusters located around $\pm 1/T_s = \pm 500$ MHz give:

$$A_{+1}(n) = \left| J_n\left(\pi + \frac{n\pi}{10}\right) \right|^2 \quad (3\text{–}15)$$

$$A_{-1}(n) = \left| J_n\left(-\pi + \frac{n\pi}{10}\right) \right|^2 \quad (3\text{–}16)$$

where $A_{+1}(n)$ and $A_{-1}(n)$ represent approximated amplitude values of the spectral lines located around $1/T_s$ and $-1/T_s$, respectively. The code for evaluating the expressions in Eqs. (3–15) and (3–16) is:

```
n=(-20:1:20);
Ap1=abs(besselj(n,pi+(pi/10).*n)).^2;
Am1=abs(besselj(n,((pi/10).*n)-pi)).^2;
figure(2)
stem(n,Ap1)
figure(3)
stem(n,Am1)
```

The graphical output generated by the above code is shown in Figures 3–7 and 3–8.

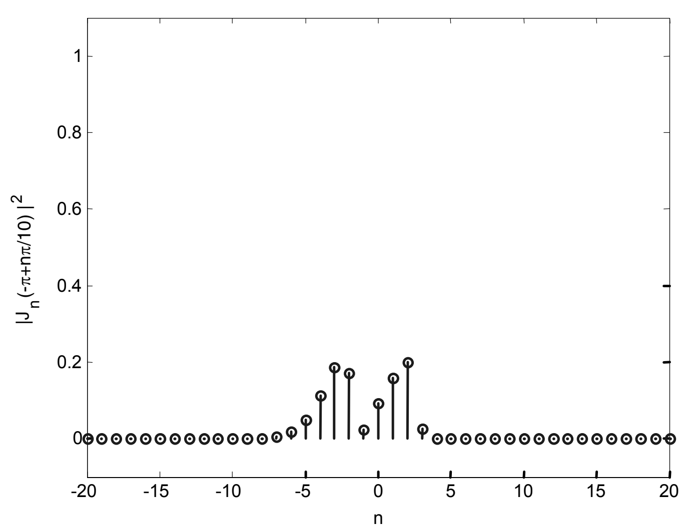

Figure 3–7 Cluster of spectral lines located around frequency -500 MHz — $|J_n(-\pi+n\pi/10)|^2$ for different values of n.

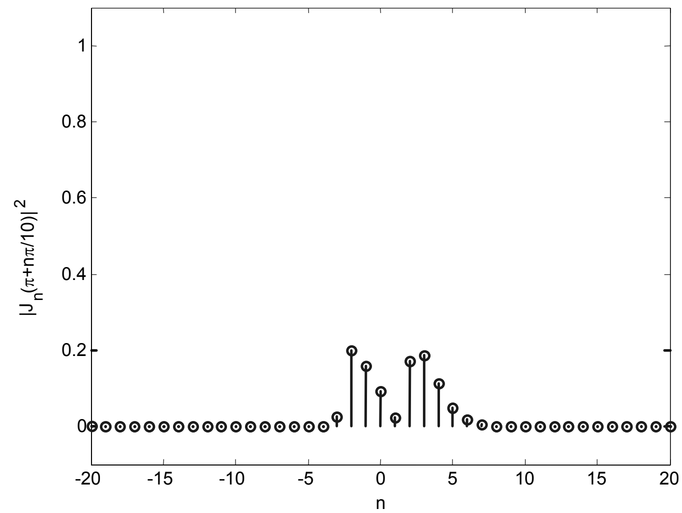

Figure 3–8 Cluster of spectral lines located around frequency +500 MHz — $|J_n(\pi+n\pi/10)|^2$ for different values of n.

Figure 3–7 represents the cluster of spectral lines located around frequency $-1/T_s = -500$ MHz, while Figure 3–8 represents the cluster of spectral lines located around frequency 1/Ts = 500 MHz. When comparing the plots in Figures 3–7 and 3–8 with the PSD in Figure 3–5, we notice the agreement between theoretical analysis and simulation results. In addition, if we compare the plots in Figures 3–7 and 3–8 vs. Figure 3–6 , we observe that the clusters at $\pm 1/T_s$ are composed of more spectral lines with respect to the cluster at the center of the PSD. The same result is verified when analyzing the pair of clusters located at $\pm 2/T_s = \pm 1$ GHz. In this case, the following expressions must be considered:

$$A_{+2}(n) = \left| J_n\left(2\pi + \frac{n\pi}{10}\right)\right|^2 \qquad (3\text{–}17)$$

$$A_{-2}(n) = \left| J_n\left(-2\pi + \frac{n\pi}{10}\right)\right|^2 \qquad (3\text{–}18)$$

which can be reproduced with the following code lines:

```
n=(-20:1:20);
Ap2=abs(besselj(n,2*pi+(pi/10).*n)).^2;
```

```
Am2=abs(besselj(n,((pi/10).*n)-2*pi)).^2;
figure(4)
stem(n,Ap2)
figure(5)
stem(n,Am2)
```

The graphical output generated by the above code is shown in Figures 3–9 and 3–10.

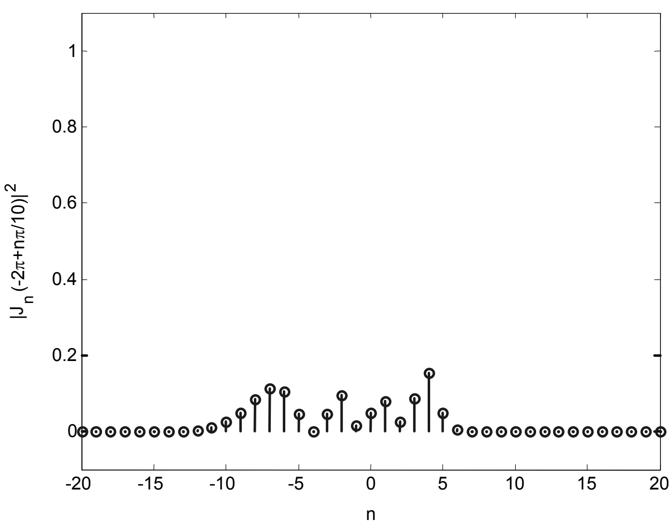

Figure 3–9 Cluster of spectral lines located around frequency -1GHz — $|J_n(-2\pi+n\pi/10)|^2$ for different values of n.

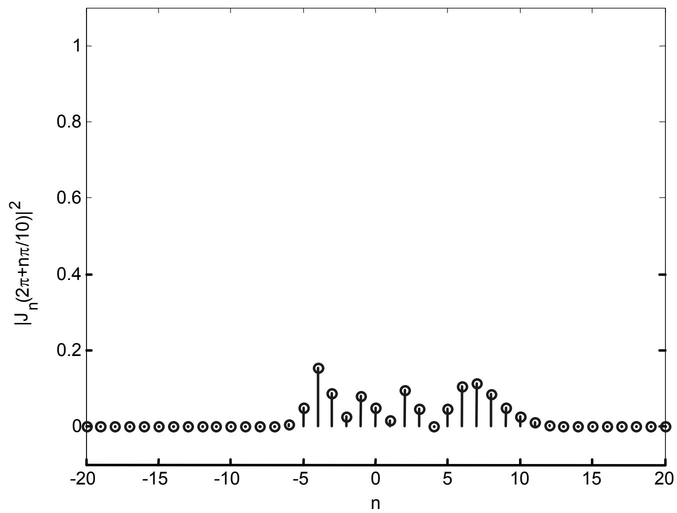

Figure 3–10 Cluster of spectral lines located around frequency +1GHz — $|J_n(2\pi+n\pi/10)|^2$ for different values of n.

Figure 3–9 represents the cluster of spectral lines located around frequency $-2/T_s = -1$ GHz, while Figure 3–10 represents the cluster of spectral lines located around frequency $2/T_s = 1$ GHz. Once again, we observe that the number of spectral lines composing these clusters is increased with respect to the previous cases, that is, the number of spectral lines composing one cluster increases when the central frequency of the cluster increases.

CHECKPOINT 3–1

3.1.2 Generic Periodic Modulating Signals

When $m(t)$ is periodic of period T_p, that is, $m(t+T_p) = m(t)$ for all t, the Fourier series representation is valid for all t and is expressed as follows:

$$m(t) = \sum_{n=-\infty}^{+\infty} m_n e^{jn2\pi t/T_p} \tag{3–19}$$

where m_n is the n-th Fourier coefficient given by:

$$m_n = \frac{1}{T_p} \int_{\alpha}^{\alpha+T_p} m(t) e^{-jn2\pi t/T_p} dt \tag{3–20}$$

If $m(t)$ is real, then one obtains:

$$m_n = m_{-n}^{*} \tag{3–21}$$

The Fourier series expansion shows that a periodic signal $m(t)$ can be represented for all t as a sum of components of different frequencies, all multiples of the fundamental frequency $1/T_p$. The n-th term of the summation corresponds to frequency n/T_p.

The analysis of sinusoidal modulating waves can be expanded to periodic modulating signals by observing that in this last case, the modulating signal is composed of the sum of sinusoidal waves at frequencies that are multiples of the fundamental. Note that in this case, the condition $A < T_s/2$ becomes:

$$\sum_{n=-\infty}^{+\infty} m_n < \frac{T_s}{2} \tag{3–22}$$

We put:

$$M = \sum_{n=-\infty}^{+\infty} m_n \tag{3–23}$$

By applying the multiple Fourier series method as proposed by (Bennett, 1933, 1944, and 1947) and as further suggested by (Rowe, 1965), we find:

$$\begin{aligned} x_{PPM}(t) = \frac{1}{T_s} \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} (-j)^n J_n\left(2\pi M\left(m\frac{1}{T_s} + nl\frac{1}{T_p}\right)\right)\cdot \\ \cdot P\left(m\frac{1}{T_s} + nl\frac{1}{T_p}\right) e^{j2\pi\left(m\frac{1}{T_s}+nl\frac{1}{T_p}\right)t} \end{aligned} \tag{3–24}$$

Equation (3–24) easily leads to the PSD of a periodic modulating PPM signal $x_{PPM}(t)$ which is expressed by:

$$\begin{aligned} P_{x_{PPM}}(f) = \frac{1}{T_s^2} \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} \left| J_n\left(2\pi M\left(m\frac{1}{T_s} + nl\frac{1}{T_p}\right)\right)\right|^2 \cdot \\ \cdot \left| P\left(m\frac{1}{T_s} + nl\frac{1}{T_p}\right)\right|^2 \delta\left(f - \left(m\frac{1}{T_s} + nl\frac{1}{T_p}\right)\right) \end{aligned} \tag{3–25}$$

Similar to the sinusoidal case, when the modulating signal is a periodic waveform, the PPM signal contains discrete frequency components located at the fundamental and pulse repetition frequencies and their harmonics, as well as at the sum and difference frequencies of the modulating signal and the pulse repetition frequency and harmonics. The amplitude of the pulses in frequency is governed by two terms: *P(f)* and $J_n(x)$. The analysis of the effect of these two terms closely follows the analysis of the sinusoidal case.

CHECKPOINT 3–2

In this checkpoint, we will use computer simulation to analyze the spectral occupation of an UWB signal implementing PPM in the case of a periodic modulating signal. The modulating signal m(t) is chosen to have a negative exponential amplitude decay within a period T_p:

$$m(t) = \sum_{k=-\infty}^{+\infty} A e^{B(kT_p - t)} rect_{T_p}\left(t - \frac{3}{2}kT_p\right) \quad (3\text{–}26)$$

where A and B are two real constant terms.

Assume, for example, T_p = 20 ns, A = $1 \cdot 10^{-9}$ V, and B = 10. We can generate the waveform in Eq. (3–26) by using the following MATLAB code:

```
A = 1e-9;
B = 10;
Tp = 20e-9;
fc = 1e11;
dt = 1 / fc;
T = 1e-6;
time = (0:dt:T);
m = A.*exp(-(B/Tp).*mod(time,Tp));
plot(time,m);
```

The above code stores in memory vector `m`, which contains the samples of signal m(t). Figure 3–11 represents m(t) in the time domain.

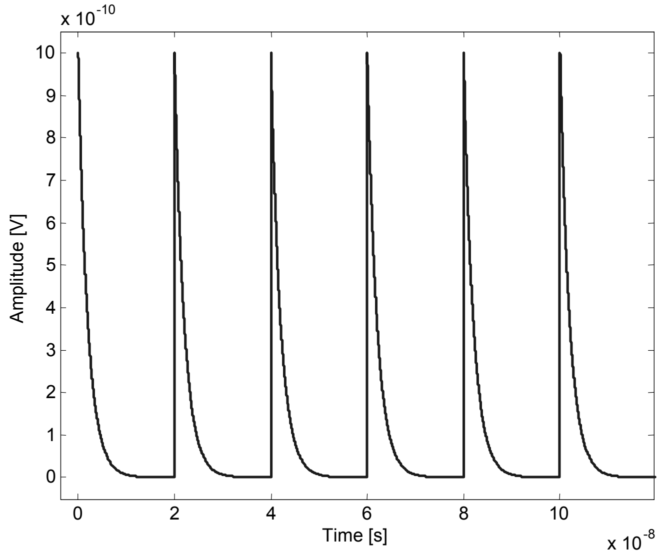

Figure 3–11 Periodic modulating signal m(t).

Given vector `m`, we can isolate a single period of the periodic signal. The following code line extracts the first period of m(t) and stores it in vector `x`:

```
x = m(1:floor(Tp/dt));
```

Given vector `x`, we can use the MATLAB function `fft(x)` to evaluate the coefficients of the Fourier series representing the periodic signal m(t):

```
X = fftshift((1/length(x)).*fft(x));
```

Figure 3–12 shows the modulus of the coefficients of the Fourier series of m(t).

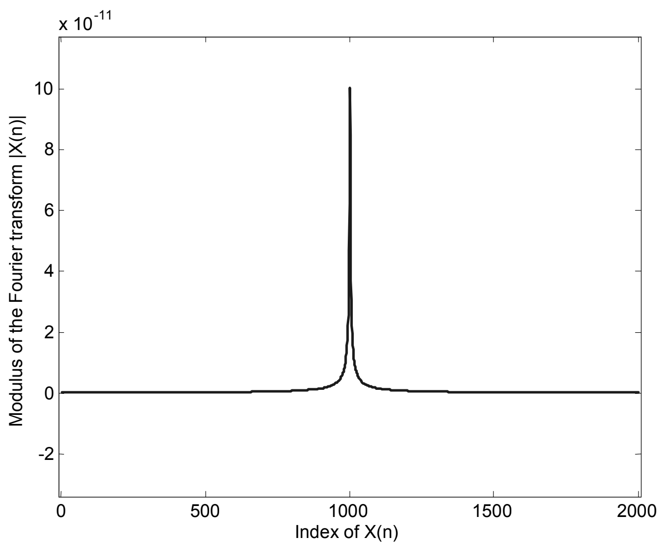

Figure 3–12 Modulus of the coefficients of the Fourier series of m(t).

Finally, we can evaluate the value in Eq. (3–23) as follows:

```
M = real(sum(X))
>> M = 1.0000e-009
```

In the above command line, the MATLAB function `real()` is necessary to take into account the approximation errors due to the sampling of the original waveform m(t). In the case of no approximation errors, the summation in Eq. (3–23) is always real if m(t) is real.

To analyze the spectral characteristics of PPM-UWB signals with periodic modulating signals, we must introduce a new MATLAB function.

Function 3.3 (see Appendix 3.A) generates a train of rectangular pulses that are modulated in position by the signal m(t) in Eq. (3–26). Within the function, the user must set the following parameters: the average transmitted power in dBm `Pow`, the sampling frequency `fc`, the number of pulses to be generated `np`, the time duration of each rectangular pulse `Tr`, the average pulse repetition period in seconds `Ts`, and finally, parameters `A`, `B`, and `Tp`, which characterize the modulating signal. Function 3.3 returns two outputs: the generated train of pulses `Stx` and the corresponding sampling frequency `fc`. The command line is:

```
[Stx,fc]=cp0302_PPM_periodic;
```

We will use Function 3.3 for generating an UWB signal with a periodic modulating signal m(t) as in Figure 3–11. The following parameters are set within the function: `Pow=-30`; `fc=1e11`; `np=10000`; `Tr=0.5e-9`; `Ts=2e-9`; `A=1e-9`; `B=10`; `Tp=20e-9`. Figure 3–13 represents a portion of the generated signal `Stx` in the time domain.

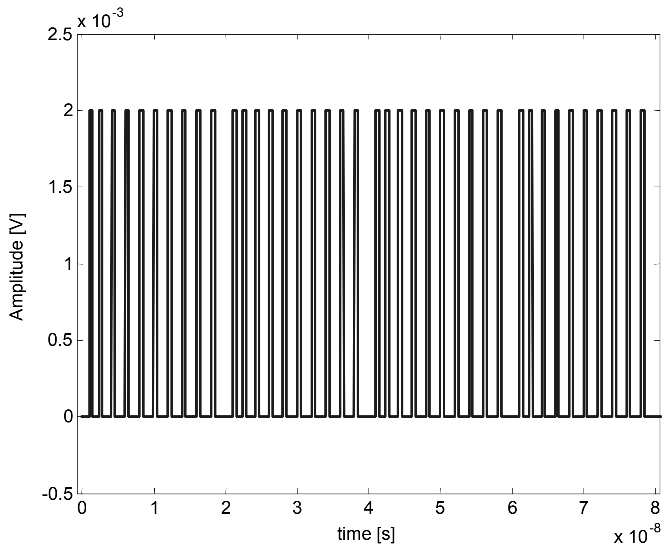

Figure 3–13 A portion of signal Stx generated with Function 3.3. The modulating signal is the periodic waveform represented in Figure 3–11.

The spectral analysis of signal `Stx` can be carried out using Function 3.2. The following command must be executed:

```
[PSD,df] = cp0301_PSD(Stx,fc);
```

which stores vector `PSD` in memory and produces the graphical output in Figures 3–14 and 3–15.

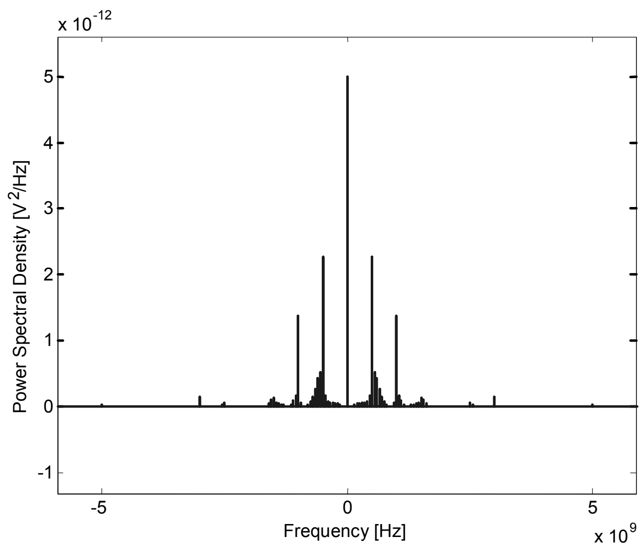

Figure 3–14 PSD of signal Stx (see Figure 3–13).

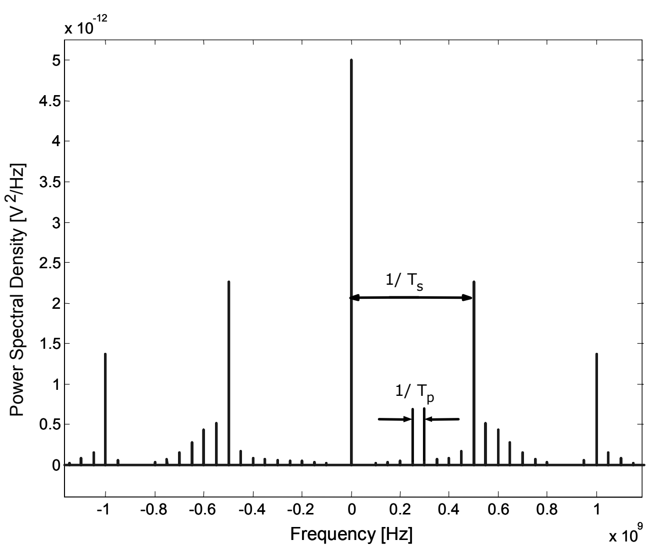

Figure 3–15 Detail of Figure 3–14: PSD of signal Stx.

Figures 3–14 and 3–15 show that the PSD of signal `Stx` is composed of spectral lines located at each integer multiple of the pulse repetition frequency $1/T_s$, and at all the sum and difference frequencies of the modulating signal frequency $1/T_p$ and the pulse repetition frequency $1/T_s$. According to Eq. (3–25), the amplitude of the spectral line at frequency $f_x = (m/T_s + nl/T_p)$ depends on both the value in f_x of the Fourier transform of the rectangular pulse and the value of the Bessel function $J_n(2\pi M f_x)$, where M is the constant term in Eq. (3–23). In the case under examination, we found that $M = 1 \cdot 10^{-9} = T_s/2$. Moreover, since $T_p = 10T_s$, we can simplify the argument of the Bessel function as follows:

$$J_n\left(2\pi\frac{T_S}{2}\left(\frac{m}{T_S}+\frac{nl}{10T_S}\right)\right)=J_n\left(m\pi+\frac{nl\pi}{10}\right) \quad (3\text{–}27)$$

As shown in Checkpoint 3–1, we can use the MATLAB function `besselj(nu,z)` to analyze each cluster of spectral lines separately. The code for generating the cluster located at the zero frequency, for example, is:

```
Jm0=zeros(1,51);
for n = -5 : 5
for l = -5 : 5
i = n * l;
index = i + 26;
Jm0(index)=Jm0(index)+abs(besselj(n,n*l*pi/10))^2;
end
end
abscissa = (-25:1:25);
figure(1)
stem(abscissa,Jm0)
```

The above code lines store vector `Jm0` in memory and produce the plot of Figure 3–16.

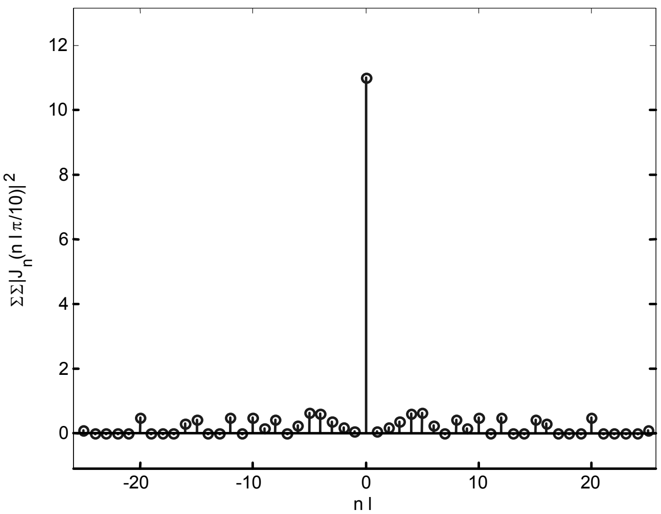

Figure 3–16 Cluster of spectral lines located around frequency zero

The two clusters at $\pm 1/T_s$ can be generated as follows:

```
Jmp1 = zeros(1,51);
Jmn1 = zeros(1,51);
for n = -5 : 5
for l = -5 : 5
i = n * l;
index = i + 26;
Jmp1(index) = Jmp1(index) + abs(besselj(n,pi + ...
    (n*l*pi/10))) ^2;
Jmn1(index) = Jmn1(index) + abs(besselj(n,(n*l*pi/10)...
    - pi)) ^2;
end
end
abscissa = (-25:1:25);
figure(2)
stem(abscissa,Jmn1);
figure(3)
stem(abscissa,Jmp1);
```

The graphical output resulting from the above code lines is shown in Figures 3–17 and 3–18.

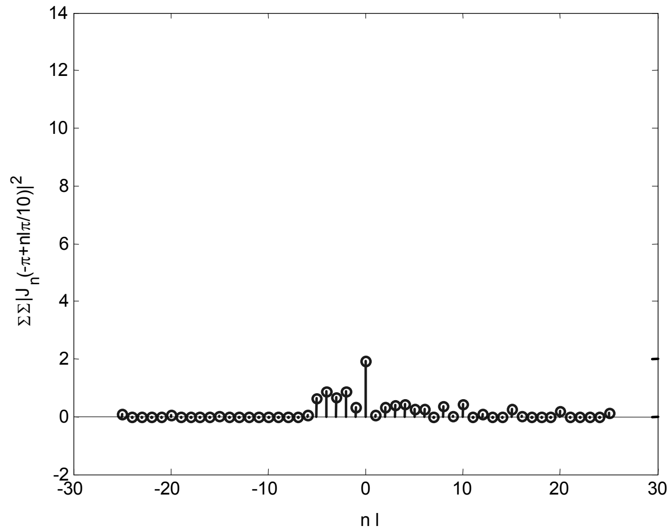

Figure 3–17 Cluster of spectral lines located around frequency -500 MHz

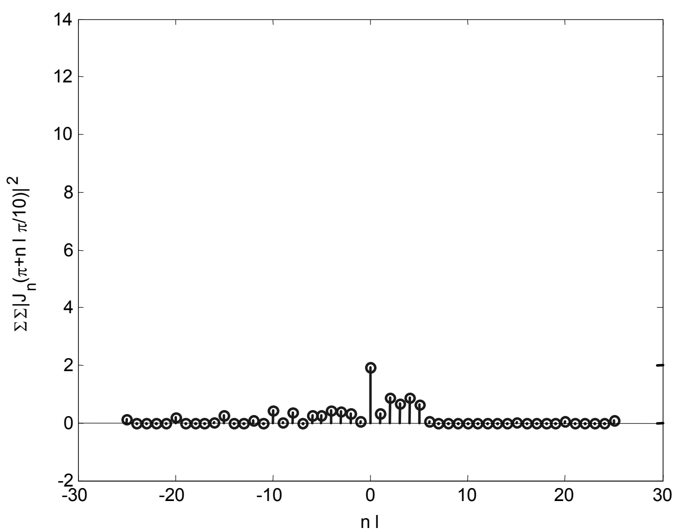

Figure 3–18 Cluster of spectral lines located around frequency +500 MHz

Finally, we can generate the clusters at $\pm 2/T_s$ with the following code lines:

```
Jmp2 = zeros(1,51);
Jmn2 = zeros(1,51);
for n = -5 : 5
for l = -5 : 5
i = n * l;
index = i + 26;
Jmp2(index) = Jmp2(index) + abs(besselj(n,2*pi + ...
   (n*l*pi/10))) ^2;
Jmn2(index) = Jmn2(index) + ...
   abs(besselj(n,(n*l*pi/10)- pi*2)) ^2;
end
end
abscissa=(-25:1:25);
figure(4)
stem(abscissa,Jmn2)
figure(5)
stem(abscissa,Jmp2)
```

The output provided by the above MATLAB code is shown in Figures 3–19 and 3–20.

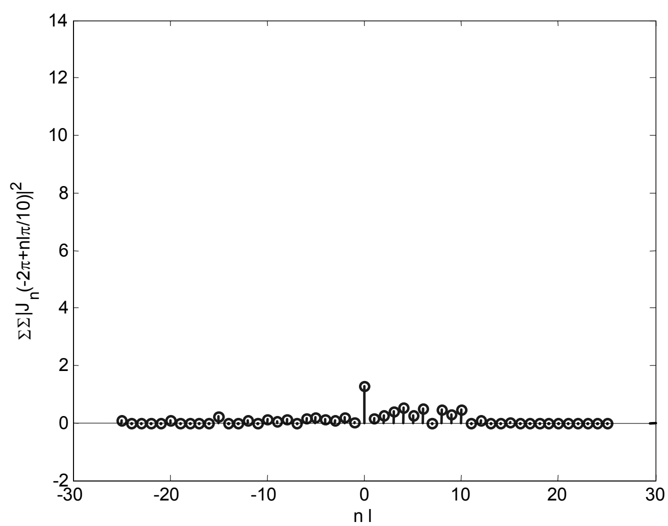

Figure 3–19 Cluster of spectral lines located around frequency -1 GHz

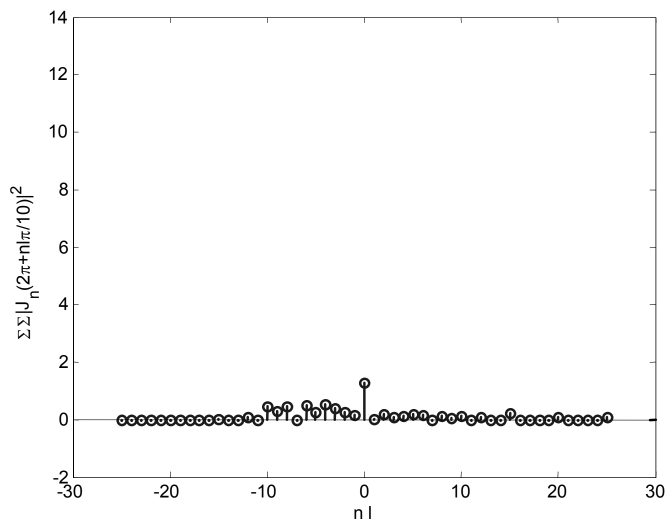

Figure 3–20 Cluster of spectral lines located around frequency +1 GHz

Remember that the clusters represented in Figures 3–16 to 3–20 only take into account the effect on the amplitude of the Bessel function. The exact values of the amplitude of the spectral lines should also take into account the effect of the Fourier transform of the rectangular pulse waveform.

CHECKPOINT 3–2

3.1.3 Random Modulating Signals

The derivation of the PSD of Eq. (3–1) can be performed under the hypothesis that $m(kT_s)$ is a strict-sense stationary discrete random process, where $m(kT_s)$ are the samples of a strict-sense stationary continuous process $m(t)$ and the different $m(kT_s)$ are statistically independent with a common probability density function $w(m(kT_s))$.

Since the signal of Eq. (3–1) is not wide-sense stationary, the PSD $P_{x_{PPM}}(f)$ for this signal can be found by applying the following steps:

1. Compute the autocorrelation function of a particular $x_{PPM}(t)$.
2. Average over the ensemble to find the ensemble average.

3. Obtain $P_{x_{PPM}}(f)$ by taking the Fourier transform of the ensemble average.

As shown by (Rowe, 1965), $P_{x_{PPM}}(f)$ can be expressed as follows:

$$P_{x_{PPM}}(f) = \frac{|P(f)|^2}{T_s}\left[1-|W(f)|^2 + \frac{|W(f)|^2}{T_s}\sum_{n=-\infty}^{+\infty}\delta(f-\frac{n}{T_s})\right] \qquad (3–28)$$

where $W(f)$ is the Fourier transform of the probability density w and coincides with the characteristic function of w computed in $-2\pi f$:

$$W(f) = \int_{-\infty}^{+\infty} w(s)e^{-j2\pi fs}ds = \left\langle e^{-j2\pi fs}\right\rangle = C(-2\pi f) \qquad (3–29)$$

Equation (3–28) shows that the spectrum of a random modulating PPM signal is composed of a continuous part controlled by the term $1-|W(f)|^2$, and of a discrete part formed by line components at frequency $1/T_s$, that is, the pulse repetition frequency and harmonics. The discrete part corresponds to a periodic component of the PPM signal of Eq. (3–1).

Since $W(f)$ is the Fourier transform of a probability density function, its value at 0 is 1, that is, $W(0) = 1$; therefore, the continuous component of the spectrum is zero at frequency zero and it necessarily rises at higher frequencies. The discrete components that are weighted by $|W(f)|^2$ are larger at low frequencies, then decrease at high frequencies. The predominance of the continuous term over the discrete term depends on the values of $m(kT_s)$. If these are small, then the PPM signal of Eq. (3–1) resembles a periodic signal and the discrete components dominate the low frequencies. If, however, the $m(kT_s)$ values are large, then the PPM signal of Eq. (3–1) loses its resemblance with a periodic signal and the continuous component dominates the low frequency as well as the high frequency range of the spectrum.

Finally, as in the case of sinusoidal modulating signals, the term $|P(f)|^2$ shapes the overall spectrum and limits bandwidth occupation to finite values.

When the different $m(kT_s)$ are not independent, the PSD is found to be (Rowe, 1965):

$$P_{x_{PPM}}(f) = \frac{|P(f)|^2}{T_s}\sum_{n=-\infty}^{+\infty}\left\langle e^{-j2\pi f\left(m((l+n)T_s)-m(lT_s)\right)}\right\rangle e^{-j2\pi fnT_s} \qquad (3–30)$$

CHECKPOINT 3–3

In this checkpoint, we will use computer simulation to analyze the spectral occupation of a PPM-UWB signal in the presence of a random modulating signal. In particular, we will assume that the samples $m(kT_s)$ are statistically independent and Gaussian distributed random variables. The transmitted UWB signal is characterized by rectangular pulses with duration T_r and an average pulse repetition period T_s. To simulate the generation of the UWB signal under examination, we will introduce Function 3.4.

Function 3.4 (see Appendix 3.A) generates a PPM-UWB signal in the case of a random modulating signal. Within the function, the user must set the following parameters: the average transmitted power in dBm `Pow`, the sampling frequency for representing the signal `fc`, the number of pulses to be generated `np`, the time duration of each rectangular pulse `Tr`, the average pulse repetition period in seconds `Ts`, and the standard deviation of the Gaussian distributed modulating signal `sigma`. Function 3.4 returns three outputs: the generated train of pulses `Stx`, the corresponding sampling frequency `fc`, and vector `M0` containing all the time shifts applied to the transmitted pulses due to the presence of the modulating signal. The command line for generating the signal is:

```
[Stx,fc,M0]=cp0303_PPM_random;
```

We will use Function 3.4 for generating the UWB signal `RS0`, which is characterized by the following parameters: `Pow=-30`; `fc=1e11`; `np=10000`; `Tr=0.5e-9`; `Ts=2e-9`; `sigma=0.1e-9`. The command line for generating signal `RS0` is:

```
[RS0,fc,M0]=cp0303_PPM_random;
```

Figure 3–21 represents a fragment of the generated signal `RS0` in the time domain.

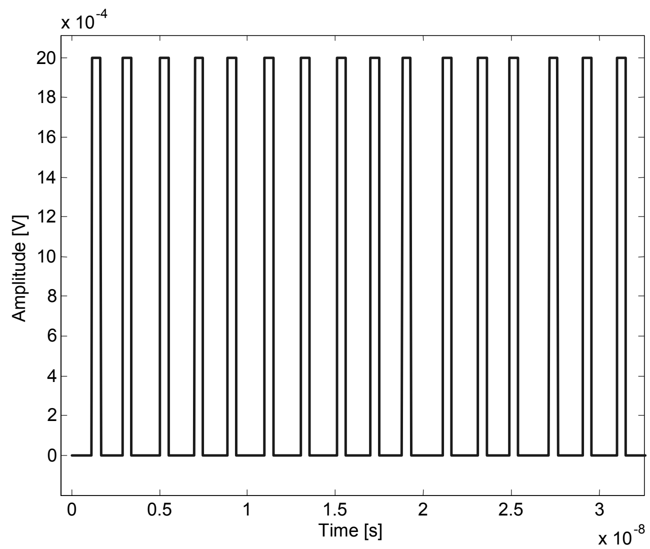

Figure 3–21 Signal RS0 in the time domain — PPM-UWB signal in the case of a random modulating signal.

Figure 3–21 shows that the effect of PPM on the position of the transmitted pulses is barely appreciable, due to the small value chosen for the standard deviation, sigma=0.1·10-9. We can analyze the spectral characteristics of signal RS0 by using Function 3.2, i.e.:

```
[PSD0,df]=cp0301_PSD(RS0,fc);
```

The above command line stores in memory the PSD PSD0 and produces the graphical output in Figure 3–22.

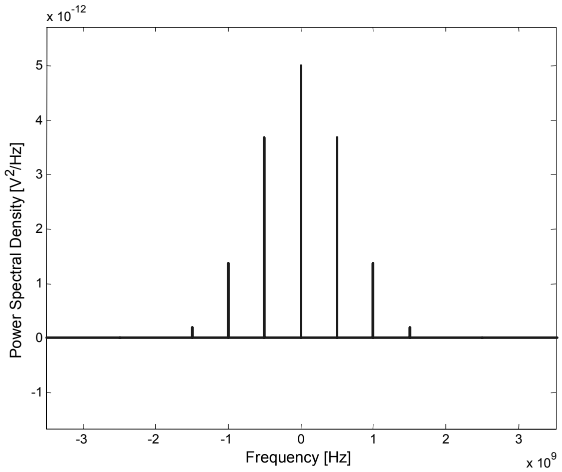

Figure 3–22 PSD of signal RS0 (see Figure 3–21).

The PSD in Figure 3–22 is composed of discrete contributions only. The discrete terms in Eq. (3–28) are thus predominant over the continuous term. This result is due to the presence of very small values for the randomly generated PPM shifts. We also observe that the envelope of the PSD is considerably different from the sin(x)/x shape since no side lobes are visible in the plot of Figure 3–22. This result can be analyzed by taking into account the statistical characteristics of the modulating signal. Given vector M0, resulting from the execution of Function 3.4, we can evaluate the probability density function of the time shifts, introduced within the signal by executing the following code lines:

```
dt = 1/fc;
NI = ((2e-9)-(0.5e-9))/dt;
h = hist(M0,NI);
h0 = (1/sum(h)).*h;
time=linspace(0,2e-9,length(h))-(2e-9/2);
stem(time,h0)
```

The above code makes use of the MATLAB function H=hist(Y,N), which groups the elements of vector Y into N equally spaced containers and returns vector H, which contains the number of elements in each container. The above set of commands also produces the plot in Figure 3–23, which represents the probability density function of the PPM shifts for signal RS0. The shape of this function resembles the well-known bell shape of a Gaussian.

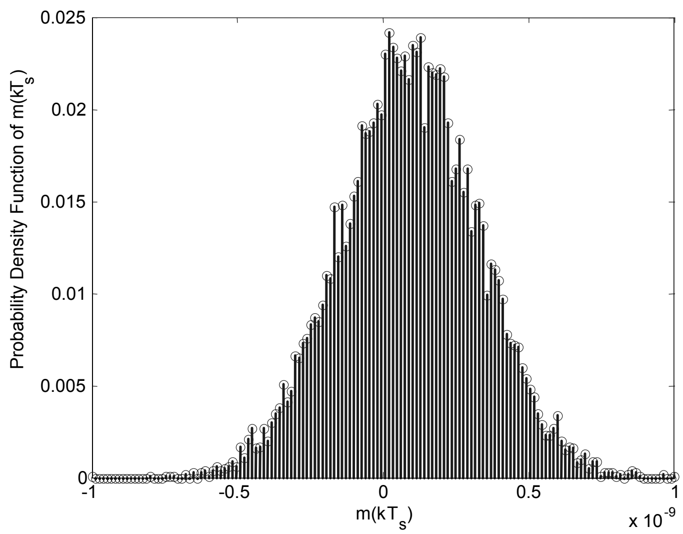

Figure 3–23 Probability density function of the PPM shifts for signal RS0.

We can now apply Function 3.2 to vector `h0`:

```
fcx=1/((2e-9)/length(h));
[W,df]=cp0301_PSD(h0,fcx);
```

The above commands store in memory vector `W`, which represents a function in the frequency domain that is proportional to the term $|W(f)|^2$ of Eq. (3–28). The graphical output provided by the above commands is represented in Figure 3–24.

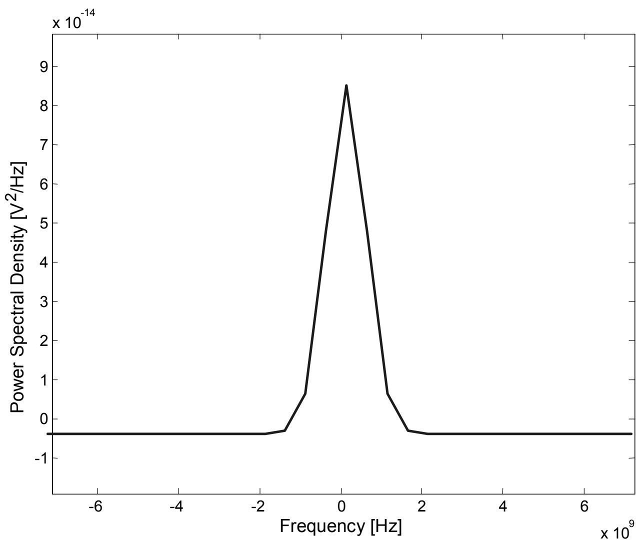

Figure 3–24 PSD of vector h0 — squared modulus of the Fourier transform of the probability density function of the PPM shifts.

The result shown in Figure 3–24 confirms the effect of the Fourier transform W(f) on the PSD of the generated UWB signal. When comparing Figures 3–24 and 3–22, we conclude that in the present case, the bandwidth of the transmitted signal is limited by the Fourier transform W(f) of the probability density of the PPM shifts, not by the Fourier transform P(f) of the pulse waveform.

To verify the presence of a continuous part in the PSD of PPM-UWB signals modulated with a random signal, we can consider the case of signal `RS1` with an increased variability in the PPM shifts. To better exploit the variability of the PPM shift, we choose higher values for both standard deviation `sigma` and average pulse repetition period `Ts`. The following parameters are set within Function 3.4: `Pow=-30`; `fc=1e11`; `np=10000`; `Tr=0.5e-9`; `Ts=10e-9`; `sigma=4e-9`. The command line for generating signal `RS1` is:

```
[RS1,fc,M1]=cp0303_PPM_random;
```

Figure 3–25 represents a section of the generated signal `RS1` in the time domain.

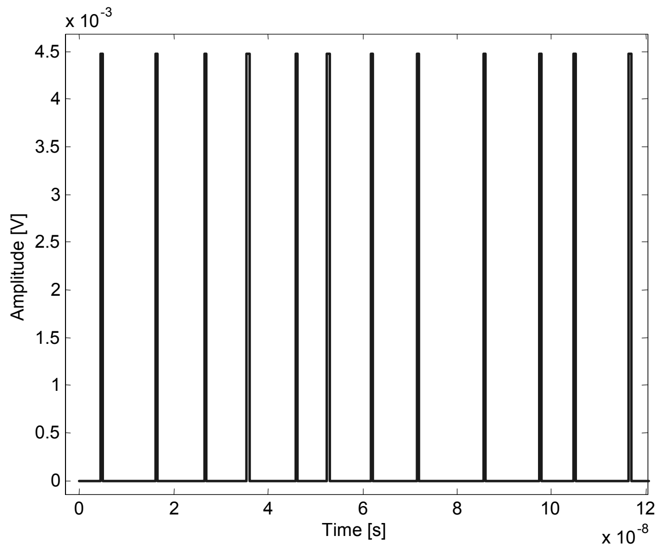

Figure 3–25 Signal RS1 in the time domain.

Figure 3–25 shows that the effect of PPM is considerable in the case of signal `RS1`. We can analyze the spectral characteristics of signal `RS1` by executing the following command line:

```
[PSD1,df]=cp0301_PSD(RS1,fc);
```

The graphical output that results from the above command is shown in Figure 3–26.

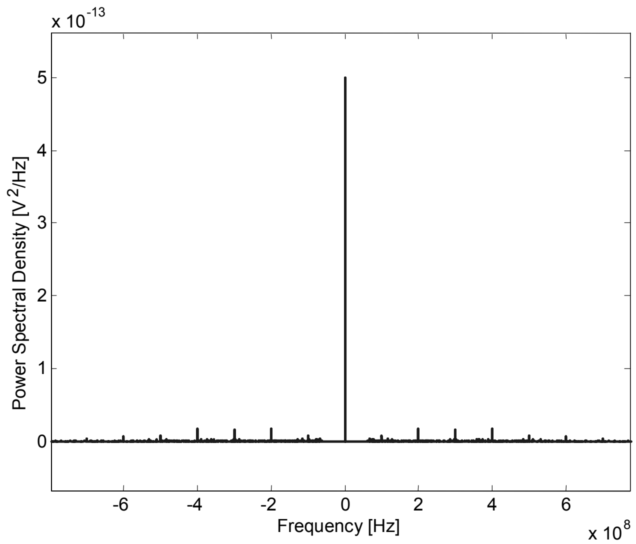

Figure 3–26 PSD of signal RS1 (see Figure 3–25).

Figure 3–26 shows that the PSD of signal `RS1` is dominated by a strong peak at zero frequency. We also observe that the PSD is not composed of spectral lines only, since we notice a few spurious contributions between these lines. This observation is confirmed by zooming in on Figure 3–26 in the region of frequencies included between ±200 MHz (see Figure 3–27).

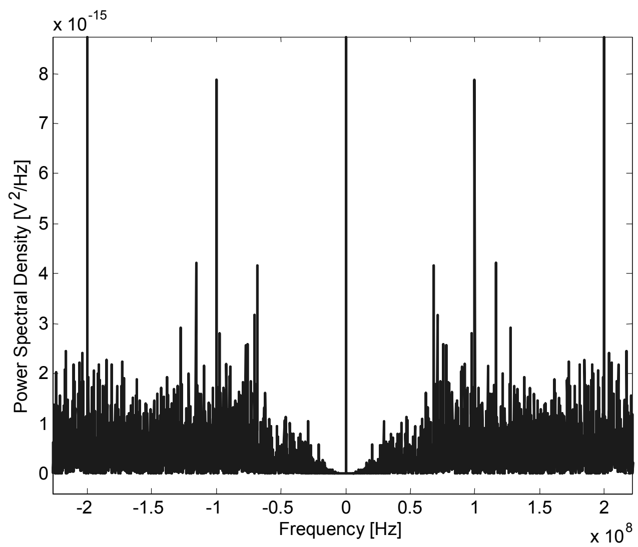

Figure 3–27 Details of the PSD of signal RS1 .

In the case of signal `RS1`, we can recognize the presence of a continuous part in the PSD, which is controlled by the function W(f).

Note that a similar effect is not observed if one tries to zooming in on Figure 3–22, as can be easily checked by the interested reader.

CHECKPOINT 3–3

3.2 THE PPM-TH-UWB CASE

We can now refer back to the PPM-TH-UWB signal, as expressed by Eq. (2–4), and establish a correspondence with the PPM signal of Eq. (3–1), that is, between the $m(kT_s)$ process and the θ time dither process, which, as previously defined, incorporates the time shift introduced by the TH code η and the time shift introduced by the PPM modulator ε.

Since ε is much smaller than η, θ is quasi-periodic and closely follows the periodicity of the TH code. We can, as a reasonable first approximation, make the hypothesis that the effect of the ε shift on the PSD is not significant with respect to η. Therefore, the signal of Eq. (2–4) is modulated by a periodic signal and its PSD follows Eq. (3–25). In other words, the PSD is discrete and contains discrete frequency components located at the fundamental. In addition, it contains components at pulse repetition frequencies and their harmonics, and at linear combinations. In the present case, the period of the modulating periodic waveform corresponds to the period of the code N_p multiplied by the pulse interval T_s, that is, $T_p = N_pT_s$; therefore, the fundamental frequency of the modulating waveform is $f_p = 1/T_p$.

We shall first note that the pulse repetition frequency $1/T_s = N_p/T_p$ is a multiple of the fundamental frequency of the periodic waveform; therefore, the PSD is composed of lines occurring at $1/T_p$ and its harmonics (see Checkpoint 3–2).

The case $N_p = 1$ corresponds to the actual absence of coding and generates a signal with a PSD composed of lines at $1/T_s$ and harmonics. Power concentrates on spectrum lines with the undesirable effect of presenting spectral line peaks. This is not a surprise since having neglected the effect of ε, Eq. (2–4) forms a periodic train of pulses occurring at multiples of T_s.

If, as common practice, N_p is set equal to N_s, that is, the periodicity of the code coincides with the number of pulses per bit, spectrum lines occur at $1/T_b$ and its harmonics, where $T_b = N_sT_s = N_pT_s$ is the bit interval. Although the spectrum is still discrete, spectrum lines occur at frequencies that are more numerous than in the previous case for equal bandwidth since $1/T_b<1/T_s$. The whitening effect of the code is visible in that power distributes over a larger number of spectrum lines and spectral peaks are less accentuated.

When we make N_p larger than N_s, the above effect is more prominent, and if N_p is not a multiple of N_s, several spectral lines generated by linear combination of $1/T_p$ and $1/T_s$ fill up the power spectrum with a beneficial smoothing effect.

The extreme case $N_p \rightarrow \infty$ corresponds to a lack of periodicity in the signal of Eq. (2–4). In this case, the time dither process can be assimilated to the random modulating signal $m(kT_s)$ and its power spectrum is given by Eq. (3–28). All comments made on Eq. (3–28) are valid here, that is, the spectrum has two components, one continuous and one discrete. The discrete term corresponds to a periodic component of the signal, which reduces with increasing variance in the position of the pulses. In addition, note that in the present case, the TH code generates time shifts that span over the entire T_s interval. Therefore, the θ values cannot be considered as small, and we can expect a reduction of the periodic component in the signal, that is, of the discrete component in the spectrum.

In the presence of many such signals, or in the case of a multi-user system, we can expect that the resulting signal shows little periodicity, and the comments corresponding to the random modulating case apply.

The case of a system composed of a few users using the same value for N_p could possibly be considered as of a periodic type, with the resulting cumulative signal having a discrete spectrum if all users were synchronized. Note, however, that under the realistic hypothesis of asynchronous users we can expect that, as is the case for several users, the multi-user signal looses its periodicity and its spectrum is well represented by Eq. (3–28).

A more detailed analysis requires relaxing the hypothesis of an inconsequential effect of the PPM time shift ε. A straightforward solution to this problem corresponding to the common case $N_p = N_s$ is to consider Eq. (2–4) in which we first neglect the effect of ε, that is, we define a signal $v(t)$ given by:

$$v(t) = \sum_{j=1}^{N_s} p\left(t - jT_s - \eta_j\right) \tag{3–31}$$

The Fourier transform of the above signal is:

$$P_v(f) = P(f)\sum_{m=1}^{N_s} e^{-j\left(2\pi f\left(mT_s+\eta_m\right)\right)} \tag{3–32}$$

If we now consider *v(t)* as the basic multi-pulse used for transmission and apply the ε PPM shift, we obtain the following expression for the transmitted signal:

$$s(t) = \sum_{j=-\infty}^{+\infty} v\left(t - jT_b - \varepsilon b_j\right) \tag{3–33}$$

which is a PPM modulated waveform in which the shift is ruled by the sequence of data symbols **b**, that is, the **b** process emitted by the source. Note that the repetition code is now incorporated in the multi-pulse. If we can assume that **b** is a strict-sense stationary discrete random process, and the different extracted random variables b_k are statistically independent with a common probability density function *w*, then the signal of Eq. (3–33) has the PSD of Eq. (3–28) in which the Fourier transform of the pulse waveform *P(f)* is substituted by the PSD of the multi-pulse given by Eq. (3–32).

Given that the multi-pulse repetition rate is T_b, one obtains the following spectrum of a PPM-TH-UWB signal:

$$P_s(f) = \frac{\left|P_v(f)\right|^2}{T_b}\left[1 - \left|W(f)\right|^2 + \frac{\left|W(f)\right|^2}{T_b}\sum_{n=-\infty}^{+\infty}\delta\left(f - \frac{n}{T_b}\right)\right] \tag{3–34}$$

Equation (3–34) shows the double effect on one side of the TH code through $P_v(f)$, and on the other side of the time shift introduced by the PPM modulator which has characteristics following the statistical properties of the source. Note that the discrete component of the spectrum has lines at $1/T_b$. The amplitude of the lines is weighted by the statistical properties of the source represented by $|W(f)|^2$. If *p* indicates the probability of emitting a 0 bit (no shift) and 1-*p* the probability of emitting a '1' bit (ε shift), one can write:

$$\left|W(f)\right|^2 = 1 + 2p^2\left(1 - \cos\left(2\pi f\varepsilon\right)\right) - 2p\left(1 - \cos\left(2\pi f\varepsilon\right)\right) \tag{3–35}$$

If the source emits equiprobable symbols 0 and 1, then Eq. (3–35) simplifies as follows:

$$\left|W(f)\right|^2 = \frac{1}{2}\left(1 + \cos\left(2\pi f\varepsilon\right)\right) \tag{3–36}$$

Note here that the time shift is small and therefore the discrete components dominate the spectrum. In the simplifying hypothesis made in the beginning of this paragraph that ε is negligible, Eq. (3–34) is periodic with period $1/T_b$. Note that Eq. (3–34) can also be applied to any type of source, not necessarily binary.

CHECKPOINT 3–4

In this checkpoint, we will use computer simulation to analyze the spectral occupation of a PPM-TH-UWB signal. The MATLAB functions that are required for such an analysis have already been introduced in previous checkpoints: Function 2.6 in Checkpoint 2–1 for generating the PPM-TH-UWB signal and Function 3.2 in Checkpoint 3–1 for representing the PSD. Different simulations will be performed to analyze the effect of the main parameters of the UWB signal under examination on the PSD.

In the first simulation, we consider the case of an UWB signal with no PPM and no TH coding, denoted as signal `u0`. To generate signal `u0`, we execute Function 2.6 with the following parameters: `Pow=-30`; `fc=50e9`; `numbits=1000`; `Ts=10e-9`; `Ns=5`; `Tc=1e-9`; `Nh=10`; `Np=1`; `Tm=0.5e-9`; `tau=0.25e-9`; `dPPM=0`; `G=0`. The command line for generating signal `u0` is:

```
[bits,THcode,u0,ref]=cp0201_transmitter_2PPM_TH;
```

The above command line stores vector `u0` in memory, to represent the signal under examination. This signal is characterized by the transmission of five pulses per bit. All pulses are equally spaced in time with a pulse repetition period T_s. Because of the value `Np=1`, all pulses occupy the same position inside each T_s interval. Each pulse has the second derivative Gaussian shape with a maximum length of 0.5 ns.

We can analyze signal `u0` in the frequency domain by executing the command:

```
[PSDu0,df]=cp0301_PSD(u0,50e9);
```

The above command stores vector `PSDu0` in memory, to represent the PSD of signal `u0` and produce Figure 3–28. An enlarged version of Figure 3–28 is shown in Figure 3–29.

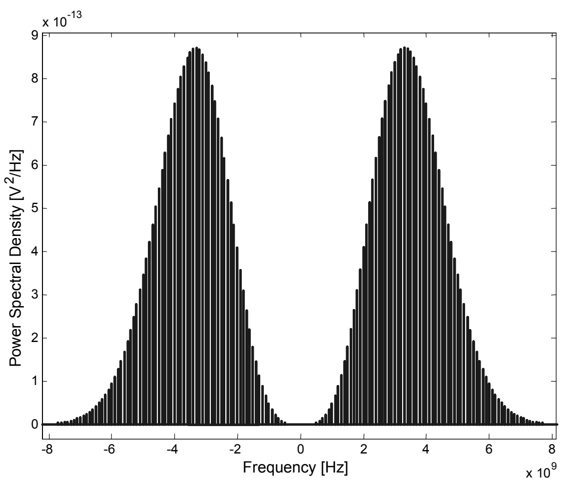

Figure 3–28 PSD of signal u0 — no PPM and no TH coding, and $N_p = 1$.

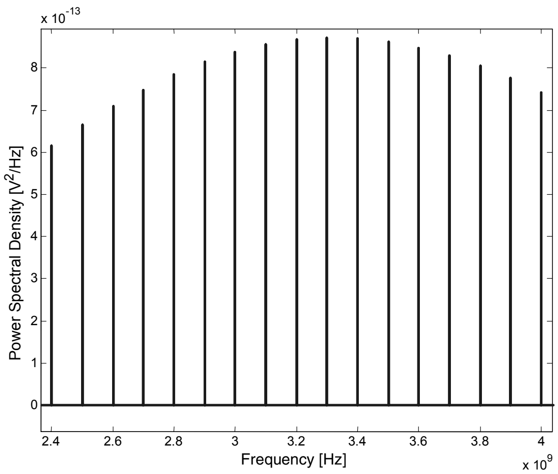

Figure 3–29 Detail of Figure 3–28 — PSD of signal u0.

As expected, Figures 3–28 and 3–29 show that the PSD of signal `u0` is composed of spectral lines occurring at $1/T_s$ = 0.1 GHz and harmonics, that is, the transmitted power is concentrated at multiples of the pulse repetition frequency. The envelope of the PSD has the shape of the Fourier transform of the second derivative Gaussian waveform.

In the second simulation, we consider the same parameters characterizing signal `u0`, but with an increased periodicity `Np` of the TH code. In particular, we set `Np=Ns`, that is, `Np=5`. The resulting signal `u1` can be generated as follows:

```
[bits,THcode,u1,ref] = cp0201_transmitter_2PPM_TH;
```

The above command line stores vector `u1` in memory, to represent the signal under examination. This signal is characterized by the transmission of five pulses per bit. The position of the pulse within each T_s interval depends on the corresponding coefficient of the TH code.

We can analyze signal `u1` in the frequency domain by executing the following command:

```
[PSDu1,df] = cp0301_PSD(u1,50e9);
```

The above command stores vector `PSDu1` in memory, to represent the PSD of signal `u1` and produce the plot in Figure 3–30 (see the detailed plot in Figure 3–31).

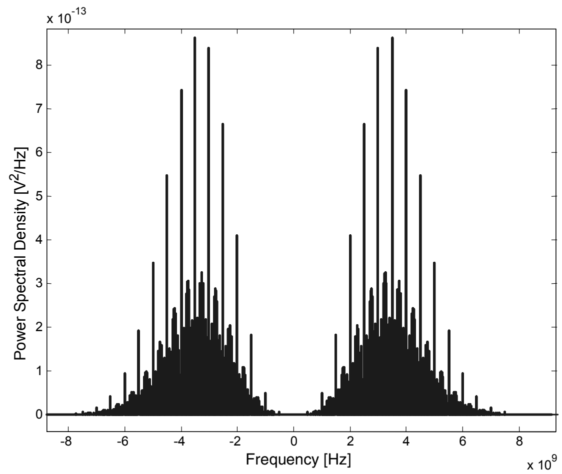

Figure 3–30 PSD of signal u1 — no PPM and with TH coding, and $N_p = 5$.

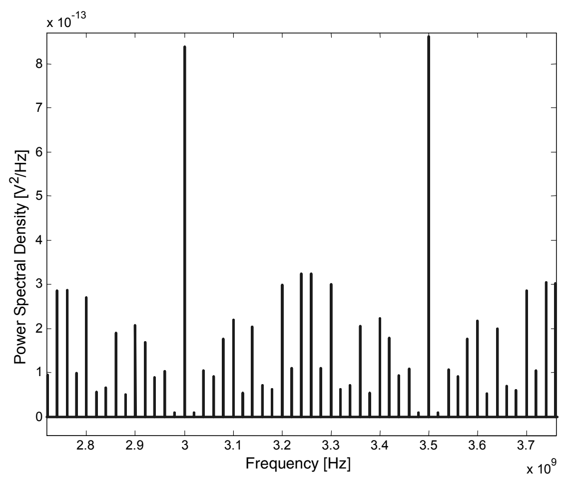

Figure 3–31 Detail of Figure 3–30 — PSD of signal u1.

Figures 3–30 and 3–31 show that the PSD of signal `u1` is composed of spectral lines at distances of $1/(N_sT_s)$ = 20 MHz, that is, the transmitted power is concentrated at multiples of the bit repetition frequency (the bit rate). This result is justified by the periodicity of the TH code, which coincides with N_s. Signal `u1` is, therefore, periodic with a period equal to the bit period. The envelope of the PSD still resembles the Fourier transform of the second derivative Gaussian waveform. When comparing the PSD of signal `u1` with the PSD of signal `u0`, we can verify that the TH code has the effect of diminishing the number of peaks with the highest power contribution since the same power is distributed over a larger number of spectral lines. This effect should be more prominent when one further increases the N_p value. This analysis can be performed by generating a new signal `u2`, with N_p equal to the total number of transmitted pulses. In the case being examined, we run Function 2.6 with the same parameters of signal `u1`, but we set N_p = 5000. The command line for generating the signal is:

```
[bits,THcode,u2,ref]=cp0201_transmitter_2PPM_TH;
```

which stores vector `u2` in memory, representing the signal under examination. This signal is still characterized by five pulses per bit. Each pulse, however, occupies a position that is given by a discrete random variable uniformly distributed between 0 and N_h-1. We can analyze signal `u2` in the frequency domain by executing:

```
[PSDu2,df]=cp0301_PSD(u2,50e9);
```

The above command stores vector `PSDu2` in memory, to represent the PSD of signal `u2` and produce the plot of Figure 3–32 .

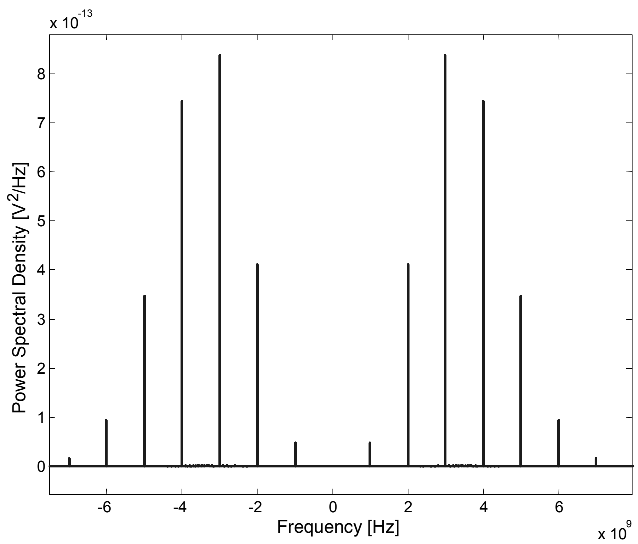

Figure 3–32 PSD of signal u2 — no PPM and with TH coding, and N_p = 5000.

Figure 3–32 shows that the PSD is still composed of a discrete part. The number of peaks, however, is smaller with respect to signals `u0` and `u1`. These peaks are located at multiples of frequency $1/T_c$ = 1 GHz. Although the TH code is not periodic, the positions of the pulses inside each T_s interval are not random. Each T_s interval is, in fact, divided into N_h slots with length T_c, and the pulses are forced to locations at the beginning of these intervals. In the case under examination, N_h = 10, there are only ten possible positions for each pulse within one T_s interval. One can conclude, therefore, that the loss of periodicity of the TH code does not guarantee by itself the possibility of removing all peaks in the PSD. The number of peaks, however, is definitely reduced. Figure 3–33 shows the detail of the PSD of signal `u2` in the range between 2 and 3 GHz. We can verify the presence of a continuous part with smaller peaks at distances $1/(N_sT_s)$ = 20 MHz.

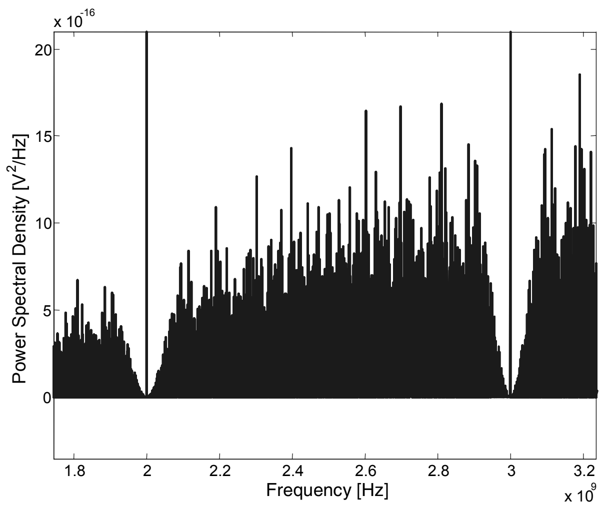

Figure 3–33 Detail of Figure 3–32 — PSD of signal u2.

The spectral analysis of signal `u2` showed that it is not possible to remove all the peaks of the PSD by only increasing the periodicity of the TH code. To decrease the peaks, we should allow each pulse to assume random positions inside each T_s interval. In the following simulation, we generate an UWB signal with the same parameters of signal `u2`, but with an increased cardinality of the TH code, that is, we divide the T_s interval into a higher number of T_c intervals. In particular, we consider the following parameters: `Pow=-30`; `fc=50e9`; `numbits=1000`; `Ts=10e-9`; `Ns=5`; `Tc=0.1e-9`; `Nh=100`; `Np=5000`; `Tm=0.5e-9`; `tau=0.25e-9`; `dPPM=0`; `G=0`. The number of slots per frame is increased tenfold, with a higher variance in the position of the pulses. We generate the resulting signal, signal `u3`, by executing the following command:

```
[bits,THcode,u3,ref] = cp0201_transmitter_2PPM_TH;
```

which stores vector `u3` in memory, representing the signal under examination. The PSD of signal `u3` is obtained as follows:

```
[PSDu3,df] = cp0301_PSD(u3,50e9);
```

The above command stores vector `PSDu3` in memory and provides the plot of Figure 3–34.

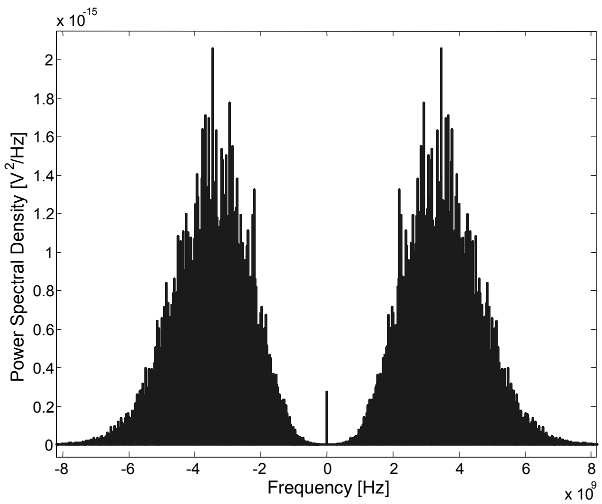

Figure 3–34 PSD of signal u3 — no PPM and with TH coding. Effect of an increased cardinality of the TH code, N_h.

When comparing the PSD in Figure 3–34 with the PSD in Figure 3–32, we conclude that the increased N_h value has the effect of removing the strong peaks in the PSD. As a matter of fact, these lines have been simply moved out of the bandwidth of the Fourier transform of the pulse waveform, that is, $1/T_c$= 10 GHz.

A final simulation can be performed to take into account the effect of the PPM. The following parameters are set within Function 2.6: `Pow=-30`; `fc=50e9`; `numbits=1000`; `Ts=10e-9`; `Ns=5`; `Tc=0.1e-9`; `Nh=100`; `Np=5000`; `Tm=0.5e-9`; `tau=0.25e-9`; `dPPM=0.25e-9`; `G=0`. The resulting signal, signal `u4`, is generated as follows:

```
[bits,THcode,u4,ref] = cp0201_transmitter_2PPM_TH;
```

The above command stores vector `u4` in memory. The PPM block is included within the transmission chain with a PPM shift of 0.25 ns for representing 1 bits. The spectral analysis of signal `u4` is performed as follows:

```
[PSDu4,df] = cp0301_PSD(u4,50e9);
```

The above command stores vector `PSDu4` in memory and provides the plot of Figure 3–35, which shows the PSD of a PPM-TH-UWB signal. This PSD is composed of a continuous part plus spectral lines located at multiples of $1/T_b$.

Figure 3–36 compares this PSD with that of signal `u0`. We can verify that the introduction of both TH coding and PPM has the effect of distorting the original Gaussian shape of the PSD. Figure 3–36 also shows that the PSD of a PPM-TH-UWB signal is fully contained within the envelope of the PSD, which results from the transmission of equally spaced pulses with the same shape and same average repetition frequency, that is, with the same average power.

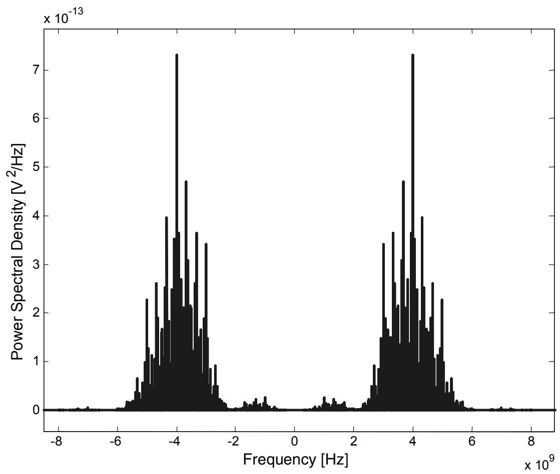

Figure 3–35 PSD of signal u4 — with PPM and TH coding.

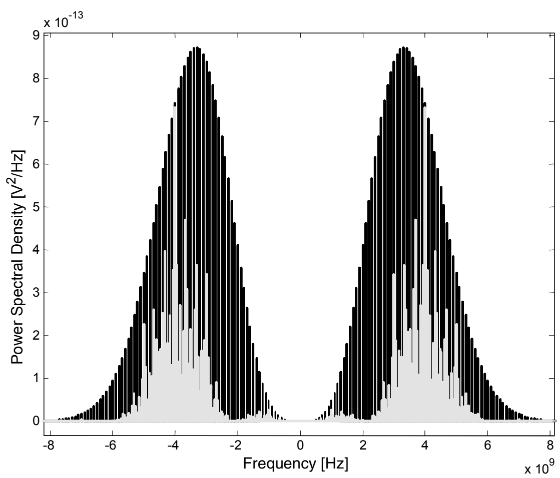

Figure 3–36 Comparison between the PSD of signal u0 (black), that is, no PPM and no TH coding, and the PSD of signal u4 (gray), that is, with PPM and TH coding.

CHECKPOINT 3–4

FURTHER READING

A PSD for PPM-TH-UWB following a different approach of the one adopted here can be found in (Kissik, 2001) and (Win, 2002), in which a unified spectral analysis for TH-SS in the presence of timing jitter is introduced. A recent paper by (Lehman and Haimovich, 2003) includes several PSD analytical expressions for TH-IR signals. The work by (Padgett, 2003) assimilates dithering to a modulation of the pulse repetition frequency.

REFERENCES

Bennett, W.R., "New results in the calculation of modulation products," *Bell System Tech. J.*, no. 12 (1933), 228–243.

Bennett, W.R., "Response of a linear rectifier to signal and noise," *J. Acoust. Soc. Am.*, no. 15 (1944), 164–172.

Bennett, W.R., "The biased ideal rectifier," *Bell System Tech. J.*, no. 26 (1947), 139–169.

Black, H.S., J.W. Beyer, T.J. Grieser, and F.A. Polkinghorn, "A multichannel microwave radio relay system," *AIEE Transaction Electrical Engineering*, Volume: 65 (1946), 798–805.

Di Benedetto, M.-G., and B.R. Vojcic, "Ultra Wide Band (UWB) Wireless Communications: A Tutorial," *Journal of Communication and Networks, Special Issue on Ultra-Wideband Communications*, Volume: 5, Issue: 4 (December 2003), 290–302.

Kissik, A.W., U.S. Department of Commerce, *The Temporal and Spectral Characteristics of Ultra-wideband Signals*, Available at *www.its.bldrdoc.gov/pub/ntia-rpt/01-383/01-383.pdf* (January 2001).

Lehmann, N.H., and A.M. Haimovich, "The Power Spectral Density of a Time Hopping UWB Signal: A Survey," *IEEE Conference on Ultra Wideband Systems and Technologies* (November 2003), 234–239.

Padgett, J.E., "The Power Spectral Density of a UWB Signal with Pulse Repetition Frequency (PRF) Modulation," *IEEE Conference on Ultra Wideband Systems and Technologies* (November 2003), 15–20.

Proakis, J.G., and M. Salehi, *Communication Systems Engineering*, Englewood Cliffs, New Jersey: Prentice-Hall International (1994).

Rowe, H.E., *Signals and Noise in Communications Systems*, Princeton, New Jersey: D. Van Nostrand Company, Inc (1965).

Win, M.Z., "A Unified Analysis of Generalized Time-Hopping Spread-Spectrum Signals in the Presence of Timing Jitter," *IEEE Journal on Selected Areas in Communications*, Volume: 20, Issue: 9 (December 2002), 1664–1676.

APPENDIX 3.A

Function 3.1 Analogue PPM with Sinusoidal Modulating Signals

Function 3.1 is composed of two steps. Step Zero contains all the parameters characterizing the signal to be generated: the average transmitted power in dBm `Pow`, the sampling frequency for representing the signal `fc`, the number of pulses to be generated `np`, the time duration of each rectangular pulse `Tr`, the average pulse repetition period in seconds `Ts`, the amplitude and frequency of the sinusoidal modulating signal, `A` and `f0`. Step One contains the code for generating the PPM signal. PPM is implemented in Function 3.1 by introducing vector `Mtot`, which collects all the time shifts that must be applied to the pulses.

```
%
% FUNCTION 3.1 : "cp0301_PPM_sin"
%
% Generation of a PPM-UWB signal in the case of a
% sinusoidal modulating signal and rectangular pulses
%
% Transmitted power is fixed at 'Pow'
% The signal is sampled with frequency 'fc'
% 'np' is the number of generated pulses
% 'Ts' is the average pulse repetition period
% Each rectangular pulse has time duration 'Tr'
% The modulating signal is a sinusoid with
% amplitude 'A' and frequency 'f0'
%
% The function returns the generated signal 'Stx'
% and the corresponding sampling frequency 'fc'
%
% Programmed by Guerino Giancola
%

function [Stx,fc]=cp0301_PPM_sin;

% ----------------------------
% Step Zero - Input parameters
% ----------------------------

Pow = -30;  % average transmitted power (dBm)
```

```
fc = 1e11;  % sampling frequency

np = 10000; % number of pulses

Tr = 0.5e-9;% time duration of the rectangular pulse [s]

Ts = 2e-9;  % average pulse repetition period [s]
A = Ts/2;   % maximum time shift provided by the
            % modulation [s]
f0 = 5e7;   % frequency of the modulating signal [Hz]

% -----------------------------------------
% Step One - Simulating transmission chain
% -----------------------------------------

dt = 1 / fc;          % sampling period
sTs = floor(Ts/dt);   % number of samples per frame
sTot = sTs * np;      % total number of samples
Stx = zeros(1,sTot);  % output vector

% pulse position modulation
j = (0:1:np-1);
M0 = A.*cos((2*pi*f0).*(j.*Ts));
M1 = j.*Ts;
Mtot = M0 + M1;
for k = 1 : np
  Stx(1+floor(Mtot(k)/dt))=1;
end

% shaping filter
sP = floor(Tr/dt);    % number of samples per pulse

p0 = (1/sqrt(Tr)).*ones(1,sP); % energy normalized rect
power = (10^(Pow/10))/1000;    % average transmitted power
                               % (watt)
Ex = power * Ts;        % energy per pulse
ptx= p0 .* sqrt(Ex);    % pulse waveform

Stx = conv(Stx,ptx);
Stx = Stx(1:sTot);
```

Function 3.2 PSD

Function 3.2 is composed of two steps. Step One contains the code for evaluating the PSD. Step Two contains the code for the graphical representation of the PSD. Note that Function 3.2 makes use of the MATLAB function `fftshift(x)`, which shifts zero frequency components of `x` to the center of the spectrum. The command `fftshift(x)` is useful for visualizing double-sided Fourier transforms.

```
%
% FUNCTION 3.2 : "cp0301_PSD"
%
% Evaluates the PSD of the
% signal represented by the input vector 'x'
% The input signal is sampled with frequency 'fc'
%
% This function returns the PSD ('PSD')
% and the corresponding frequency resolution ('df')
%
% Programmed by Guerino Giancola
%

function [PSD,df]=cp0301_PSD(x,fc)

% -------------------------------
% Step One - Evaluation of the PSD
% -------------------------------

dt=1/fc;
N=length(x);
T=N*dt;
df=1/T;
X = fft(x);
X = X / N;
mPSD=abs(X).^2/(df^2);
PSD = fftshift(mPSD);
PSD = (1/T).*PSD;

% ----------------------------------
% Step Two - Graphical representation
% ----------------------------------

frequency = linspace(-fc/2,fc/2,length(PSD));
PF=plot(frequency,PSD);
set(PF,'LineWidth',[2]);
```

```
AX=gca;
set(AX,'FontSize',12);
X=xlabel('Frequency [Hz]');
set(X,'FontSize',14);
Y=ylabel('Power Spectral Density [V^2/Hz]');
set(Y,'FontSize',14);
```

Function 3.3 Analog PPM with Generic Periodic Modulating Signals

Function 3.3 is composed of two steps. Step Zero contains all the parameters characterizing the signal to be generated, that is, the average transmitted power in dBm `Pow`, the sampling frequency `fc`, the number of pulses to be generated `np`, the time duration of each rectangular pulse, `Tr`, the average pulse repetition period in seconds `Ts`, and finally, parameters `A`, `B`, and `Tp`, characterizing the modulating signal (see Checkpoint 3–2). Step One contains the code for generating the PPM signal. Similarly to Function 3.1, the PPM scheme is implemented by introducing vector `Mtot` which collects all the time-shifts that must be applied on the transmitted pulses (see also Function 3.1).

```
%
% FUNCTION 3.3 : "cp0302_PPM_periodic"
%
% Generation of a PPM-UWB signal in the case of a generic
% periodic modulating signal and rectangular pulses
% Modulating signal is chosen to be characterized by
% an exponential decay exp(-t)
%
% Transmitted power is fixed at 'Pow'
% The signal is sampled with frequency 'fc'
% 'np' is the number of generated pulses
% 'Ts' is the average pulse repetition period
% Each rectangular pulse has time duration 'Tr'
% The periodic signal is characterized by
% shape parameters 'A' and 'B', and period 'Tp'
%
% The function returns the generated signal 'Stx'
% and the corresponding sampling frequency 'fc'
%
% Programmed by Guerino Giancola
%

function [Stx,fc]=cp0302_PPM_periodic;

% ----------------------------
% Step Zero - Input parameters
% ----------------------------

Pow = -30;     % average transmitted power (dBm)

fc = 1e11;     % sampling frequency
```

```
np = 10000;   % number of pulses

Tr = 0.5e-9;  % time duration of the rectangular pulse [s]

Ts = 2e-9;    % average pulse repetition period [s]

A = 1e-9;     % first shape parameter
B = 10;       % second shape parameter
Tp = 20e-9;   % period of the modulating signal [s]

% ----------------------------------------
% Step One - Simulating transmission chain
% ----------------------------------------

dt = 1 / fc;        % sampling period
sTs = floor(Ts/dt); % number of samples per frame
sTot = sTs * np;    % total number of samples
Stx = zeros(1,sTot);% output vector

% PPM
j = (0:1:np-1);
M0 = A.*exp(-(B/Tp).*mod(j*Ts,Tp));
M1 = j.*Ts;
Mtot = M0 + M1;
for k = 1 : np
  Stx(1+floor(Mtot(k)/dt))=1;
end

% shaping filter
sP = floor(Tr/dt);       % number of samples per pulse

p0 = (1/sqrt(Tr)).*ones(1,sP); % energy normalized rect
power = (10^(Pow/10))/1000;    % average transmitted power
                               % (watt)
Ex = power * Ts;        % energy per pulse
ptx= p0 .* sqrt(Ex);    % pulse waveform

Stx = conv(Stx,ptx);
Stx = Stx(1:sTot);
```

Function 3.4 Analog PPM with Random Modulating Signals

Function 3.4 is composed of two steps. Step Zero contains all the parameters characterizing the signal to be generated, that is, the average transmitted power in dBm `Pow`, the sampling frequency for representing the signal `fc`, the number of pulses to be generated `np`, the time duration of each rectangular pulse `Tr`, the average pulse repetition period in seconds `Ts`, and the standard deviation of the Gaussian distributed modulating signal `sigma`. Step One contains the code for generating the PPM signal. The PPM scheme is implemented by introducing vector `Mtot`, which collects all time shifts that must be applied on the transmitted pulses (see also Functions 3.1 and 3.3). The generation of the random shifts is performed by means of the MATLAB function `randn(1,N)`, which generates a vector of `N` random entries, chosen from a normal distribution with mean zero, variance one, and standard deviation one. To avoid pulse overlapping, the values of the random shifts are limited within the range $[0,T_s\text{-}T_r]$.

```
%
% FUNCTION 3.4 : "cp0303_PPM_random"
%
% Generation of a PPM-UWB signal in the case of
% a random modulating signal and rectangular pulses
% The modulating signal is characterized by
% a normal distribution
%
% Transmitted Power is fixed at 'Pow'
% The signal is sampled with frequency 'fc'
% 'np' is the number of generated pulses
% 'Ts' is the average pulse repetition period
% Each rectangular pulse has time duration 'Tr'
% The random modulating signal is characterized
% by standard deviation 'sigma'
%
% The function returns the generated signal 'Stx',
% the corresponding sampling frequency 'fc',
% and vector 'M0' of all the PPM time shifts
%
% Programmed by Guerino Giancola
%

function [Stx,fc,M0]=cp0303_PPM_random;

% ----------------------------
% Step Zero - Input parameters
% ----------------------------
```

```
Pow = -30;    % average transmitted power (dBm)

fc = 1e11;    % sampling frequency

np = 10000;   % number of pulses

Tr = 0.5e-9;  % time duration of the rectangular pulse [s]

Ts = 2e-9;    % average pulse repetition period [s]

sigma = 0.1e-9;  % standard deviation of the modulating
signal

% ----------------------------------------
% Step One - Simulating transmission chain
% ----------------------------------------

dt = 1 / fc;             % sampling period
sTs = floor(Ts/dt);      % number of samples per frame
sTot = sTs * np;         % total number of samples
Stx = zeros(1,sTot);     % output vector

% PPM
j = (0:1:np-1);
M0 = max(zeros(1,np), min((Ts -Tr).*...
   ones(1,np),((Ts/2)+sigma.*randn(1,np))));
M1 = j.*Ts;
Mtot = M0 + M1;
for k = 1 : np
  Stx(1+floor(Mtot(k)/dt))=1;
end

% shaping filter
sP = floor(Tr/dt);              % number of samples per
                                % pulse
p0 = (1/sqrt(Tr)).*ones(1,sP);  % energy normalized rect
power = (10^(Pow/10))/1000;     % average transmitted power
                                % (Watt)
Ex = power * Ts;         % energy per pulse
ptx= p0 .* sqrt(Ex);     % pulse waveform

Stx = conv(Stx,ptx);
Stx = Stx(1:sTot);
```