Spread Spectrum Systems

HTE - 15.04.2013

1. Basics

Up to now, four axes have been discovered and used to multiplex message signals. Basically each uses the idea of orthogonality along one of these axes. As shown in Fig. 1.1, these are

a) Frequency division multiplexing (FDM), the one of the oldest, it simply corresponds to placing message signals onto different sinusoidal time signals, carriers (orthogonal delta functions along frequency axis). This process can also be regarded as shifting the original spectrums of the message signals to different around frequencies along frequency axis.
b) Time division multiplexing (TDM) is achieved by sampling the message signals at different times, thus putting the message signal in orthogonal order along time axis.
c) Physical division multiplexing (PDM), considering the diffractive nature of propagation of electromagnetic waves and the reduction of signal intensity with propagation distance, multiplexing can also be achieved along spatial axis by placing message signals far apart. Another way of PDM is electrically isolating the message signals or inserting insulating material between the propagation paths of message signals, like the one in cables
d) Code division multiplexing (CDM), this is basically assigning (multiplying) message signals by different (spreading) codes which are assumed to be orthogonal to each other.

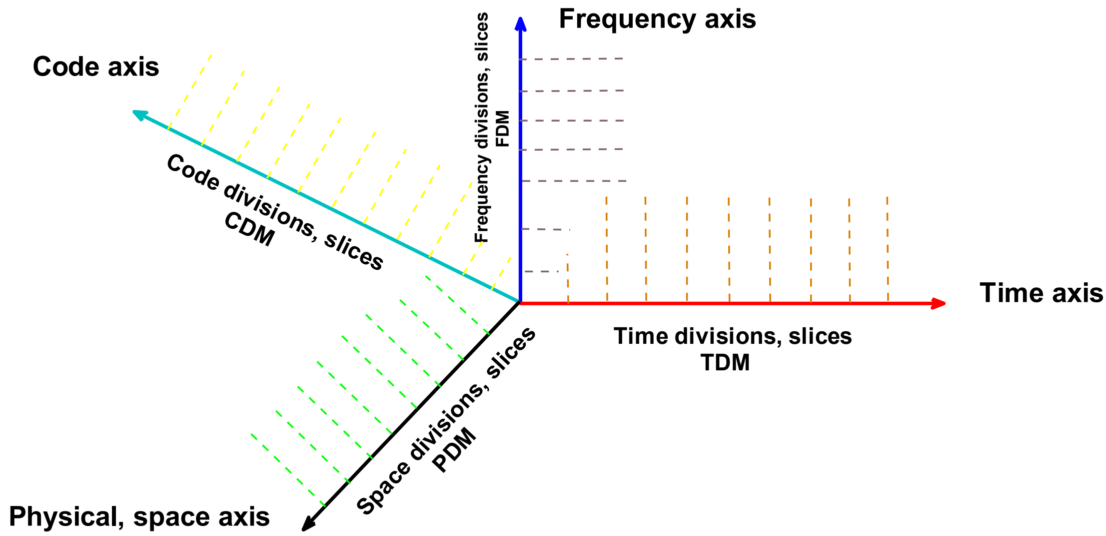

Fig. 1.1 Illustration of four types of multiplexing.

Spread spectrum (SS) systems are based on CDM. Although for a given communication system, the use of one single multiplexing scheme is sufficient to achieve separation of signals, they can also be combined to increase the number of signals multiplexed, like the combined use of FDM, TDM and PDM in GSM. For spread spectrum systems, since CDM is considered sufficient to establish the desired level of multiplexing (i.e. separation of signal), we will carry out the analysis under the assumption that the message signals are overlaid along frequency, time and physical (spatial) axis. So our SS message signals will be concurrent, coincident and collocated along time, frequency and spatial axes.

Another point that needs to be clarified is that, the act of placing our message signal onto a high frequency carrier serves only to shift the spectrum of the message signal or signals so that the transmitted signal or signals are better suitable for antenna dimensions, the communication medium. In this sense, the use of such a carrier has no effect on our analysis and will be excluded from our treatment.

Here we concentrate on single carrier spread spectrum systems otherwise called as direct sequence (DS) systems and leave out frequency hopping (FH).

We assume a message signal in the form of binary, i.e. two waveform representation of $M = 2$, amplitude shift keying (ASK) or phase shift keying (PSK) expressed in the following manner

$$v(t) = \sum_{n=-\infty}^{\infty} a_n g\left(t - nT_b\right) \qquad (1.1)$$

where a_n are the symbols of the message signal with $a_n \in \{1,\ -1\}$ or $a_n = \pm 1$ indicating that the message symbols are the binary antipodal waveforms, while $g(t)$ is the shaping waveform, mostly a rectangular pulse, scaled to have unit energy in the binary waveform duration T_b of the message signal such that

$$g(t) = \begin{cases} T_b^{-0.5} & 0 \le t \le T_b \\ 0 & \text{otherwise} \end{cases} \qquad (1.2)$$

$g\left(t - nT_b\right)$ in (1.1) represents the time advance and the delayed versions of $g(t)$.

Now, for spreading sequence, also named as signature sequence or PN (pseudo noise) sequence, we limit our time interval zero to T_b and define

$$e(t) = \sum_{i=0}^{L_c - 1} e_i p\left(t - iT_c\right) \qquad 0 \le t \le T_b \qquad (1.3)$$

where $e_i \in \{1,\ -1\}$ or $e_i = \pm 1$, $L_c = T_b / T_c$ is the length of spreading sequence, also called the processing gain, with T_c indicating the duration of one pulse of the spreading waveform, $e(t)$. T_c is also called the chip duration. $p(t)$ is again a rectangular waveform such that

$$p(t) = \begin{cases} 1 & 0 \le t \le T_c \\ 0 & \text{otherwise} \end{cases} \qquad (1.4)$$

If scaling is important we can revised (1.4) and put it in the form similar to (1.2), thus (1.4) becomes

$$p(t)=\begin{cases} T_c^{-0.5} & 0 \le t \le T_c \\ 0 & \text{otherwise} \end{cases} \tag{1.5}$$

To save on bandwidth, we can convert $p(t)$ into the (half) sine as shown below

$$p(t)=\begin{cases} \sin^2(\pi t / T_c) & 0 \le t \le T_c \\ 0 & \text{otherwise} \end{cases} \tag{1.6}$$

Exercise 1.1 : Find the scaled version of (1.6) so that

$$\text{Energy in } p(t)=\varepsilon_p=\int_0^{T_c} p^2(t)\, dt=1 \tag{1.7}$$

Example 1.1. : To prove that (1.6) actually more bandwidth efficient than (1.4), we wish to run a simple MATLAB code, given in ptspectrum.m. By running this code we see that most of the energy in $p(t)$ is concentrated around low frequencies, thus $p(t)$ of (1.6) occupies less bandwidth than $p(t)$ of (1.4) or (1.5).

Exercise 1.2 : In the m file ptspectrum.m, change $p(t)$ to the followings and find whether there are any bandwidth saving improvements

$$p(t)=\begin{cases} 1-\cos^2(\pi t / T_c) & 0 \le t \le T_c \\ 0 & \text{otherwise} \end{cases} \tag{1.8}$$

$$p(t)=\begin{cases} \exp\left(-\dfrac{t^2}{A^2 T_c^2}\right) & 0 \le t \le T_c \\ 0 & \text{otherwise} \end{cases} \tag{1.9}$$

where A is a factor standing for the variance of the exponential, examine the bandwidth variation of against A as well.

Returning to the PN (signature) sequence given in (1.3), we normally express the e_i coefficients in the following vectorial form.

$$\mathbf{e}=[\underbrace{\overbrace{e_0, \quad e_1}^{T_c}, \quad e_2, \; e_3, \; \cdots \;, e_{L_c-1}}_{T_b}]$$

$$\text{Sample PN sequence } =\begin{bmatrix}1 & -1 & -1 & 1 & \cdots & 1\end{bmatrix} \tag{1.10}$$

It is important to realize that the time spacing (decimation) between the elements of vector $\mathbf{e}$ is T_c, while the whole duration of $\mathbf{e}$ is T_b. To create effective code division multiplexing, we expect the cross correlations of the PN sequences used for different message signals to approach (normalized) zero. We analyse this subject in more details later.

Now a DS spread signal, $s(t)$ is obtained by multiplying $v(t)$ from (1.1) by $e(t)$ of (1.3), hence

$$\begin{aligned} s(t) = v(t)e(t) &= \sum_{n=-\infty}^{\infty} a_n g(t - nT_b)\, e(t) \\ &= \sum_{n=-\infty}^{\infty} a_n g(t - nT_b) \sum_{i=0}^{L_c-1} e_i p(t - iT_c) \end{aligned} \tag{1.11}$$

Assuming $g(t)$ and $p(t)$ are both rectangular waveform with unity amplitude during the time interval of existence, then $s(t)$ will become

$$s(t) == \sum_{n=-\infty}^{\infty} a_n \sum_{i=0}^{L_c-1} e_i \tag{1.12}$$

Under a single summation covering the message symbols, (1.12) will turn into

$$\begin{aligned} s(t) &= \sum_{n=-\infty}^{\infty} a_n \left(e_0 + e_1 + e_2 + \cdots + e_{L_c-1} \right) \\ &= \sum_{n=-\infty}^{\infty} a_n e_0 p(t - nT_b) + a_n e_1 p(t - nT_b - T_c) + a_n e_2 p(t - nT_b - 2T_c) + a_n e_3 p(t - nT_b - 3T_c) \\ &\quad + \cdots + a_n e_{L_c-1} p\left[t - nT_b - (L_c - 1)T_c \right] \end{aligned} \tag{1.13}$$

where on the second and third lines of (1.13), the time shifted copies of $p(t)$ is reinserted to demonstrate clearly the time decimation in increments of T_c. Thus in a time increment of T_c, $s(t)$ can only take on the value of ± 1, as shown below

$$s(t) = a_n e_i p(t - nT_b - iT_c) = \pm 1 \qquad \text{for} \qquad nT_b + iT_c \le t \le nT_b + (i+1)T_c \tag{1.14}$$

It is important to realize that n counts in increments of T_b, while i counts in increments of T_c and in practice, $T_b \gg T_c$ or $L_c = T_b / T_c \gg 1$.

It is instructive to analyse the typical time waveforms of $v(t)$, $e(t)$ and $s(t)$ and their related frequency spectrums, i.e. $V(f)$, $E(f)$ and $S(f)$. This is done in Fig. 1.2

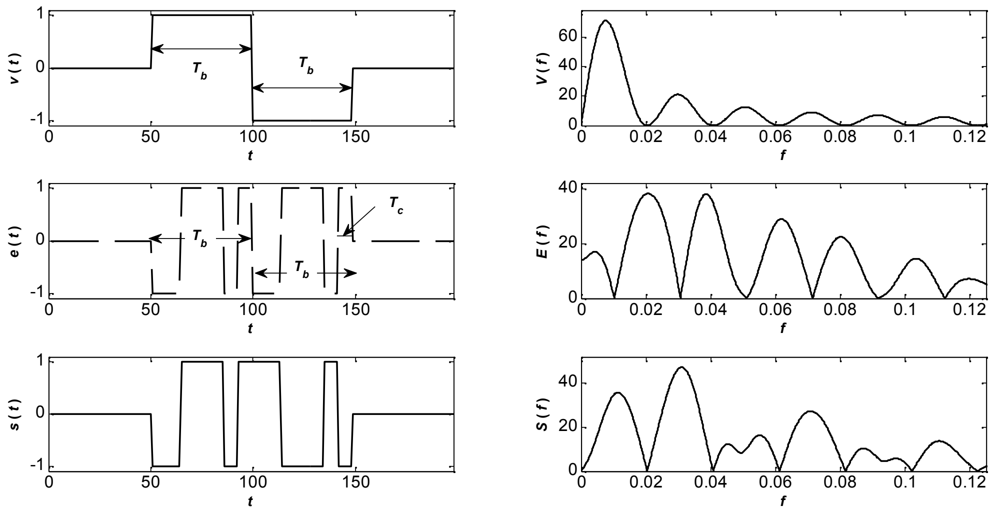

Fig. 1.2 Time waveforms and frequency spectrums of (sample) message signal, spreading signal and the spread signal.

As observed from Fig. 1.2, after the spreading operation, i.e. after multiplying $v(t)$ by $e(t)$ the spread signal $s(t)$ has a bandwidth nearly the same as $e(t)$. This means by the spreading operation, we have expanded the bandwidth of the original message signal.

Bearing in mind the relation $T_b \gg T_c$, we approximate the spectrums of $V(f)$, $E(f)$ and $S(f)$ to rectangular shape rather than the sinc profiles seen in Fig. 1.2. By assuming such bandwidths (BW) will be given by $B_b \simeq 1/T_b$, $B_c \simeq 1/T_c$, then $V(f)$, $E(f)$ and $S(f)$ will become as shown in Fig. 1.3. Note that the spectrum of $S(f)$ can be deduced from the relations

$$\text{If} \quad s(t) = v(t)e(t) \quad , \quad \text{then} \quad S(f) = V(f) \otimes E(f) \quad , \quad \otimes : \text{convolution operator} \qquad (1.15)$$

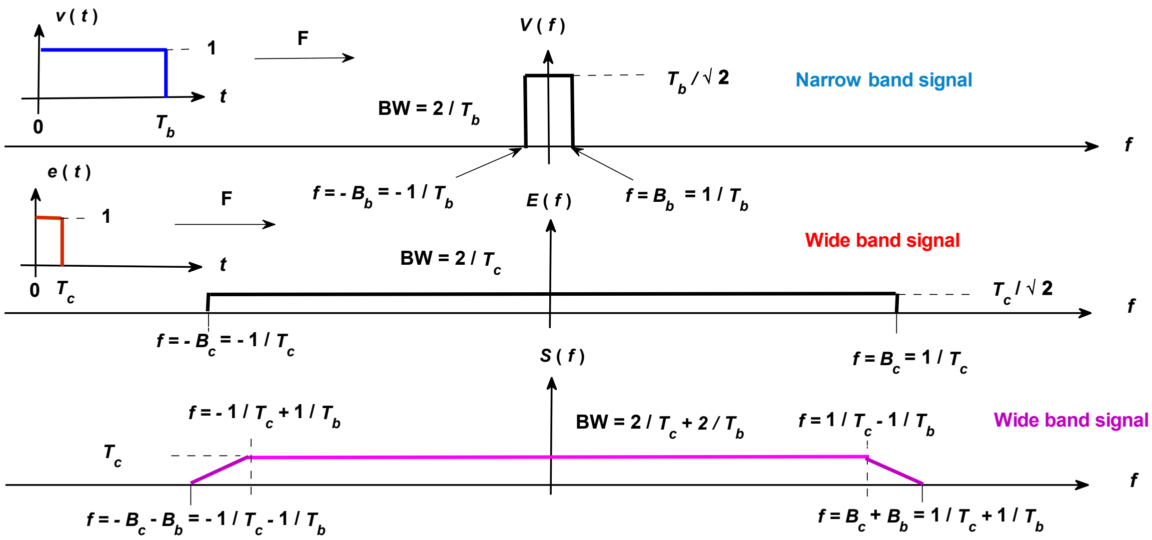

Fig. 1.3 Approximated rectangular spectral representations for $V(f)$, $E(f)$ and $S(f)$.

Exercise 1.3 : Prove analytically that the spectrum of $S(f)$ is as shown in Fig. 1.3 via the use of convolution.

Before we conclude this section, we want to write the mathematical expression of the DS signal with a carrier. So after multiplying $s(t)$ by a sinusoidal carrier at frequency of f_c, we get

$$u(t) = A_c s(t)\cos(2\pi f_c t) \tag{1.16}$$

Since we know that (1.14) that $s(t) = \pm 1$ in any given T_c time interval, then (1.16) can also be converted into

$$u(t) = A_c \cos\left[2\pi f_c t + \theta(t)\right]$$
$$\theta(t) = \begin{cases} 0 & \text{when} \quad s(t) = 1 \\ \pi & \text{when} \quad s(t) = -1 \end{cases} \tag{1.17}$$

(1.17) reveals that DS signal with a carrier is exactly in the form of binary phase shift keying (PSK).

Below we continue without the sinusoidal carrier, since it has no effect on our analysis as stated before.

2. Demodulation (Detection) of DS Signal

We assume that the received signal is $r(t)$. To recover the message signal, we first multiply it by the locally generated PN sequence, that is supposed to be time synchronised (or phase locked) to the $e(t)$ used at transmitter. This multiplication corresponds to dispreading operation, then the resultant is integrated from zero up to T_b. As a whole act is called correlating the received signal with the locally generated PN sequence of the receiver. The output of the correlator is named z, which is considered to contain sufficient statistics that will enable a decision on the transmitted message signal. For this purpose we feed z to a decision making device, i.e. a detector.

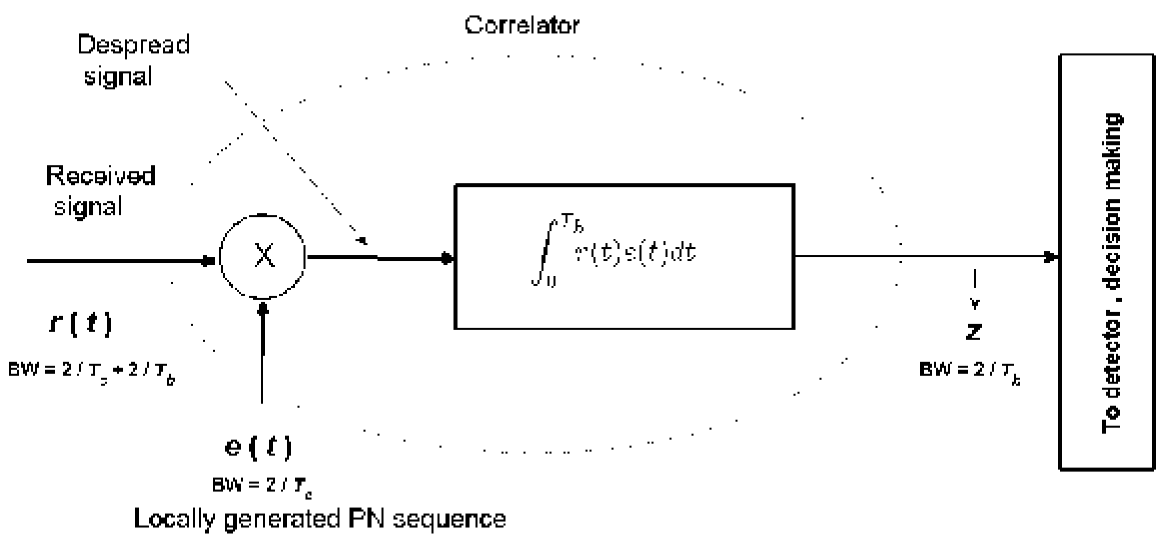

Fig. 2.1 Demodulation of DS signal.

To simplify the description of the operations in Fig. 2.1, we assume an all pass (band unlimited) channel with time and frequency responses of $c(t)=\delta(t),\ C(f)=1$, this means that if we also exclude additive white Gaussian noise (AWGN), then we receive an exact replica of the transmitted signal, thus $r(t)$ will be

$$r(t)=s(t)=v(t)e(t) \tag{2.1}$$

The output of the correlator will be

$$z=\int_0^{T_b} r(t)e(t)\,dt=\int_0^{T_b} v(t)e^2(t)\,dt \tag{2.2}$$

If we are considering the time interval zero up to T_b as the integral limits in (2.2) indicate, then according to (1.1), we can set $n=0$. In this time interval, we have $v(t)_{n=0}=a_0$, if $g(t)$ is rectangular, then

$$z=a_0\int_0^{T_b} e^2(t)\,dt \tag{2.3}$$

In (2.3), the integral is simply the energy in $e(t)$. From the definition given underneath (1.3), we know that in $e(t)$, the coefficient $e_i=\pm1$, thus $e^2(t)=1$ for any T_c time interval and also for the whole duration of T_b. Taking into the length of $\mathbf{e}$ is L_c and we turn the integration (2.3) into a vector inner product or matrix multiplication with dt being T_c, then we get

$$z=a_0\int_0^{T_b} e^2(t)\,dt=a_0T_b \tag{2.4}$$

Or if we apply normalization, then $z=a_0$. So with all pass channel and without noise (and with the presence of only one message signal), our correlator is able to recover the transmitted symbol exactly and our decision device will always give the correct decision.

Next we analyse the effect of noise. With noise signal, $n(t)$ added to $s(t)$, $r(t)$ will become

$$r(t)=s(t)+n(t)=v(t)e(t)+n(t) \tag{2.5}$$

Then, following the above development, z of (2.4) will in this case turn into

$$z=a_0T_b+\int_0^{T_b} n(t)e(t)\,dt=a_0T_b+n_1 \tag{2.6}$$

where n_1 is the instantaneous noise sample. Noise is statistical, hence it is best to continue with variance, i.e. noise power, defined by taking the expectation denoted by $E(\ )$ over two noise samples, which will be given by

$$E(n_1 n_2) = E\left[\int_0^{T_b} n(t)e(t)\,dt \int_0^{T_b} n(\tau)e(\tau)\,d\tau\right] = \int_0^{T_b}\int_0^{T_b} E[n(\tau)n(t)]e(t)\,e(\tau)\,dtd\tau$$
$$= \frac{N_0}{2}\int_0^{T_b}\int_0^{T_b} \delta(t-\tau)e(t)\,e(\tau)\,dtd\tau = \frac{N_0}{2}\int_0^{T_b} e^2(t)\,dt = \frac{N_0}{2}T_b \tag{2.7}$$

Using (2.4) or (2.6) and (2.7), we find the signal to noise ratio (SNR) as

$$\text{SNR} = \frac{a_0^2 T_b^2}{(N_0/2)T_b} = \frac{a_0^2 T_b}{N_0/2} = \frac{\varepsilon_b}{N_0/2} = \frac{\text{Energy in } a_0}{\text{Noise spectral density (two sided)}} \tag{2.8}$$

(2.8) shows that the SNR of DS spread spectrum systems is no different from binary PSK (or ASK) in a narrow band system.

Now we consider the interference rejection capability of the spread spectrum systems, so we express the received signal as

$$r(t) = s(t) + I_i = v(t)e(t) + I_i \tag{2.9}$$

where the interference is represented by the term I_i which is a DC term, is overlapping with $s(t)$ along frequency axis (along time axis as well). In this case we calculate the decision statistics, as follows

$$z = \int_0^{T_b} v(t)e^2(t)\,dt + \int_0^{T_b} I_i e(t)\,dt = a_0 T_b + I_i \int_0^{T_b} e(t)\,dt \tag{2.10}$$

As we shall see later, the spreading sequence, $e(t)$ mostly consists of odd length of $L_c = 2m+1$, out of which m number of e_i coefficients will for instance be $+1$ and $m+1$ number of e_i coefficients will for instance be -1 or the other way round. In any case the integral of $e(t)$ over the whole interval of T_b will be $\pm T_c$. So after rearrangement (2.10) will become

$$z = a_0 T_b\left(1 + \frac{I_i}{a_0 T_b}\int_0^{T_b} e(t)\,dt\right) = a_0 T_b\left(1 + \frac{I_i T_c}{a_0 T_b}\right) = a_0 T_b\left(1 + \frac{I_i}{a_0 L_c}\right) \tag{2.11}$$

(2.11) means that during the demodulation (dispreading) process at the receiver, the transmitted symbol is recovered without loss, but the interfering signal is reduced in amplitude by factor of L_c, where $L_c \gg 1$. In other words, the demodulation process at the receiver restores the signal from the wide band spectrum (extending from $f = -1/T_c - 2/T_b$ to $f = 1/T_c + 2/T_b$) back to its original narrow band spectrum (extending from $f = -1/T_b$ to $f = 1/T_b$), without any loss, but this action will also spread the interfering signal to a wide spectrum (extending from $f = -1/T_c$ to $f = 1/T_c$), so its power remaining within the message signal bandwidth (extending from $f = -1/T_b$ to $f = 1/T_b$), will be much reduced.

Example 2.1 : Assume that we have a DS system $a_0 = 1 \text{ V}$ and an interference of $I_i = 10 \text{ V}$. If $T_b = 1 \text{ msec}$ and $T_c = 1$, find the (message) signal power to interference power ratio (SIR) prior to demodulation (despreading) and after demodulation.

Solution : Normalized power is calculated by taking the square of the voltage, thus before demodulation, we will have

$$\begin{aligned} &\text{Prior to demodulation} \\ &\text{Signal power} : P_s = a_0^2 = 1 \text{ W} \;\;, \;\text{Interference power} : P_i = I_i^2 = 100 \text{ W} \\ &\text{SIR} = \frac{P_s}{P_i} = \frac{1}{100} = 0.01 \quad \text{or} \quad -20 \text{ dB} \end{aligned} \tag{2.12}$$

We know from (2.11) that after demodulation, the amplitude level of the interference will be reduced by $L_c = T_b / T_c = 1000$, thus

$$\begin{aligned} &\text{After demodulation} \\ &\text{Signal power} : P_s = a_0^2 = 1 \text{ W} \;\;, \;\text{Interference power} : P_i = \left(I_i / L_c\right)^2 = 1 \times 10^{-4} \text{ W} \\ &\text{SIR} = \frac{P_s}{P_i} = \frac{1}{1 \times 10^{-4}} = 10^4 \quad \text{or} \;\; 40 \text{ dB} \end{aligned} \tag{2.13}$$

From the comparison of (2.12) to (2.13), we understand that demodulation (despreading) operation has resulted in an SIR improvement of 60 dB.

Now we investigate the interference coming from another user (message signal). Such an investigation is important in the sense that, DS spread spectrum message signals are overlaid in time and frequency. Such analysis will also reveal valuable information on the properties on PN sequences. For simplicity, we take the case of two users as illustrated in 2.2.

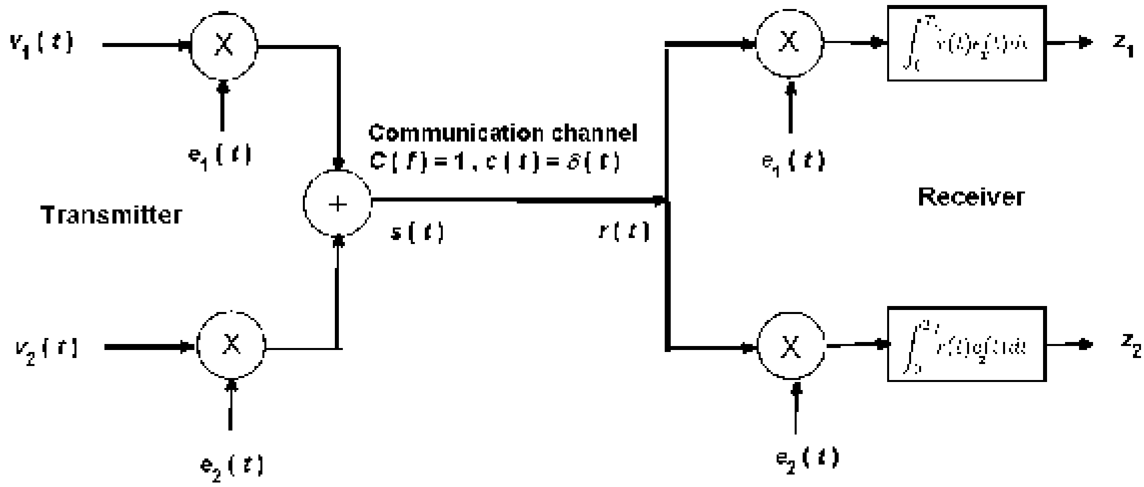

Fig. 2.2 Block diagram of a DS system consisting of two users.

As seen in Fig. 2.2, we have utilized different PN sequences for the two users. Normally the block diagram of Fig. 2.2 would extend to , we have K number of users. As in the case of Fig. 2.1, we have assumed an all pass channel.

For the configuration of Fig. 2.2, the received signal, $r(t)$ will become

$$r(t)=s(t)=v_1(t)e_1(t)+v_2(t)e_2(t) \tag{2.14}$$

After the correlation, the output on the upper branch, i.e., z_1 takes the form of

$$z_1=\int_0^{T_b} v_1(t)e_1^2(t)\,dt+\int_0^{T_b} v_2(t)e_1(t)\,e_2(t)\,dt \tag{2.15}$$

If

$$\begin{aligned} v_1(t)&=\sum_{n=-\infty}^{\infty} {}_1a_n g(t-nT_b) \\ v_2(t)&=\sum_{n=-\infty}^{\infty} {}_2a_n g(t-nT_b) \end{aligned} \tag{2.16}$$

which means, there is no time delay between $v_1(t)$ and $v_2(t)$, i.e. synchronous operation, then in the zeroth time interval of $0\le t\le T_b$, (2.15) will be

$$z_1={}_1a_0T_b+{}_2a_0\int_0^{T_b} e_1(t)\,e_2(t)\,dt \tag{2.17}$$

Similarly for the lower branch we obtain z_2 as

$$z_2 = {}_2a_0 T_b + {}_1a_0 \int_0^{T_b} e_1(t)\, e_2(t)\; dt \tag{2.18}$$

For correct demodulation, we expect that (2.17) delivers ${}_1a_0$, while (2.18) delivers ${}_2a_0$. For this to happen, the second terms, i.e. the terms involving integration should go to zero or approximate to zero. For simplification recalling that the symbols of the transmitted signal, in this case, ${}_1a_0$ and ${}_2a_0$ are either $+1$ or -1, we can rewrite (2.17) and (2.18) as

$$z_1 = \pm T_b \left[1 + \frac{1}{T_b} \int_0^{T_b} e_1(t)\, e_2(t)\; dt \right] \quad , \quad z_2 = \pm T_b \left[1 + \frac{1}{T_b} \int_0^{T_b} e_1(t)\, e_2(t)\; dt \right] \tag{2.19}$$

So we need to examine (T_b normalized) cross correlation of $e_1(t)$ and $e_2(t)$. Ideally we would like this cross correlation to be zero, which means that $e_1(t)$ and $e_2(t)$ should be orthogonal or maximally dissimilar to each other. Below we analyse different options.

3. Spreading Sequences, Pseudo Noise (PN) Sequences

A PN sequence can be generated, by serially connecting shift registers and having some feedback arrangement between them. In such an arrangement, we wish to achieve a sequence that repeats itself every T_b interval, but contains no subsection replicas within the interval of T_b, meaning that we should have maximum length sequences in the interval T_b, or the maximum number of nonperiodic chips (L_c). The time flow of these sequences together with the message symbols is shown below in Fig. 3.1.

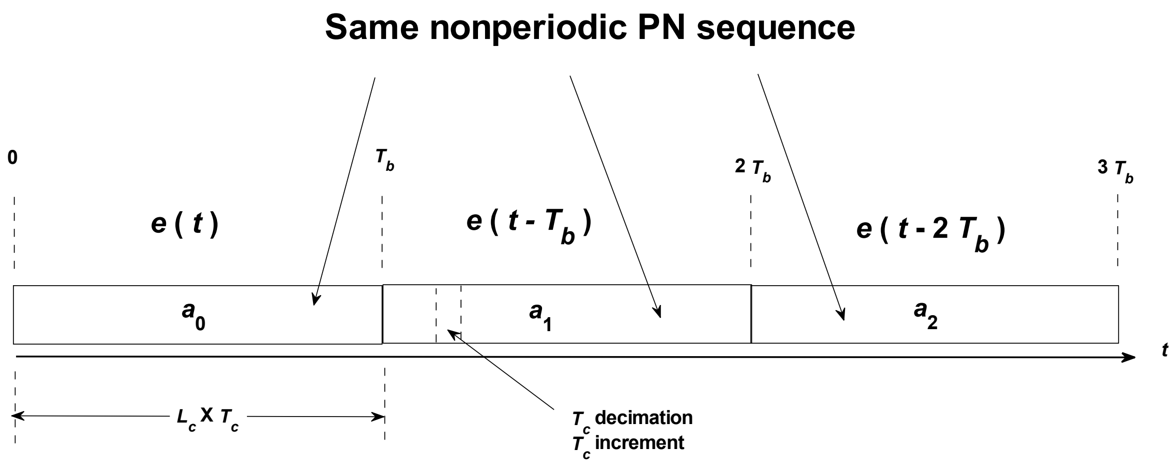

Fig. 3.1 Time flow of PN sequences with message symbols.

Maximal (or maximum) length shift register based PN sequences are based on Hamming codes which are a class of linear block codes with $(n,\ k)$ arranged as $n = 2^m - 1,\ k = 2^m - m - 1$, for some $m \geq 3$, where $m = n - k$. Maximum length codes are on the other hand the dual of Hamming codes which means that $(n,\ k)=(2^m-1,\ m)$ for some $m\geq 3$. Thus we need m number of shift registers with feedback connections expressed in the form of a generator polynomial.

Initially, we take the shift register configuration given on page 462 of Proakis 2008 [5] for $m=3$ (number of shift registers) and trace the output with an initial loading of 001.

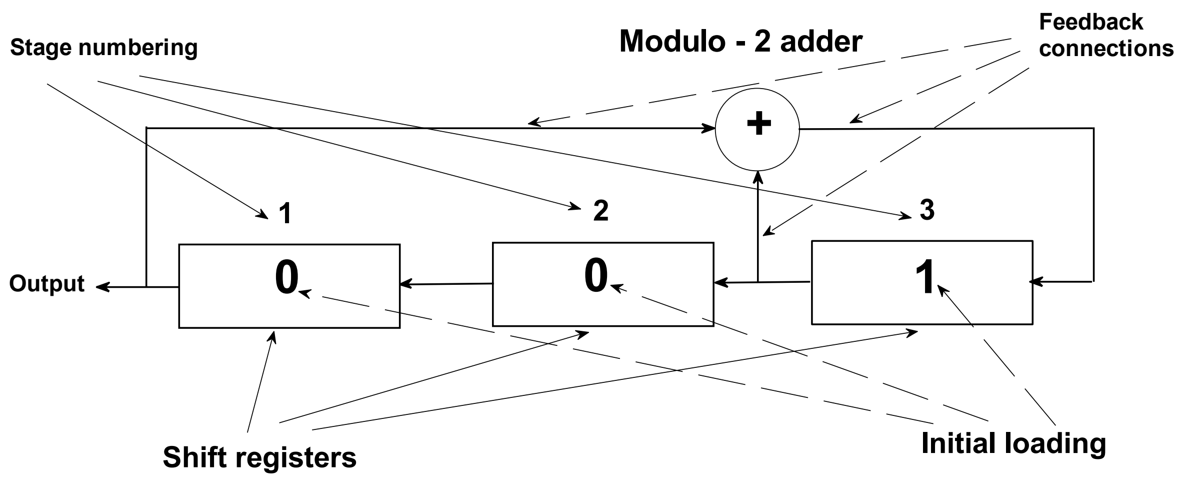

Fig. 3.2 The shift register configuration given on page 462 of Proakis 2008 [5] for $m=3$.

Starting with the initial loading of 001 as shown in Fig 3.2 and assuming that the contents of the shift registers shift to the left at each time increment (decimation) of T_c, we get the following output as time advances

	Content of shift registers			Ouput
	Register 1	Register 2	Register 3	
At $t=0$	0	0	1	$e_0=0$
At $t=T_c$	0	1	1	$e_1=0$
At $t=2T_c$	1	1	1	$e_2=1$
At $t=3T_c$	1	1	0	$e_3=1$ } PN sequence with $L_c=7$
At $t=4T_c$	1	0	1	$e_4=1$
At $t=5T_c$	0	1	0	$e_5=0$
At $t=6T_c$	1	0	0	$e_6=1$

At $t=7T_c$	0	0	1	$e_0=0$
At $t=8T_c$	0	1	1	$e_1=0$
.	.		.	
.	.		.	
.	.		.	

$$\mathbf{e}=[e_0,\ e_1,\ e_2,\ e_3,\ e_4,\ e_5,\ e_6]=[0\ 0\ 1\ 1\ 1\ 0\ 1]\rightarrow[-1\ \ -1\ 1\ 1\ 1\ \ -1\ 1] \quad (3.1)$$

(3.1) shows that the shift register configuration of Fig. 3.1 will produce an PN sequence of $L_c = 2^m - 1 = 7$. The actual sequence is stated at the bottom of (3.1) both in unipolar (i.e. in terms of 0s and 1s) and bipolar (i.e. in terms of -1s and 1s) versions.

Exercise 3.1 : Assume an initial loading of 100 for the shift register configuration of Fig. 3.2 and find the output by tracing. Comment on its difference from the one written on the last line of (3.1).

Now we come to the representation of the shift register configuration of Fig. 3.2 in Matlab. This is important, from the point of establishing equivalence and learning how to generate longer PN sequences which are the ones used practically. In fact, from this point onwards, we deal with the PN sequence generator of Matlab. The first point about the shift register configuration of Fig. 3.2 is that it is drawn in Matlab notation by rotating everything 180^0, as shown in Fig. 3.3.

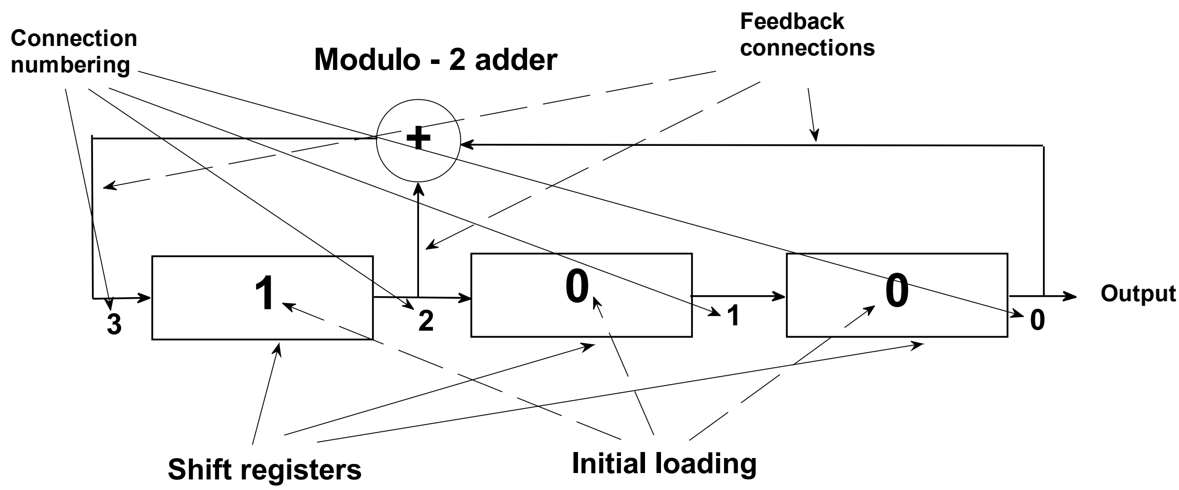

Fig. 3.3 Shift register arrangement in Matlab notation for $m = 3$.

The feedback arrangement of Fig. 3.3 is specified in terms of connections numbers and inserted in the Matlab poly function as follows

Connection number

$$
\begin{array}{llll}
 & 3 & 2 & 1 & 0 \\
\text{poly} = [& 1 & 1 & 0 & 1] \\
 & \uparrow & & \uparrow & \\
 & \text{presence of} & & \text{absence of} & \\
 & \text{a connection} & & \text{a connection} & \\
 & \text{to Modulo - 2} & & \text{to Modulo - 2} & \\
 & \text{adder} & & \text{adder} &
\end{array}
$$

Alternative representation

Connection number

$$
\begin{array}{lllll}
 & 3 & 2 & 1 & 0 \\
\text{poly} = [& 1 & 1 & 0 & 1] \\
\text{poly} = [& 3\times 1 & 2\times 1 & 1\times 0 & 0\times 1] = [\,3\ 2\ 0]
\end{array}
$$

$$g_p(p) = 1\times p^3 + 1\times p^2 \ + \ 0\times p \ + \ 1\times p^0 \tag{3.2}$$

The arrangement of Fig. 3.3 with the initial loading of 100 again produces the same PN sequence on the last line of (3.1), which is also given below and vectorially named $\mathbf{e}_1$.

$$\mathbf{e}_1 = [{}_1e_0,\ {}_1e_1,\ {}_1e_2,\ {}_1e_3,\ {}_1e_4,\ {}_1e_5,\ {}_1e_6] = [0\ 0\ 1\ 1\ 1\ 0\ 1] \rightarrow [-1\ -1\ 1\ 1\ 1\ -1\ 1] \tag{3.3}$$

The interesting characteristics of the sequence in (3.3) is that no matter what cyclic shift, we subject it to, we cannot get the original sequence. Furthermore in $\mathbf{e}_1$, there are no repetitions of subsequences of length 3. This way we say $\mathbf{c}_1$ satisfies the maximum length property. For such a PN sequence, the cyclic autocorrelation would look like the one displayed in Fig. 3.4.

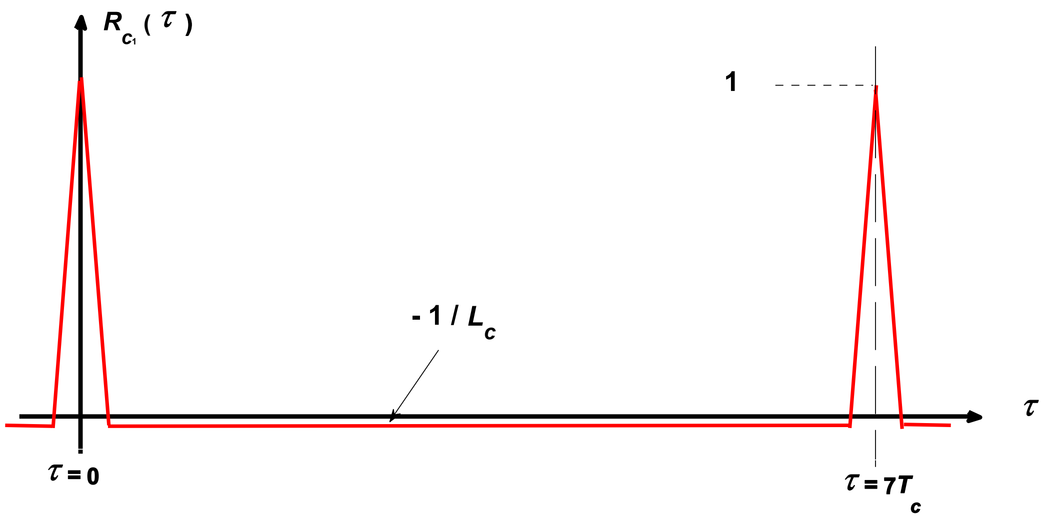

Fig. 3.4 Cyclic autocorrelation of the PN sequence in (3.3).

Deferring the description of cyclic autocorrelation function for the moment, we concentrate on the three shift register arrangement and try to find what other possible feedback connection would generate, in particular if maximum length PN sequences other than the one in (3.3) are available. One possible variation from the configuration given in Fig. 3.3 is the one drawn in Fig. 3.5 (note that to highlight the necessary items, we have implemented simplification in Fig. 3.5.

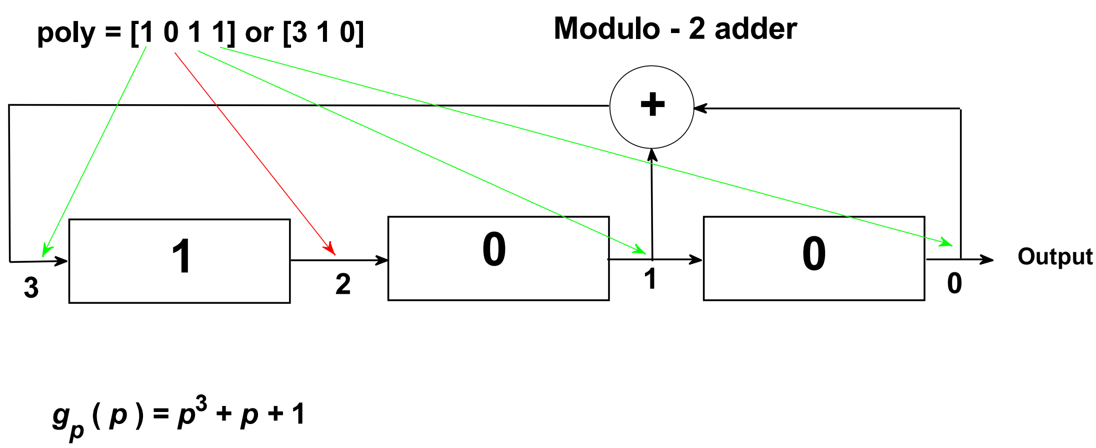

Fig. 3.5 Another shift register configuration to generate max length PN sequence $m = 3$.

In Fig 3.5, the polynomial in Matlab and in modulu-2 notation is also typed in the figure. The output from the configuration of Fig. 3.5 is (again with the initial loading of 100)

$$\mathbf{e}_2 = \left[{}_2e_0,\ {}_2e_1,\ {}_2e_2,\ {}_2e_3,\ {}_2e_4,\ {}_2e_5,\ {}_2e_6\right] = \left[0\ 0\ 1\ 0\ 1\ 1\ 1\right] \rightarrow \left[-1\ \ -1\ 1\ \ -1\ 1\ 1\ 1\right] \qquad (3.4)$$

The comments made for $\mathbf{e}_1$ are also valid for $\mathbf{e}_2$. Therefore, in $\mathbf{e}_2$, there is no repetitions of subsequence of length 3, this way the cyclic autocorrelation for $\mathbf{e}_2$ will also be like the one shown in Fig. 3.4.

4. Cyclic Auto Correlation, Cross Correlation

In this section, we give formulations of cyclic autocorrelation, crosscorrelation.

Now given PN sequence $e(t)$ (in fact any time function), that exists over a time interval of T_b, the autocorrelation is defined as

$$R_e(\tau) = \frac{1}{T_b}\int_0^{T_b} e(t)\, e(t+\tau)\, dt \qquad (4.1)$$

The natural assumption here is that it is sufficient for τ to sweep from $-T_b$ to T_b, so that $R_e(\tau)$ covers a time range of $2T_b$. For crosscorrelation, we need two sequences, let those be $e_1(t)$ and $e_2(t)$, then

$$R_{e_1e_2}(\tau)=\frac{1}{T_b}\int_0^{T_b} e_1(t)\, e_2(t+\tau)\; dt \tag{4.2}$$

Another important parameter for PN sequences is the mean (or the DC) value, which can be computed as

$$m_e=\frac{1}{T_b}\int_0^{T_b} e(t)\, dt \tag{4.3}$$

A desirable PN sequence should have the following characteristics

a) $m_e=0$

b) $R_e(\tau)=\begin{cases}1 & \tau=nT_b \quad , \quad n=-\infty,\ \cdots\ -1,0,1,\ \cdots\ \infty \\ 0 & \tau\neq nT_b\end{cases}$

c) $R_{e_1e_2}(\tau)=0 \qquad -\infty\leq\tau\leq\infty$

d) For an m (number of shift registers) value, it should be possible to get the largest number of maximum length PN sequences.

We know from the above developments that m number of shift registers is able to generate sequences of $L_c=2^m-1$, additionally these sequences will contain 2^{m-1} number of -1s and $2^{m-1}-1$ number of 1s or vice versa. As m increases, m_e will approach zero more rapidly. So this satisfies condition a). It is also know that, if the generated PN sequence is maximum length, then condition b) is also satisfied. We also know that getting the largest number of maximum length PN sequences depends on the number of irreducible polynomials that exist for a given m . But we have the cross correlation properties stated c). Below we study this together with the computation of autocorrelation and crosscorrelation for the discrete forms of the PN sequences as we have for DS spread spectrum systems.

It is well know that integration of the continuous world corresponds to approximating the interval to rectangles, then finding the area of each and finally summing these individual rectangular areas to arrive at the result. So bearing in mind that

$$e(t)=\sum_{i=0}^{L_c-1} e_i p(t-iT_c) \quad \rightarrow \quad \mathbf{e}=\underbrace{[\overbrace{e_0,\quad e_1}^{T_c},\ e_2,\ e_3,\ \cdots\ ,\ e_{L_c-1}]}_{T_b} \tag{4.4}$$

Then at $\tau=0$ (4.1) will turn into

$$R_e(\tau=0) \;\rightarrow\; r_{\mathbf{e}} = \frac{1}{T_b}\mathbf{e}\,\mathbf{e}^t = \left[e_0,\, e_1,\, e_2,\, e_3,\, \cdots ,\, e_{L_c-1}\right] \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ \cdot \\ \cdot \\ \cdot \\ e_{L_c-1} \end{bmatrix}$$

$$= \frac{T_c}{T_b}\left(e_0^2 + e_1^2 + e_2^2 + e_3^2 + \;\cdots\; + e_{L_c-1}^2\right) = \frac{T_c}{T_b} L_c = 1 \tag{4.5}$$

where the subscript t denotes the transpose operation. In order to create the shifts in τ, i.e. for $\tau > 0$, we have to cyclically shift $\mathbf{e}^t$ at intervals of T_c to obtain the complete cyclically shifted autocorrelation function. For a general shift of jT_c times, $r_{\mathbf{e}}(jT_c)$ will be given by

$$r_{\mathbf{e}}(jT_c) = \sum_{i=0}^{L_c-1-j} e_i e_{i+j} + \sum_{i=L_c-j}^{L_c-1} e_i e_{i+j-L_c} \quad , \quad 0 \le j \le L_c - 1 \tag{4.6}$$

Assuming $j > L_c - 1 - j$ or $2j > L_c - 1$, then the corresponding coefficients (elements) of $\mathbf{e} = \mathbf{e}^{(0)}$ and $\mathbf{e}^{(jT_c)}$ would be aligned for multiplication to be performed in (4.6) as shown below in (4.7)

$$\begin{array}{ll} \mathbf{e}^{(0)} & = \left[e_0 \;\; e_1 \;\; e_2 \;\; e_3 \;\cdots\; e_{L_c-1-j} \;\; e_{L_c-1-j} \;\cdots\; e_j \;\cdots\cdots\; e_{L_c-1}\right] \\ & \quad\;\; \downarrow \;\;\; \downarrow \;\;\; \downarrow \;\;\; \downarrow \qquad\quad \downarrow \qquad\;\; \downarrow \qquad\quad \downarrow \qquad\quad \downarrow \\ \mathbf{e}^{(jT_c)} & = \left[e_j \;\; e_{j+1} \;\; e_{j+2} \;\; e_{j+3} \;\cdots\; e_{L_c-1} \;\;\; e_0 \;\cdots\cdots\; e_{2j-L_c} \;\cdots\; e_{j-1}\right] \end{array} \tag{4.7}$$

In the end, we will obtain the discrete equivalent, i.e. cyclic autocorrelation of $R_e(\tau)$ by placing $r_{\mathbf{e}}(jT_c)$ in a row array as follows

$$R_e(\tau) \rightarrow \mathbf{R}_{\mathbf{e}} = \left\{r_{\mathbf{e}}(0),\, r_{\mathbf{e}}(T_c),\, \cdots\; r_{\mathbf{e}}(jT_c) \;\cdots\; r_{\mathbf{e}}\left[(L_c-1)T_c\right]\right\} \tag{4.8}$$

In matrix form, (4.8) can be visualized as

$$\mathbf{R}_{\mathbf{e}} = \overbrace{\begin{bmatrix} e_0 & \cdots\cdots & e_{L_c-1} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ e_0 & \cdots\cdots & e_{L_c-1} \end{bmatrix}}^{\text{Rows are } \mathbf{e}^{(0)}} \overbrace{\begin{bmatrix} e_0 & \cdots & e_j & \cdots & e_{L_c-1} \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot \\ e_{L_c-1} & \cdots & e_{j-1} & \cdots & e_{L_c-2} \end{bmatrix}}^{\text{Columns are } \mathbf{e}^{(jT_c)} \text{ for } 0 \le j \le L_c-1} \tag{4.9}$$

Now taking the simple case of $\mathbf{e} = \mathbf{e}_1$ from (3.6), below we perform a sample calculation of $r_{\mathbf{e}}(3T_c)$

$$\begin{array}{c} e_0 \quad e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \end{array}$$

$$\mathbf{e} = [\underbrace{-1 \;\; -1}_{T_c} \;\; 1 \;\; 1 \;\; 1 \;\; -1 \;\; 1]$$

$$r_{\mathbf{e}}(jT_c) = \frac{1}{T_b}\mathbf{e}^{(0)}\;\mathbf{e}^{t(3T_c)} = \frac{1}{T_b}[e_0,\, e_1,\, e_2,\, e_3,\, e_4,\, e_5,\, e_6]\begin{bmatrix} e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_0 \\ e_1 \\ e_2 \end{bmatrix}$$

$$= \frac{1}{T_b}[-1 \;\; -1 \;\; 1 \;\; 1 \;\; 1 \;\; -1 \;\; 1]\begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} = \frac{-1}{7} \tag{4.10}$$

Continuing in this manner, we will find

$$\begin{array}{cccccccccccc} & r_{\mathbf{e}}(0) & r_{\mathbf{e}}(T_c) & r_{\mathbf{e}}(2T_c) & r_{\mathbf{e}}(3T_c) & r_{\mathbf{e}}(4T_c) & r_{\mathbf{e}}(5T_c) & r_{\mathbf{e}}(6T_c) & r_{\mathbf{e}}(7T_c) & r_{\mathbf{e}}(8T_c) & r_{\mathbf{e}}(9T_c) \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \mathbf{R}_{\mathbf{e}} = & \multicolumn{7}{c}{\underbrace{\left[1 \quad -\frac{1}{7} \quad -\frac{1}{7} \quad -\frac{1}{7} \quad -\frac{1}{7} \quad -\frac{1}{7} \quad -\frac{1}{7}\right]}_{T_b}} & 1 & -\frac{1}{7} & -\frac{1}{7} \end{array} \tag{4.11}$$

Expressed in the usual manner, we have the following representation

$$R_e(\tau : 0 \rightarrow 6T_c) = \begin{cases} 1 & 0 \le \tau \le T_c \\ -\frac{1}{7} = -\frac{1}{L_c} & T_c \le \tau \le (L_c - 1)T_c \end{cases} \tag{4.12}$$

So the plot of (4.11) or (4.12) will look exactly like the one given in Fig. 3.4. This is the typical behaviour of auto correlation of a PN code which has maximum length. Therefore Fig. 3.4 will also applicable for $\mathbf{e}_2$ of (3.6).

Exercise 4.1 : By using the Matlab files, PNrun_Exp3.m together with PNGenExp3.mdl, it is possible to generate PN sequences of any length. Here we are interested in generating $\mathbf{e}_1$ given in (3.3) and $\mathbf{e}_2$ given in (3.4). Confirm that in PNrun_Exp3.m, poly functions for $\mathbf{e}_1$ and $\mathbf{e}_2$ are written as stated in

(3.2) and Fig. 3.5. Also confirm the sequences are as given in (3.3) and (3.4). Finally confirm that cyclic autocorrelations of $\mathbf{e}_1$ and $\mathbf{e}_2$ are as displayed in Fig. 3.4.

These notes are based on

1) Michael B. Pursley, "Introduction to Digital Communications", International Edition, Prentice Hall 2005, ISBN : 0-13-123392-0.
2) S. Verdu, "Multiuser Detection ", Cambridge University Press 2005, ISBN : 0-521-59373-5.
3) John G. Proakis, Masoud Salehi, "Communication Systems Engineering" 2nd Ed. 2002, ISBN : 0-13-061793-8.
4) John G. Proakis, Masoud Salehi, "Fundamentals of Communication Systems", Prentice Hall 2005, ISBN : 0-13-147135-X.
5) John G. Proakis, Masoud Salehi, "Digital Communications", McGraw Hill 2008, ISBN : 978-007-126378-8.
6) MATLAB help files.
7) My own Lecture Notes.